Chapter 1 - Introduction To MVS/QuickRef ................................................................. 7

Preface .......................................................................................................................... 8
Road Map for This Guide ............................................................................................. 8
What's New For This Release: ..................................................................................... 9
Overview ...................................................................................................................... 9
What is MVS/QuickRef? ............................................................................................... 10
   The Philosophy Behind MVS/QuickRef ................................................................. 11
   MVS/QuickRef Usage Examples ............................................................................ 11

MVS/QuickRef Data Base Contents ......................................................................... 12
   MVS, OS/390, and z/OS Messages and Codes ...................................................... 12
   MVS JCL Syntax .................................................................................................... 13
   Assembler Language Syntax ............................................................................... 14
   MVS Utility Reference Data ................................................................................ 14
   CICS/MVS & CICS/ESA Reference Information ................................................ 15
   IMS/VS & IMS/ESA Reference Information ......................................................... 16
   REXX Language Syntax ...................................................................................... 18
   COBOL, PL/1, C++, and C Language Syntax ...................................................... 18
   DB2 Messages and Codes .................................................................................... 18
   Independent Software Vendor Product Messages ............................................. 19

Chapter 2 - Accessing and Using MVS/QuickRef ...................................................... 21

Introduction ................................................................................................................ 22
Information Storage and Retrieval Overview ............................................................ 22
Invocation Commands ............................................................................................... 24
Invocation Techniques .............................................................................................. 24
   Menu-Driven Invocation ...................................................................................... 25
   Fast-Path Invocation .......................................................................................... 26
   Cursor-Driven Invocation .................................................................................. 28
Terminating/Exiting MVS/QuickRef .......................................................................... 30
Restoring Previous Display ...................................................................................... 30
Using The Help Facility ............................................................................................. 30

Chapter 3 - MVS/QuickRef Features ........................................................................ 31

Introduction ............................................................................................................... 32
Additional On-Line Features ..................................................................................... 32
User-Directed Selection List Processing .................................................................. 33
   General Rules ................................................................................................... 34
   Types Of User-Directed Selection List Processing .......................................... 36
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using LIBDEF</td>
<td>95</td>
</tr>
<tr>
<td>Customizing MVS/QuickRef</td>
<td>96</td>
</tr>
<tr>
<td>Setting MVS/QuickRef Global Installation Options</td>
<td>97</td>
</tr>
<tr>
<td>Customizing The Vendor/Product Specific Selection List Table</td>
<td>105</td>
</tr>
<tr>
<td>Expiration Date Zaps</td>
<td>109</td>
</tr>
<tr>
<td>Preparing to Test MVS/QuickRef</td>
<td>109</td>
</tr>
<tr>
<td>Testing MVS/QuickRef</td>
<td>110</td>
</tr>
<tr>
<td>Troubleshooting MVS/QuickRef</td>
<td>111</td>
</tr>
<tr>
<td>Production Implementation</td>
<td>114</td>
</tr>
<tr>
<td>Production Implementation Considerations</td>
<td>114</td>
</tr>
<tr>
<td>User Documentation Considerations</td>
<td>114</td>
</tr>
<tr>
<td>Updating the MVS/QuickRef Data Base</td>
<td>115</td>
</tr>
<tr>
<td>Refreshing the MVS/QuickRef Data Base</td>
<td>116</td>
</tr>
<tr>
<td>Data Base Merge Facility</td>
<td>116</td>
</tr>
<tr>
<td>Selective Data Base Load Facility</td>
<td>120</td>
</tr>
<tr>
<td>Selective Load Versus User-Directed Selection List Processing</td>
<td>126</td>
</tr>
<tr>
<td>Correcting the Text of an MVS/QuickRef Item</td>
<td>128</td>
</tr>
<tr>
<td>Merging Data From a Previous Release</td>
<td>128</td>
</tr>
<tr>
<td>Using MVS/QuickRef Under CA-ROSCOE</td>
<td>130</td>
</tr>
<tr>
<td>Users of CA-ROSCOE 5.7 or Above</td>
<td>131</td>
</tr>
<tr>
<td>Users of CA-ROSCOE 5.6 or Below</td>
<td>131</td>
</tr>
<tr>
<td>CA-ROSCOE Considerations</td>
<td>132</td>
</tr>
<tr>
<td>Chapter 5 - Displaying Your Own Reference Information</td>
<td>133</td>
</tr>
<tr>
<td>Introduction</td>
<td>134</td>
</tr>
<tr>
<td>Overview</td>
<td>134</td>
</tr>
<tr>
<td>Deciding how many user data bases you need</td>
<td>135</td>
</tr>
<tr>
<td>Building User Data Base(s)</td>
<td>135</td>
</tr>
<tr>
<td>Step 1 - Allocate and Load User Data Base(s)</td>
<td>136</td>
</tr>
<tr>
<td>User Data Base JCL DD Statements</td>
<td>137</td>
</tr>
<tr>
<td>QWIKREF2 Return Codes</td>
<td>139</td>
</tr>
<tr>
<td>Key Indicator Records</td>
<td>139</td>
</tr>
<tr>
<td>Alias Indicator Records</td>
<td>141</td>
</tr>
<tr>
<td>Text Records</td>
<td>143</td>
</tr>
<tr>
<td>Copyright Indicator Records</td>
<td>144</td>
</tr>
<tr>
<td>Category Code Indicator Records</td>
<td>148</td>
</tr>
<tr>
<td>Start Access Based On Content Indicator Records</td>
<td>151</td>
</tr>
<tr>
<td>Automatic Lookup Based On Pointers</td>
<td>153</td>
</tr>
<tr>
<td>Input File Processing Considerations</td>
<td>155</td>
</tr>
<tr>
<td>User Data Base Member Selection Control Statement Syntax</td>
<td>155</td>
</tr>
<tr>
<td>User Data Base JCL Examples</td>
<td>156</td>
</tr>
<tr>
<td>Step 2 - Define Name(s) of User Data Base(s)</td>
<td>159</td>
</tr>
<tr>
<td>Determine If User Menu Panel Is Needed</td>
<td>160</td>
</tr>
</tbody>
</table>
Preface

This is the user's guide for Release 6.9 of MVS/QuickRef, a productivity tool for MVS programmers and operators. This guide will help you understand the structure and function of MVS/QuickRef, but more importantly, how to use the product effectively to make your day-to-day interface with MVS, OS/390, z/OS, and other MVS-based products from IBM and other software vendors more productive. MVS/QuickRef is very easy to use, as you will see in this guide. As you use the product, remember that on-line help information is provided. The on-line help information can be accessed by pressing the ISPF HELP PF key, or by typing HELP on the command line of any MVS/QuickRef panel.

Please note that there is a feedback form at the end of this guide. If you determine that some of the information you need is not available within MVS/QuickRef, please use the feedback form to let us know. We want to make this product as helpful to you as possible, so don't hesitate to send us your product improvement suggestions. You can best communicate with us via email to our Internet email address, support@quickref.com.

If you need additional copies of this guide, it is available in Adobe Acrobat™ format on our web site at http://www.quickref.com. The Adobe Acrobat™ Reader is free and is available from the Adobe web site at http://www.adobe.com.

Road Map for This Guide

It would be best if you read this user's guide in its entirety. However, if you do not have time to read the entire manual, use the "road map" below to determine exactly which portions you do need to read.

- If you are a first-time user of MVS/QuickRef, you should read at least the first two chapters. This will give you enough basic information to "get started". As time permits, you should also read chapter three (which describes certain additional features of MVS/QuickRef which you may want to use).

- If you are a casual or first-time user who does not have time to read the first two chapters, read the sections titled "Invocation Commands" and "Using The Help Facility" in Chapter Two. This will give you enough information to bring up the MVS/QuickRef main menu. From there, you can use the menu options to find the information you need or you can use the MVS/QuickRef help facility to learn more about MVS/QuickRef. The help item named QUICK-START, which provides a "quick reference sheet" for MVS/QuickRef, may provide enough information to get you started using MVS/QuickRef.

- If you are a first-time user who has installation and/or technical support responsibilities for MVS/QuickRef, you should read at least the first four chapters.

- If you are an experienced MVS/QuickRef user, read the section on "What's New For This Release" in chapter one. Based on the description of the new enhancements provided by this
release, you should be able to determine the other portions of the manual that you need to review.

- If you are an experienced user who has installation and/or support responsibilities for MVS/QuickRef, read at least the section on "What's New For This Release", the other sections of this user's guide which the new enhancements may suggest to you, and the installation information in chapter four.

**What's New For This Release:**

**Program Enhancements:**

-  

**Data Base Content Enhancements:**

- Additional data base content changes consisting of corrections, deletions, and additions of new content from participating Independent Software Vendors have been made. A full list of these changes, which are extensive, can be found by invoking the “What’s New” option on the MVS/QuickRef R6.8 main menu.

**Overview**

This is the user's guide for MVS/QuickRef; it is organized as follows:

- Chapter One provides an introduction to MVS/QuickRef, including examples of the available reference information provided by MVS/QuickRef and how that information can be used.

- Chapter Two describes how you access and use MVS/QuickRef.

- Chapter Three describes certain additional features of MVS/QuickRef and includes specifics on how to use each additional feature.

- Chapter Four contains the steps to follow to install and customize MVS/QuickRef.

- Chapter Five tells you how to define and implement your own user data bases with MVS/QuickRef. With a user data base, you can store and present any kind of locally produced reference text, from telephone extension lists to complete program reference documents.

- Chapter Six presents a discussion of the MVS/QuickRef override parameter feature, which allows you to logically add, replace, or prevent access to items in the main reference data base or your own user data base.

- Appendix A. explains how you can find descriptions of the various messages issued by MVS/QuickRef. Appendix B. describes the user abend codes issued by MVS/QuickRef.
What is MVS/QuickRef?

MVS/QuickRef is an ISPF information storage and retrieval system which can be used to provide quick, convenient, on-line access to the types of reference information frequently needed by MVS application programmers, system programmers, and operators. It comes with a data base, referred to as the MVS/QuickRef main data base, which is pre-loaded with information which includes descriptions for the following topics:

- MVS messages and codes
- MVS JCL syntax
- Assembler language syntax
- MVS Utilities JCL and Usage
- MVS Reference Summary
- TSO CLIST Language Syntax
- CICS Messages & Abend Codes
- IMS User Abends & Messages
- REXX Language Syntax
- COBOL Language Syntax
- DB2 Messages and codes
- VTAM messages and sense codes
- Independent Software Vendor Messages
- MVS, JES2, JES3, and VTAM commands syntax
- C, C++, and PL/I Language Syntax

The reference information is presented on a scrollable display panel. For messages and codes, the reference information will normally provide a general description of the item, including its format, meaning, possible causes, and necessary corrective actions. The reference information for items like commands, JCL parameters, utility programs, and programming language elements normally includes an explanation of the item's function and syntax requirements as well as examples showing how it can be used.

In addition to the information in the main data base, MVS/QuickRef provides an optional facility which can be used to create and access one or more installation-defined data bases. Such installation-defined data bases are referred to as user data bases. Up to nine user data bases can be created and accessed. User data bases can be used to store and display installation-specific information like local job submission standards, local JCL coding standards, etc.

MVS/QuickRef is designed to provide fast, convenient access to the information in the data base. This is accomplished by providing rapid data base access and display techniques which:

- require an absolute minimum number of key strokes
- cause minimal interruption to the work currently in progress
- use "obvious" key values whenever possible
- do not require that the absolute key value be known in order to find some needed information

To ensure ease of use, MVS/QuickRef is designed as a "pop-up" application. When invoked, it "pops-up" on top of the current ISPF application display and, when you are through reviewing the information you need and MVS/QuickRef is terminated, the previous ISPF application
display is restored. In this way, MVS/QuickRef can be invoked to get some needed information, whenever you need it as you work at your terminal, without a significant interruption to the work that was in progress.

In addition to the information supplied by the data base, MVS/QuickRef provides "seamless" access to the information in the partitioned data set(s) concatenated to the SYSHELP DD statement. This provides reference information for TSO commands and for any other products in your ISPF installation that provide on-line assistance via the TSO HELP command.

MVS/QuickRef also provides on-line, real-time information on DASD free space. This includes the ability to generate and display lists of DASD volumes based on various selection and sorting criteria. Each DASD volume serial listed on a DASD free space display shows certain DASD characteristics for that volume (mount attribute, device type, density, etc.) as well as the number of free cylinders and tracks on that volume.

**The Philosophy Behind MVS/QuickRef**

MVS/QuickRef was not designed to be an encyclopedic reference; rather, it presents in a concise form the information you need to quickly solve the most frequently occurring MVS related operational problems. MVS/QuickRef does not store reference text using the book metaphor common with some other on-line text storage and retrieval systems. Instead, information in the MVS/QuickRef data base is stored as individual discrete items, such as an individual message description, a syntax overview for a particular JCL keyword, or an example of how to use a specific verb in a programming language. MVS/QuickRef is designed to give you a quick answer to a specific question with a minimum number of interactions with the system, so that you can go on with the task at hand as you work at your terminal.

**MVS/QuickRef Usage Examples**

MVS/QuickRef can be used in a variety of ways; some examples are described below:

- If you are an application programmer or MVS scheduler, you can use MVS/QuickRef to get immediate explanations for MVS messages associated with batch job output, while you are reviewing the job output. No matter what product or facility is used to examine job output, MVS/QuickRef can be accessed whenever an unusual message or abend is encountered, without leaving the output review function. No interruption in your work flow or thinking process need occur to locate the proper reference manual and look up the message - MVS/QuickRef provides the answer after a few quick keystrokes. And, for most messages and abend codes, in addition to a clear and concise description of the message or abend, MVS/QuickRef also provides a set of recommended actions for the user to follow in order to solve the problem.

- If you are an MVS operator familiar with ISPF, use the DASD free space display feature of MVS/QuickRef to locate the largest free DASD space quantity when sort work or public work space deficiencies cause production job abends. Many MVS console messages or batch job abends can also be rapidly analyzed without the need for a messages and codes manual.
As an Assembler language application programmer or MVS system programmer, you can use the assembler language reference functions of MVS/QuickRef as an aid during the program coding process, without consulting the IBM Principles of Operations manual. You need not stop coding or interrupt your train of thought to locate an explanation of how a particular machine instruction is specified or executes.

As you use TSO and ISPF daily, you will find MVS/QuickRef helpful when you need to know how to invoke or use a particular TSO command, or to answer questions on REXX or CLIST language syntax when creating or debugging a REXX exec or CLIST under ISPF.

Each one of the above examples illustrates the value of MVS/QuickRef - you as an ISPF user need not stop what you are doing at the TSO terminal, locate the proper reference manual (assuming you can find it at all), then find information on the desired topic. MVS/QuickRef is instantly available to provide the necessary information or recommended user action. Your productivity under ISPF increases, and more useful work is done in less time.

MVS/QuickRef Data Base Contents

The sections which follow give examples of the types of reference information carried in the MVS/QuickRef main data base.

MVS, OS/390, and z/OS Messages and Codes

The MVS/QuickRef main data base contains reference information for all MVS, OS/390, and z/OS system messages and abend codes. An example of the reference information provided for the IEC149I message is shown in Figure 1.
Message Format: IEC149I 813-rc,mod,jjj,sss,ddn(-#),ser,dsn

Description: An error occurred during OPEN processing for a data set on a magnetic tape. The 'rc' value in the message is a return code that describes the error in detail. Possible 'rc' values and their meanings are provided below.

For rc=04, an OPEN was issued for a data set on magnetic tape, but the data set name in the tape's header label did not match the data set name in the JCL.

User Action: Ensure that the data set name and volume serial on the DD statement are correct; if they are, verify that the JFCB was not modified incorrectly by the program that issued the OPEN.

---

**Figure 1 - IEC149I Message Description**

**MVS JCL Syntax**

Complete descriptions of the key elements of MVS Job Control Language (JCL) statements are available. JCL information includes descriptions of JOB, EXEC, DD, OUTPUT, IF, INCLUDE, JCLLIB, and JES2/JES3 JECL statement formats and keywords. The reference information provided for the JOB statement CLASS parameter is shown in Figure 2.

---

* ***The CLASS Parameter*** *

The CLASS parameter is used to assign your job to a job processing class. The class you should request depends on the characteristics of the job and your installation's rules for assigning classes. Consult your installation's operations staff or MVS System Programmer for a list of valid job classes and their processing characteristics.

Syntax:

CLASS=jobclass

where "jobclass" is a single alphanumeric character that specifies the desired job class, in the range A-Z or 0-9.

In a JES2 system, the job class used affects whether or not a job is executed. A job class on a JES2 system can be:

---

**Figure 2 - JOB CLASS Description**
### Assembler Language Syntax

Syntax and usage descriptions for the most frequently needed IBM System 370 and System 390 assembler and machine instructions is also available in the MVS/QuickRef data base. Figure 3 is a display of the reference information for the BASSM machine instruction.

### MVS Utility Reference Data

JCL and usage information is provided for the most frequently used MVS utility programs, including IEBCOPY, IEBGENER, IEHLIST, AMASPZAP, AMBLIST, ASMA90 (High Level Assembler), IEBIMAGE, IDCAMS, IEBPTPCH, DFSORT, SYNCSORT, IEHPROGM,IEWL, IEBUPDTE, IEV90 (Assembler H), ICKDSF, IEFR14, and the Binder/Loader. The reference text for each utility includes examples of the JCL used to invoke that utility. You can use the MVS/QuickRef CUT and PASTE feature to transfer these examples into your own JCL library when you need to invoke one of the documented utilities. Figure 4 is a display of some of the MVS/QuickRef information for the IEBCOPY utility.

---

**Figure 3 - BASSM Assembler Instruction Description**

<table>
<thead>
<tr>
<th>Instruction Format: BASSM R1,R2</th>
<th>Operation Code (Hex): 0C</th>
</tr>
</thead>
</table>

**Examples:**

- BASSM R14,R15: BRANCH TO ADDRESS IN R15
- BASSM R2,R2: BRANCH TO ADDRESS IN R2 AFTER COMPUTING LINKAGE ADDRESS FROM R2

**Description:**

The BRANCH AND SAVE AND SET MODE instruction causes program execution to branch to the address provided in the second operand register. Linkage information consisting of bits 32-63 (right half) of the current PSW is saved in the first operand register after the branch address has been computed. When the R2 value specifies R0, no branch is taken. Bit 0 of the branch address is used to set the new addressing mode (0=24 bit mode, 1=31).
The IEBCOPY Utility

IEBCOPY is used to copy all or part of a Partitioned Data Set (PDS). Selected members of a PDS can be copied to another or the same PDS and/or renamed. A sequential backup copy of a PDS can be made. A PDS can be subsequently be refreshed from a sequential copy created by an unload. Finally, IEBCOPY is used to "compress" a PDS when all of its unused internal space has been exhausted. The compress operation reorganizes a PDS so that all previously unused space inside the PDS is reclaimed.

Sample JCL for IEBCOPY appears below; following the JCL is a discussion of each DD statement required by IEBCOPY, and what each is used for. An explanation of IEBCOPY control statements follows the JCL discussion.

Sample IEBCOPY JCL:
//JS10 EXEC PGM=IEBCOPY,REGION=1024K,
//      PARM='SIZE=nnnnnnnnK' Optional PARM
//SYSPRINT DD SYSOUT=* IEBCOPY Messages

Figure 4 - IEBCOPY Utility Description

CICS/MVS & CICS/ESA Reference Information

MVS/QuickRef also provides a complete explanation for all of the messages and transaction abend codes issued internally by CICS. These are the messages and abend codes that CICS application programmers encounter and solve frequently. Figure 5 is the descriptive information for the ASRA abend code.
MVS/QuickRef also includes a complete description of all the status codes, user abends, and messages issued by IMS during its processing. Figure 6 is an example of the information supplied by MVS/QuickRef for the IMS “AC” status code.
IMS/VS Status Codes

Status Code: AC
Description: A ISRT or get call contains a hierarchy error. One of the following errors has occurred:

- two SSA's were specified for the same hierarchic level
- a segment name exists in the DB PCB, and the SSA that specifies the same segment name is not in the correct hierarchic sequence
- DL/I could not locate a segment in the DB PCB that matched the SSA segment name
- A STAT call was issued with an invalid statistics function requested

User Action: The responsible programmer should correct the call so

The INDEX Function
(TSO/E Version 2 or above only)

The INDEX function is used to find the position of one character string within another character string. The format of the INDEX function is:

```
INDEX(string,substring{,start})
```

where 'string' is the string to be searched, and 'substring' is the string to search for within 'string'. If 'substring' is not found, INDEX returns a 0. If 'substring' is found, INDEX returns the position, relative to 1, of the first character of 'substring' within 'string'. 'start' is an optional starting character position for the search within 'string' - the default for 'start' is 1. 'start' must be a positive integer.

INDEX usage examples:

- `INDEX('hello','ll')` returns a 3
- `INDEX('say what','w ')` returns a 0
- `INDEX('zyxwvu','vu',6)` returns a 0
- `INDEX('zyxwvu','vu',2)` returns a 5

Figure 6 - IMS AC Status Code Description

Figure 7 - REXX INDEX Function Description
REXX Language Syntax

The MVS/QuickRef data base provides a complete description of the syntax and function of all of the elements of the REXX language. Figure 7 is part of the description of the REXX INDEX function.

COBOL, PL/1, C++, and C Language Syntax

MVS/QuickRef also includes a complete description of the COBOL, PL/1, C++, and C programming languages. Figure 8 is an example of the description for the COBOL “PICTURE” Clause.

---

**ITEM ==>**                         * MVS/QuickRef *                     **Col 1 Line 1 of 89**
**Command ==>**                                              **Scroll ==>** PAGE
You may scroll the information below UP, DOWN, LEFT, and/or RIGHT as needed
----------------------------------------------------------------------------------

*** PICTURE CLAUSE ***

The PICTURE clause is an optional clause which can be specified as part of a data description entry. It specifies the general characteristics and editing requirements for an elementary data item.

The PICTURE clause is formatted as shown below:

    PICTURE/PIC {IS} character-string

The PICTURE clause must be specified for every elementary item other than an index item and the subject of a RENAMES clause. It may not be specified for a group item.

The PICTURE character-string is made up of certain COBOL characters used as symbols. Up to 30 of these symbols may be used within a single PICTURE character-string. The symbols actually used within a given PICTURE character-string determine the category of the elementary data item being described.

---

Figure 8 - COBOL PICTURE Clause Description

DB2 Messages and Codes

DB2 messages and SQL code descriptions are also provided. This includes a description of each of the messages issued by DB2 and its various attachments, as well as an explanation for all of the SQL return codes and system state codes passed back to application programs that issue SQL requests. Complete descriptions for all of the DB2 error reason codes are also provided. Figure 9 is an example of the reference information for the +558 SQL return code.
Independent Software Vendor Product Messages

The MVS/QuickRef data base also contains descriptions for messages issued by many different software products from most of the major Independent Software Vendors (ISVs). Through a special arrangement with each ISV, message descriptions for many of their products are provided to Chicago-Soft for inclusion in the MVS/QuickRef data base.
ISVs with product messages in the MVS/QuickRef data base include:

- 21ST CENTURY
- ACTION SOFTWARE
- ALLEN SYSTEMS
- AMERICAN SOFT.
- ASPG
- ASTRAC
- B I MOYLE
- BETA SYSTEMS
- BMC
- COMPUTER ASSOCIATES
- CDB SOFTWARE
- CHANEY SYSTEMS
- CHICAGO INTERFACE GROUP
- CHICAGO-SOFT
- CINCOM
- COMPUTER CORP. OF AMERICA
- COMPUTWARE
- CYBERMATION INC
- DATA21
- DILIGENT
- DINO-SOFTWARE
- DTS SOFTWARE
- E-NET
- EMC
- GT SOFTWARE
- IBM
- INFORMATICA
- INNOVATION
- LEVI, RAY & SHOUP
- MACKINNEY SYSTEMS
- MACRO4 PLC
- MAINSTAR
- MANTISSA
- MICRO FOCUS
- MVS SOLUTIONS
- NCR
- NEON SYSTEMS
- NESI
- NETEC
- OPEN SOFTWARE
- OPSERVER
- PASSGO TECHNOLOGIES
- PHOENIX SOFTWARE
- PKWARE
- PRINCETON SOFTECH
- PROGINET
- ROCKET SOFTWARE
- RSD S.A.
- SEA
- SERENA
- SMA
- SOFTBASE
- SOFTEK STORAGE
- SOFTWARE AG
- STERLING COMM.
- STONEBRANCH
- STORAGE TEC.
- SYNCSORT
- SYSTEMWARE
- TACT
- TONE SOFTWARE
- TOP DOWN SYSTEMS
- TRIDENT SRVCS
- UNICOM SYSTEMS
- VANGUARD
- WILLIAMS DATA
Chapter 2 - Accessing and Using MVS/QuickRef
Introduction

This chapter provides a brief description of how to access and use MVS/QuickRef. Its purpose is to give you a quick introduction to MVS/QuickRef invocation and access techniques. It contains just enough information to "get you started" with MVS/QuickRef.

For a more complete, detailed description of MVS/QuickRef, you should review the information provided by the MVS/QuickRef on-line help facility. The help facility provides a complete on-line tutorial for MVS/QuickRef. In addition to comprehensive descriptions of MVS/QuickRef processing concepts, functions, and facilities, the on-line tutorial also provides many detailed examples showing how MVS/QuickRef and its primary features can be used. For this reason, you may prefer to review the more comprehensive information in the tutorial before attempting to use MVS/QuickRef. If this is the case, review the information in the sections titled "Invocation Commands" and "Using the Help Facility" later in this chapter.

If you prefer the "quick-start" method, continue reading the information which follows. But remember to go back and review the on-line tutorial at a later time. This is the only way to get a comprehensive understanding of MVS/QuickRef and the only way to ensure that you are using it in an effective manner.

Information Storage and Retrieval Overview

The information in the MVS/QuickRef data base is divided into products which are further divided into discrete items of information which can be individually retrieved and displayed. A data base product is identified by a vendor name, a product name, and a release number.

To make access easier, some products in the data base are grouped into categories through the assignment of optional product category codes. For example, PROGLANG is the product category code assigned to products providing information on programming languages.

An individual item in the data base generally contains information about a single error message, abend code, JCL parm, etc. and will usually be assigned an item name that reflects the type of information stored in that item. For example, item IDC3009I describes system error message IDC3009I; item S0C4 describes system abend code 0C4; etc.

You access an individual item in the data base by specifying the item name and, optionally, the vendor name, product name, and/or the release number of the product containing that item. You access the products assigned to a product category by specifying the associated product category code. You can list the vendors and products stored in the data base by specifying the associated vendor name, product name, and/or the release number.

The vendor name, product name, release number, and item name together comprise the 'key' for a particular data base item. When accessing a data base item, the specified "key" can be a full key or a generic key (i.e., a key prefix followed by an asterisk). For example, IDC3009I is a full
item name; IDC* is a generic item name and would match all item names that start with the characters 'IDC'. IBM is an example of a full vendor name; IB* is an example of a generic vendor name and would match all vendor names that start with the characters 'IB'.

If a key value is specified as a single asterisk, then it is considered to match all key values of the same type. If an applicable key value is unspecified, then it is treated as if it were specified as a single asterisk and is considered to match all key values of the same type. For example, if the product name is specified as a single asterisk, then all products in the data base will be considered to match the "specified" product name. If the vendor name is unspecified, then all vendors in the data base will be considered to match the "unspecified" vendor name.

If the specified keys match a single element in the data base, then the information related to that single element is displayed. If the specified keys match more than one element in the data base, then a list of all matching elements is displayed. For example, if you specify item name IDC*, a list of all item names that start with the characters 'IDC' will be shown. If you specify item name PROC* and product name C*, a list of all item names that start with the characters 'PROC' in all products with product names that start with the character 'C' will be shown. If you request a product list and specify a vendor name of IBM, a list of all IBM products will be shown.

Such a list of "matching elements" is referred to as a selection list because you can select any individual element on the list in order to request a display of the information related to that element. For example, you can select an individual item from an item selection list to request a display of the reference information provided by that item. You can select a product category code from a product category code selection list to request a display of the products associated with that product category code. You can select a vendor from a vendor selection list to request a display of the products belonging to that vendor. You can select a product from a product selection list to request a display of the items associated with that product.

If you request information on a single item in the data base and that item appears in more than one release of a given product, then you will generally be presented with an item selection list showing all releases of the product that contains the requested item. However, through a facility referred to as user-directed selection list processing, you (or your installation) can cause the appropriate release to be automatically selected for you. For more information, see "User-Directed Selection List Processing" in Chapter Three.

If your installation has one or more user data bases defined, then, when requesting an information display, you can also specify the data base to be searched. You specify the data base to be searched using a data base indicator. The letter M is used as the data base indicator for the main data base; a single digit from one to nine is used for each of the (up to nine) user data bases. If your installation has one or more user data bases defined and you do not specify the data base to be searched, then the main data base and all defined user data bases will be searched for the requested information.
Invocation Commands

Depending upon how it is installed, there are two basic ISPF commands which can be used to invoke MVS/QuickRef:

**QW** and **QWS**

The QW command invokes MVS/QuickRef as a "pop-up" application. As such, it "pops-up", in the same logical screen, over the top of the current ISPF application and panel, and, when it terminates, the current ISPF application and panel are restored.

The QWS command invokes MVS/QuickRef after creating a new logical screen split. MVS/QuickRef will then execute in the new logical screen split and, when it terminates, the new logical screen split will be eliminated.

For either command, you simply type the command in the command line of any ISPF panel currently on display and press the ENTER key. MVS/QuickRef will be immediately invoked. You also have the option of equating either command with a PF key and then using that PF key to invoke MVS/QuickRef.

The only difference in the two commands is the way in which MVS/QuickRef is invoked (as a "pop-up" application or in a new logical screen split). Otherwise, the two commands are functionally equivalent and you can use them interchangeably.

Throughout this user guide, the QW command is used almost exclusively in all descriptions and examples. However, since you can use the two commands interchangeably, you can substitute the QWS command anyplace the QW command is shown.

Note: Depending upon how MVS/QuickRef is installed at your site, your installation may support only the QW command or only the QWS command. Of course, it may support both commands. In this case, you have the option of using either command.

Invocation Techniques

There are three basic techniques which you can use to invoke MVS/QuickRef:

- **Menu-driven invocation:** uses the QW command to provide a series of menu and display request panels which you can use to indicate the type of reference information which you need

- **Fast-path invocation:** uses the QW command to go directly to a display of the requested information (bypassing the menus and display request panels) by allowing the specification of a parameter which indicates the type of information to be displayed

- **Cursor-driven invocation:** uses the QW command, when you place the cursor underneath any character string shown on any ISPF panel, to do an immediate search for the item name represented by the indicated character string
As you can see from the above, the QW command is used for all three invocation techniques. The way in which MVS/QuickRef determines which invocation technique is being used is as follows:

- if the QW command is used with an accompanying parameter, then the fast-path invocation technique is used
- if the QW command is used without an accompanying parameter and the cursor is positioned underneath a blank character, then the menu-driven invocation technique is used
- if the QW command is used without an accompanying parameter and the cursor is positioned underneath a non-blank character, then the cursor-driven invocation technique is used

### Menu-Driven Invocation

If the QW command is used without an accompanying parameter and the cursor is positioned underneath a blank character, then the menu-driven invocation technique is used. In this case, the MVS/QuickRef main menu panel, shown in Figure 10 will be immediately displayed.

```
* MVS/QuickRef - Main Menu *
Command ==> 

Please enter one of the options listed below:
   C - Request Reference Information by Category
   R - Request Reference Information by Name
   L - List Vendors, Products, and Releases
   S - Request DASD Free Space Information
   ? - What's New with MVS/QuickRef?
   X - Exit MVS/QuickRef
```

**Figure 10 - MVS/QuickRef Main Menu**

Option 'C' displays a product category code selection list showing all defined product category codes.

Option 'R' brings up the Request Reference Information panel. This display request panel allows you to request an item selection list or the display of the reference information for an individual item.

Option 'L' brings up the List Vendors/Products/Releases panel. This display request panel allows you to request a vendor or product selection list.
Option 'S' displays the Request DASD Free Space Information panel. This display request panel allows you to request a DASD free space display.

Each of the display request panels described above will prompt you for the key-values that you need to enter for that specific type of information request.

Option '?' provides a display showing what's new in the current release of MVS/QuickRef.

Option 'X' causes MVS/QuickRef to terminate.

**Fast-Path Invocation**

Fast-path invocation uses the QW command to go directly to a display of the requested information (bypassing the main menu and the display request panels) by allowing the specification of a parameter which indicates the type of information to be displayed. The parameter specified along with the QW command is referred to as a fast-path string.

The format of the fast-path string depends on the type of information being requested.

To request an item selection list or the display of an individual item, the QW command and fast-path string can be formatted as follows:

```
QW I=iii
```

where iii represents the item name and I= is optional.

For example:

```
QW I=IEC145I   would display information on system message id IEC145I
QW S0C7        would display information on system abend code 0C7
QW I=JOB       would display information on the JCL JOB statement
QW IEBCOPY     would display information on the IEBCOPY utility
QW MOVE        would display information on the COBOL MOVE statement
QW I=IEC*      would list all item names beginning with the characters 'IEC'
QW P*          would list all item names beginning with the character 'P'
```

You can restrict the list of matching item names with the additional fast-path string components:

```
V=vvv P=ppp R=rrr D=d
```

where vvv represents the vendor name

ppp represents the product name

rrr represents the release number

d represents the data base, with M for the main data base and 1-9 for a user data base

These additional fast-path string components are all optional and can be specified in any order.
For example:

QW I=PROC* V=IBM would list all item names starting with the characters 'PROC' in all products belonging to IBM

QW V=ALLEN* P=JCLP* I=* would list all items associated with product names that start with 'JCLP' and vendor names that start with 'ALLEN' (like JCLPREP from Allen Systems Group)

QW P=M* D=1 would list all items in user data base #1 in products with product names that start with the character 'M'

To request a vendor or product selection list, the fast-path string is formatted as:

V=vvv P=ppp R=rrr D=d L=l

where the V=, P=, R=, and D= components are as described above and L= must be specified as:

V for a list of products in vendor sequence
P for a list of products in product sequence
O for a list of vendors only

For example:

QW L=V would list all products (in vendor name sequence)
QW L=P would list all products (in product name sequence)
QW L=O would list all vendors
QW V=CA L=P would list all products belonging to vendor CA
QW P=C* L=V would list all product names starting with character 'C'
QW V=I* L=O would list all vendor names starting with character 'I'
QW L=O D=2 would list all vendor names in user data base #2

To request a product category code display, the fast-path string is formatted as:

C=ccc D=d

where ccc represents the product category code and d represents the data base.

For example:

QW C=PROGLANG would display the list of products associated with product category code PROGLANG

QW C=* would display a list of all defined product category codes
To request a DASD free space display, the fast-path string is formatted as:

    S=sss

where sss represents a full or generic DASD volume serial.

For example:

    QW S=SCR101   would produce a free space display for volume SCR101
    QW S=SCR*     would display all volume serials starting with 'SCR'

The volume serial can also be specified as:

- a generic unit name like '3380'
- an esoteric unit name like 'SYSDA'
- an SMS storage group name
- 'PRIVATE', 'PUBLIC', or 'STORAGE' to display only DASD volumes with those specific mount attributes
- 'SMVOLS' to display only DASD volumes under SMS management
- 'NONSMS' to display only DASD volumes not under SMS management

For example:

    QW S=3390         would display all 3390 volumes
    QW S=PUBLIC   would display all public volumes

**Cursor-Driven Invocation**

If the QW command is used without an accompanying parameter and the cursor is positioned underneath a non-blank character, then the cursor-driven invocation technique is used. In this case, MVS/QuickRef treats the character string indicated by the position of the cursor as an item name and does an immediate search for that item name.

With cursor-driven invocation, you can place the cursor under any non-blank character string shown on any ISPF panel. The cursor can be placed under any non-blank character in the string, from the first character in the string to the last character in the string. If you have a PF key equated to QW, then all you have to do is place the cursor underneath the desired item name and press the appropriate PF key. This makes cursor-driven invocation very convenient to use.

As an example of cursor-driven invocation, suppose you are browsing the output from a batch job using the ISPF OUTLIST facility, PDF option 3.8 and that, within this ISPF application, you have equated PF key 21 to 'QW'. If a portion of the job log for the batch job should appear as shown below:
then you could invoke MVS/QuickRef and get an immediate display of the reference information available for system error message IEC149I by placing the cursor underneath the IEC149I message id, as shown below, and pressing PF key 21.

Figure 11 - ISPF OUTLIST Display

Figure 12 - Cursor-Driven Invocation
Terminating/Exiting MVS/QuickRef

You can terminate and exit the current invocation of MVS/QuickRef, no matter how it was invoked, and no matter which MVS/QuickRef panel is on display, by using the RETURN command (or a PF key equated to 'RETURN').

You can also terminate and exit the current invocation of MVS/QuickRef, when the main menu is on display:
- by using the END command
- by entering option 'X' on the main menu

Restoring Previous Display

The END command (or a PF key equated to 'END') generally restores the previous MVS/QuickRef display. For example, from any one of the display request panels that are accessed from the main menu, the END command will take you back to the main menu. From the display which results when a selection is made from a selection list, the END command will return you to the display of that selection list.

Using The Help Facility

Help for MVS/QuickRef is requested by specifying the HELP command, which can be specified on any MVS/QuickRef panel.

When specified, the HELP command displays a list of all the "help" items which are available for review. You can select any item on the list produced by the HELP command in order to review the help information provided by that item. Select the item named USING-HELP and it will provide a complete explanation of how to use the help facility.
Chapter 3 - MVS/QuickRef Features
Introduction

In addition to the invocation and information retrieval facilities described in Chapter Two, MVS/QuickRef provides a number of other important features.

Additional On-Line Features

MVS/QuickRef provides a number of additional features when executing in the ISPF environment. These include:

- A FIND command, which allows you to find a specified character string in some displayed reference information
- A FINDCODE command, which facilitates finding numeric codes in a list format (like lists of return codes, lists of reason codes, lists of error codes, etc.)
- A SEARCH command, which allows you to search for a specified character string over some specified portion of the entire data base
- A "cut and paste" facility, which allows you to "cut" some displayed reference information and then "paste" it into an edited data set
- A textmark facility, which allows you set, list, delete, and redisplay textmarks, which are pointers to "marked" sections of reference text
- Flexible printing capabilities, which allow you to print selected reference information or to place it in a specified output file
- GETNEXT and GETPREV commands, which allow you to retrieve certain related types of information in "content" sequence
- A SORT command, which allows you to sort DASD free space displays on-line
- A QINFO command, which allows you to display certain information about your MVS/QuickRef installation
- Recursive invocation, which allows MVS/QuickRef to be invoked "on top of itself"

These features are fully described in the information provided by the MVS/QuickRef on-line help facility. For information on using the help facility, see the section titled "Using The Help Facility" in Chapter Two.
User-Directed Selection List Processing

A request to retrieve the reference text for a given item often results in a selection list showing that the requested item is carried in two or more releases of the same product. User-directed selection list processing is a facility that allows installations and/or individual users to indicate the "preferred" release for a given product in the MVS/QuickRef data base. With this facility, MVS/QuickRef can automatically select and display the preferred release from an item selection list.

As an example of user-directed selection list processing, suppose you attempt to lookup the S214 system abend code with a fast-path string like QW S214. Without user-directed selection list processing (and depending upon the exact content of the main data base you are using), MVS/QuickRef might display an item selection list like that shown in Figure 13.

Note: The item selection lists shown throughout this section on “User-Directed Selection List Processing” are examples only. The exact content of the item selection lists which you will actually see depend on the release of MVS/QuickRef you are running and the exact content of your main data base. Later releases of MVS/QuickRef, for example, will contain later releases of the products associated with the OS/390 and Z/OS operating systems.

<table>
<thead>
<tr>
<th>Item</th>
<th>Vendor</th>
<th>Product</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>_S214</td>
<td>IBM</td>
<td>OS/390 SYSTEM CODES</td>
<td>V1R3</td>
</tr>
<tr>
<td>_S214</td>
<td>IBM</td>
<td>OS/390 SYSTEM CODES</td>
<td>V2R4</td>
</tr>
<tr>
<td>_S214</td>
<td>IBM</td>
<td>OS/390 SYSTEM CODES</td>
<td>V2R5</td>
</tr>
<tr>
<td>_S214</td>
<td>IBM</td>
<td>SYSTEM CODES</td>
<td>V5R2M2 &amp; PRIOR</td>
</tr>
</tbody>
</table>

Figure 13 - S214 Item Selection List Without User-Directed Selection List Processing

In this case, the main data base has four products which contain an item named S214 and, in order to select the appropriate one, you have to know which product and release to select. If your installation is running OS/390 V2R4, then you would probably want to select the second item on the list above.

With user-directed selection list processing, the "preferred" release for a given product can be predefined to MVS/QuickRef and, once the "preferred" release for a given product is defined, MVS/QuickRef will automatically select and display the reference text for the item associated with that preferred release. Suppose for example, that V1R3 were defined as the preferred release for products containing information on "system codes". Then, instead of the item selection list shown in Figure 13, MVS/QuickRef would automatically select and display the reference text for item S214 associated with OS/390 SYSTEM CODES V1R3.
General Rules

User-directed selection list processing is controlled by the following general rules:

- User-directed selection list processing only applies to item selection lists. It does not apply to vendor selection lists, product selection lists, or product category selection lists.

- User-directed selection list processing only applies when the requested item name is a full, non-generic item name. If you specify a generic item name, like S214*, then user-directed selection list processing, even if defined for your processing environment, will not be applied. This provides an easy way to override any user-directed selection list processing that may be defined for your processing environment. So, if you really want to see every release of every product containing item name S214, use a fast-path string like QW S214*.

- As long as the requested item name is non-generic, user-directed selection list processing, if defined for your processing environment, will be applied even if other key operands are specified as either generic or non-generic. For example,

  QW S214  QW S214 V=IBM  QW S214 V=IB* D=M

  are all fast-path strings to which user-directed selection list processing could be applied.

- Since cursor-driven invocation results in a non-generic item name request, user-directed selection list processing, if defined for your processing environment, will always apply to cursor-driven invocation. The only exception to this rule occurs when the cursor is placed underneath a character string ending with an asterisk.

- User-directed selection list processing applies only to base products, where a base product is defined as any product which has two or more releases stored in the data base. As an example, suppose the data base contains the three releases of product CA-1 from vendor CA shown in Figure 14.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>CA-1</td>
<td>V5R1</td>
</tr>
<tr>
<td>CA</td>
<td>CA-1</td>
<td>V5R2</td>
</tr>
<tr>
<td>CA</td>
<td>CA-1</td>
<td>V5R3</td>
</tr>
</tbody>
</table>

  Figure 14 - CA-1 Product List

In this case, then, CA-1 is a base product with three releases in the data base.

For IBM products in the main data base, a base product is further defined as a product with the same product name, with or without the "OS/390" or “Z/OS” prefixes which are applied
to some IBM products. As shown in Figure 13 on page 33, SYSTEM CODES is also a base product. In this case, the base product SYSTEM CODES actually has four releases in the data base:

- OS/390 SYSTEM CODES   V1R3
- OS/390 SYSTEM CODES   V2R4
- OS/390 SYSTEM CODES   V2R5
- SYSTEM CODES                V5R2M2 & PRIOR

In some cases, an item selection list may contain more than one base product. In other cases, an item selection list may contain single releases of one or more other products along with one or more base products. In these situations, user-directed selection list processing, if defined for your environment, will keep the preferred release of each base product on the list while removing all other base product releases from the list. As an example, without user-directed selection list processing, a fast-path string like QW CD might produce an item selection list like that shown in Figure 15.

<table>
<thead>
<tr>
<th>Item</th>
<th>Vendor</th>
<th>Product</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD</td>
<td>CHICAGO-SOFT</td>
<td>GLOSSARY</td>
<td>V1R1M0</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>IMS/ESA MESSAGES</td>
<td>V5R1 &amp; PRIOR</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>MVS COMMANDS</td>
<td>V5R2M2 &amp; PRIOR</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>OS/390 MVS COMMANDS</td>
<td>V1R3</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>OS/390 MVS COMMANDS</td>
<td>V2R4</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>OS/390 MVS COMMANDS</td>
<td>V2R5</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>OS/390 UNIX(R) SYNTX</td>
<td>V2R4</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>UNIX(R) SYNTX</td>
<td>V5R2M2 &amp; PRIOR</td>
</tr>
</tbody>
</table>

Figure 15 - CD Item Selection List Without User-Directed Selection List Processing

In this case, MVS COMMANDS and UNIX(R) SYNTX are both base products. GLOSSARY and IMS/ESA MESSAGES are single releases of other products. If user-directed selection list processing was used to define V2R4 as the preferred release for base product MVS COMMANDS as well as for base product UNIX(R) SYNTX, then the other releases of these two base products would be removed from the list, as shown in Figure 16.

<table>
<thead>
<tr>
<th>Item</th>
<th>Vendor</th>
<th>Product</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD</td>
<td>CHICAGO-SOFT</td>
<td>GLOSSARY</td>
<td>V1R1M0</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>IMS/ESA MESSAGES</td>
<td>V5R1 &amp; PRIOR</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>OS/390 MVS COMMANDS</td>
<td>V2R4</td>
</tr>
<tr>
<td>CD</td>
<td>IBM</td>
<td>OS/390 UNIX(R) SYNTX</td>
<td>V2R4</td>
</tr>
</tbody>
</table>

Figure 16 - CD Item Selection List With User-Directed Selection List Processing
User-directed selection list processing, if defined for your processing environment, is applied only after the key fields you have specified are used to determine the products that would normally be shown on the item selection list. In the example illustrated in Figure 15, if you specified a fast-path string like QW CD V=IBM, then the CHICAGO-SOFT GLOSSARY product would not show up on the resulting item selection list, either before or after user-directed selection list processing was applied. The specification of additional key fields can also override user-directed selection list processing. In the example illustrated in Figure 15, if you specified a fast-path string like QW CD P=UNIX*, then only one item in the database would match your request - the item named CD associated with IBM UNIX(R) SYNTX V5R2M2 & PRIOR. In this case, the reference text for this specific item would be shown and user-directed selection list processing would not come into play.

Types Of User-Directed Selection List Processing

There are three types of user-directed selection list processing:

♦ operating system level selection list processing
♦ vendor/product specific selection list processing
♦ user specific selection list processing

Each of these is described in the sections which follow.

Operating System Level Selection List Processing

Operating system level selection list processing is the first of the three types of user-directed selection list processing. It is controlled by the QWIKOPTS operating system level selection list (OSLSLO=) option. QWIKOPTS is an options table that is assembled and linked into the MVS/QuickRef load library by the product installer. For more information on QWIKOPTS, see the section on "Setting MVS/QuickRef Global Installation Options" on page 97 in Chapter Four.

The QWIKOPTS OSLSLO= option may be set to Y (for Yes), N (for No), or H (for High). If set to 'Yes' (OSLSLO=Y), which is the default setting, operating system level selection list processing will be in effect. If set to 'No' (OSLSLO=N), operating system level selection list processing will not be used. The 'High' (OSLSLO=H) setting is designed to be used when your installation is running on a release of the operating system which is more current than the most current release of the operating system covered in the MVS/QuickRef main data base. The Y (Yes) and H (High) settings are more fully described in the paragraphs which follow.

You can determine how the QWIKOPTS OSLSLO= option is set for your installation by using the QINFO command and paging down until you see the OSLSLO= setting.
Operating system level selection list processing only applies to base products in the main data base which are associated with vendor IBM. Furthermore, at least one of the releases of the base product must have a product name which contains the "OS/390" prefix or the “Z/OS” prefix. The "OS/390" and “Z/OS” prefixes generally identify products that are "operating system specific"; in other words, that are updated with each new release of the operating system. As shown in Figure 13 on page 33, SYSTEM CODES is a base product which is "operating system specific".

When the QWIKOPTS operating system level selection list option is set to Yes (OSLSLO=Y), then operating system level selection list processing will automatically select the release of a given base product which "best" matches the release of the operating system which your installation is currently running (or, more specifically, the release of the operating system on which MVS/QuickRef is currently executing).

The "best" matching release is determined according to the following rules, which are applied in the order shown:

- when the release number of a given base product matches the release level of the operating system in use, that release is automatically selected

- when the release of the operating system in use is more recent than one or more releases of the base product, the most current release of the base product prior to the release of the operating system in use is automatically selected

- so long as at least one base product release is left on the selection list, all releases of the base product more current than the release of the operating system in use are eliminated

As an example, consider the item selection list shown in Figure 13 on page 33 the QWIKOPTS operating system level selection list option is set to Yes (OSLSLO=Y) and MVS/QuickRef is running in an OS/390 V2R4 environment, then reference text for the item named S214 associated with OS/390 SYSTEM CODES V2R4 would be automatically selected and shown in place of the selection list in Figure 13 on page 33.
As another example, suppose the QWIKOPTS operating system level selection list option is set to Yes (OSLSLO=Y) and MVS/QuickRef is running in an OS/390 V1R1 environment. Then, since reference text for OS/390 V1R1 and OS/390 V1R2 is generally carried in the MVS/QuickRef main data base in those IBM products with release number V5R2M2 & PRIOR, reference text for the item named S214 associated with SYSTEM CODES V5R2M2 & PRIOR would be automatically selected and shown in place of the selection list in Figure 13 on page 33.

As another example, suppose a fast-path string like QW $A_A were specified. Without user-directed selection list processing, this might produce an item selection list like that shown in Figure 17. Notice that, since IBM did not update JES2 COMMANDS for OS/390 V2R5, there is no entry for OS/390 JES2 COMMANDS V2R5.

<table>
<thead>
<tr>
<th>Item</th>
<th>Vendor</th>
<th>Product</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_A</td>
<td>IBM</td>
<td>JES2 COMMANDS</td>
<td>V5R2M2 &amp; PRIOR</td>
</tr>
<tr>
<td>$A_A</td>
<td>IBM</td>
<td>OS/390 JES2 COMMANDS</td>
<td>V1R3</td>
</tr>
<tr>
<td>$A_A</td>
<td>IBM</td>
<td>OS/390 JES2 COMMANDS</td>
<td>V2R4</td>
</tr>
</tbody>
</table>

Figure 17 - $A_A Item Selection List Without User-Directed Selection List Processing

Now suppose the QWIKOPTS operating system level selection list option is set to Yes (OSLSLO=Y) and MVS/QuickRef is running in an OS/390 V2R5 environment. Since OS/390 JES2 COMMANDS V2R4 is the most current release of the base product prior to the release of the operating system, reference text for the item named $A_A associated with OS/390 JES2 COMMANDS V2R4 would be automatically selected and shown in place of the selection list in Figure 17.

As stated earlier, the QWIKOPTS operating system level selection list option High setting (OSLSLO=H) is designed to be used when your installation is running on a release of the operating system which is more current than the most current release of the operating system covered in the main data base. When the QWIKOPTS operating system level selection list option is set to High (OSLSLO=H), operating system level selection list processing will automatically select the most current release available for a given base product.

As an example, consider the item selection list shown in Figure 13 on page 33. In this case, the most current release of the operating system covered in the main data base is OS/390 V2R5. Now suppose you are running in an OS/390 V2R6 environment and the QWIKOPTS operating system level selection list option is set to High (OSLSLO=H). In this case, reference text for the item named S214 associated with OS/390 SYSTEM CODES V2R5 would be automatically selected and shown in place of the selection list in Figure 13 on page 33.
Vendor/Product Specific Selection List Processing

Vendor/product specific selection list processing is the second of the three types of user-directed selection list processing. It is controlled by the QWIKOPTS vendor/product specific selection list (VPSSLO=) option. QWIKOPTS is an options table that is assembled and linked into the MVS/QuickRef load library by the product installer. For more information on QWIKOPTS, see the section on "Setting MVS/QuickRef Global Installation Options" in Chapter Four.

The QWIKOPTS VPSSLO= option may be set to Y (for Yes) or N (for No). If set to Yes (VPSSLO=Y), vendor/product specific selection list processing will be in effect. If set to No (VPSSLO=N), which is the default setting, vendor/product specific selection list processing will not be used.

Before setting the QWIKOPTS vendor/product specific selection list option to Yes (OSLSLO=Y), your product installer will customize, assemble, and link a load module called the Vendor/Product Specific Selection List Table (QWIKVPST). This table will indicate the preferred release for each base product which your product installer chooses to add to the table.

When the QWIKOPTS vendor/product specific selection list option is set to Yes (VPSSLO=Y), then, when a base product in the Vendor/Product Specific Selection List Table appears in an item selection list, vendor/product specific selection list processing will automatically select the release of that base product that matches the preferred release specified in the Vendor/Product Specific Selection List Table.

You can determine how the QWIKOPTS VPSSLO= option is set for your installation by using the QINFO command and paging down until you see the VPSSLO= setting. If the VPSSLO= option is set to Yes (VPSSLO=Y), then, on the same QINFO display, you can continue to page down until you see the entry which says "Vendor/Product Specific Selection List Table follows". This will show you the preferred releases for the base products in your installations Vendor/Product Specific Selection List Table.

As an example, consider the QINFO display in Figure 18. As shown in Figure 18, the Vendor/Product Specific Selection List Table specifies, for a given base product, the data base containing that product (with M for the main data base, 1-9 for one of the user data bases), the associated vendor name, the associated product name, and the preferred release number. As indicated by the presence of the data base indicator in the table, vendor/product specific selection list processing can be applied to products in a user data base as well as to products in the main data base.
In the example shown in Figure 18, the preferred release for base product CA-1 in the main data base is V5R1. Now suppose the main data base contains the three releases of CA-1 shown in Figure 14 on page 34. If you specified a fast-path string like QW CTS001 then, without user-directed selection list processing, you would get an item selection list like that shown in Figure 19.

<table>
<thead>
<tr>
<th>Item</th>
<th>Vendor</th>
<th>Product</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>_CTSO01</td>
<td>CA</td>
<td>CA-1</td>
<td>V5R1</td>
</tr>
<tr>
<td>_CTSO01</td>
<td>CA</td>
<td>CA-1</td>
<td>V5R2</td>
</tr>
<tr>
<td>_CTSO01</td>
<td>CA</td>
<td>CA-1</td>
<td>V5R3</td>
</tr>
</tbody>
</table>

Figure 19 - CTS001 Item List Without User-Directed Selection List Processing

Now suppose the QWIKOPTS vendor/product specific selection list option is set to Yes (VPSSLO=Y) and the Vendor/Product Specific Selection List Table is setup as shown in Figure 18. Then reference text for the item named CTS001 associated with CA-1 V5R1 would be automatically selected and shown in place of the selection list in Figure 19.

In the example shown in Figure 18, the preferred release for base product JES2 COMMANDS in the main data base is V1R3. As stated earlier, for IBM products in the main data base, a base product is defined as a product with the same product name, with or without the "OS/390" or "Z/OS" prefix. Now suppose the QWIKOPTS vendor/product specific selection list option is set to Yes (VPSSLO=Y) and the Vendor/Product Specific Selection List Table is setup as shown in Figure 18. If you specify a fast-path string like QW $A_A, then, instead of the item selection list shown in Figure 17 on page 38, vendor/product specific selection list processing would automatically select and display the reference text for item $A_A associated with OS/390 JES2 COMMANDS V1R3.

In the example shown in Figure 18, the preferred release for base product MACROS from vendor RAMAC in user data base number one is R5. Now suppose there are two releases of
RAMAC MACROS in user data base number one (R5 and R6), that a macro named ADDMAC appears in both releases of this base product, that the QWIKOPTS vendor/product specific selection list option is set to Yes (VPSSLO=Y), and that the Vendor/Product Specific Selection List Table is setup as shown in Figure 18. If you specify a fast-path string like QW ADDMAC D=1, then, instead of an item selection list showing both releases of the RAMAC MACROS in user data base number one, vendor/product specific selection list processing would automatically select item ADDMAC associated with RAMAC MACROS R5.

The entries in the Vendor/Product Specific Selection Table are processed in the order they appear in the table. If the same base product appears in the table more than one time, the first entry will dictate the preferred release to be used and the second and all subsequent entries for that base product will be ignored.

**User Specific Selection List Processing**

User specific selection list processing is the third type of user-directed selection list processing. The other two types of user-directed selection list processing, operating system level selection list processing and vendor/product specific selection list processing, are both "installation-wide"; when active, they affect every user at your installation. User specific selection list processing, on the other hand, allows you to setup your own personal selection list processing. This type of user-directed selection list processing affects only you (or, more specifically, your TSO user id); it will not affect any other TSO user id at your installation.

User specific selection list processing is controlled by your Selection List Profile. You setup and control your Selection List Profile using the SLPROFILE command, which can be abbreviated as SLP. Whenever the SLPROFILE (or SLP) command is specified, the Selection List Profile panel, shown in Figure 20, is displayed.
As shown in Figure 20, the first field following the command line on the Selection List Profile panel is the Operating System Level Selection List Option field. This field can be used to override the QWIKOPTS operating system level selection list (OSLSLO=) option. It can be entered with the same values (Y for Yes, N for No, and H for High) as used for the QWIKOPTS OSLSLO= option and these values have the same meanings as when used with the QWIKOPTS OSLSLO= option. Until you enter this field, it is shown with and defaults to the same value set for the QWIKOPTS OSLSLO= option. For more information on the QWIKOPTS OSLSLO= option and its values, see the previous section in this chapter on "Operating System Level Selection List Processing".

As an example of using the Selection List Profile panel Operating System Level Selection List Option field, suppose the QWIKOPTS operating system level selection list option is set to Yes (OSLSLO=Y). Now suppose you want to turn off operating system level selection list processing for yourself. Then you could enter an N in the Operating System Level Selection List Option field on the Selection List Profile panel, as shown in Figure 20.

The next field on the Selection List Profile panel is the Vendor/Product Specific Selection List Option field. This field can be used to override the QWIKOPTS vendor/product specific selection list (VPSSLO=) option. It can be entered with the same values (Y for Yes and N for No) as used for the QWIKOPTS VPSSLO= option and these values have the same meanings as when used with the QWIKOPTS VPSSLO= option. Until you enter this field, it is shown with and defaults to the same value set for the QWIKOPTS VPSSLO= option. For more information
on the QWIKOPTS VPSSLO= option and its values, see the previous section in this chapter on "Vendor/Product Specific Selection List Processing".

As an example of using the Selection List Profile panel Vendor/Product Specific Selection List Option field, suppose the QWIKOPTS vendor/product specific selection list option is set to Yes (VPSSLO=Y). If you have not previously entered the Selection List Profile panel Vendor/Product Specific Selection List Option field, then it will be shown with a value of Y (the same value as carried in QWIKOPTS). Now suppose you want to continue to have vendor/product specific selection list processing as specified in the Vendor/Product Specific Selection List Table. Then you could simply leave the Y in the Vendor/Product Specific Selection List Option field on the Selection List Profile panel, as shown in Figure 20.

The next field on the Selection List Profile panel is the User Specific Selection List Option field. This field can be entered with a Y (for Yes) or an N (for No). If entered as N (for No), user specific selection list processing will not be in effect. If entered as Y (for Yes), user specific selection list processing will be in effect. Until you enter this field, it is shown with and defaults to a value of N (for No).

As shown in Figure 20, the body of the Selection List Profile panel has room for you to specify preferred releases for up to 10 base products. For each base product, you specify the data base containing that product (with M for the main data base, 1-9 for one of the user data bases), the associated vendor name, the associated product name, and the preferred release number. As indicated by the presence of the data base indicator, user specific selection list processing can be applied to products in a user data base as well as to products in the main data base.

When the User Specific Selection List Option field is set to Y (for Yes), then, when a base product in the body of the Selection List Profile panel appears in an item selection list, user specific selection list processing will automatically select the release of that base product that matches the preferred release specified in the body of the Selection List Profile panel.

In the example shown in Figure 20, the preferred release for base product CA-1 in the main data base is V5R2. Now suppose the main data base contains the three releases of CA-1 shown in Figure 14 on page 34. If you specified a fast-path string like QW CTS001 then, without user-directed selection list processing, you would get an item selection list like that shown in Figure 19 on page 40. Now suppose the Selection List Profile panel User Specific Selection List Option field is set to Y (for Yes) and that the body of the Selection List Profile panel is setup as shown in Figure 20. Then reference text for the item named CTS001 associated with CA-1 V5R2 would be automatically selected and shown in place of the selection list in Figure 19 on page 40.

As shown by the comments at the bottom of the panel, any changes entered on the Selection List Profile panel can be saved "temporarily" or "permanently". If saved temporarily, the changes will be available only for the remainder of the current MVS/QuickRef session. If saved permanently, they will be saved in your ISPF application profile for MVS/QuickRef and will be available for the remainder of this MVS/QuickRef session and all future sessions.
To save changes entered on the Selection List Profile panel temporarily, simply press ENTER. The changes will then be available only for the remainder of the current MVS/QuickRef session. The next time you invoke MVS/QuickRef the changes will be discarded.

To save changes entered on the Selection List Profile panel permanently, type SAVE in the command line and press ENTER. The changes will then be available until the Selection List Profile panel is used again to assign new values.

If you want to discard any changes entered on the Selection List Profile panel, then, before pressing ENTER, type END or CANCEL in the command line.

As shown by the comment at the very bottom of the panel, you can delete any specific preferred release entry in the body of the panel by replacing the underbar in front of that entry with the letter D. If you want the entry temporarily deleted, press ENTER. If you want the entry permanently deleted, type SAVE in the command line before pressing ENTER.

Preferred release entries which have been deleted, or which have been left blank, are simply ignored. They have no effect on higher numbered entries. In other words, preferred release entry number six, if present, will be utilized even if preferred release entries 1 through 5 have been deleted or left blank.

If you want to turn off user specific selection list processing but do not want to discard your preferred release entries, simply set the User Specific Selection List Option field to N (for No). In this case, preferred release entries in the body of the panel will be ignored. The preferred release entries in the body of the Selection List Profile panel are processed in the order they appear in the panel. If the same base product appears in the panel more than one time, the first entry will dictate the preferred release to be used and the second and all subsequent entries for that base product will be ignored.

The vendor name, product name, and preferred release number specified in the body of the Selection List Profile panel must be specified, formatted and spelled exactly as they appear in the associated data base. Otherwise, user specific selection list processing will not work correctly.

Since it depends on the ISPF application profile, user specific selection list processing is applied only when MVS/QuickRef is running under ISPF. It will not be used when MVS/QuickRef is executed in a batch environment.

**Order Of Precedence**

The three types of user-directed selection list processing, when active, are applied in the following order:

- user specific selection list processing
- vendor/product specific selection list processing
- operating system level selection list processing
So, in effect, user specific selection list processing, when active, takes precedence over vendor/product specific selection list processing which, when active, takes precedence over operating system level selection list processing.

As an example, suppose the QWIKOPTS vendor/product specific selection list option is set to Yes (VPSSLO=Y) and the Vendor/Product Specific Selection List Table is set up as shown in Figure 18 on page 40. Now suppose you specify a fast-path string like QW CTS001. Instead of the item selection list shown in Figure 19 on page 40, item CTS001 associated with CA-1 V5R1 would automatically be selected and shown. Now suppose you have your own Selection List Profile set up as shown in Figure 20 on page 42 and you specify a fast-path string like QW CTS001. Then, since user specific selection list processing takes precedence over vendor/product specific selection list processing, item CTS001 associated with CA-1 V5R2 would automatically be selected and shown.

**Example of Using User-Directed Selection List Processing**

The preceding sections contain several examples showing how the different types of user-directed selection list processing can be used. Here is one additional, more comprehensive example.

Suppose your installation is running OS/390 V2R4 and that your system installer wants MVS/QuickRef to automatically select this same release for the "operating system specific" products in the main data base (products like OS/390 SYSTEM CODES, OS/390 SYSTEM MSGS, OS/390 MVS COMMANDS, OS/390 UNIX(R) SYNTX, etc.). Then he would set the QWIKOPTS operating system level selection list option to Yes (OSLSLO=Y).

So now V2R4 is designated as the preferred release for all operating system specific products in the main data base for all users of MVS/QuickRef in this processing environment.

Now suppose your installation is running OS/390 JES2 V1R3. So, for products like OS/390 JES2 COMMANDS and OS/390 JES2 MESSAGES, the product installer wants to designate V1R3 as the preferred release and have this override the preferred release (V2R4) specified for these products by operating system level selection list processing. So he would set the QWIKOPTS vendor/product specific selection list option to Yes (VPSSLO=Y) and set up the Vendor/Product Specific Selection List Table as shown in Figure 21.

<table>
<thead>
<tr>
<th>Data Base</th>
<th>Vendor</th>
<th>Product</th>
<th>Preferred Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>IBM</td>
<td>OS/390 JES2 MESSAGES</td>
<td>V1R3</td>
</tr>
<tr>
<td>M</td>
<td>IBM</td>
<td>OS/390 JES2 COMMANDS</td>
<td>V1R3</td>
</tr>
</tbody>
</table>

----End of Vendor/Product Specific Selection List Table----

**Figure 21 - QINFO Vendor/Product Specific Selection List Table Display**
Since vendor/product specific selection list processing takes precedence over operating system level selection list processing, OS/390 V1R3 is now designated as the preferred release for JES2 MESSAGES and JES2 COMMANDS for all users of MVS/QuickRef in this processing environment.

Now suppose that, while OS/390 JES2 V1R3 is still the production release of OS/390 JES2, your installation decides to start testing OS/390 JES2 V2R4 and that you have been chosen to participate in this testing effort. While testing OS/390 JES2 V2R4, you would like for V2R4 to be the preferred release of OS/390 JES2 for you personally but you do not want this to affect any other users of MVS/QuickRef. Then you could set up your Selection List Profile as shown in Figure 22.

```
* MVS/QuickRef - Selection List Profile *
Command ===> 

Operating System Level Selection List Option => Y (Y-Yes N-No H-High)
Vendor/Product Specific Selection List Option=> Y (Y-Yes N-No)
User Specific Selection List Option => Y (Y-Yes N-No)

#  Data Base Vendor             Product                                   Preferred Release
   1 M          IBM                 OS/390 JES2 MESSAGES     V2R4
   2 M          IBM                 OS/390 JES2 COMMANDS   V2R4
   3
   4
   5
   6
   7
   8
   9
  10

Press ENTER to save changes temporarily (for remainder of this Q/R session)
Type  SAVE   to save changes permanently (in profile)
Type  END or CANCEL to discard current screen changes
Replace underbar with 'D' to delete entry
```

Figure 22 - Selection List Profile Panel

Since user specific selection list processing takes precedence over vendor/product specific selection list processing, OS/390 V2R4 is now designated as the preferred release for JES2 MESSAGES and JES2 COMMANDS for you personally. Meanwhile, no other users of the system are affected and you still have operating system level selection list processing and vendor/product specific selection list processing available for any other products to which they might apply.
Now suppose that, when you are not testing OS/390 JES2 V2R4, you want to go back to having V1R3 as your preferred release for JES2. In this case, you could just change the Selection List Profile panel User Specific Selection Option field to N (for NO), leave the preferred release entries as shown in the body of the panel, and type SAVE in the command line before hitting ENTER to make this a permanent change. Then, when you are ready to start testing OS/390 JES2 V2R4 again, simply change the User Specific Selection List Option field back to Y (for Yes) and type SAVE in the command line before hitting ENTER to make this a permanent change. This way you can reactivate V2R4 as the preferred release for OS/390 JES2 MESSAGES and OS/390 JES2 COMMANDS without having to reenter the information in the body of the panel.

**Predisplay Analysis**

Predisplay analysis refers to certain types of automatic processing done by MVS/QuickRef before certain types of reference items are displayed. Predisplay analysis applies to the main data base as well as to any user data bases which may be defined.

There are two basic types of predisplay analysis:

- automatic lookup
- automatic FINDCODE processing

Automatic lookup is a predisplay analysis feature which causes additional relevant items to be automatically read in and displayed following the reference text for the current item. There are two types of automatic lookup:

- automatic lookup based on pointers, which are placed in the reference text by the person preparing the data for inclusion in the data base

- automatic lookup based on text content, in which MVS/QuickRef determines which additional items to bring in by analyzing the content of the reference text about to be displayed

The purpose of the automatic lookup feature is to have additional relevant information automatically presented - without the need for a separate lookup request.

Automatic FINDCODE processing is a predisplay analysis feature which, under certain circumstances, automatically places an appropriate FINDCODE command in the ISPF command line of the current MVS/QuickRef display panel. The purpose of automatic FINDCODE processing is to allow you to find a relevant numeric code in a list (like a list of return, reason, or error codes) without having to manually type in the appropriate command and the code to be found. With automatic FINDCODE processing, once you are through reviewing the initial display of the reference text for a message or abend code which contains a list of codes, all you have to do is press ENTER to find the relevant code.
Due to the great volume and variety of data in the MVS/QuickRef data base, predisplay analysis cannot be guaranteed to work correctly in every single case. There may be occasions when automatic lookup brings in an additional item that is not really relevant to the current display and there may be occasions when automatic lookup fails to bring in an appropriate relevant item. There may be occasions when FINDCODE processing fails to find the appropriate numeric code, when it finds the wrong numeric code, or when it finds the right numeric code in the wrong list of codes. However, in most of the cases where predisplay analysis fails to work correctly, it will be obvious from the context that it did not work correctly. Moreover, it should work correctly often enough (and save you time often enough) to more than compensate for the infrequent occasions when it does not work correctly. For more information on predisplay analysis, see item PREDSPLY-ANLYSIS in the MVS/QuickRef on-line help facility.

**Console Support Feature**

MVS/QuickRef can be invoked from an MVS console as a started task in order to display reference information directly on the console. JCL for a started task to invoke MVS/QuickRef is supplied as member “QWC” in the MVS/QuickRef JCL library. The started task is invoked with a single parameter, the fast-path invocation string indicating the reference information to be returned (generally, this is the message ID or abend code for which help is desired). The requested reference information will be written to the console via a WTO with a ROUTCDE=2. You can use the MVS/QuickRef options table, QWIKOPTS, to change this route code value if route code two is not desirable.

Below are two examples of console invocation of MVS/QuickRef:

S QWC,PARM='IEC141I'  <!-- Help with IEC141I message -->
S QWC,PARM='S213'  <!-- Help with S213 abend -->

**Data Base Customization**

MVS/QuickRef allows you to specify overriding customization parameters to replace, augment, or deny access to reference information in the MVS/QuickRef main data base or in your own local user data base(s). Specification of override customization parameters is optional. The override parameter definition and implementation process is fully described in chapter six of this guide.

**Direct Program Call**

Through a facility referred to as direct program call, MVS/QuickRef may be called by another program in order to extract information from the MVS/QuickRef data base. The calling program passes a parm list specifying the information to be returned and where it is to be stored; MVS/QuickRef extracts the requested information from the data base and passes it back in the specified storage area.
Direct Program Call Parmlist

On a direct program call, the invoking program must call or link to module QWIKREF1 and must pass two parameters.

♦ the fullword address of a storage area mapped by copy member QWIKDPC1
♦ the fullword address of a storage area mapped by copy member QWIKDPC2

When the call to module QWIKREF1 is made, register one must contain the address of this two-word parm list. The fullword containing the address of the second parm must have its leftmost bit set to one to indicate that it is the last parameter in the list. Copy members QWIKDPC1 and QWIKDPC2 can be found in the MVS/QuickRef macro library and are further described below.

The text below shows the format of the parameter list which must be passed:

R1 contents: Address of two-word parameter list, with X’80’ bit turned on in leftmost byte of the second word.

Parameter list format:

1st word - fullword address of storage area mapped by copy member QWIKDPC1

2nd word - fullword address of storage area mapped by copy member QWIKDPC2.

The leftmost bit in this fullword must be set to one to indicate that it is the last parameter in the list.

If you review the QWIKDPC1 copy member as it appears in the MVS/QuickRef macro library, you will see that it contains the fields described below:

QWDC1EYE - nine-byte "eyecatcher" constant. This field is required; leave it as is.

QWDC1REL - three-byte character field which must contain the release number of an MVS/QuickRef release which is compatible with the direct program call interface to be used for this call. In other words, this should be the release number of the MVS/QuickRef User's Guide from which the direct program call interface specifications are (or were) taken. As an example, if you want to use the direct program call interface specified in the Release 5.3 User's Guide, then you should move '5.3' to QWDC1REL. (The release number used must be 5.3 or later.)

If you review the QWIKDPC2 copy member as it appears in the MVS/QuickRef macro library, you will see that it contains the fields described below:

QWDC2FPA - fullword address of a fast-path invocation string, consisting of a 2-byte field containing the length, in hexadecimal, of the following fast-path string. For a
description of the format of a fast-path string, see the section titled "Fast-Path Invocation".

As an example, consider the fast-path string below which is used to request a display of the IDC3009I message:

```
0008C9C4C3F3F0F0F9C9
IDC3009I
```

"IDC3009I" is the desired item to fetch.

The X'0008' halfword in the example above is the length of the IDC3009I string that follows.

If the fast-path string contains a component containing embedded spaces, then the component containing the embedded spaces must be enclosed in single quote marks. For example, fast-path string:

```
S0C1 P='SYSTEM CODES'
```

would be formatted as follows:

```
0015E2F0C3F140D77E7DE2E8E2E3C5D440C3D6C4C5E27D
S 0 C 1   P = ' S Y S T E M   C O D E S '
```

QWDC2SAA - fullword address of a storage area where the requested reference information is to be returned. This area:
- can reside above or below the 16mb address line
- must be at least 2,000 bytes in length
- should be large enough to contain the requested reference information
An area 1,000,000 bytes in length should be large enough to support extraction of any single item in the main data base.

QWDC2SAL - fullword length in bytes of the area pointed to by the QWDC2SAA field. This fullword is changed by module QWIKREF1 to the actual length in bytes of the reference information which is returned.

QWDC2LL - fullword which is set by module QWIKREF1 to indicate the line length of the text lines being returned.

QWDC2RAL - fullword which is set by module QWIKREF1 only when the length of the passed storage area (as indicated by the QWDC2SAL field) is not large enough to contain the requested reference information. In this situation, QWDC2RAL is set to the minimum number of bytes of storage that would be required to hold all of the requested reference information. The return code is also set to 4 and the
QWDC2SAL field is set to indicate the number of bytes of information that were actually returned.

QWDC2TIR - one-byte character field which is set by module QWIKREF1 to indicate the type of reference information being returned, where:

T  indicates reference text for an individual item in the data base
I  indicates a regular item selection list (one item per line)
G  indicates a generic item selection list (four items per line)
V  indicates a product list in vendor sequence
P  indicates a product list in product sequence
O  indicates a vendor only list
C  indicates a product category code list
S  indicates a DASD free space display
Q  indicates a QINFO display

QWDC2ITM - 16-byte character field which is set by module QWIKREF1, when the QWDC2TIR field is set to T, to indicate the item name associated with the reference text being returned

QWDC2VEN - 15-byte character field which is set by module QWIKREF1, when the QWDC2TIR field is set to T, to indicate the vendor name associated with the reference text being returned

QWDC2PRD - 20-byte character field which is set by module QWIKREF1, when the QWDC2TIR field is set to T, to indicate the product name associated with the reference text being returned

QWDC2REL - 15-byte character field which is set by module QWIKREF1, when the QWDC2TIR field is set to T, to indicate the release number associated with the reference text being returned

For each different type of reference information which may be returned, each line of returned text is formatted exactly as it would appear in the corresponding ISPF display. The only exception to this rule has to do with the fact that the returned reference information is formatted so that each text line is the same length. The "fixed line length" used for this purpose is dictated by the length of the longest text line currently being returned. Any returned text lines shorter than this length are padded on the right with blanks as required to match this "fixed line length". The "fixed line length" being used is then returned to the calling program in the QWDC2LL field.

As an example of using the direct program call interface parms, suppose the fast-path string pointed to by the QWDC2FPA field specifies an item name that appears in the data base in three different products. Now suppose that this results in a regular item selection list containing three lines, one for each product in which the specified item name appears, that the longest line in the list happens to be 80 characters long, and that, including the "header" line and the line marking
the end of the list, the total number of characters to be returned is 400. In this case, the QWIKREF1 module will place an 'I' in the QWDC2TIR field (to indicate a regular item selection list), will store the resulting regular item selection list in the area pointed to by the QWDC2SAA field, will store a value of 80 in the QWDC2LL field, and, finally, will store a value of 400 in the QWDC2SAL field.

Now suppose the fast-path string pointed to by the QWDC2FPA field specifies an item name that appears just one time in the data base, that the longest reference text line carried in the data base for this item happens to be 72 characters long, and that the total number of characters to be returned is 3672. In this case, the QWIKREF1 module will place a 'T' in the QWDC2TIR field (to indicate reference text for an individual item in the data base), will store the reference text associated with the specified item in the area pointed to by the QWDC2SAA field, will store a value of 72 in the QWDC2LL field, and, finally, will store a value of 3672 in the QWDC2SAL field. The QWIKREF1 module will also place the associated item name, vendor name, product name, and release number in the QWDC2ITM, QWDC2VEN, QWDC2PRD, and QWDC2REL fields, respectively.

**Direct Program Call Considerations**

When using direct program call to extract information from MVS/QuickRef, the following points should be considered:

1. MVS/QuickRef executes AMODE(31), RMODE(24). The data area parameters passed may reside above or below the 16mb address line, but module QWIKREF1 must be invoked in 31-bit addressing mode on MVS/XA MVS/ESA, OS/390, or Z/OS systems; this can be accomplished by invoking the product using a LINK macro to invoke module QWIKREF1 or by issuing a LOAD for the module and then branching to it after setting AMODE(31).

2. Module QWIKREF1 executes in problem program state and key; for this reason, the storage area and the parameter list passed must reside in non-fetch-protected key 8 storage, since both will be modified by QWIKREF1 if the invocation is successful.

3. If reference information is to be extracted from the MVS/QuickRef main data base, then the main data base must be preallocated to the QWREFDD DD statement or the name of the main data base must be specified in the QWIKOPTS table. If reference information is to be extracted from one or more user data bases, then the single user data base from which information is to be extracted must be preallocated to the QWREFDDU DD statement or the name(s) of the user data base(s) must be specified in the QWIKOPTS table. If override parameters are being used as described in chapter six of this guide, then the override parameter data set must be preallocated to the QWPARMS DD statement or the name of the override parameter data set must be specified in the QWIKOPTS table. If access to TSO SYSHELP information is desired, then a SYSHELP DD should be preallocated to the same TSO help data sets that are concatenated to the SYSHELP DD statement in your TSO logon procedure.
Direct Program Call Return Codes

One of the decimal return codes listed below will be passed back in register 15 to the caller of module QWIKREF1, to indicate the success or failure of the invocation. If an error other than a parameter list error occurs, a message describing the error will normally be placed in the passed storage area.

0 - success; the requested reference information was placed in the storage area pointed to by the QWDC2SAA field; the QWDC2SAL field was updated to contain the length in bytes of the reference information which was returned; the QWDC2LL field was updated to indicate the "fixed line length" of the returned text lines

4 - partial success; the supplied area was not large enough to contain the requested reference information; increase the size of the supplied storage area to match that specified in the QWDC2RAL field

8 - the requested information could not be found; verify that the fast-path invocation string and all data set names are correctly specified

12 - you cannot access the requested information due to local security restrictions; check with your systems programmer if you think this is incorrect

16 - required parm is missing or invalid; check to make sure all parms are correctly specified

20 - data base name not found or dynamic allocation failed; check to make sure that all data set names are correctly specified

24 - severe error encountered when attempting to open or access data base; make sure all data set names are correctly specified and that none of the data sets being accessed have been corrupted
   -this return code can also indicate that the QWIKOPTS load module is from an earlier release of MVS/QuickRef and is not compatible with the current release; make sure you are using the version of QWIKOPTS associated with the current release of MVS/QuickRef

28 - MVS/QuickRef usage period has expired or you are attempting to run on an unauthorized CPU; have your systems programmer check the status of your MVS/QuickRef installation

32 - termination was forced by installation security exit; check with your systems programmer if you think this is incorrect

36 - invalid batch commands file; check the validity of the batch commands file (ddname QWCMDSDD) and the commands it contains

40 – there is a problem with the license key file; either the data set name could not be determined or the file could not be dynamically allocated, could not be opened, is fixed length, or
contains no input; check to make sure the license key file is valid and properly defined to MVS/QuickRef

Direct Program Call Example

For an example of a program which calls MVS/QuickRef using the direct program call interface, see member QWIDPCDR in the MVS/QuickRef JCL library.

Direct Program CallParms For Releases Prior To 5.3

Prior to Release 5.3, MVS/QuickRef direct program call used a different set of interface parms, which are described below. You can continue to use these direct program call interface parms with MVS/QuickRef 5.3 and later releases; however, they do not provide as much data on the type of reference information being returned.

When using the pre-5.3 direct program call interface to MVS/QuickRef, the invoking program must call or link to module QWIKREF1 and must pass four parameters:

♦ the address of a fast-path invocation string
♦ the address of a storage area in which the reference information generated by MVS/QuickRef is to be placed
♦ the length of the passed storage area
♦ an area where MVS/QuickRef can indicate the length of the returned text lines

The text below shows the format of the parameter list which must be passed:

R1 contents: Address of four word parameter list, with X’80’ bit turned on in leftmost byte of the fourth word.

Parameter list format:

1st word - address of a fast-path invocation string, consisting of a 2-byte field containing the length, in hexadecimal, of the following fast-path string. For a description of the format of a fast-path string, see the section titled "Fast-Path Invocation".

As an example, consider the fast-path string below which requests a display of the IDC3009I message:

```
0008 C9C4C3F3F0F0F9C9
IDC3009I
```

“IDC3009I” is the desired item to fetch.

The X’0008’ halfword in the example above is the length of the IDC3009I string that follows.
2nd word - fullword address of a storage area where the requested reference information is to be returned. This area:
- can reside above or below the 16mb address line
- must be at least 2,000 bytes in length
- should be large enough to contain the requested reference information
An area 1,000,000 bytes in length should be large enough to support extraction of most individual items in the main data base.

3rd word - fullword length in bytes of the area pointed to by the 2nd parameter. This fullword is changed by module QWIKREF1 to the actual length in bytes of the reference information which is returned.

4th word - fullword which is set by module QWIKREF1 to indicate the line length of the text lines being returned. The leftmost bit in this fullword must be set to one to indicate that it is the last parameter in the list.

The reference information returned in the area pointed to by the second parameter is formatted so that each text line is the same length. The "fixed line length" used for this purpose is dictated by the length of the longest text line currently being returned. Any returned text lines shorter than this length are padded on the right with blanks as required to match this "fixed line length". The "fixed line length" being used is then returned to the calling program in the fourth parameter.

User Security Exit

MVS/QuickRef supports a user security exit which can be used in addition to the PREVENT and ALLOW statements described in chapter six of this guide. The security exit is a program that you or someone at your installation writes to interface with your local security system. The exit will be used if it can be found in the LOGON PROC STEPLIB, in PLPA, on the MVS link list, or in the ISPLLIB concatenation when MVS/QuickRef is invoked. The exit is called:
- each time a vendor, product, or item is about to be included in a selection list
- prior to a direct key read for an individual item
- each time a DASD volume is about to be included in a DASD free space display

The exit is passed a single parameter, which is the address of a storage area mapped by the QWISPRMS copy member. Each time the exit is called, the storage area mapped by the QWISPRMS copy member will indicate whether or not the call is for a vendor name, a product name, an item name, or a DASD volume. It will also supply the name of the vendor, product, item, or DASD volume which is about to be displayed. (See the QWISPRMS copy member in the MVS/QuickRef JCL library for more details on the layout of the storage area that is passed to the exit.)
The exit must then pass back one of the return codes shown below in order to indicate whether or not the vendor, product, item, or DASD volume specified in the QWISPRMS storage area should be displayed:

0 - allow the user to view the specified vendor, product, item, or DASD volume

4 - deny the user access to the specified vendor, product, item, or DASD volume

8 - deny the user access to the specified vendor, product, item, or DASD volume and, on a direct key read for an individual item, terminate MVS/QuickRef immediately

If the exit passes back a return code value of 4 for an item name for which a direct key read is about to be done, then, instead of displaying the reference information for the specified item, MVS/QuickRef will display message QWRFM036. If the exit passes back a return code value of 8 for an item name for which a direct key read is about to be done, then, instead of displaying the reference information for the specified item, MVS/QuickRef will immediately terminate.

If the exit passes back a return code value of 4 or 8 for a vendor name, product name, item name, or DASD volume about to be added to a selection list or DASD free space display, then the specified vendor, product, item, or DASD volume will be dropped from the selection list or DASD free space display that is about to be shown. If all of the vendors, products, items, or DASD volumes are dropped from a selection list or DASD free space display due to the exit module, then message QWRFM036 will be shown.

A sample security exit is included in the MVS/QuickRef JCL library as member QWIKEXIT.

The security exit must be a load module named QWIKEXIT; the module must reside in a STEPLIB in your LOGON procedure, in PLPA, in a data set that is part of the MVS link list, or in the ISPLLIB concatenation. The module is LOAD’ed when MVS/QuickRef is first invoked and then CALL’ed each time it is needed. If a module named QWIKEXIT cannot be successfully LOAD’ed, then MVS/QuickRef assumes that no user security module has been provided and that none is needed.

The exit must be coded and link edited reentrant and reusable, but it can execute in any addressing or residency mode. Since MVS/QuickRef is not APF-authorized, the exit will be invoked in problem program state and key, non-APF-authorized. If the exit dynamically switches to an APF-authorized state, it MUST switch back before returning to MVS/QuickRef to avoid integrity exposures.

The storage area mapped by the QWISPRMS copy member, as well as the parm list containing the address which points to this area, resides in virtual storage below the 16mb address line.

**Selective Data Base Load Facility**

MVS/QuickRef includes a program that will allow you to discard those elements of the main data base that you do not want or need. You discard elements by creating utility control
Batch Execution

MVS/QuickRef can also be executed as a batch job. When executed in this mode, the resulting reference information is written to the file allocated to the QWPRINT DD statement.

Below is a sample of the JCL required to execute MVS/QuickRef in batch. Each component of the JCL is explained below the sample. This sample also appears in the MVS/QuickRef JCL library in member QWBATCH.

```plaintext
//JS10 EXEC PGM=QWIKREF1,
  REGION=0K,
  PARM='fast-path-string'
//STEPLIB DD DSN=quickref.linklib.dsn,
  DISP=SHR
//QWPRINT DD SYSOUT=*,
  DCB=(RECFM=VBA,LRECL=133,BLKSIZE=6000)
//QWREFDD DD DSN=quickref.database.dsn,
  DISP=SHR
//QWREFDDU DD DSN=quickref.user.database.dsn,
  DISP=SHR
//QWPARMS DD DSN=quickref.override.parmdsn,
  DISP=SHR
//SYSHELP DD DSN=SYS1.HELP,
```

Description of Batch JCL Elements

PARM='fast-path-string' - specifies the fast-path invocation string which dictates the information to be retrieved from MVS/QuickRef. Please note that, since this JCL parm takes the form of an MVS/QuickRef fast-path invocation string, commas should not be used to separate the components of the fast-path string. For example, use: PARM='V=IBM P=C* I=ADD'. Do not use: PARM='V=IBM,P=C*,I=ADD'.

STEPLIB - specifies the name of the program library in which module QWIKREF1 resides.

QWPRINT (optional) - the DD statement defining the output file to which the requested reference information is to be written. If this DD statement is not provided, the requested reference information will be WTO'd to the job log. If a QWPRINT DD statement is provided:

♦ the associated file can be allocated to SYSOUT or to a sequential DASD or tape data set
♦ the associated file must have RECFM=VBA (variable block with ANSI printer control characters)
the associated file must have an LRECL large enough to handle the longest line in the reference information to be written out

any line in the reference information which is longer than the maximum record length allowed for the associated file will be truncated on the right as required to match the longest allowed record, a warning message will be issued, and a return code of 4 will be set

QWREFDD (optional) - the DD statement defining the MVS/QuickRef main data base to be accessed in this batch run. If this DD statement is not provided, MVS/QuickRef will dynamically allocate the main data base using the name of the main data base specified in the QWIKOPTS table. If no name is provided for the main data base in the QWIKOPTS table, then this DD statement is required in order for any reference information to be extracted from the main data base.

QWREFDDU (optional) - the DD statement defining the single user data base to be accessed in this batch run. If this DD statement is not provided, MVS/QuickRef will dynamically allocate the required user data base(s) using the user data base name(s) specified in the QWIKOPTS table. If no user data base name(s) are provided in the QWIKOPTS table, then this DD statement is required in order for any reference information to be extracted from a user data base.

QWPARMS (optional) - the DD statement defining the MVS/QuickRef override parameter data set to be accessed in this batch run. If this DD statement is not provided, MVS/QuickRef will dynamically allocate the override parameter data set using the name of the override parameter data set specified in the QWIKOPTS table. If no name is provided for the override parameter data set in the QWIKOPTS table, then this DD statement is required if overrides are to be processed.

SYSHELP (optional) - the DD statement defining the TSO help data sets that are to be accessed in this batch run. This DD statement should reference the same TSO help data sets concatenated to the SYSHELP DD statement in your TSO logon procedure. If this DD statement is not provided, then no TSO help information can be returned.

Examples of Batch Execution

Here are two examples of executing MVS/QuickRef as a batch job. The first example requests display of the reference text for item IDC3009I (a VSAM Access Method Services message description) and requests that the item be written to SYSOUT. No user data base, override parameter, or TSO help information is desired.

```
//JS10    EXEC PGM=QWIKREF1,REGION=0K,
//               PARM='IDC3009I'
//STEPLIB   DD DSN=SYS2.QUICKREF.LINKLIB,
//             DISP=SHR
//QWPRINT   DD SYSOUT=*,
//             DCB=(RECFM=VBA,LRECL=133,BLKSIZE=6000)
//QWREFDD   DD DSN=SYS2.QUICKREF.DATABASE,
//             DISP=SHR
```
The job below requests a DASD free space display for all DASD volumes with volume serial numbers beginning with TSO. The resulting DASD free space information is written to the QWPRINT file, which in this case is allocated to a PDS member.

```plaintext
//JS10 EXEC PGM=QWIKREF1,
//      REGION=0K,
//      PARM='S=TSO*'
//STEPLIB DD DSN=SYS2.QUICKREF.LINKLIB,
//      DISP=SHR
//QWPRINT DD DSN=SYS2.WORK.PDS(TSODASD),
//      DISP=SHR
//QWREFDD DD DSN=SYS2.QUICKREF.DATABASE,
//      DISP=SHR
```

**Batch Execution Return Codes**

When executing in batch, MVS/QuickRef returns the following return codes:

- **0** - success; the requested reference information was output to the QWPRINT DD statement
- **4** - partial success; one or more lines of output text were truncated; increase the LRECL of the QWPRINT output file
- **8 or higher** - see the return codes described in the section "Direct Program Call Return Codes" on page 53 of this document.

**Batch Commands**

MVS/QuickRef supports a number of commands which can be used when executing in the ISPF environment. Certain of these commands can also be used when executing in batch, whether using direct program call or executing MVS/QuickRef as a batch job. See the previous sections in this chapter on “Direct Program Call” and “Batch Execution” for more information on these types of processing environments.

The commands which are supported in batch are the following:

<table>
<thead>
<tr>
<th>Command</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>QINFO</td>
<td>QINFO</td>
</tr>
<tr>
<td>SORT</td>
<td>SO</td>
</tr>
<tr>
<td>GETNEXT and GETPREV</td>
<td>GN and GP</td>
</tr>
<tr>
<td>SEARCH</td>
<td>SE</td>
</tr>
<tr>
<td>QPRINT</td>
<td>QP</td>
</tr>
</tbody>
</table>
For general information on these commands, see the section on “Additional On-Line Features” in Chapter Three; for more detailed information, look up the descriptions of these commands in the MVS/QuickRef Help facility.

You “invoke” commands while running in batch by placing them in a file referred to as the batch commands file. The ddname used for the batch commands file is QWCMDSDD. When this ddname is present in the batch environment, it must specify a valid batch commands file. When this ddname is not present, it is assumed that no batch commands are to be executed.

The batch commands file:

♦ may be specified as a physical sequential data set or a specific member of a PDS
♦ may be fixed or variable length
♦ must have an LRECL no greater than 80 bytes (84 if the file is variable length)
♦ must not contain any blank records

Each command must be placed in a separate record within the batch commands file. Commands cannot be continued into the next record and any part of a command which extends beyond column 64 is ignored.

When a command is placed in the batch commands file, the command and its parameters (if any) should be formatted just as they would be if the command were being specified on the ISPF command line.

Those commands (like QINFO, HELP, and SEARCH) which, in the ISPF environment, do not relate directly to the text currently being displayed, can be used without specifying a fast-path string. When executing MVS/QuickRef as a batch job, this means that these commands can be used without specifying a JCL parm. For direct program call, this means that these commands can be used with the QWDC2FPA field of the QWIKDPC2 copy member pointing to a halfword containing a length value of zero.

Those commands (like SORT, GETNEXT, and GETPREV) which, in the ISPF environment, do relate directly to the data currently being displayed can be used only when a fast-path string is specified. Furthermore, the fast-path string that is used must be one that causes some data to be retrieved that relates to the command being used. For example, if you use the SORT command, the fast-path string must cause some DASD free space information to be retrieved. If you use the GETNEXT command, the fast-path string must cause an individual item to be retrieved which belongs to a product which has “getnext based on content” processing applicable to it.

When the batch commands file is specified and contains a single valid command, the information returned is the information relating to that specific command. When the batch commands file contains multiple commands, the information returned is the information relating to the last command in the file.

The information returned for each command is shown below:
QINFO       - same information as QINFO command in ISPF environment
SORT        - DASD free space information sorted as requested
GETNEXT     - next item in “content” sequence
GETPREV     - previous item in “content” sequence
HELP        - item selection list showing items belonging to the MVS/QuickRef help product
SEARCH      - item selection list showing items containing the specified character string
QPRINT      - immediate printing of the information currently available (based on either the fast-path string specified or other commands which may have been specified)

When the batch commands file is specified and contains a single valid command, the information returned is the information relating to that specific command. When the batch commands file contains multiple commands, the information returned is the information relating to the last command in the file. The only exception to this rule is the QPRINT command; it causes immediate printing of the "currently available" information.

Although it is possible to place multiple commands in the batch commands file, you must take care to make sure that the commands “make sense” in the context they appear. For example, if you place the GETNEXT command after the QINFO command, this will result in an error since GETNEXT processing does not apply to the information produced by the QINFO command.

It should also be noted that, if the QPRINT command is used to get more than one "print-out", then, if the QWPRINT dd statement is allocated to a data set instead of SYSOUT, that data set should be specified with DISP=MOD in the JCL. Otherwise, each subsequent "print-out" will overlay and thus replace the previous "print-out" and only the last "print-out" will appear in the output data set.

### Batch Command Examples

As an example of using batch commands, suppose you are executing MVS/QuickRef as a batch job and you want to review the information provided by the QINFO command. Then you could simply add these two statements to your JCL:

```
//QWCMDSDD DD *
QINFO
```

This would specify the batch commands file as an instream data set containing a single command (QINFO). Since the QINFO command is one which, in the ISPF environment, does not relate to the information currently on display, you could eliminate the JCL parm that is normally required when executing in batch. This means that you could specify the EXEC statement as shown below:

```
//JS10   EXEC PGM=QWIKREF1
```
As another example, suppose you are using direct program call to retrieve DASD free space information and you want the information to be sorted into descending sequence based on the fourth column of data being returned. Then you could allocate the QWCMDSDD statement to a sequential file containing the single record shown below:

```
SORT 4 D
```

As another example, suppose you want to retrieve DASD free space information for all online DASD volumes (JCL PARM is S=*) and you want three different listings of the information sorted 3 different ways as indicated by the following 3 SORT commands:

```
SORT 4 D (sorted by free DASD extent count, descending)
SORT 4 A (sorted by free DASD extent count, ascending)
SORT 5 A (sorted by free track count, ascending)
```

Then you would allocate the QWCMDSDD statement to a sequential file containing the records shown below:

```
SORT 4 D
QPRINT
SORT 4 A
QPRINT
SORT 5 A
```

In this case, the first QPRINT command would print the listing resulting from the first SORT command; the second QPRINT command would print the listing resulting from the second SORT command. There is no need for a third QPRINT command because, as a default, batch processing always outputs the results of the last batch command in the QWCMDSDD file.

### Batch Command Errors and Return Codes

If you receive a zero return code when executing in batch with a batch commands file, this indicates that the command(s) in your batch commands file were successfully executed.

If you receive a non-zero return code other than 36, this indicates an error of some sort. Look at the return codes for the particular batch environment in which you are running (documented in the sections on “Batch Execution Return Codes” and “Direct Program Call Return Codes” earlier in this chapter).

If you receive return code 36, this indicates a problem with the batch commands file or the command(s) in the batch commands file. In this case, check to make sure the batch commands file is specified as a physical sequential data set or a member of a PDS with an LRECL no greater than 80 or, if the file is variable length, no greater than 84. Make sure all commands and all command parameters in the batch commands file are correctly specified, that they do not extend beyond column 64, and that the batch commands file does not contain any blank records.
For the SORT, GETNEXT, and GETPREV commands, make sure a fast-path string is specified which retrieves the type of data required for the type of command(s) being used.
## Installing MVS/QuickRef

You install MVS/QuickRef by following the information presented in this chapter. Before proceeding with the installation process please review the 'Installation Considerations' and 'Upgrade Information for Existing Users' sections presented below.

Please note that the section titled "Troubleshooting MVS/QuickRef" later in this chapter contains a list of common problems and corrective actions. **Please review the information in this section before calling MVS/QuickRef product support.**

### Installation Considerations

The MVS/QuickRef installation process copies the elements of the product from the distribution tape to DASD data sets that are allocated during the installation process. All necessary data sets are allocated during the product installation process, so you do not need to manually allocate any data sets before beginning the installation process. For your use in planning for product installation, the DCB attributes and *approximate* 3390 DASD space requirements for the data sets used by MVS/QuickRef are provided below:

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Non-SMP/E Installation</th>
<th>SMP/E Installation</th>
<th>REC/LRE/BLK</th>
<th>390 Tracks</th>
<th>Directory Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>hlq.JCL</td>
<td>hlq.JCL</td>
<td>FB/80/6160</td>
<td>77</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>hlq.PANELS</td>
<td>hlq.QWIPNL</td>
<td>FB/80/6160</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>hlq.MESSAGES</td>
<td>hlq.QWIMG</td>
<td>FB/80/6160</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>hlq.TABLES</td>
<td>hlq.QWITBL</td>
<td>FB/80/6160</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>hlq.MACLIB</td>
<td>hlq.QWIMAC</td>
<td>FB/80/6160</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>hlq.SOURCE</td>
<td>hlq.QWISRC</td>
<td>FB/80/6160</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>hlq.LINKLIB</td>
<td>hlq.QWILINK</td>
<td>U/0/6233</td>
<td>40</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>hlq.DATABASE</td>
<td>hlq.DATABASE</td>
<td>F/6160/6160</td>
<td>18,552</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Total:</td>
<td>18,678</td>
<td>32</td>
</tr>
</tbody>
</table>

1 These figures are estimates only; they are not meant to be exact.

The prefix ‘hlq’ on the data set names in the previous table can be any data set name high-level qualifier you choose such as 'SYS2.QUICKREF.R620'. The data sets are allocated and cataloged by the jobs in the installation process, but you can copy MVS/QuickRef elements into existing PDS’s at your installation rather than allocate new ones by modifying the installation JCL.

Please note that the ‘hlq.DATABASE’ data set is the MVS/QuickRef main data base. This data set is NOT a PDS; it is a direct-access data set that is read with the EXCP access method, so no directory space is allocated to it.

All MVS/QuickRef data bases, including the main data base and any user data bases you may create, can occupy one or more DASD volumes (with one or more extents on each volume). All volumes used for a single data base must be of the same DASD type (all 3380, all 3390, etc.).
The maximum number of volumes/extents that can be allocated for an MVS/QuickRef data base is the lesser of:

- 16 volumes with a maximum of 16 extents per volume
- OR
- the maximum number of volumes/extents allowed for a non-VSAM physical sequential data set in your particular processing environment.

All of the PDS members copied to DASD during the installation process begin with the characters ‘QWI’ except the PASTE program. Remember this if you want to move or copy them to the production ISPF libraries and system link list when implementing the system into production.

MVS/QuickRef includes an options table load module named QWIKOPTS. In order to make sure that MVS/QuickRef functions correctly in your processing environment, you should review the section entitled "Setting MVS/QuickRef Global Installation Options" on page 97 of this guide in order to determine whether or not you need to make any changes to the default options supplied with the QWIKOPTS table; in particular, the OSLSLO= option which controls the MVS/QuickRef automatic operating system level selection feature.

MVS/QuickRef can be accessed either by using the ISPF LIBDEF technique or by placing the product's elements in permanently allocated data sets. The LIBDEF technique provides you with an easier testing environment, however, using permanently allocated data sets offers better performance, and, for this reason, is the preferred method. You must decide during the installation process which method you will use.

MVS/QuickRef can either use a main data base that is allocated in your TSO logon procedure or it can dynamically allocate the main data base. Since the main data base can be inadvertently freed by a TSO user, allowing MVS/QuickRef to dynamically allocate the main data set is the recommended method. In this way, MVS/QuickRef should always be able to access the main data base. You must decide in the installation process which method you will use.

If you decide to allow MVS/QuickRef to dynamically allocate the main data base instead of pre-allocating it in the TSO logon procedure, the data base name must be provided in the MVS/QuickRef options table, QWIKOPTS.

The real power of MVS/QuickRef as an instant reference tool is that it can be invoked from any panel within any application under ISPF. In order for this to occur, the QW or QWS command must be added to the ISPF system command table. If you are running ISPF 4.2 or a later release of ISPF, then alternatively you could add the QW or QWS command to the ISPF command table or to the ISPF site command table. Since MVS/QuickRef will not be available from all panels under ISPF unless the ISPF system or site command table is modified, updating the ISPF system or site command table to add the QW or QWS command is highly recommended. If you choose not to modify the ISPF commands table, you can add MVS/QuickRef as an option to an existing application selection menu at your installation. See the topic “Alternative
MVS/QuickRef Invocation Methods” on page 95 in this chapter for a description of using the panel-based MVS/QuickRef invocation method.

**IMPORTANT: Upgrade Information for Existing Users**

If you will be installing this release of MVS/QuickRef as an upgrade rather than for the first time at your installation, then please read the information presented in the cautionary notes below. If you are installing MVS/QuickRef for the first time at your site, you may proceed directly to the installation steps that appear after this topic.

1. If you are currently using MVS/QuickRef R5.7 or higher, then you may perform either a full install or a ‘data base only’ install of this release.

2. Please note that as of MVS/QuickRef R6.2, the data set names used on the MVS/QuickRef distribution tape(s) all begin with the same high-level qualifier, ‘CSL’. If you are performing a full product install, you must use a version of the QWLOAD JCL procedure from a R6.2 or later JCL library. Older versions of the QWLOAD JCL procedure will not work with a R6.2 or later distribution tape.

3. If you are upgrading to this new release from a release older than release 5.3, please note that the format of the distribution tape has changed from that used in prior releases; in particular, the JCL library which was previously file one on the tape, is now file eight. You must use the initial tape download JCL found in this User's Guide on page 70 to download the MVS/QuickRef JCL library.

4. If you are upgrading to this release from a release older than release 5.3, please note that all modules, macros, source elements, panels, messages, and tables that comprise MVS/QuickRef have now been renamed to start with the prefix 'QWI', in support of SMP/E packaging and installation. This 'QWI' prefix has been registered with IBM so that no other IBM products or registered ISV products may use this prefix. One implication of this change is that the ISPF application ID used by MVS/QuickRef has been changed from QWRF to QWIK. You will need to execute the QWCMDS CLIST during the process of installing this release, so that the new ISPF application ID will be stored in the ISPF command table's QW and QWS entries (or the ISPF site command table's QW and QWS entries if applicable).

5. Release 5.2 or earlier users should note that two new data sets are now allocated during non-SMP/E install processing, an MVS/QuickRef macro library and a source library. These two data sets contain the macros and source, previously contained within the JCL library, that are used to assemble various MVS/QuickRef elements, such as the QWIKOPTS table and QWIKEEXIT module.

6. If you are upgrading to this release from release 5.1 or earlier, please note that we have converted the functions previously performed by the QWSDSF REXX exec into mainline Assembler code within MVS/QuickRef itself. For this reason, the QWSDSF command and the QWSDSF REXX exec have both been made obsolete. As a conversion aid for your use, we have provided a 'dummy' version of the QWSDSF REXX exec with this release. The 'dummy' version writes a message to the terminal when it is invoked, reminding the user to change their QWSDSF PF key equate to a QW PF key equate. The MVS/QuickRef support
group recommends that you copy this 'dummy' QWSDSF REXX exec to your CLIST/EXEC library after this release is installed, so that your users can easily transition to using the 'QW' command for all modes of access to MVS/QuickRef. The QW command is now used for cursor-driven invocation as well as for menu-driven and fast-path invocation. For this reason, you should have all of your MVS/QuickRef users change their PF key assignments so that the PF keys currently equated to QWSDSF are changed to the QW command. Implementing the new 'dummy' QWSDSF REXX exec into production along with R5.3 will remind them to do this.

7. If you are upgrading to this release from a release prior to release 5.3, please note that the MVS/QuickRef distribution tape is now in standard label (SL) format, not non-labeled (NL) format; also, three 3480 tapes or one 3490 tape are now needed. The tapes are plainly marked on their external labels. You must use the QWLOAD job or the SMP/E jobs that are shipped with this release to unload the product from the distribution tapes. Older versions of QWLOAD will not work!

8. Users upgrading from releases prior to release 5.3 must note that the MVS/QuickRef data base now contains information for more than one release of some products. This change was made to permit you to support multiple releases of various products with one copy of MVS/QuickRef, if you so desire. For example, the data base's MVS messages information includes messages for z/OS V1R2, z/OS V1R3, z/OS V1R4, and another set of MVS messages that apply to prior releases of MVS. Further, within the MVS message set, we have three different copies of the IEC1611 message; one for each previously mentioned MVS release. The most obvious effect of this change is that cursor-driven invocation of MVS/QuickRef may produce a list of items, rather than going directly to the item the cursor was on. You will probably want to closely examine the QEXCLUDE member in the MVS/QuickRef JCL library and consider excluding products from the data base that you do not have installed in your shop. Also, please note that as of R5.3, we have added a new facility called 'operating system level selection list processing', which will automatically choose the correct item from an item list for many data base items. You should read the discussion of this new feature on page 36 before moving this release of MVS/QuickRef into production.

9. If you are upgrading to this release from a release prior to release 5.0, you need to know that the names of the MVS/QuickRef main data base, user data base, and override parameter data sets can no longer be defined in the )ATTR section of the QWIKREFA panel. Support for this feature has been dropped. You must use the QWKOPTS options table to define the names of these data sets to MVS/QuickRef, or preallocate them in your logon procs, or use the LIBDEF CLIST to allocate them.

10. If you are upgrading from a release of MVS/QuickRef prior to release 5.0, then in addition to rebuilding your user data bases with the release 5.9 version of QWIKREF2, you may also need to consider converting the key indicator records in your user data base input files to release 5.3 format. If you have a relatively large user data base, we recommend converting your user data base key indicators to release 5.x format. This will allow your user data bases to take full advantage of the data base enhancements provided by release 5.3. A conversion utility has been provided to facilitate conversion of your user data base input from the release
4.3 and prior format to release 5.x format. The documentation for the conversion utility can be found in member QR5CVRT in the MVS/QuickRef JCL library. However, conversion is not absolutely required. If necessary, you can leave your key indicator records as they are and run QWIKREF2 with the OLDKEYS=Y parameter. If you do this, QWIKREF2 will put the Release 4.3 topic indicator in the product name field and leave the vendor name and release number fields blank. See the description of the QWIKREF2 JCL parameters in the section titled "Step 1 - Allocate and Load User Data Base(s)" in Chapter 5 for more details.

11. If you are upgrading from an MVS/QuickRef release prior to release 5.0, please note that the syntax and structure of the MVS/QuickRef override parameter statements has changed. If you are using override parameters, you MUST modify them so that they are in the format required for release 5.9. The data display programs for this release cannot read override parms formatted for a release prior to release 5.0. If you are upgrading to this release from a release prior to release 3.2, then you must also add the UDBN= parameter to any REPLACE, AUGMENT, PREVENT, and ALLOW override statements you have in use for your user data base. The UDBN= parameter is described in detail on page 199 of this guide.

12. If you are upgrading to this MVS/QuickRef release from a release prior to 5.0, please note that the MVS/QuickRef direct program call interface has had a fourth parameter added to the parm list that is passed to MVS/QuickRef. If you are using direct program call, then you will need to change the parm list being passed to use the new format.

13. If you are upgrading from a release prior to 4.2, please note that the panel changes previously required when installing the QWSDFS cursor driven invocation facility are no longer necessary, since the QWSDFS REXX exec has been eliminated. Any panel updates made when you installed a prior MVS/QuickRef release do not have to be removed, but can be removed if you so desire.

CA-ROSCOE Users

If your installation uses CA-ROSCOE, you will want to make MVS/QuickRef available under CA-ROSCOE, and the distribution tape contains all the elements you need to do this. First, you must install MVS/QuickRef in the MVS environment as outlined in "The Installation Process" section which follows. Then, you will download the CA-ROSCOE support file and make MVS/QuickRef available in your CA-ROSCOE environment as discussed in "Using MVS/QuickRef Under CA-ROSCOE" section on page 130 later in this chapter.

The Installation Process

In the installation process description that follows, you may see references to background material which is contained later in this chapter. The background material is designed to provide more information on specific topics you will need to understand while installing MVS/QuickRef. You will prevent problems and ensure a smoother product installation process if you take the time to read each background material item when it is referred to.
Unload MVS/QuickRef JCL

To begin the installation process, key in and submit the JCL below to copy the MVS/QuickRef JCL elements from the distribution tape to DASD. Don’t forget to add a JOB statement, change "hlq" to a valid high-level qualifier, and change the unit names to comply with your local JCL standards. Please note that the distribution tape containing the MVS/QuickRef data sets is a standard-labeled (SL) tape.

When the job ends, check it for a successful completion. A good way to do this is to browse the output listing online using SDSF (or some equivalent product) and to issue a find command for 'COMP CODE'. The return code will be shown and should be 0.

If you do not get a zero return code, then you must look at the job output to determine what happened. Often, a return code of 4 means the job ran successfully but something unexpected occurred. If the return code is 8 or above, you usually must resolve the problem and rerun the job. To determine the cause of an error, first look in the JES JOB LOG for messages or other indications that something unusual happened. If you find nothing out of the ordinary, then look at the output of the program (in this case IEBCOPY) which is at the end of the job. Look for messages or error conditions that affected the job. Correct any errors and re-execute the job if necessary.

When the JCL dataset has been copied, be sure to browse the $QWINFO member loaded to DASD and review the information it contains for last-minute notes on this release.

Choosing an Installation Method

At this point, you will need to determine which method you will use to perform the installation. MVS/QuickRef can be installed in any of the following ways:

- Using our installation assistance ISPF dialog, QWIAD, to automate the install process. The installation assistance dialog QWIAD supports both the SMP/E and non-SMP/E installation methods. It automatically tailors batch job JCL according to your specifications and guides you through the entire installation, and it is fully restartable. This is the recommended install method!
- Manually submitting jobs to perform the install using SMP/E for the MVS/QuickRef elements
- Manually submitting jobs to perform the install using the standard, non-SMP/E method
- Manually submitting a job to download only the MVS/QuickRef data base.

As shown by the installation methods outlined above, you must also decide whether or not to use SMP/E to do the install. MVS/QuickRef does not normally require a lot of maintenance once it has been installed. This means that SMP/E’s “maintenance tracking” capabilities are not really needed for MVS/QuickRef and, for this reason, the use of SMP/E is entirely optional. Unless your installation has a standard requiring the use of SMP/E or you are a knowledgable SMP/E user, use of SMP/E to install MVS/QuickRef is not recommended. If you are required to use SMP/E due to an installation standard and are not a knowledgable SMP/E user, then it is recommended that you find an experienced SMP/E user to assist you.

Each of the installation methods is presented below. You should now choose one of these methods and follow the steps listed within it to install MVS/QuickRef.

**Installation Method 1 - Using QWIAD**

QWIAD was designed to provide assistance with installation. It consists of panels, messages, and REXXX routines which are contained within the JCL data set. When invoked, QWIAD prompts you for required information which it then uses to construct and tailor JCL or control statements. It will then guide you through the MVS/QuickRef installation process. Several batch jobs are tailored using information you supply, after which you will be asked to submit the jobs for execution. QWIAD will wait for each job to end; you should split screen and ensure that each batch job executes successfully before proceeding.

Prior to starting the installation process, you should answer all questions in the QWIAD Installation Checklist section which follows. Once you have done so, invoke QWIAD by following the instructions in the "Invoking QWIAD" section which follows the checklist.

**QWIAD INSTALLATION CHECKLIST**

QWIAD will prompt you for the following information. Although QWIAD provides default responses for most of the questions, if you take the time to indicate your responses now, it will make the installation process proceed much more quickly.

1. Job statement Information: Programmer Name______________________________
   Account Number ____________________________
   Job Class ________________________________
Message Class
Notify Value
Other keyword= Value

2. Would you like QWIAD to generate JCL to support SMS (Y/N)?: __________
   If you answer 'N', skip to checklist item number 4, otherwise continue with next question.

3. Under SMS, which JCL keywords should be used for output data sets? (Y or N)
   PERMANENT ___ TEMPORARY ___
   DSN=data-set-name ___ DISP=disposition ___
   DATACLASS=data-class ___ MGMTCLASS=management-class
   STORCLASS=storage-class ___ UNIT=unit-name ___
   VOL=SER=volume ___ DCB=dcb-specification ___
   SPACE=space-designation ___

4. MVS/QuickRef Data Set Information:
   non-SMS
   DASD Unit on which to create datasets ________________________________
   DASD Volser on which to create data sets ______________________________
   SMS
   PERMANENT _________________________________________________________
   TEMPORARY _________________________________________________________
   DATACLASS _________________________________________________________
   MGMTCLASS _______________________________________________________
   STORCLASS _________________________________________________________
   TAPE Unit for distribution tape _________________________________
   TAPE Volser of distribution tape _________________________________
   High-Level Qualifier of Q/R data sets _______________________________
   Foreign tape indicator (1 or 2):
   { 1 if LABEL=(n,SL,EXPDT=98000) or 2 if SPACE=(1,(1,1)) }

5. Type of Install (choose 1): Full Install ___ Database Only ___ CA-Roscoe-only ___

6. Do you wish to use SMP/E to install MVS/QuickRef (Y/N)?: __________
   If you answer 'N', skip to checklist item number 8, otherwise continue with next question.

7. Do you wish to create new SMP/E data sets for MVS/QuickRef use (Y/N)?: __________
   If not, specify the names of existing SMP/E data sets along with their target and DLIB boundary names. If yes, you may specify the new data set names and boundary names or let QWIAD provide default choices:
   SMP/E CSI DSN ________________________________
   SMP/E LOG DSN ________________________________
   SMP/E LOGA DSN ________________________________
   SMP/E SCDS DSN ________________________________
   SMP/E MTS DSN ________________________________
SMP/E PTS DSN
SMP/E STS DSN
SMP/E Target Boundary Name
SMP/E DLIB Boundary Name

non-SMS
SMP/E DASD Unit
SMP/E DASD Volume Serial Number

SMS
PERMANENT
TEMPORARY
SMP/E DATACLAS
SMP/E MGMTCLAS
SMP/E STORCLAS

8. Do you wish to update either the ISPF or site command Table (Y/N)?
   If not, continue with checklist item number 10, otherwise continue with next step.
   (For additional information that can help you make this decision, see the "ISPF Command Tables" section on page 94.)

9a. If you wish to update the ISPF Command Table, provide this information:
   Which table, the System or Site command table? (Y/N)?
   Will you use LIBDEF (Y/N)?
   (if you don't know what LIBDEF is, refer to the section titled 'Using LIBDEF' on page 96)
   If Site Table, Site Table ID
   If Site Table, are you creating or are you updating?
   Which commands (choose 1) QW
   QWS
   QW & QWS

9b. If updating System or Site Table, data set where ISPCMDS/site table resides:

9c. If you elect to apply our USERMOD to update the ISPF Cmd Table with SMP/E, you must indicate where ISPF's SMP/E information resides:
   ISPF FMID
   ISPF Global CSI DSN
   ISPF Global Boundary
   ISPF Target Boundary

10. If you are not updating the ISPF Cmd Table or the Site command table, will you use LIBDEF to invoke MVS/QuickRef? (Y/N)?

11. After the install process is completed, do you want QWIAD to assist you in making the MVS/QuickRef elements available to your users (Y/N)?
If not, continue with checklist item 13, else go on to the next question.

12. Since you answered 'Y' to checklist question 11, how will you be making MVS/QuickRef elements available to the system? (choose one of the following three, either A, B, or C):
   A) _____ Using LIBDEF  CLIST library where the CLISTs will reside _______________
       Name of the tables library where the tables will reside ________________________
   B) _____ Copying elements to existing system data sets defined in LOGON PROCs
       Name of the panels library where the panels will reside _______________________
       Name of the messages library where the messages will reside__________________
       Name of the tables library where the tables will reside_______________________
       Name of the link library where the load modules will reside__________________
   C) _____ Adding Q/R data sets to logon procedure or to CLIST invoked at LOGON
       (If so, you must manually update the logon procs)

13. Do you wish to change the MVS/QuickRef options table QWIKOPTS (Y/N)?___
   (see the “Setting MVS/QuickRef Global Installation Options” section starting on page 97 )
   If not, continue with checklist question 15, else go on to next question.

14. Which QWIKOPTS keywords do you wish to modify : __________________________
     _______________________________________________________________________

15. Do you wish to apply the expiration date/CPU ID zap (Y/N)? ____________________
   If not, continue with question 17, else go on to the next question.

16. What is the REP portion of the zap?
   For non-SMP/E installation, what is the CHECKSUM?________________________

17. Do you wish to download the full database or exclude various vendors, products, and / or releases (FULL/PARTIAL)? ________________________________
   If FULL, continue with checklist question 19, else go on to the next checklist question.

18. Which data base items do you wish to include or exclude? (You must modify JCL library member QEXCLUDE to include or exclude products from the data base; see the "Selective Data Base Load Facility" on page 120 and member QEXCLUDE in the MVS/QuickRef JCL library for more information):

19. Do you wish to copy the CA-Roscoe support files to DASD (Y/N)? ______________

20. Do you wish to have the ability to invoke MVS/QuickRef as a started task from an operator's console (Y/N)? _____ If yes, what is the name of the JES proclib where the MVS/QuickRef started task JCL should be stored? ____________________________

**INVOKING QWIAD**

Once you have answered the questions listed in the checklist above, invoke QWIAD by entering the following command from the ISPF command line or under ISPF option 6:
TSO EXEC 'hlq.JCL(QWIAD)'

where 'hlq' is the high-level qualifier of the JCL data set. QWIAD will then execute and guide you through the required product installation steps.

If you terminate QWIAD without executing it to its full completion, QWIAD will restart at the closest possible point to where you left off (unless you chose to restart from the beginning). When you restart, the values you entered previously will be used as defaults whenever possible.

You can force a restart of QWIAD by deleting all lines in the CHKPOINT member in the JCL data set.

You should now refer to the following sections for additional information you must consider: 'Security System Considerations' on page 95, 'Customizing MVS/QuickRef' on page 96, 'Setting MVS/QuickRef Global Installation Options' on page 97, 'Customizing the Vendor/Product Specific Selection List Table' on page 106, 'Preparing to Test MVS/QuickRef' on page 109, 'Testing MVS/QuickRef' on page 110, and 'User Documentation Considerations' on page 114.

When QWIAD completes execution, the MVS/QuickRef installation process is complete.

Installation Method 2 - Manual Installation with SMP/E

This section describes manual installation of MVS/QuickRef using SMP/E. The MVS/QuickRef Support Group highly recommends that instead of manual installation, you use the MVS/QuickRef Installation Assistance Dialog QWIAD. To use the dialog, go to the "Installation Method 1 - Using QWIAD" section on page 71. If you prefer to perform a manual installation, follow step one below through step eleven on page 84.

After you have successfully unloaded the MVS/QuickRef JCL library from tape, you will need to edit and submit several jobs in order to set up the SMP/E environment. You can install MVS/QuickRef into existing program product SMP/E data sets or create new ones for MVS/QuickRef. The MVS/QuickRef Support Group recommends that you create a new SMP/E environment for MVS/QuickRef use; this will facilitate an easier transition between releases.

Follow these steps in this order:

1. SMPALLOC
   This job is located in the MVS/QuickRef JCL library as member SMPALLOC. It allocates the MVS/QuickRef target and distribution data sets. Although you could specify existing MVS/QuickRef data sets, this is not recommended as each release may add or remove modules.
Read the comments in the top of SMPALLOC and make changes as directed. Ensure you have a valid JOB statement and change the symbolic parameters as appropriate to meet your local standards. Then, submit SMPALLOC.

Browse the output of this job and locate the return code. The highest acceptable return code is 0.

2. SMPCSI
This job is located in the MVS/QuickRef JCL library as member SMPCSI. It allocates the data sets required by SMP/E. Although you could specify existing SMP/E data sets, in order to facilitate an easier transition between releases, this is not recommended.

Read the comments in the top of SMPCSI and make changes as directed. Ensure you have a valid JOB statement and change the symbolic parameters as appropriate to meet your local standards. In addition, be sure to change the VOLUMES parameter in the IDCAMS statements. Then, submit SMPCSI.

Browse the output of this job and locate the return code. The highest acceptable return code is 0.

3. SMPDDDEF
This job is located in the MVS/QuickRef JCL library as member SMPDDDEF. It initializes the SMP/E data sets created by SMPCSI. If you are using existing SMP/E data sets, this job is not required.

Read the comments in the top of SMPDDDEF and make changes as directed. Ensure you have a valid JOB statement and change the symbolic parameters as appropriate to meet your local standards. Be sure you specify the correct target and DLIB boundary names. Then, submit SMPDDDEF.

Browse the output of this job and locate the return code. The expected return code is 4.

4. SMPRECV
This required job is located in the MVS/QuickRef JCL library as member SMPRECV. It performs the SMP/E RECEIVE function on the MVS/QuickRef FUNCTION SYSMOD.

Read the comments in the top of SMPRECV and make changes as directed. Ensure you have a valid JOB statement and change the tape unit name, tape volume serial number, and data set names as appropriate to meet your local standards. Then, submit SMPRECV.

Browse the output of this job and locate the return code. The highest expected return code is 0.
5. SMPAPPLY
This required job is located in the MVS/QuickRef JCL library as member SMPAPPLY. It performs the SMP/E APPLY function on the MVS/QuickRef FUNCTION SYSMOD.

Read the comments in the top of SMPAPPLY and make changes as directed. Ensure you have a valid JOB statement and change to target boundary name and CSI data set name as appropriate to meet your local standards. Then, submit SMAPPLY.

Browse the output of this job and locate the return code. The highest expected return code is 0.

6. SMPACCEPT
This required job is located in the MVS/QuickRef JCL library as member SMPACEPT. It performs the SMP/E ACCEPT function on the MVS/QuickRef FUNCTION SYSMOD.

Read the comments in the top of the SMPACEPT member and make changes as directed. Ensure you have a valid JOB statement and change to target boundary name and CSI data set name as appropriate to meet your local standards. Then, submit SMACEPT for execution.

Browse the output of this job and locate the return code. The highest expected return code is 0.

7. QWUPDATE
The SMP/E process does not download the MVS/QuickRef data base; thus, QWUPDATE is a required job. The JCL is located in the MVS/QuickRef JCL library as member QWUPDATE. It executes the QWIKSLCT program to copy the MVS/QuickRef data base to DASD.

Read the comments in the top of QWUPDATE and make changes as directed. Ensure you have a valid JOB statement and change the tape and DASD unit names, the tape and DASD volume serial numbers, and the name of the MVS/QuickRef data base as appropriate to meet your local standards.

Before submitting the QWUPDATE job for execution, consider whether you want to drop some information from the MVS/QuickRef main data base for those products that you do not want or need. If you want to omit some of the reference information from the MVS/QuickRef main data base, you should stop now and read the section “Selective Data Base Load Facility” on page 120. In order to make the best decision about whether or not to drop information from the main data base, as well as what to drop, you should also review the section "Selective Load Versus User-Directed Selection List Processing" on page 125. After you have read the information in both of these sections, then, if you do want to omit some reference information from the main data base, do the following:
• customize the QEXCLUDE member in the MVS/Quickref JCL data set to match your exclusion criteria, as described beginning on page 121 of this guide.

• change the SYSIN DD DUMMY statement in PS10 of the QWUPDATE job so that it refers to the QEXCLUDE member containing the include/exclude control statements you customized.

Note: If you do not provide the QWIKSLCT program with one or more include/exclude control statements, it will simply copy the entire MVS/QuickRef main data base "as is". If you do want the entire data base loaded to DASD, leave the SYSIN DD statement dummied out in PS10 of the QWUPDATE job.

Submit the QWUPDATE job for execution.

After the QWUPDATE job has executed, review its output and locate the return code. The desired return code is 0; however, a return code of 4 is acceptable. If the return code is greater than 4, determine the cause of the error, correct it, and re-run the job.

8. DEFINE DATA SET NAME OF MAIN DATA BASE
If you will be using the ISPF LIBDEF technique to access MVS/QuickRef’s elements, do **not** perform this step as the MVS/QuickRef main data base will be allocated by the LIBDEF CLIST. LIBDEF users should refer to the discussion of the “QW” CLIST on page 96 then continue with the next step.

The data set name of the MVS/QuickRef main data base is defined to the product in the MVS/QuickRef options table, QWIKOPTS. The data base data set specified in the QWIKOPTS table will be dynamically allocated at execution time.

Note: Dynamic allocation of the data base can be bypassed by preallocating it in your TSO logon procs with a DD name of QWREFDD, as follows:

```
//QWREFDD DD DSN=xxxx.xxxx.xxxx,DISP=SHR
```

If you elect to preallocate the QWREFDD DD statement in your TSO logon procs, then you do not have to specify the main data base data set name in the QWIKOPTS options table; however, this can be done as a safety measure in case the QWREFDD DD is freed by an ISPF user.

To specify the name of the MVS/QuickRef main data base in the QWIKOPTS options table, create valid IEBUPDTE control statements to modify the QDB= keyword parameter (along with any other desired parameter). Then, incorporate these IEBUPDTE control statements as a source change in the QWINUOPT member in the MVS/QuickRef JCL library. After updating QWINUOPT, save the member.
Then, use member SMPRAP3 in the MVS/QuickRef JCL library to receive and apply a usermod to reassemble the module. Read the comments in the top of SMPRAP3 and make changes as directed. Ensure you have a valid JOB statement, add the MVS/QuickRef target boundary, and change the data set names as appropriate to meet your local standards. Then, submit SMPRAP3.

Browse the output of this job and locate the return code. The highest expected return code is 0.

9. MODIFY ISPF SYSTEM COMMAND TABLE OR SITE COMMAND TABLE

In this step, we will add the ISPF command(s) required to invoke MVS/QuickRef (QW and/or QWS) to the appropriate ISPF command table. We will do this by using the QWCMDS CLIST in the MVS/QuickRef JCL library. This CLIST will prompt you for the information required to complete the command table update and will make sure the updates are done in the correct format.

Before continuing with this step, please review the information in the section titled "ISPF Command Tables" on page 94 later in this chapter. It is essential that you understand how ISPF command tables are used before attempting this update.

Once you have completed reviewing the information in the section on "ISPF Command Tables", the next thing you must do is determine if you want to add both commands (QW and QWS) to the command table. The function provided by the QWS command is only available for ISPF release 4.2 or higher. So, if you are running an ISPF release prior to 4.2, then your only option is to add the QW command. If you are running ISPF release 4.2 or a later release, then you have the option of adding either one or both of the two commands. For a description of the two commands and their functions, see the section titled "Invocation Commands" on page 25 in Chapter Two.

The next thing you must do is determine if you will be adding the command(s) to the ISPF system command table or the ISPF site command table. The site command table is only available with ISPF release 4.2 or higher; therefore, if you are utilizing an ISPF release prior to 4.2, then your only option is to update the system command table. If you plan to use the site command table, then you must determine if you are going to be updating an existing site command table or creating the site command table for the first time. Site command tables and how to define them to ISPF are documented in the ISPF Planning and Customization Guide.
If you want to update the system command table, then you will need to use
ISPF/PDF menu option 3.3 to copy the ISPCMDS table member from the production
ISPF table library1 (ISPTLIB) to a partitioned data set over which you can obtain
exclusive control. If you want to update an existing site command table, then you
will need to copy your site command table to a partitioned data set over which you
can obtain exclusive control.

Next, invoke the QWCMDS CLIST from the MVS/QuickRef JCL library under
ISPF with the TSO EXEC command, as follows:

EXEC 'local.QUICKREF.JCL(QWCMDS)'

You MUST execute this CLIST under ISPF because it uses ISPF commands. You
MUST be able to obtain exclusive control over the data set or the TSO RENAME
command in the QWCMDS CLIST will fail.

The QWCMDS CLIST will prompt you for the following information:

▪ whether or not you are updating the system command table or the site command
table

▪ if you specify the site command table, then it will prompt you for the applid
assigned to the site command table and will ask if you want to update an existing
site command table or create a new one

▪ if you indicate that you are updating the system command table or an existing
site command table, then it will prompt you for the name of the PDS containing
the working copy of the specified command table member; if you indicate that
you are creating a new site command table, then it will prompt you for the name
of the PDS where you want the new site command table member to be stored

▪ whether or not you will be using the LIBDEF CLIST method to invoke
MVS/QuickRef; if you don't know or are not sure, respond "No".

▪ whether or not you want to add the QW command to the command table being
updated or created

▪ whether or not you want to add the QWS command to the command table being
updated or created

1The ISPCMDS copy you use should be the production copy, which will be the first one found
in the list of data sets concatenated to the ISPTLIB DD in your TSO LOGON procedures. If your
company has NOT made any prior changes to the ISPCMDS table, then the “vanilla” copy of
it can be found in data set ISP.VnRnMn.ISPTENU for most releases of ISPF.
The actual questions asked by QWCMDS will be dictated by the release of ISPF that you are using. If you are using a release of ISPF prior to 4.2, then the site command table and the function provided by the QWS command are not supported. In this case, QWCMDS will assume that the QW command is the only command option and that it is to be added to the system command table.

Based upon the release of ISPF that you are using and your responses to the questions which it asks, QWCMDS will either update the existing system or site command table member in place or create a new site command table member and place it in the specified PDS.

WARNING: Do not attempt to update the production copy of the system or site command table; update your working copy instead.

Once the specified ISPF system or site command table update is completed, the QWCMDS CLIST will then ask if you want to copy the MVS/QuickRef application command table (QWIKCMDS) to a PDS in the ISPTLIB concatenation. We recommend that you use QWCMDS to copy the tables. You can take care of adding QWIKCMDS to the ISPTLIB concatenation manually if you prefer; the QWCMDS CLIST simply provides a convenient means of getting this done (and a reminder that it must be done in order for MVS/QuickRef to function properly).

Note 1: Any time you use the QWCMDS CLIST to update an existing command table, make sure you use it to update a "fresh" copy of the command table (that is, one which does not already contain the QW or QWS command entries).

Note 2: The data set updated by the QWCMDS CLIST cannot be closed by the QWCMDS CLIST. The data set remains allocated after QWCMDS completes processing. For this reason, you must log off of TSO and log back on again before using the QWCMDS CLIST for a second time so that the data set containing the working copy of the command table will be closed and freed.

If you updated ISPCMDS and not a site command table, you may use member SMPRAP2 in the MVS/QuickRef JCL library to receive and apply a usermod to "protect" ISPCMDS from inadvertent updates.

First, you must edit member QWIISPF in the MVS/QuickRef JCL library and specify the correct FMID for the release of ISPF that you run at your installation. The FMID must be placed in the FMID parameter on the ++VER statement as shown here:

```
++VER(Z038) FMID(HIF5A02)    /* For these ISPF FMIDs */.
```
If you don't know what the FMID is for your release of ISPF, here is a list of some of them; you can see that the fourth and fifth characters of the FMID are generally the ISPF release number:

<table>
<thead>
<tr>
<th>FMID</th>
<th>Release</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIF1102 ISPF V1 R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF2102 ISPF V2 R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF2104 ISPF V2 R1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF2105 ISPF V2 R1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF2202 ISPF V2 R2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF2302 ISPF V2 R3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF3102 ISPF V3 R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF3202 ISPF V3 R2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF3302 ISPF V3 R3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF3502 ISPF V3 R5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF4102 ISPF V4 R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF4202 ISPF V4 R2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF4302 ISPF V4 R3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF4402 ISPF V4 R4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF4502 ISPF V4 R5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF4602 ISPF V4 R6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF4702 ISPF V4 R7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF4802 ISPF V4 R8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF4902 ISPF V4 R9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF5A02 ISPF V5 R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF5B02 ISPF V5 R2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIF5C02 ISPF V5 R3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Save the changed QWIISPF member.

Next, read the comments in the top of SMPRAP2 and make changes as directed. Ensure you have a valid JOB statement, add the MVS GLOBAL CSI data set name and the target boundary where ISPF is defined, and change the data set names as appropriate to meet your local standards. Then, submit SMPRAP2.

Browse the output of this job and locate the return code. The highest expected return code is 0.

10. EQUATE PF KEYS

Note: ***If you are upgrading to this release of MVS/QuickRef from a release prior to R5.3, please note that the functions previously performed by the QWSDFS REXX exec have been merged into the main body of MVS/QuickRef code. As a conversion aid for your user community, you should copy the QWSDFS member from the JCL library to your production CLIST or EXEC library. The QWSDFS member reminds your users to change their PF key equates for the QWSDFS command to equates for the QW command.***

To make MVS/QuickRef even easier to use, you should equate a PF key to the MVS/QuickRef invocation command (QW or QWS) that you intend to use most often, and remind each one of the MVS/QuickRef users in your enterprise to do the same. This is especially true for cursor-driven invocation. With a PF key equated to your preferred MVS/QuickRef invocation command, all you have to do to use
cursor-driven invocation is place the cursor underneath a character string representing an MVS/QuickRef item name and then press that PF key.

Of course, since PF keys are defined separately for each different ISPF application, this means that you must equate a PF key to your preferred MVS/QuickRef invocation command in each ISPF application where you want to use that PF key. If you want to invoke MVS/QuickRef via a PF key in PDF EDIT, for example, then you must do the PF key equate while you are in PDF EDIT. If you also want to invoke MVS/QuickRef via a PF key while in SDSF, then you must do another PF key equate while in SDSF. Once you are in a given ISPF application, you can use the KEYS command or, if the application is a CUA-style application, the KEYLIST command to establish the appropriate PF key equate.

As an example, suppose, for a given ISPF application, that you decide to use PF key 23 to invoke MVS/QuickRef and that you want to use the QW command. From within that application, you issue the KEYS or KEYLIST command and get a display of the currently defined PF keys. The figure which follows shows how this PF key display might appear after you type in the appropriate PF key equate for PF key 23.

--- PF KEY DEFINITIONS AND LABELS - PRIMARY KEYS ---
COMMAND ===>
NUMBER OF PF KEYS ===> 24                  TERMINAL TYPE ===>
3278
PF13 ===> HELP
PF14 ===> SPLIT
PF15 ===> END
PF16 ===> RETURN
PF17 ===> IFIND
PF18 ===> FIND ' - COND CODE'
PF19 ===> UP
PF20 ===> DOWN
PF21 ===> SWAP
PF22 ===> LEFT
PF23 ===> QW                              /* <= PF key equate for MVS/QuickRef */
PF24 ===> RETRIEVE

Note: It is best if you do not use a PF key equated to 'TSO %QW' since this method of invocation bypasses ISPF parsing of the command line and hence is not reliable in all cases.
11. APPLY THE EXPIRATION DATE / CPU ID ZAP

The MVS/QuickRef product is distributed in an "expired" status. The distribution package will contain a yellow sheet containing your expiration date and / or CPU ID zap. Locate this sheet and use the appropriate following method to apply the zap.

Edit member QWIEXPDT in the MVS/QuickRef JCL library and key in the REP portion of the zap on the REP 05 statement in place of the question marks as shown here:

```
REP 05 ?????,????,????,??
```

Please note that it is normal for this zap's VER and REP offsets not to match. Save the QWIEXPDT member.

Edit member SMPRAP4 in the MVS/QuickRef JCL library. This member is used to receive and apply the QWI0003 usermod you just edited in member QWIEXPDT, causing SMP/E to apply the MVS/QuickRef expiration date / CPU ID zap.

Read the comments in the top of the member and make changes as directed. Remember to add a valid JOB statement, add the MVS/QuickRef target boundary name, and change the MVS/QuickRef data set names as appropriate to meet your local standards, then submit the job for execution. Expect a return code of 0 from this job.

12. MAKE MVS/QUICKREF ELEMENTS AVAILABLE TO SYSTEM

If you will be LIBDEF'ing MVS/QuickRef, you should refer to the discussion of LIBDEF found on page 96 of this guide.

If you are not going to use the LIBDEF invocation technique, then you may now use member PRODCOPY in the MVS/QuickRef JCL library to manually copy the entire MVS/QuickRef panel, message, table, and program libraries into the existing central ISPF application libraries at your installation, or add them to the ISPPLIB, ISPMLIB, ISPTLIB, and ISPLLIB concatenations in your LOGON procs.

Regardless of whether you move the MVS/QuickRef elements or use them directly from the target libraries created by the SMPALLOC job and the QWUPDATE job, all elements must be accessible before MVS/QuickRef can use them. The following table shows the MVS/QuickRef data set name along with the DD statement name that must be used for each product data set.

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>DD Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>hlq.QWIPNL  (panel library)</td>
<td>ISPPLIB</td>
</tr>
<tr>
<td>hlq.QWIMSG  (message library)</td>
<td>ISPMLIB</td>
</tr>
<tr>
<td>hlq.QWITBL  (table library)</td>
<td>ISPTLIB</td>
</tr>
<tr>
<td>hlq.QWILINK  (load module library)</td>
<td>ISPLLIB</td>
</tr>
</tbody>
</table>
You should now refer to the following sections for additional information you must consider: 'Security System Considerations' on page 95, 'Customizing MVS/QuickRef' on page 96, 'Setting MVS/QuickRef Global Installation Options' on page 97, 'Customizing the Vendor/Product Specific Selection List Table' on page 106, 'Preparing to Test MVS/QuickRef' on page 109, 'Testing MVS/QuickRef' on page 110, and 'User Documentation Considerations' on page 114.

This ends the MVS/QuickRef manual installation method using SMP/E.

Installation Method 3 - Manual Installation without SMP/E

This section describes manual installation of MVS/QuickRef using the regular, non-SMP/E method. The MVS/QuickRef Support Group highly recommends that instead of manual installation, you use the MVS/QuickRef Installation Assistance Dialog QWIAD. To use the dialog, go to the "Installation Method 1 - Using QWIAD" section on page 71. If you prefer to perform a manual installation without SMP/E, follow steps one on this page through step five on page 93.

After you have successfully unloaded the MVS/QuickRef JCL library from tape, you will need to edit and submit the following jobs in the order listed to completely install MVS/QuickRef:

1. **QWLOAD**

   Edit member QWLOAD, which will allocate the other MVS/QuickRef data sets and copy the remaining elements of MVS/QuickRef as outlined in the proc step descriptions below:

   - **PS10** - Uses IEBCOPY to copy the MVS/QuickRef panels to the MVS/QuickRef panel library or an existing ISPF panel library.
   - **PS20** - Uses IEBCOPY to copy the MVS/QuickRef messages to the MVS/QuickRef message library or an existing ISPF message library.
   - **PS30** - Uses IEBCOPY to copy the MVS/QuickRef command table to the MVS/QuickRef table library or your local ISPF table library.
   - **PS40** - Uses IEBCOPY to copy required MVS/QuickRef macros to the MVS/QuickRef macro library or an existing macro library.
   - **PS50** - Uses IEBCOPY to copy required MVS/QuickRef source to the MVS/QuickRef source library or an existing source library.
   - **PS60** - Uses IEBCOPY to copy the MVS/QuickRef programs to the MVS/QuickRef program library or an existing program library.
   - **PS70** - Uses QWIKSLCT to copy the MVS/QuickRef main data base to DASD.

   Read the comments in the top of the QWLOAD member and make changes as directed. Remember to add a valid JOB statement and change the unit names as appropriate to meet your local standards.
Before submitting the QWLOAD job for execution, consider whether you want to drop some information from the MVS/QuickRef main data base for those products that you do not want or need. If you want to omit some of the reference information from the MVS/QuickRef main data base, you should stop now and read the section “Selective Data Base Load Facility” on page 120. In order to make the best decision about whether or not to drop information from the main data base, as well as what to drop, you should also review the section "Selective Load Versus User-Directed Selection List Processing" on page 126. After you have read the information in both of these sections, then, if you do want to omit some reference information from the main data base, do the following:

- customize the QEXCLUDE member in the MVS/Quickref JCL data set to match your exclusion criteria, as described beginning on page 120 of this guide

- change the SYSIN DD DUMMY statement in PS70 so that it refers to the QEXCLUDE member containing the include/exclude control statements you just customized.

Submit the QWLOAD job for execution.

Note: If you do not provide the QWIKSLCT program with one or more include/exclude control statements, then it will simply copy the entire MVS/QuickRef main data base as is. If you do want the entire data base loaded to DASD, leave the SYSIN DD statement dummied out in PS70.

After the QWLOAD job has executed, review its output to ensure that all elements were successfully copied from the distribution tape and that all steps ended with a return code of zero.

2. DEFINE DATA SET NAME OF MAIN DATA BASE

If you will be using the ISPF LIBDEF technique to access MVS/QuickRef’s elements, do not perform this step as the MVS/QuickRef main data base will be allocated by the LIBDEF CLIST. LIBDEF users should refer to the discussion of the “QW” CLIST on page 95 and then continue with the next step.

The data set name of the MVS/QuickRef main data base is defined to the product in the MVS/QuickRef options table, QWIKOPTS. The data base data set specified in the QWIKOPTS table will be dynamically allocated at execution time.

Note: Dynamic allocation of the data base can be bypassed by preallocating it in your TSO logon procs with a DD name of QWREFDD, as follows:

```bash
//QWREFDD DD DSN=xxxx.xxxx.xxxx,DISP=SHR
```
If you elect to preallocate the QWREFDD DD statement in your TSO logon procs, then you do not have to specify the main data base data set name in the QWIKOPTS options table; however, this can be done as a safety measure in case the QWREFDD DD is freed by an ISPF user.

To specify the name of the MVS/QuickRef main data base in the QWIKOPTS options table, modify the QDB= keyword parameter (along with any other desired parameter) and assemble the QWIKOPTS member in the MVS/QuickRef source library after reviewing the information on the QWIKOPTS table contained in the section "Setting MVS/QuickRef Global Installation Options" on page 97 in this chapter.

Member QWOPTASM in the MVS/QuickRef JCL library can be used to perform the reassembly and link. Read the comments in the top of the member and make changes as directed. Remember to add a valid JOB statement, then submit the job for execution. Expect a return code of 0 from this job.

3. MODIFY ISPF SYSTEM COMMAND TABLE OR SITE COMMAND TABLE

In this step, we will add the ISPF command(s) required to invoke MVS/QuickRef (QW and/or QWS) to the appropriate ISPF command table. We will do this by using the QWCMDS CLIST in the MVS/QuickRef JCL library. This CLIST will prompt you for the information required to complete the command table update and will make sure the updates are done in the correct format.

Before continuing with this step, please review the information in the section titled "ISPF Command Tables" on page 93 later in this chapter. It is essential that you understand how ISPF command tables are used before attempting this update.

Once you have completed reviewing the information in the section on "ISPF Command Tables", the next thing you must do is determine if you want to add both commands (QW and QWS) to the command table. The function provided by the QWS command is only available for ISPF release 4.2 or higher. So, if you are running an ISPF release prior to 4.2, then your only option is to add the QW command. If you are running ISPF release 4.2 or a later release, then you have the option of adding either one or both of the two commands. For a description of the two commands and their functions, see the section titled "Invocation Commands" on page 24 in Chapter Two.

The next thing you must do is determine if you will be adding the command(s) to the ISPF system command table or the ISPF site command table. The site command table is only available with ISPF release 4.2 or higher. So, if you are running an ISPF release prior to 4.2, then your only option is to update the system command table. If you plan to use the site command table, then you must determine if you are going to be updating an existing site command table or creating the site command table for the first time. Site command tables and how to define them to ISPF are documented in the ISPF Planning and Customization Guide.
If you want to update the system command table, then you will need to use ISPF/PDF menu option 3.3 to copy the ISPCMDS table member from the production ISPF table library\(^2\) (ISPTLIB) to a partitioned data set over which you can obtain exclusive control. If you want to update an existing site command table, then you will need to copy your site command table to a partitioned data set over which you can obtain exclusive control.

Next, invoke the QWCMDS CLIST from the MVS/QuickRef JCL library under ISPF with the TSO EXEC command, as follows:

```
EXEC 'local.QUICKREF.JCL(QWCMDS)'
```

You **MUST** execute this CLIST under ISPF because it uses ISPF commands. You **MUST** be able to obtain exclusive control over the data set or the TSO RENAME command in the QWCMDS CLIST will fail.

The QWCMDS CLIST will prompt you for the following information:

- whether or not you are updating the system command table or the site command table

- if you specify the site command table, then it will prompt you for the applid assigned to the site command table and will ask if you want to update an existing site command table or create a new one

- if you indicate that you are updating the system command table or an existing site command table, then it will prompt you for the name of the PDS containing the working copy of the specified command table member; if you indicate that you are creating a new site command table, then it will prompt you for the name of the PDS where you want the new site command table member to be stored

- whether or not you will be using the LIBDEF CLIST method to invoke MVS/QuickRef; if you don't know or are not sure, respond "No".

- whether or not you want to add the QW command to the command table being updated or created

- whether or not you want to add the QWS command to the command table being updated or created

\(^2\)The ISPCMDS copy you use should be the production copy, which will be the first one found in the list of data sets concatenated to the ISPTLIB DD in your TSO LOGON procedures. If your company has NOT made any prior changes to the ISPCMDS table, then the "vanilla" copy of it can be found in data set ISP.VnRnMn.ISPTENU for most releases of ISPF.
The actual questions asked by QWCMDS will be dictated by the release of ISPF that you are running. If you are running a release of ISPF prior to 4.2, then the site command table and the function provided by the QWS command are not supported. In this case, QWCMDS will assume that the QW command is the only command option and that it is to be added to the system command table.

Based upon the release of ISPF that you are running and your responses to the questions which it asks, QWCMDS will either update the existing system or site command table member in place or create a new site command table member and place it in the specified PDS.

Once the specified ISPF system or site command table update is completed, the QWCMDS CLIST will then ask if you want to copy the MVS/QuickRef application command table (QWIKCMDS) to a PDS in the ISPTLIB concatenation. **We recommend that you use QWCMDS to copy the tables.** You can take care of adding QWIKCMDS to the ISPTLIB concatenation manually if you prefer using the COPYISPC member discussed later; the QWCMDS CLIST simply provides a convenient means of getting this done (and a reminder that it must be done in order for MVS/QuickRef to function properly).

**Note 1:** Any time you use the QWCMDS CLIST to update an existing command table, make sure you use it to update a "fresh" copy of the command table (that is, one which does not already contain the QW or QWS command entries).

**Note 2:** The data set updated by the QWCMDS CLIST cannot be closed by the QWCMDS CLIST. The data set remains allocated after QWCMDS completes processing. For this reason, you must log off of TSO and log back on again before using the QWCMDS CLIST for a second time so that the data set containing the working copy of the command table will be closed and freed.

After you have successfully updated ISPCMDS or your site command table, you must replace the production copy. Member COPYISPC in the MVS/QuickRef JCL library may be used for this purpose. Read the comments in the top of the member and make changes as directed. Ensure that the JOB statement is valid, and that the correct data set names are specified. Submit the job for execution to copy the table into production.

4. **EQUATE PF KEYS**
To make MVS/QuickRef even easier to use, you should equate a PF key to the MVS/QuickRef invocation command (QW or QWS) that you intend to use most often, and remind each one of the MVS/QuickRef users in your enterprise to do the same. This is especially true for cursor-driven invocation. With a PF key equated to your preferred MVS/QuickRef invocation command, all you have to do to use cursor-driven invocation is place the cursor underneath a character string representing an MVS/QuickRef item name and then press that PF key.

Of course, since PF keys are defined separately for each different ISPF application, this means that you must equate a PF key to your preferred MVS/QuickRef invocation command in each ISPF application where you want to use that PF key. If you want to invoke MVS/QuickRef via a PF key in PDF EDIT, for example, then you must do the PF key equate while you are in PDF EDIT. If you also want to invoke MVS/QuickRef via a PF key while in SDSF, then you must do another PF key equate while in SDSF. Once you are in a given ISPF application, you can use the KEYS command or, if the application is a CUA-style application, the KEYLIST command to establish the appropriate PF key equate.

As an example, suppose, for a given ISPF application, that you decide to use PF key 23 to invoke MVS/QuickRef and that you want to use the QW command. From within that application, you issue the KEYS or KEYLIST command and get a display of the currently defined PF keys. The figure which follows shows how this PF key display might appear after you type in the appropriate PF key equate for PF key 23.

Note: **If you are upgrading to this release of MVS/QuickRef from a release prior to R5.3,** remember that the functions previously performed by the QWSDSF REXX exec have been merged into the main body of MVS/QuickRef code. As a conversion aid for your user community, you should copy the QWSDSF member from the JCL library to your production CLIST or EXEC library. The QWSDSF member reminds your users to change their PF key equates for the QWSDSF command to equates for the QW command.
Note: It is best if you do not use a PF key equated to 'TSO %QW' since this method of invocation bypasses ISPF parsing of the command line and hence is not reliable in all cases.

5. APPLY THE EXPIRATION DATE / CPU ID ZAP

The MVS/QuickRef product is distributed in an "expired" status. The distribution package will contain a yellow sheet containing your expiration date and / or CPU ID zap. Locate this sheet and use the appropriate following method to apply the zap.

Member EXPIREDT in the MVS/QuickRef JCL library can be tailored to execute the AMASPZAP utility program. Read the comments in the top of the member and make changes as directed. Ensure that the JOB statement is valid, the SYSLIB DD statement references the MVS/QuickRef linklib, and the REP 05 statement was modified to contain the zap.

6. MAKE MVS/QUICKREF ELEMENTS AVAILABLE TO SYSTEM

If you will be LIBDEF'ing MVS/QuickRef, refer to the discussion of LIBDEF found on page 95 of this guide.

If you are not going to use the LIBDEF invocation technique, then you may now use member PRODCOPY in the MVS/QuickRef JCL library to manually copy the entire MVS/QuickRef panel, message, table, and program libraries into the existing central ISPF application libraries at your installation, or add them to the ISPPLIB, ISPMLIB, ISPTLIB, and ISPLLLIB concatenations in your LOGON procs.

Regardless of whether you move the MVS/QuickRef elements or use them in the data sets created by the QWLOAD job, the elements must be accessible before MVS/QuickRef can use them. The following table shows the MVS/QuickRef data set name along with the DD statement name that must be used.
You should now refer to the following sections for additional information you must consider: 'Security System Considerations' on page 95, 'Customizing MVS/QuickRef' on page 96, 'Setting MVS/QuickRef Global Installation Options' on page 97, 'Customizing the Vendor/Product Specific Selection List Table' on page 106, 'Preparing to Test MVS/QuickRef' on page 109, 'Testing MVS/QuickRef' on page 110, and 'User Documentation Considerations' on page 114.

This ends the MVS/QuickRef installation process using the non-SMP/E method.

**Installation Method 4 - Data Base Only Install**

You must be using MVS/QuickRef R5.7 or higher in order perform the steps below. If you are using R5.6 or earlier, you must perform a full product installation and you cannot use the 'data base only' install procedure.

1) Key in the JCL below; customize the OUTPUT1 data set name and add a JOB statement, then use the JCL to unload the MVS/QuickRef JCL library from the enclosed distribution tapes:

```jcl
//JS10 EXEC PGM=IEBCOPY,REGION=1024K
//SYSPRINT DD SYSOUT=*
//INPUT1 DD DISP=OLD,
//    DSN=CSL.QUICKREF.JCL,
//    UNIT=3490, <=3480 users change unit name
//    VOL=SER=QR????,
//    LABEL=(8,SL,EXPDT=98000)
//OUTPUT1 DD DSN=local_data_set_name,
//    DISP=(,CATLG,DELETE),
//    UNIT=3390,VOL=SER=??????,
//    SPACE=(TRK,(60,15,25))
//SYSIN DD *
C O=OUTPUT1,I=((INPUT1,R))
//
```

**WARNING -** If you intend to preprocess the MVS/QuickRef panels, please remember that panels QWIKREFC and QWIKREFE, since they contain an ISPF dynamic area, cannot be stored as preprocessed panels; the panels must be stored in non-preprocessed form. Preprocessed panels are panels that have been converted to their internal execution form ahead of time, to save resources and improve response when the panel is displayed. All other MVS/QuickRef panels may be preprocessed.
2) If you want to install the full MVS/QuickRef data base, go to step 3) below. If you want to perform a selective data base load, then edit the QEXCLUDE member in the MVS/QuickRef JCL library that you just unloaded from tape. Follow the editing instructions in the QEXCLUDE member to choose which data base product content to load from the installation tape. Save the changed member, then go on to the next step.

3) Edit the QWUPDATE member in the JCL library that you just unloaded from tape. Follow the edit and customization instructions in the member. The STEPLIB DD should specify your production MVS/QuickRef linklib. If you are performing a selective data base load, make sure that the '/SYSIN DD DUMMY' JCL statement is changed to refer to the QEXCLUDE member that you customized in step 2) above, and that the SORTWKnn DD statements are not commented out. Submit the job for execution; the job will create a new MVS/QuickRef data base on DASD using the distribution tape(s) as input.

4) When the QWUPDATE job ends, logon to TSO/E and get into ISPF. Using the TSO/E FREE command, free the QWREFDD DD statement; then using the TSO/E ALLOCATE command, allocate a QWREFDD DD statement to point to the new MVS/QuickRef data base you just created. Get into MVS/QuickRef and issue the QINFO command. QINFO output should indicate that the data base release level matches the release that you are installing. Review the "What's New" item, which you can access by typing a question mark on the command line of the MVS/QuickRef main menu. The "What's New" item will only be available if you did NOT exclude the Chicago-Soft products from the data base in step 2) above.

5) If you have successfully created your own copy of the new MVS/QuickRef data base, implement it into production by doing ONE of the following, as is appropriate for your site:

- o add the new data base name to the QWIKOPTS options table using the QDB=
  keyword
- o rename your current production main data base and then rename the new data base to the old name; this must be done while no one is using MVS/QuickRef, possibly during dedicated system test time
- o if you are preallocating the QWREFDD DD statement in your TSO/E logon procs instead of using the QWIKOPTS table, then change the QWREFDD DD to refer to the new data base data set name
- o if you are using the QW LIBDEF CLIST, change the allocation for the QWREFDD file in the CLIST to point to the new data base data set name

---

**Background Information**

**ISPF Command Tables**

Each application that executes under ISPF may have its own command table that is accessed when the application is invoked. The commands that may be used for that application are stored in its application command table. Each application command table takes the form of a member of a PDS.
When an application is started, ISPF searches for the application command tables in the PDS's concatenated to the ISPTLIB DD statement. The member name ISPF uses for each application command table is formatted as follows:

aaaaCMDS

where "aaaa" is the application identifier.

If installed as outlined in this user's guide, MVS/QuickRef will be assigned an ISPF applid of "QWIK" and ISPF will expect to find the application command table for MVS/QuickRef in the first member named QWIKCMDS in the PDS's concatenated to the ISPTLIB DD statement. The command table provided for this purpose is copied from the distribution tape during the product installation process. This table must be the first member named QWIKCMDS in the Partitioned Data Sets concatenated to the ISPTLIB DD statement. Otherwise, none of the commands processed internally by MVS/QuickRef (like HELP and REPEAT FIND) will be available.

ISPF itself has its own system command table where the definitions for the primary commands that ISPF supports such as FIND, PRINT, SPLIT, and SWAP are stored. The name of ISPF’s system command table is ISPCMDS; one master copy of ISPCMDS is used by all of the ISPF users at your installation.

Starting with release 4.2 of ISPF, you have the option of creating a site command table. The site command table is designed to allow you to add commands to ISPF without modifying the ISPF system command table. To make use of the site command table, you must update the ISPF ISRCONFG table so that the SITECMDS variable specifies an application id for the optional site command table. (For more details, see the ISPF Planning and Installation Guide.)

To fully install MVS/QuickRef, you must update the ISPF system command table member (ISPCMDS) or the site command table to include at least one of the commands required to invoke MVS/QuickRef (QW and/or QWS).

Normally, PDF option 3.9 is used to add, delete, or change entries in an application command table. Unfortunately, the 3.9 option will not allow you to change entries in the ISPCMDS table. For this reason, the QWCMDS CLIST is provided with MVS/QuickRef to automatically perform the steps necessary to add the QW and/or the QWS command to either the ISPF systems command table or the ISPF site command table.

**Note on CA Products**

If you are using the CA-OPS/MVS II or CA-SYSVIEW/E products, please note that both of these products have a built-in interface to MVS/QuickRef that is used when these products are invoked outside of an ISPF environment. You should consult your CA-OPS/MVS documentation or your CA-SYSVIEW/E Command Reference for information on how to define and use these products' direct interface to MVS/QuickRef. For CA-SYSVIEW/E, you will need to set the GSVXGEN options table parameters QRFDSN (to indicate the name of the
MVS/QuickRef main data base) and QRFLNCT (to indicate the number of text lines to be returned.)

Security System Considerations

MVS/QuickRef users need READ access to the MVS/QuickRef main data base, to all defined user data bases, and to the data sets concatenated to the SYSHELP DD in their LOGON proc or to the SYS1.HELP data set if SYSHELP is not preallocated. If your security system allows you to limit access to SYS1 or other data sets to specific program names, the program name to use for MVS/QuickRef is QWIKREF1.

Some security systems require (even for READ access) that each program be identified that is to have READ access to a certain data set or data set type. This is sometimes done by placing the program name in a table. If your security system has this requirement for any of the data sets to be accessed by MVS/QuickRef (like SYSHELP), be sure that program QWIKREF1 is properly identified as having READ access.

Alternative MVS/QuickRef Invocation Methods

You should read this section if you do not want to modify the ISPF commands table in order to add the QW or QWS command.

MVS/QuickRef can be invoked as an application from an existing ISPF application selection panel by adding the line below to the ZSEL portion in the )PROC section of the panel:

```
option,'PGM(QWIKREF1) NEWAPPL(QWIK)'
```

where ‘option’ is the option number or letter that you choose to use. If you prefer to use the LIBDEF service to invoke MVS/QuickRef, then add the line below to the ZSEL in the )PROC section of the panel:

```
option,'CMD(%QW) NEWAPPL(QWIK)'
```

The disadvantage of this installation technique (as opposed to modifying the ISPF command table in order to add the QW or QWS command) is that MVS/QuickRef can then only be invoked from one panel under ISPF. Your ISPF users will have to end the current application and transfer to the MVS/QuickRef panel in order to get instant reference information. This technique nullifies much of the value of MVS/QuickRef as an instant reference tool.

Using LIBDEF

MVS/QuickRef can be invoked using the ISPF LIBDEF service to access its panels, messages, and programs; however, there are several points that must be considered:

1. LIBDEF lets you establish MVS/QuickRef in its own libraries and then point to those libraries at execution time. With LIBDEF, you do not need to change your TSO/E logon
procedures or copy MVS/QuickRef elements into your existing ISPF libraries. A LIBDEF implementation makes overall product maintenance easier since the product is segregated in its own libraries; however, invocation and initial startup of MVS/QuickRef will be slower, compared to a non-LIBDEF product implementation.

2. If you want to use the LIBDEF service to invoke MVS/QuickRef, you may do so by using the QW CLIST supplied in the MVS/QuickRef JCL library. This CLIST supports nested invocation of MVS/QuickRef as well as the fast-path invocation technique. The CLIST must be edited so that the names of the MVS/QuickRef data sets at your installation are used, and then it must be copied to the SYSPROC concatenation before use.

3. The MVS/QuickRef commands table member QWIKCMDS must be copied into your logon proc ISPTLIB concatenation even if you are using the QW LIBDEF CLIST to invoke MVS/QuickRef. This is because ISPF must be able to find the QWIKCMDS table when the QW command is fetched for execution from the ISPF command table. Command fetching and NEWAPPL parameter processing is done by ISPF before the QW CLIST actually starts executing. Failure to copy the QWIKCMDS table from the MVS/QuickRef table library to your logon proc ISPTLIB concatenation will cause ISPF not to honor the commands MVS/QuickRef has defined in its QWIKCMDS table, and necessary functions (like the HELP command and the reference information it displays) will not be available at MVS/QuickRef execution time.

4. The PASTE portion of the MVS/QuickRef CUT and PASTE feature normally executes as a program-form edit macro and issues messages that must be available in the ISPF message library when PASTE is invoked. If MVS/QuickRef is invoked via LIBDEF, then the PASTE function must also be invoked via LIBDEF. Member QWPASTE in the MVS/QuickRef JCL library is a CLIST that can be used as a model for invoking the PASTE function as a CLIST form edit macro; the CLIST must be edited to reflect the MVS/QuickRef data set names in use at your installation. When using QWPASTE, you must apply the zap documented at the top of the QWPASTE CLIST, edit the CLIST to reflect local data set names, then copy the CLIST to your SYSPROC concatenation.

**Customizing MVS/QuickRef**

You can use the override parameter facility of MVS/QuickRef to replace, augment, or prevent access to any item in the MVS/QuickRef main data base or in a user data base. Refer to chapter six of this guide for a description of the override parameter facility.

Some MVS/QuickRef sites may find it desirable to alter the default parameters for the MVS/QuickRef DASD free space feature. This feature, option “S” on the MVS/QuickRef main menu, allows users to quickly determine the amount of free space available on online DASD volumes. The default action for this option, if no DASD volume serial number or volume serial number prefix is entered, is to require entry of a volser or volser prefix.

To change the default for the DASD free space option so that the user need not enter a volume serial number or volume serial number prefix, edit the MVS/QuickRef secondary menu panel
QWIKREFB. You will find two statements commented out in the )PROC section of the panel that relate to this function. Blank out the ‘/*’ in columns one and two of these two statements and save the QWIKREFB panel. Your MVS/QuickRef users will now no longer have to enter either a full DASD volume serial number or a generic volume serial number prefix when the DASD free space look-up function is invoked. All they have to do is press enter, and free space information for all online DASD will be shown automatically.

Setting MVS/QuickRef Global Installation Options

MVS/QuickRef includes an options table load module named QWIKOPTS. You can set certain global MVS/QuickRef installation options at product install time using this options table.

The QWIKOPTS options table load module must be available when MVS/QuickRef executes and it must reside in the same load library with the other MVS/QuickRef load modules. A copy of the QWIKOPTS load module with all default options set is supplied in the MVS/QuickRef load library and will be copied to DASD when you install this release. However, in order to make sure MVS/QuickRef functions correctly in your processing environment, you should review the information in this section in order to determine whether or not you need to make any changes to the default options supplied with the QWIKOPTS table. If you decide to make any changes, then you will need to reassemble and relink the QWIKOPTS load module (non-SMP/E) or run a job to apply a usermod to rebuild the QWIKOPTS module (SMP/E).

Options you can set using the QWIKOPTS table include:

- the MVS/QuickRef main data base data set name
- the data set name(s) of your user data base(s)
- the name of the MVS/QuickRef override parameter data set
- global defaults for the QPRINT command, such as output class, output destination, and HOLD/NOHOLD disposition
- default route code used for WTOs issued by MVS/QuickRef when it executes in batch or as a started task
- default names for the MVS/QuickRef main menu and highest-level user menu
- amount of virtual storage allocated for MVS/QuickRef main data base reference information
- amount of virtual storage allocated for user data base reference information
- order of fields in the DASD Free Space report
- character translation for some special display characters
- DDname used for SYSHELP processing
- action taken when PF3 is pressed
♦ order of the fields shown on a user item selection list
♦ alternate JES2 command and message prefix character
♦ whether character translation is to be in effect
♦ characters that are to be translated when character translation is in effect
♦ for cursor driven invocation, the characters that are translated into spaces during on-screen item name processing
♦ options controlling user-directed selection list processing
♦ whether data base information is to be upper cased

If you are not using SMP/E to modify the QWIKOPTS table, you must manually modify and reassemble member QWIKOPTS in the MVS/QuickRef SOURCE library. The QWIKOPTS member invokes the MVS/QuickRef QWIKOPT macro, passing option settings to it at assembly time. To set a particular option that you may be using, edit the QWIKOPTS member and set the appropriate macro keyword to the value desired, then reassemble the QWIKOPTS table using member QWOPTASM in the MVS/QuickRef JCL library.

If you are using SMP/E to modify the QWIKOPTS option table, you should specify source changes in the QWINUOPT member in the MVS/QuickRef JCL library. After updating QWINUOPT, use member SMPRAP3 in the MVS/QuickRef JCL library to receive and apply a usermod to reassemble the module.

After modifying QWIKOPTS, you should refresh ISPF before using MVS/QuickRef.

The first time you use MVS/QuickRef after modifying QWIKOPTS, issue the QINFO command to verify that your changes have been made successfully.

A description of each keyword option supported by the QWIKOPT macro is described below:

**QDB=**

This keyword is used to define the name of the MVS/QuickRef main data base. The data set name specified for this option must be the fully qualified data set name you have chosen for the MVS/QuickRef main data base at your site. For example, if your main data base data set name is SYS2.QUICKREF.DATABASE, then the specification for this option on the QWIKOPTS macro would be QDB=SYS2.QUICKREF.DATABASE. Please note that the main data base opened and used by MVS/QuickRef is determined in this sequence:

1. If the main data base is preallocated to the QWREFDD DD statement when MVS/QuickRef is invoked, then the preallocated copy is used, regardless of the setting for the QDB= parameter in the QWIKOPTS table.
2. If the main data base is not preallocated, then the data set name specified by the QDB= parameter in the QWIKOPTS table is used to dynamically allocate the main data base. This is true for non-LIBDEF users and for LIBDEF users who do not preallocate the QWREFDD file in their LIBDEF CLIST.

**PARMDSN** = This keyword identifies the override parameter data set, if you are using MVS/QuickRef's optional override parameters. The data set name specified for this option must be the fully qualified data set name you have chosen for your MVS/QuickRef override parameter PDS. For example, if the name or your override parameter PDS is SYS2.QUICKREF.PARMS, then the value to specify for this keyword is PARMDSN=SYS2.QUICKREF.PARMS. Please note that the override parameter PDS data set name to use is determined in this sequence:

1. If the override parameter PDS is preallocated to the QWPARMS DD statement when MVS/QuickRef is invoked, then the preallocated copy is used, regardless of the setting for the PARMDSN= parameter in the QWIKOPTS table.

2. If the override parameter PDS is not preallocated, then the data set name specified by the PARMDSN= parameter in the QWIKOPTS table is used to dynamically allocate the override parameter data set. This is true for non-LIBDEF users and for LIBDEF users who choose not to preallocate the PARMDSN file in their LIBDEF CLIST.

**OUTCLAS** = This keyword identifies the default SYSOUT output class to use for all printed output produced by the MVS/QuickRef QPRINT command. The default for this keyword is **A**. The SYSOUT class used for a specific QPRINT request comes first from the CLASS operand on the QPRINT command if it is present; if no CLASS operand is specified on the QPRINT command, the SYSOUT class to use comes from the QPRINT profile information last set with the QPRINT PROFILE command; if no QPRINT profile class is found, the setting for the OUTCLASS= keyword in the QWIKOPTS table is used. Finally, if the QWIKOPTS table is not in use, a program default of **A** is set.

**OUTDEST** = This keyword is used to set the default output destination to use for all printed output produced by the MVS/QuickRef QPRINT command. The default for this keyword is **LOCAL**. The destination used for a specific QPRINT request comes first from the DEST operand on the QPRINT command if it is present; if no DEST operand is specified on the QPRINT command, the destination to use comes from the QPRINT profile information last set with the QPRINT PROFILE command; if no QPRINT profile destination is found, the setting for the OUTDEST= keyword in the QWIKOPTS table is used. Finally, if the QWIKOPTS table is not in use, a program default of **LOCAL** is set.
**OUTDISP=** This keyword is used to set the default HOLD disposition to use for all printed output produced by the MVS/QuickRef QPRINT command. The default for this keyword is **HOLD**. The disposition used for a specific QPRINT request comes first from the HOLD/NOHOLD operand on the QPRINT command if it is present; if no HOLD/NOHOLD operand is specified on the QPRINT command, the disposition to use comes from the QPRINT profile information last set with the QPRINT PROFILE command; if no QPRINT profile disposition is found, the setting for the OUTDISP= keyword in the QWIKOPTS table is used. Finally, if the QWIKOPTS table is not in use, a program default of **HOLD** is set.

**ROUTCDE=** This keyword sets the default WTO routing code to use for WTO messages written to the MVS console and system log when MVS/QuickRef is executed as a batch job (with no QWPRINT DD in the JCL) or as a started task. The default for this keyword is ROUTCDE=2 (route code 2).

**MAINMNU=** This keyword is used to set the default name of the MVS/QuickRef main menu. The default for this name is QWIKREFA.

**USERMNU=** This keyword is used to set the default name of the MVS/QuickRef user menu. The default for this name is QWIKREFU.

**TEXTSTR=** This keyword sets the size of the virtual storage area (in bytes, allocated above the 16mb line) used for housing reference information retrieved from the MVS/QuickRef main data base. This keyword also sets the size of the virtual storage area (in bytes, allocated above the 16mb line) used for housing DASD information collected from the system for a DASD Free Space Report. If a main data base reference item, selection list, or DASD Free Space Report contains more than the number of bytes specified by this keyword, then only the number of bytes specified by this keyword will be displayed. The default for this keyword is 940,000 decimal.

**USERSTR=** This keyword sets the size of the virtual storage area (in bytes, allocated above the 16mb line) used for housing reference information retrieved from a user data base. If a user data base reference item or selection list contains more than the number of bytes specified by this keyword, then only the number of bytes specified by this keyword will be displayed. The default for this keyword is 640,000 decimal.

**DASDFRE=** This keyword is used to specify the order of the fields that appear on the DASD Free Space Report. This parameter consists of a list of numbers separated by commas and enclosed in parenthesis. This is illustrated in the default parameter shown here: DASDFRE=(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17).

The fields are as follows:
The DASD Free Space report supports display of more information than will fit on a regular 80-character wide display screen. The DASDFRE= parameter allows you to reorder the fields to make the ones most useful to you visible without scrolling right. In addition, some installations may wish to eliminate some of the fields that are displayed in the DASD Free Space report. As an example, suppose you wish to "hide" the Mount Attribute (field 2), Device Type (field 3), and the SMS Indicator (field 11). You would specify the DASDFRE= option as follows:

\[
\text{DASDFRE}=(1,4,5,6,7,8,9,10,12,13,14)
\]

Please note that the Volume Serial field should always be specified.

**HELPDD=** This keyword is used to specify the DDName to be used for HELP processing. If your company does not use the IBM default DD name SYSHELP for access to TSO/E command help information, then you can use this keyword to define the name of the DD statement that MVS/QuickRef should use to fetch TSO/E command help information if it is requested. The default for this keyword is HELPDD=SYSHELP.

**PF3=** This keyword is used to specify the action taken when the PF3 key is pressed. The PF3= parameter is honored only when MVS/QuickRef is invoked through cursor-driven invocation. This parameter consists of two choices: END or RETURN. The default is END as shown here:

\[
\text{PF3=}\text{END}
\]

Setting the PF3= parameter to END causes MVS/QuickRef to return to the previous panel when PF3 is pressed, if MVS/QuickRef was entered via cursor-driven invocation. A value of RETURN causes MVS/QuickRef to terminate processing when PF3 is pressed during cursor-driven invocation.

**UISLOPT=** This is the user item selection list option. It indicates the order of the fields shown on each line of a user data base item selection list. If set to N, the default, then the user data base title field will be the last field shown on the right. If set to Y, then the user data base title field will be shown just to the right of the item name field and the other fields will be moved over to the right.
CHRXLAT= Designed for use primarily by European customers, this keyword determines if character translation should be performed on certain special characters when the text for a given item is displayed. The default value of NO indicates that no character translation will be performed. The option YES causes the characters specified in the CHRXTAB parameter to be translated.

WARNING: This option, as of MVS/QuickRef R6.2, if set to YES, will translate not only the text of individual items as it did in prior releases, but it will also translate the text of individual item names as well.

CHRXTAB= This keyword is used in conjunction with the CHRXLAT=YES parameter. When CHRXLAT=YES is specified, the CHRXTAB= parameter provides a list of up to 16 characters that will be translated before they are displayed on-screen. If CHRXLAT=NO is specified, then the CHRXTAB= parameter setting is ignored. Do not confuse this parameter with the CDIXLAT parameter, which is used during cursor-driven invocation processing.

The syntax for the CHRXTAB= parameter is illustrated in the default parameter shown here:

CHRXTAB=((C0,4C),(D0,6E),(E0,5F),(5A,4F),
(6A,4F),(00,00),(00,00),(00,00),
(00,00),(00,00),(00,00),(00,00),
(00,00),(00,00),(00,00),(00,00)),

To better understand the EBCDIC characters, the following information is presented:

<table>
<thead>
<tr>
<th>Character Before Translation</th>
<th>Character After Translation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Character Before Translation</td>
<td>Character After Translation</td>
</tr>
<tr>
<td></td>
<td>(chr) (hex)</td>
<td>(chr) (hex)</td>
</tr>
<tr>
<td>{</td>
<td>C0  &lt;</td>
<td>4C</td>
</tr>
<tr>
<td>}</td>
<td>D0 &gt;</td>
<td>6E</td>
</tr>
<tr>
<td>\</td>
<td>E0 ¬</td>
<td>5F</td>
</tr>
<tr>
<td>!</td>
<td>5A</td>
<td>4F</td>
</tr>
<tr>
<td></td>
<td>6A</td>
<td>4F</td>
</tr>
</tbody>
</table>

The parameter consists of a series of sub-parameters of the format (xx,yy) where the "xx" value is the hexadecimal value of the character to be translated and "yy" is the hexadecimal value of the new character. For example, in (C0,4C), the character 'X'C0' will be translated into 'X'4C'. From one to 16 sub-parameters can be specified in the CHRXTAB= parameter.

Note: Each group of sub-parameters are enclosed within parentheses, and the entire set of sub-parameters is enclosed in one set of parentheses.
**JES2CHR=** This keyword is used to determine the character used as the prefix character for JES2 commands and messages. The dollar sign ("$") is always accepted to look-up JES2 commands and messages and is the default value for the JES2CHR parameter. By setting the JES2CHR to something other than the "$" character, another prefix character is also accepted as input for an item request and this character is used as the prefix in displaying all JES2 commands and messages.

**NOTE:** By default, cursor-driven invocation will not work for JES2 messages if the JES2 prefix character is ",", "==", "*=", "/", "("", ")", or "?" because MVS/QuickRef translates these characters to spaces. To resolve this problem, remove the JES2CHR from the CDIXLAT= parameter to prevent this translation to spaces.

**CDIXLAT=** When you place the cursor on a field on an ISPF screen and request cursor-driven invocation of MVS/QuickRef, certain characters in the field are translated to blanks before the word the cursor is on is extracted for data base lookup. CDIXLAT= indicates the characters to be translated to blanks. The default values for this parameter are as follows (Note that the leading and trailing apostrophe are part of the syntax):

CDIXLAT='!',"*/()? {};!' You made need to change the CDIXLAT= parameter defaults if you use an alternative JES2 command and message prefix character, such as an asterisk (*). In order to perform cursor driven invocation of JES messages prefixed with an asterisk instead of a dollar sign, remove the "*" from the CDIXLAT parameter. In general, the value specified by the JES2CHR= parameter should not be specified in the CDIXLAT= parameter. Do not confuse this parameter with the CHRXTAB= parameter.

**OSLSLO=** This keyword controls the operating system level selection list option. Along with the QWIKOPTS VPSSLO= option, it is used to setup and control user-directed selection list processing for your entire installation. The OSLSLO= option may be set to Y (for Yes), N (for No), or H (for High). If set to Yes, which is the default setting, operating system level selection list processing will be in effect. If set to No, operating system level selection list processing will not be used. The H (or High) setting should be used when you want to turn on operating system level selection list processing but are running on a release of the operating system which is more current than the most current release of the operating system covered in the main data base. You can determine the most current release of the operating system covered in the main data base by browsing the QEXCLUDE member in the MVS/QuickRef JCL member and looking at the Z/OS SYSTEM CODES entries. The most current release listed in the QEXCLUDE member for Z/OS SYSTEM CODES is the most current release of the operating system covered in the main data base. If your installation is running a more current release of the operating system than that covered in the main data base, then you must either turn operating system level selection list processing off (OSLSLO=N) or use the High setting (OSLSLO=H). Otherwise, operating
system level selection list processing will not work correctly. For more information on operating system level selection list processing, see the section titled "User-Directed Selection List Processing" in Chapter Three.

**VPSSLO=**

This keyword controls the vendor/product specific selection list option. Along with the QWIKOPTS OSLSLO= option, it is used to setup and control user-directed selection list processing for your entire installation. The VPSSLO= option may be set to Y (for Yes) or N (for No). If set to No, which is the default setting, vendor/product specific selection list processing will not be used. If set to Yes, vendor/product specific selection list processing will be in effect. Before setting the VPSSLO= option to Yes, you should customize and assemble the Vendor/Product Specific Selection List Table (QWIKVPST). For more details on vendor/product specific selection list processing, see the section titled "User-Directed Selection List Processing" in Chapter Three. For more details on customizing QWIKVPST, see the section which follows on "Customizing The Vendor/Product Specific Selection List Table".

**FOLD=**

This keyword indicates whether or not data base information is to be presented as if it were upper cased. It can be set to Y (for Yes) or N (for No), which is the default. If set to Yes (FOLD=Y), then all data base information will be folded to upper case before it is displayed. More specifically, all data to be shown in the dynamic area of the QWIKREFE and QWIKREFC panels will be converted to upper case. This includes user data base information, DASD free space information, all types of selection lists, information provided by the QINFO command, information "added" to the data base by override parameters, and information taken from the SYSHELP concatenation. This also means that all information accessed via direct program call or batch execution, printed with the PRINT command, or cut with the CUT command will be in upper case letters.

**AUTOLUP=**

This keyword indicates whether or not automatic lookup based on text content is to be turned off. If set to N (for No), automatic lookup based on text content will not be active. If set to Y (for Yes, which is the default), then automatic lookup based on text content will be active.

**AUTOFC=**

This keyword indicates whether or not automatic FINDCODE processing is to be turned off. If set to N (for No), automatic FINDCODE processing will not be active. If set to Y (for Yes, which is the default), then automatic FINDCODE processing will be active.
LOCLDB7=
LOCLDB8=
LOCLDB9=

These keywords identify the data set names of your local user data bases, if you choose to define any user data bases. The LOCLDB1= through LOCLDB9= parameters must be specified in ascending sequence; i.e., if LOCLDB2= is used, then LOCLDB1= must also be used. Please note that the user data base opened and used by MVS/QuickRef is determined in this sequence:

1. If a user data base is preallocated to the QWREFDDU DD statement when MVS/QuickRef is invoked, then the preallocated copy is used, regardless of the setting for the LOCLDBn= parameters in the QWIKOPTS table.

2. If a user data base is not preallocated, then the data set name(s) specified by the LOCLDBn= parameters in the QWIKOPTS table are used to dynamically allocate the user data base(s). This is true for non-LIBDEF users and for LIBDEF users who choose not to preallocate the QWREFDDU file in their LIBDEF CLIST.

BATPDSRD= This keyword is used to specify, for batch jobs, whether or not the dataset name list function will read the directory information for each partitioned data set on the volume being processed. The PDS information includes the following fields (see Notes 1, 2 & 3 below):

DIRA ... PDS directory blocks allocated - not available for PDS-E dataset
DIRU ... PDS directory blocks used - not available for PDS-E dataset
MEMS ... number of PDS members
ALIS ... number of PDS aliases

The possibles values are 'Y' for "yes, read the directory information", and 'N' for "no, do not read the directory information".

Note 1: These are optional fields that contains PDS directory information. To obtain PDS directory information, each PDS on the volume must be read and analyzed; thus, to give the best performance, this information is NOT displayed by default.

Note 2: To obtain the PDS directory information, each PDS must allow the program to have read access. If the Security Access Facility does not allow the program to read the directory, these columns will contain the value 'SAF'.

Note 3: *** IMPORTANT *** - Use of this keyword to read the directory information of a PDS will update the Last Referenced Date. If the Last Referenced Date is being used to control backups or SMS archiving, you should NOT set this keyword to 'Y'.

ONLPDSRD= This keyword is used to specify, for online TSO users, whether or not the dataset name list function will be allowed to read the directory information for each partitioned data set on the volume being processed. The PDS information includes the following fields (see Notes 1, 2 & 3 in the BATPDSRD keyword above):

- DIRA ... PDS directory blocks allocated - not available for PDS-E dataset
- DIRU ... PDS directory blocks used - not available for PDS-E dataset
- MEMS ... number of PDS members
- ALIS ... number of PDS aliases

The possible values are 'Y' for "yes, read the directory information", and 'N' for "no, do not read the directory information". If 'N' is specified, the online user WILL NOT see an option allowing the directory information to be read. If 'Y' is specified, the program will still default to "no, do not read the directory information"; however, the online user WILL see an option allowing the directory information to be read.

Customizing The Vendor/Product Specific Selection List Table

The Vendor/Product Specific Selection List Table is a load module, named QWIKVPST, which is used in conjunction with vendor/product specific selection list processing. Vendor/product specific selection list processing, one of the three types of user-directed selection list processing, uses the QWIKVPST table module to determine the preferred release number for each base product which you choose to add to the table.

Vendor/product specific selection list processing is controlled by the QWIKOPTS vendor/product specific selection list processing (VPSSLO=) option. Before you set the QWIKOPTS vendor/product specific selection list option to Yes (VPSSLO=Y), you must customize, assemble, and link the QWIKVPST table module. The QWIKVPST table module must be available when the QWIKOPTS VPSSLO= option is set to Yes and it must reside in the same load library with the other MVS/QuickRef load modules.

For a general description of vendor/product specific selection list processing, see the section on "User-Directed Selection List Processing" on page 33 in Chapter Three. For details on the QWIKOPTS VPSSLO= option, see the previous section on "Setting MVS/QuickRef Global Installation Options". You should review the information in both of these sections before attempting to customize the QWIKVPST table module.

Source code for the QWIKVPST table, which is an assembler module, can be found in member QWIKVPST in the MVS/QuickRef source library. You assemble and link the QWIKVPST table module using the JCL in member QWIKVPSA in the MVS/QuickRef JCL library.
You customize the QWIKVPST table module by modifying the Vendor/Product Specific Selection List Copy Member, QWIKVPSC. The QWIKVPSC copy member can be found in the MVS/QuickRef macro library.

A sample of the QWIKVPSC copy member is shown in Figure 23. As shown in Figure 23, the QWIKVPSC copy member contains commented assembler language statements showing the release numbers for each base product in the main data base. For each base product, the copy member shows the data base containing that base product (with M for the main data base, 1-9 for one of the user data bases), the associated vendor name, the associated product name, and the possible preferred release numbers.

You specify the preferred release for a given base product in the main data base by simply uncommenting the assembler statement in the QWIKVPSC copy member that contains the preferred release number for that specific base product.

Note: You uncomment an assembler statement in the QWIKVPSC copy member by replacing the asterisk in column 1 with a blank. The first character in an uncommented line must, in this case, be a blank. So do not delete the asterisk; make sure you replace it with a blank. The QWIKVPSC copy member also contains comment lines which provide instructions, column headings, and blank lines for readability. Make sure you do not uncomment any of these lines.

You can also add entries to the QWIKVPSC copy member to specify preferred releases for other base products, such as those that might appear in a user data base. If you do add entries, then they must be formatted exactly like the existing entries in the QWIKVPSC copy member. In other words, the data base, vendor name, product name, and preferred release number must all start in the same column used for the existing entries. The vendor name, product name, and preferred release number specified in each entry in the QWIKVPSC copy member must be specified, formatted and spelled exactly as they appear in the associated data base. Otherwise, vendor/product specific selection list processing will not work correctly.

Uncommented entries in the QWIKVPSC copy member are processed in the same order as they appear in the copy member. If there is more than one uncommented entry for the same base product in the copy member, the first entry will dictate the preferred release to be used and the second and all subsequent uncommented entries for that base product will be ignored.

As an example of customizing the QWIKVPST table, suppose the first three entries in the QWIKVPSC copy member appear as shown in Figure 23.
Now suppose you want to use vendor/product specific selection list processing and, in conjunction with vendor/product specific selection list processing, you want to designate V5R1 as the preferred release for base product CA-1, V1R3 as the preferred release for base product JES2 COMMANDS, and R5 as the preferred release for base product MACROS associated with vendor RAMAC in user data base number one. Suppose also that you do not want to use vendor/product specific selection list processing against base product SYSTEM CODES. To implement this change, you could modify the QWIKVPSC copy member shown in Figure 23 so that it looks like the QWIKVPSC member shown in Figure 24.

As shown in Figure 23 the asterisk in column one has been replaced by a blank in the entry for CA-1 V5R1, making this the preferred release for base product CA-1. The asterisk in column
one has also been replaced by a blank in the entry for OS/390 JES2 COMMANDS V1R3, making this the preferred release for base product JES2 COMMANDS. An entry has been added to the copy member for base product MACROS from vendor RAMAC in user data base number one making R5 the preferred release for this base product. Since vendor/product specific selection list processing is not to be used for base product SYSTEM CODES, none of the asterisks in column one in the entries for this base product have been replaced by a blank.

After the QWIKVPSC copy member has been modified as shown in Figure 23 you would assemble and link the QWIKVPST table module using the JCL in member QWIKVPSA in the MVS/QuickRef JCL library. After the QWIKVPST table module is successfully assembled and linked, you would set the QWIKOPTS vendor/product specific selection list option to Yes (VPSSLO=Y) and then reassemble and relink the QWIKOPTS options table.

Now vendor/product specific selection list processing will be in effect for your entire installation.

To verify that vendor/product specific selection list processing is set up correctly, use the QINFO command and page down until you see the VPSSLO option and verify that it is set to Yes (VPSSLO=Y). Then continue to page down until you see the entry which says "Vendor/Product Specific Selection List Table follows". This will show you the preferred releases defined for the base products in your Vendor/Product Specific Selection List Table for the modified QWIKVPSC copy member shown in Figure 24.

**Expiration Date Zaps**

MVS/QuickRef will not execute properly unless an expiration date extension zap is applied to it. The zap you should use was supplied with the product documentation on a separate sheet of paper. If no zap information sheet was supplied with the product documentation, you should contact MVS/QuickRef product support for assistance.

Once you have obtained the proper expiration date extension zap, it must be applied with the AMASPZAP utility as found in the EXPIREDT member of the MVS/QuickRef JCL library. The expiration date extension zap consists of three AMASPZAP control statements: a NAME statement identifying module QWIKREF1, a VER statement, and a REP statement. The offset values used on the VER and REP statement are not the same; this is normal for this type of zap and is not a cause for concern. If the zap is applied by AMASPZAP with a condition code of zero and the VER statement value is accepted, then the zap is properly applied.

If, after the zap is applied, your copy of MVS/QuickRef is still expiring prematurely, please review the information in the section titled "Troubleshooting MVS/QuickRef" on page 111 before calling MVS/QuickRef product support.

**Preparing to Test MVS/QuickRef**

After the installation steps for MVS/QuickRef have been completed, you should test the system to ensure that it is functioning properly.
Logon to TSO and before going into ISPF, make sure that the MVS/QuickRef panels, application command table, and messages unloaded from the MVS/QuickRef distribution tape are in data sets concatenated to the ISPPLIB, ISPTLIB, and ISPMLIB DD statements allocated in your TSO session. The PDS member that contains the modified copy of the ISPF system or site command table should be concatenated first to the ISPTLIB DD statement so that the QW or QWS command can be used to invoke MVS/QuickRef from any panel.

The program library containing the MVS/QuickRef load modules should be allocated to the ISPLLIB DD statement for this test session only. ISPLLIB will be searched first for load modules by ISPF if it is allocated. If it is not allocated, the normal MVS program search through the STEPLIB/PLPA/Link List will be performed. Using ISPLLIB lets you test MVS/QuickRef in your TSO session only, and without affecting any other ISPF users.

Note: If you are using the QW LIBDEF CLIST, then the MVS/QuickRef panel, message, and program load libraries, normally concatenated to the ISPPLIB, ISPMLIB, and ISPLLIB DD statements, will not need to be concatenated to these DD statements since the QW CLIST will "LIBDEF" them.

Make sure that you are testing MVS/QuickRef in an environment that does not include elements of the product from a prior release. For example, if you are testing with a LOGON PROC or CLIST that includes a load module library from a previous MVS/QuickRef release, you may experience problems; therefore, make sure that all MVS/QuickRef programs, panels, messages, tables, and the main data base are from the new release of the product.

Testing MVS/QuickRef

After the allocations described above are done, enter ISPF and type the appropriate MVS/QuickRef invocation command (QW or QWS) on the command line of any panel. The MVS/QuickRef main menu should appear. Exercise each of the options on the main menu and make sure that each of the resulting display request panels can be used to display reference information and/or the appropriate type of selection list.

Note: While testing MVS/QuickRef, remember that you can generally use the ISPF END command to "backup" to the previous display and that you can always use the ISPF RETURN command to exit and terminate the current invocation of MVS/QuickRef.

Once you are satisfied that menu-driven invocation is working properly, try fast-path invocation. For example, type 'QW IDC3009I' in the command line of any panel to "fast-path" to a display of the reference information for the IDC3009I message. Try 'QW L=V' get a list of all products carried in the data base in vendor name sequence. Try 'QW L=P' to get a list of all products carried in the data base in product name sequence. Try 'QW L=O' to get a list of vendors only.
Once you are satisfied that fast-path invocation is working properly, it is time to test cursor-driven invocation. Using your spool data display product (SDSF, IOF, etc.), display a job on the held or output queue or view the system log, and put the cursor on any character of any message id you might want to look up. When the cursor is on the desired message, press the PF key you have set to the QW or QWS command. You could also type the command on the command line, then move the cursor down to the desired message and press ENTER, but the PF key technique requires fewer keystrokes. MVS/QuickRef will then be invoked and will display reference information for the indicated message.

Next, go into BROWSE or EDIT for a PDS member containing JCL, put the cursor on any JCL keyword (e.g., SYSOUT=, DCB=, etc.), and press the PF key you equated to QW or QWS. MVS/QuickRef will then be invoked and will display reference information for the indicated JCL keyword.

This completes the initial installation and testing of MVS/QuickRef.

If MVS/QuickRef does not function as expected, then, **before calling MVS/QuickRef product support**, review the installation steps and the section on "Troubleshooting MVS/QuickRef" on page 111 which follows.

**Troubleshooting MVS/QuickRef**

If MVS/QuickRef does not function as expected, review the list of common problems and corrective actions outlined below.

1. **MVS/QuickRef commands like HELP or REPEAT FIND do not function correctly.**
   Make sure the MVS/QuickRef application command table (QWIKCMDS) is in the ISPTLIB concatenation. You can check this by going to ISPF option 3.9 and entering application id QWIK (the application id assigned to MVS/QuickRef). You can then browse the MVS/QuickRef application command table actually being used by ISPF. If no commands are shown, then QWIKCMDS is not in the ISPTLIB concatenation. If one or more commands are shown, then check the individual command you are having a problem with. If it is not in the table, then you are running with a copy of QWIKCMDS from a prior release of MVS/QuickRef. If the command is shown, make sure it is specified with PASSTHRU. If the command is shown and specified with PASSTHRU, then the MVS/QuickRef application id (QWIK) is not getting set. If you are using a TSO command like TSO %QW to invoke MVS/QuickRef, then you are not going through the command table and the required application id for MVS/QuickRef is not getting set. If you are using TSO %QW because you do not want to update the ISPF system command table, consider creating or updating a site command table. This is a facility provided by 4.2 and later releases of ISPF specifically so you can add commands to ISPF without changing the ISPF system command table. You can also consider using the QWX CLIST in the MVS/QuickRef JCL library, which can be invoked by specifying TSO %QWX and will set the required QWIK application id as it calls the QW CLIST. If you are going through the command table to invoke MVS/QuickRef, then refer to #3 above for information on how you can use ISPF option 3.9 to check the definition of the QW or QWS command you are using. Make sure the command is defined with NEWAPPL(QWIK).
2. **MVS/QuickRef is expired or about to expire even though an expiration date zap has been applied.** First make sure the zap was typed correctly with all characters and digits in the right position. Then make sure the zap was applied correctly and that the AMASPZAP utility ended with a zero return code. Then check to make sure you are not executing some other copy of program QWIKREF1 other than the one you zapped. You can check this by using the MVS/QuickRef QINFO command. (If MVS/QuickRef is expired, the QINFO display is automatically presented.) The QINFO display shows the expiration date associated with the copy of the QWIKREF1 module actually being executed. It also shows the hex code at the rep location within the copy of the QWIKREF1 module actually being executed. If this does not match the rep portion of the zap you applied, then you zapped some other copy of QWIKREF1 other than the copy being executed. The QINFO display also shows where other copies of the QWIKREF1 module may be located. You need to find the downlevel copy of the QWIKREF1 load module and eliminate it.

3. **The MVS/QuickRef invocation command (QW or QWS) is indicated by ISPF to be invalid.** Make sure the QW or QWS command you are attempting to use was added to the appropriate ISPF system or site command table and that the version of the command table that was updated is the version actually being used. You can check this by going to ISPF option 3.9 and entering application id 'ISP' if you updated the ISPF system command table. If you updated or created a site command table, then you will have to enter the application id assigned to the site command table (the QINFO command displays the name of your site command table if one is defined for use). Once you enter the proper application id, option 3.9 will allow you to browse all the commands in the specified command table. If the QW or QWS command you are attempting to use does not appear in the specified command table, then that command table was never updated, the PDS member containing the updated command table is not in any of the libraries allocated to ISPTLIB, or the PDS member containing the updated version is preceded by another member of the same name in the libraries concatenated to ISPTLIB. If necessary, you can refresh your understanding of ISPF command tables by reviewing the section titled "ISPF Command Tables" on page 93.

4. **The MVS/QuickRef invocation command (QW or QWS) does not function correctly.** Use ISPF option 3.9 (as outlined in #2 above) to browse the appropriate command table and to see how the QW or QWS command you are attempting to use is defined. Make sure the command definition matches one of the command definitions specified in the QWCMDS CLIST. You can browse the QWCMDS CLIST and do repeat finds for 'ZCTACT' to see how the QW and QWS commands are defined by this CLIST. With the exception of the parm operand, which appears as '&PARM' in the command table and as '&&PARM' in the CLIST, the command definition in the command table should match one of the command definitions in the QWCMDS CLIST. If they do not match, then the command table was not updated by the QWCMDS CLIST and you should update the ISPF or site command table using the QWCMDS CLIST. If the command definition does match one of those in the QWCMDS CLIST, as outlined in the installation method you chose to use for MVS/QuickRef. After this is done, check to see if the action specified for that command is to select the QW CLIST. (i.e., CMD(%QW...). If so, then you responded yes when asked by the QWCMDS CLIST if you wanted to use the LIBDEF facility. If this was not what you intended, then you need to update the ISPF or site command
table again, indicating that you do not want to use LIBDEF services. If you did intend to use the LIBDEF facility, then refer to #4 below.

5. **MVS/QuickRef does not function properly when using the LIBDEF facility.** Refer to the information in #2 and #3 above to determine that the invocation command you are using (QW or QWS) is properly defined to ISPF and that it selects the QW CLIST. Make sure you are using a copy of the QW CLIST that has been properly customized for your installation and that it starts with the following statement: PROC 0 PRM(). Change the CONTROL statement in the QW CLIST from 'CONTROL MAIN NOMSG' to 'CONTROL MAIN MSG LIST', try invoking MVS/QuickRef again, and then check the diagnostic information which is produced. If you do not get any diagnostic information, then you are not executing the copy of the QW CLIST which you modified. You need to find the other copy of the QW CLIST and eliminate it.

6. **MVS/QuickRef does not function properly when being invoked by a PF key.** Use the KEYS or KEYLIST command (in the ISPF application where you are having the problem) to check the definition of the PF key you are attempting to use. Do not assume that the PF key is defined as expected. Remember that non-default PF key definitions have to be established within each ISPF application where you want to use that PF key, and that PF key definitions are sometimes lost when system upgrades are performed. If the PF key is properly defined, then the problem is not related to using a PF key and you need to review the other problems listed in this section.

7. **Cursor-driven invocation does not find an item expected to be in the data base.** First make sure the item really is in the data base by trying to access it with a fast-path string. If the item really is in the data base, then, if cursor-driven invocation is being invoked by a PF key, refer to #5 above to make sure that PF key is properly defined. If the PF key is properly defined, then check to see if the item the cursor was positioned under is prefixed or suffixed by a special character that is not translated to blanks by the QWIKOPTS CDIXLAT= parm. You can use the MVS/QuickRef QINFO command to see how the CDIXLAT= parm is currently defined. If the item is prefixed or suffixed by a special character that is not included in the list specified by the CDIXLAT= parm, then you may want to consider adding this special character to the CDIXLAT= specification. Of course, the problem may be that the item contains a special character that is translated to blanks by the CDIXLAT= parm. In this case, you may want to consider removing this special character from the list specified by the CDIXLAT= parm. As shown by the discussion above, while adding or removing special characters from the specification of the CDIXLAT= parm may cause some additional items to be found, it may also result in additional items that cannot be found. You will have to make a judgement call, based on the types of information your users most often access, on which characters can best be added or removed from the list specified by the CDIXLAT= parm.

8. **'Getnext based on cursor position' does not find an item expected to be in the data base.** First make sure the item really is in the data base by trying to access it with a fast-path string. If the item really is in the data base, then check to make sure it is in the same product currently on display. (Getnext based on cursor position only looks for the indicated item in the same product as the item currently on display.) If the item is in the same product, then refer to the information in #6 about checking the setting of the QWIKOPTS CDIXLAT= parm.
9. **Panel QWIKREFA causes display of an ISPF "Panel too large" message.** Browse panel QWIKREFA in the panel library you are currently using. Check the body of the panel to see if the **U-Local User Menu** option line was added to the body of the panel. If it was and a blank line was not taken out, then you have 25 lines in the body section of the panel which is one too many on a mod 2 terminal. Delete one of the blank lines from the body of the panel, get out of ISPF, get back into ISPF and then the problem should be solved.

**Production Implementation**

To implement MVS/QuickRef into production for all ISPF users, make sure that the modified ISPF system or site command table member is included in the concatenation for your production ISPF table library (ISPTLIB). (Also make sure that the modified system or site command table member either replaces or is concatenated ahead of the original, non-modified version.) You should also ensure that the MVS/QuickRef application command table unloaded from the installation tape is copied into or concatenated to the ISPF production table library (ISPTLIB).

When the QW LIBDEF CLIST is **not** being used, you should also make sure that the MVS/QuickRef panel and message libraries unloaded from the installation tape are copied into or concatenated to your production ISPF panel (ISPPLIB) and message (ISPMLIB) libraries, respectively. Also, the MVS/QuickRef load module library should be copied into or concatenated to your ISPF production link library (ISPLLIB) or copied into a data set in the system link list. Don’t forget to perform an LLA refresh afterwards if you are working on an MVS/ESA, OS/390, or z/OS system.

If you **are** using LIBDEF to invoke MVS/QuickRef, copy your QW LIBDEF CLIST into the production CLIST library.

**Production Implementation Considerations**

Please note that MVS/QuickRef is invoked as a NEWAPPL under ISPF, and therefore is assigned a new default set of PF key definitions by ISPF the very first time MVS/QuickRef is invoked by an ISPF user. For this reason, new users of MVS/QuickRef may have to set new PF key definitions once MVS/QuickRef is invoked for the first time, if they have changed their PF key definitions from the normal ISPF defaults.

**User Documentation Considerations**

There are several alternative methods which you can use to get MVS/QuickRef documentation into the hands of your users. These are described below.

First of all, you can have this user's guide copied and distributed to all potential MVS/QuickRef users. If you choose this alternative, then you should also consider distributing a memo describing the various installation-dependent options which you may have chosen to install or to eliminate, including:
♦ an indication of which invocation commands (QW and/or QWS) were installed

♦ an indication of whether or not user data bases were installed and, if so, a description of the type of information contained in each one and, if more than one, the user data base number assigned to each one

♦ any other installation-dependent options which affect how MVS/QuickRef is accessed and used in your particular installation

A memo like this will eliminate a lot of user confusion (and save you a lot of questions and phone calls).

If you prefer not to copy and distribute the entire user's guide, then you can obtain a copy in Adobe Acrobat™ format from our web site, http://www.quickref.com. The Adobe Acrobat™ reader can be used to view and print the copy of the user's guide from our web site. The reader is free and is available from Adobe's web site at http://www.adobe.com.

As a last alternative for providing user documentation, you can use the QPRINT command to print and distribute the MVS/QuickRef Reference Sheet found in the help facility (under item name QUICK-START). This option can be used in conjunction with any of the other alternatives outlined above. The QPRINT command can also be used to copy the reference sheet to a data set. If you do this, then you can make an "edited" copy for distribution. This edited version of the reference sheet could also include information on the type of installation-specific options described above.

**Updating the MVS/QuickRef Data Base**

New releases of MVS/QuickRef are generally made available twice a year. These new releases always contain a new, updated copy of the MVS/QuickRef main data base. From time to time, however, it may be necessary to update or correct the information in the main data base more frequently. There are three ways in which this may be accomplished:

1. a full, refresh copy of the MVS/QuickRef main data base may be provided

2. an update data base (which can be merged into the existing main data base) may be provided

3. you can use the override parameter REPLACE statement to provide corrected information for any individual reference item in the data base

In general, option #1 would be used when the updates or corrections are extensive enough that the entire data base needs to be replaced. Option #2 would be used when the updates or corrections are not extensive enough to merit replacing the whole data base but are more extensive than what can conveniently be handled via the override parameter facility. Option #3 would be used when just one or two reference items need updating or correcting.
These three options are fully explained in the following sections titled "Refreshing the MVS/QuickRef Data Base", "Data Base Merge Facility", and "Correcting the Text of an MVS/QuickRef Item".

**Refreshing the MVS/QuickRef Data Base**

The information available from MVS/QuickRef is constantly being updated and improved. As part of this process, you may periodically receive a refresh copy of the MVS/QuickRef main data base. When you do receive a data base refresh, you can use member QWUPDATE in the MVS/QuickRef JCL data set to update your DASD copy of the main data base. To perform the data base update, edit member QWUPDATE in the MVS/QuickRef JCL data set and change the data set name symbolic parameter to match the name of the MVS/QuickRef main data base at your installation. Change the symbolic parameter for the distribution tape volume serial number to match the volser of the data base refresh tape you were sent. Add a job card and submit the job for execution.

Please note that the QWUPDATE job allocates the DASD version of the new data base as DISP=(NEW,...) so it will first be necessary to delete or rename the current MVS/QuickRef main data base. Also, MVS/QuickRef cannot be in use by any TSO user when this job executes. Alternatively, you might want to create the new data base under a completely different name and, after the QWUPDATE job runs successfully (and you do some initial testing to verify it), you can then scratch the old data set and rename the new one as required for production.

If you used the QWIKSLCT selective data base load utility when MVS/QuickRef was installed, then you should execute it again from the refresh tape or from the output file produced by the QWUPDATE job.

**Data Base Merge Facility**

The MVS/QuickRef development staff may periodically provide 'update' data bases containing corrections or additions to the information in the main data base. You can use the data base merge facility to merge the new or corrected information contained in an update data base into your main data base.

As an example of how to use an update data base, suppose an important new release of a given product comes out just after a new release of MVS/QuickRef is distributed. If enough users express an immediate need to have this new release documented in MVS/QuickRef, then, rather than forcing the user community to wait for the next general release of MVS/QuickRef, the MVS/QuickRef development staff may make documentation for this new release available via an update data base.

When any update data bases are available, a link will be added to the www.quickref.com web site. This link will provide a list showing the update data bases which are available along with a general description of each update data base.
The update data base link will also show the unique time stamp assigned to each update data base. The time stamp assigned to each update data base is in the form YYYYDDD HHMMSSSTH, where YYYY is the year, DDD is the julian day, and HHMMSSSTH is the hour, minute, second, and thousands of a second when the update data base was created. The timestamp provides a unique identifier for each update data base which is independent of the data set name assigned as it is moved into your mainframe environment.

The MVS/QuickRef web site will also provide information on how to transfer a given update data base into your mainframe environment.

If you decide to use a given update data base, then, before transferring it to your mainframe environment, you should make a note of the time stamp assigned to that update data base on the MVS/QuickRef web site.

Once you have a given update data base in your mainframe environment, you should review the information in that update data base. This is important for several reasons. First of all, it will allow you to check and make sure you have the right update data base. It will also allow you to see exactly what added or updated product information is contained in that update data base. Finally, it will allow you to review certain special processing instructions that may be needed for a particular update data base.

Once you have it in your mainframe environment, you can use MVS/QuickRef to review the information in an update data base. You can do this by setting up an ISPF test environment where the update data base is treated as if it were the MVS/QuickRef main data base. When you invoke MVS/QuickRef in your test environment, you will be viewing the information in the update data base rather than your existing main data base.

If you are not using the libdef technique to invoke MVS/QuickRef, then you can view the update data base using the TSO FREE and ALLOC commands. Go to the TSO ready prompt and type:

```
FREE F(QWREFDD)
```

and then press ENTER. This will free the main data base if it is preallocated. If it is not preallocated, the FREE command will indicate that the QWREFDD DD statement is not allocated. (This will not hurt anything.) Then type:

```
ALLOC F(QWREFDD) DA('xxx.xxx.xxx') SHR REUS
```

and press ENTER, where xxx.xxx.xxx is the data set name for the update data base you want to view. Then type ISPF and press ENTER to get into ISPF. Now when you invoke MVS/QuickRef, you will looking at the update data base as if it were the main data base.

If you are using the LIBDEF facility to invoke MVS/QuickRef, then you will need to place a copy of the QW LIBDEF CLIST in a CLIST library which is ahead of the library containing the normal QW CLIST in the SYSPROC concatenation. Then change the ALLOC statement for the main data base in the copied version of the QW CLIST so that it allocates the update data base.
Then when you invoke MVS/QuickRef, you will be looking at the update data base as if it were
the main data base.

Note: The techniques outlined above for viewing the update data base are based on the
assumption that MVS/QuickRef was installed in a normal manner as outlined in the user
guide. If a non-standard installation was done, then the techniques above may have to be
modified accordingly.

Once you have set up the test environment required to view the update data base, invoke
MVS/QuickRef and request a list of all products in the main data base. You can do this by
selecting option L off of the main menu or by using the L=V fast-path string. This will show
you a list of all products in the update data base (i.e., all the products being added or updated by
this update data base). You can select each product from the list in turn in order to see the
individual items being added or updated for each product.

Be sure to check for a product with a vendor name of CHICAGO-SOFT and a product name of
UPDATE DATABASE INFO. This is a "dummy" product that, if present in an update data
base, provides special information about that update data base. It will not be merged into your
main data base by the data base merge facility.

The type of information provided by the UPDATE DATABASE INFO product will generally
include a more detailed description of the purpose of the update data base (i.e., more detail on
what type of information it will add or correct in the main data base). It may also contain special
processing instructions for this update data base.

In general, an update data base can be merged into your main data base by simply using the data
base merge facility. However, some update data bases may require additional special processing
before they can be merged into your main data base. For these type of update data bases, the
UPDATE DATABASE INFO product will contain an item providing the special processing
instructions that you need to follow.

While viewing your update data base, be sure to use the MVS/QuickRef QINFO command. This
command can be entered on the command line of any MVS/QuickRef panel. It will display
certain information about the installation of MVS/QuickRef in your processing environment. In
particular, it will display the time stamp assigned to your main data base. The time stamp
appears several lines down in the display.

When an update data base is allocated, the QINFO command will display the time stamp
assigned to that update data base. You should check this time stamp against the one that was
shown for that update data base on the MVS/QuickRef web site to make sure you are using the
correct update data base.

Once you have verified that you have the correct update data base in your mainframe enviroment
and you have taken care of any special processing instructions that may be contained in the
UPDATE DATABASE INFO product, you are ready to use the data base merge facility to
merge the update data base into your main data base.
The data base merge facility consists of program QWIKMRGE and JCL member QWIKMRGJ.

Program QWIKMRGE merges the contents of one or more update data bases into your main data base. It does this by taking your existing main data base and from one to five update data bases as input. It then outputs a new main data base which contains all the information in the original main data base that was not updated or corrected plus all the new, updated, or corrected information contained in the specified update data bases.

JCL member QWIKMRGJ provides the JCL required to execute program QWIKMRGE. The QWIKMRGJ member is stored in the MVS/QuickRef JCL library. The QWIKMRGE program is stored in the MVS/QuickRef linklib. Be sure to add a job card and customize the QWIKMRGJ JCL as required for your processing environment before submitting it for execution.

A sample of the QWIKMRGJ JCL is shown and described below:

```plaintext
//*********************************************************************
//*                                                                   *
//*      This member contains sample JCL for the data base            *
//*      merge program (QWIKMRGE).                                    *
//*                                                                   *
//*********************************************************************
//JS10    EXEC PGM=QWIKMRGE,REGION=0M
//STEPLIB   DD DSN=quickref.linklib,            Program Library
//          DISP=SHR
//SYSPRINT  DD SYSOUT=*                         Report Output
//INDB      DD DSN=quickref.input.database,     Existing main data base
//          DISP=SHR
//*          One update data base must be provided via the UPDATE1 dd statement
//UPDATE1   DD DSN=quickref.update.database.number1,DISP=SHR
//*          Update data bases 2 through 5 are optional
//UPDATE2   DD DSN=quickref.update.database.number2,DISP=SHR
//UPDATE3   DD DSN=quickref.update.database.number4,DISP=SHR
//UPDATE4   DD DSN=quickref.update.database.number4,DISP=SHR
//UPDATE5   DD DSN=quickref.update.database.number5,DISP=SHR
//*          New main data base
//OUTDB     DD DSN=quickref.output.database,      New main data base
//          DISP=(,CATLG,DELETE),UNIT=SYSDA,
//          SPACE=(TRK,(6000,1000),RLSE),
//          DCB=(RECFM=F,BLKSIZ=6160,LRECL=6160) <---Do Not Change
//SORTIN    DD UNIT=3380,SPACE=(TRK,(2500,200),RLSE) Sort Input
//SORTOUT   DD UNIT=3380,SPACE=(TRK,(2500,200),RLSE) Sort Output
//SORTWK01  DD UNIT=3380,SPACE=(TRK,(2000,200),RLSE) Sort
//SORTWK02  DD UNIT=3380,SPACE=(TRK,(2000,200),RLSE) Work
//SORTWK03  DD UNIT=3380,SPACE=(TRK,(2000,200),RLSE) Files
//SYSOUT    DD SYSOUT=*                         Sort Listing
//SYSUDUMP  DD SYSOUT=*                         
//STEPLIB DD - this required DD statement specifies the MVS/QuickRef linklib at your site
```
SYSPRINT DD - this required DD statement contains a report titled the "Data Base Merge Report"; it shows the products that were added or updated by each update data base being processed

INDB - this required DD statement specifies the existing main data base

UPDATE1 - this required DD statement specifies the first update data base to be used

UPDATE2 through UPDATE5 - these are optional DD statements that can be used to specify additional update data bases (for a total of 5); however, they must be used in order (UPDATE2 must be used to specify the second update data base; UPDATE3 must be used for the third update data base, etc.)

OUTDB - this required DD statement specifies the new main data base

SORTIN, SORTOUT, SORTWKnn, SYSOUT - these required DD statements are SORT work files; they should not require any modification unless the default space allocations turn out to be too small

After running the QWIKMRGE program, you should check for a return code of zero and review the information in the "Data Base Merge Report". If the return code is not zero, check the "Data Base Merge Report" for error or warning messages.

You should also use MVS/QuickRef to review the information in the new main data base that was produced. You can do this setting up an ISPF test environment where the new data base is treated as the MVS/QuickRef main data base. You can do this in the same manner described previously for an update data base.

Once you have set up the test environment required to view the new main data base, invoke MVS/QuickRef and request a list of all products in the main data base. You can do this by selecting option L off of the main menu or by using the L=V fast-path string. This will show you a list of all products in the new main data base. You should check each product that was supposed to be added or updated to make sure that it is present.

After you are satisfied with the new main data base, you can scratch the existing main data base and rename the new one as required to make it your production main data base.

Once the data base merge facility has been used to update the main data base, the QINFO command will show the number of QWIKMRGE executions that have been performed against the main data base and will also list the time stamps of the last (up to 5) update data bases that have been applied to that main data base.

**Selective Data Base Load Facility**

The main data base supplied with MVS/QuickRef contains a vast amount of reference material, stored under many different products. Within each product, reference information is stored as
one or more items. You can use the utility described in this section to selectively load to DASD only those products and items you need. For example, if you are a CICS shop but not an IMS shop, you will probably want to discard the IMS messages, status codes, and user abends from the MVS/QuickRef main data base to decrease the amount of DASD space it occupies.

The selective data base load process involves executing the QWIKSLCT utility. The utility reads in the master copy of the MVS/QuickRef main data base (not a user data base!) and a set of control statements indicating which products and items to discard (or which to keep), and outputs a new copy of the data base from which the unwanted reference information has been excluded. Execution JCL for the QWIKSLCT utility appears below; it is followed by a description of each element of the JCL. A copy of the JCL below is stored in member QWIKSLCJ in the MVS/QuickRef JCL data set. You can also execute module QWIKSLCT during the MVS/QuickRef installation process. See member QWLOAD in the MVS/QuickRef JCL data set for more information.

QWIKSLCT selective data base load utility JCL:

```
//JS10    EXEC PGM=QWIKSLCT,REGION=0K
//STEPLIB DD DSN=quickref.linklib,DISP=SHR
//SYSPRINT DD SYSOUT=* 
//SYSUT1 DD DSN=input.database,DISP=SHR
//SYSIN DD *
    include/exclude control statements
/*
//SYSUT2 DD DSN=output.database,
//       DISP=(,CATLG,DELETE), 
//       UNIT=SYSDA,
//       SPACE=(TRK,(1400),RLSE),
//       DCB=(RECFM=F,BLKSIZE=6160,LRECL=6160}
//SORTIN DD UNIT=3380,SPACE=...
//SORTOUT DD UNIT=3380,SPACE=...
//SORTWK01 DD UNIT=3380,SPACE=...
//SORTWK02 DD UNIT=3380,SPACE=...
//SORTWK03 DD UNIT=3380,SPACE=...
//SYSOUT DD SYSOUT=* 
```

STEPLIB DD - this DD points to the MVS/QuickRef load module library at your site

SYSPRINT DD - this DD is required; an output report and any issued error messages are written to this DD.

SYSUT1 DD - this required DD allocates the input master copy of the MVS/QuickRef main data base. This file can be allocated to a DASD copy of the file, or to the master copy of the data base on the product distribution tape.

SYSIN DD - this DD points to the INCLUDE/EXCLUDE control statements that are used to indicate which items in the input data base are to be discarded (or which are to be kept). You will find a “sample set” of control statements in member QEXCLUDE in the MVS/QuickRef JCL data set. This member must be modified before it is used; for more information, see the description of the INCLUDE/EXCLUDE control statements in the paragraphs which follow.
SYSUT2 DD - this required DD allocates the output file into which the input file is copied, after all items indicated by the INCLUDE/EXCLUDE control statements in the SYSIN file are discarded. DCB attributes must be as coded in the sample JCL above. This file MUST be allocated to a DASD device.

SORTIN, SORTOUT, SORTWKnn, SYSOUT - These are required SORT work files. They should not require any modification unless the disk space allocations should turn out to be too small.

**INCLUDE/EXCLUDE Control Statements:**

Here are some sample INCLUDE/EXCLUDE control statements:

```
ALTAI  ZACK MESSAGES  2.1  *
ALTAI  ZACK MESSAGES  2.2  *
IBM    C LANGUAGE SYNTAX  2.1  *
```

As you can see from the sample control statements above, an INCLUDE/EXCLUDE control statement generally "names" a product to be included in or excluded from the output file.

The first non-blank character, other than an asterisk, encountered in column one in the QEXCLUDE member will indicate the type of statements being used (INCLUDES or EXCLUDES) as well as the type of processing to be performed by the QWIKLCT program.

To activate a control statement as an INCLUDE statement (and to include the product named on that statement), you must place the letter 'I' in column one of that statement.

To activate a control statement as an EXCLUDE statement (and to exclude the product named on that statement), you must place the letter 'E' in column one of that statement.

If the first non-blank, non-asterisk character encountered in column one is an 'E', then QWIKLCT will only process and report on EXCLUDE statements. In this case:
- any product on a statement with an 'E' in column one will be excluded
- any statement containing any other character in column 1 will be ignored
- any product not specifically excluded will be automatically included

If the first non-blank, non-asterisk character in column one is an 'I', then QWIKLCT will only process and report on INCLUDE statements. In this case:
- any product on a statement with an 'I' in column one will be included
- any statement containing any other character in column 1 will be ignored
- any product not specifically included will be automatically excluded
There is no need to mix INCLUDES and EXCLUDES. For example, if the first non-blank, non-asterisk character in column one is an 'E', then only EXCLUDE statements will be processed. In this case, there is no need to put an 'I' in column one for a product to be included since any product not specifically excluded is automatically included.

Here is the complete set of syntax rules for coding INCLUDE/EXCLUDE control statements:

1. Columns 4 through 18 of a given control statement must contain the vendor name (left-justified) associated with the product identified on that control statement.

2. Columns 20 through 39 must contain the product name (left-justified) associated with the product identified on that control statement.

3. Columns 41 through 55 must contain the release number (left justified) associated with the product identified on that control statement.

4. Columns 57 through 72 must contain the item name (left justified) of the specific item to be included or excluded from the product identified on that control statement.

5. The sequence in which the INCLUDE/EXCLUDE control statements are ordered does not matter to the QWIKSLCT utility.

6. You can comment out an INCLUDE/EXCLUDE control statement by coding an asterisk (*) in column one.

7. The item field can contain a full item name or a generic item name. A generic item name is a partial item name ending with an asterisk (*). If it contains a full item name, then only that specific item within the indicated releases of the indicated products will be included or excluded. If it contains a generic item name, then all matching items within the indicated releases of the indicated products will be included or excluded. If it contains a single asterisk, then all items in the indicated releases of the indicated products will be included or excluded.

8. The release field can contain a full release number or a generic release number. A generic release number is a partial release number ending with an asterisk (*). If it contains a full release number, then all matching items in that specific release of the indicated products will be included or excluded. If it contains a generic release number, then all matching items within all matching releases of the indicated products will be included or excluded. If it contains a single asterisk, then all matching items in all releases of the indicated products will be included or excluded.

9. The product field can contain a full product name or a generic product name. A generic product name is a partial product name ending with an asterisk (*). If it contains a full product name, then all matching items in matching releases of that specific product will be included or excluded. If it contains a generic product name, then all matching items within all matching releases of all matching products will be included or excluded. If it contains
a single asterisk, then all matching items in all matching releases of all products from the indicated vendor will be included or excluded.

10. The vendor field can contain a full vendor name or a generic vendor name. A generic vendor name is a partial vendor name ending with an asterisk (*). If it contains a full vendor name, then all matching items in matching releases of matching products belonging to that specific vendor will be included or excluded. If it contains a generic vendor name, then all matching items within all matching releases of all matching products belonging to matching vendors will be included or excluded. If it contains a single asterisk, then all matching items in all matching releases of all matching products from all vendors will be included or excluded.

You will find a sample set of control statements in member QEXCLUDE in the MVS/QuickRef JCL data set. This member MUST be modified before use; as distributed, it contains control statements for every release of every product in the MVS/QuickRef main data base, but the statements do not contain the 'I' or 'E' indicator in column one.

Below is an example of the type of INCLUDE/EXCLUDE statements provided in member QEXCLUDE:

```
* Vendor        Product            Release       Item
ALTAI         ZACK MESSAGES        2.1           *
ALTAI         ZACK MESSAGES        2.2           *
IBM           C LANGUAGE SYNTAX    2.1           *
IBM           COBOL370 SYNTAX      1.1           *
IBM           PL/I SYNTAX          3.0           *
```

As you can see from this example, a comment statement precedes the actual INCLUDE/EXCLUDE control statements and provides column headings for the vendor name, product name, release number, and item name fields. The item name field is shown as a single asterisk in each case because the INCLUDE/EXCLUDE control statements are setup to allow for including or excluding all items in one specific release of each specific product carried in the main data base.

If the sample INCLUDE/EXCLUDE statements were modified as shown below:

```
* Vendor        Product            Release       Item
ALTAI         ZACK MESSAGES        2.1           *
E ALTAI         ZACK MESSAGES        2.2           *
IBM           C LANGUAGE SYNTAX    2.1           *
IBM           COBOL370 SYNTAX      1.1           *
E IBM           PL/I SYNTAX          3.0           *
```

then all items in release 2.2 of the product named 'ZACK MESSAGES' from vendor 'ALTAI' would be excluded as would all items in release 3.0 of 'PL/I SYNTAX' from vendor 'IBM'. All other items in other releases of all other products would be automatically included.

If the sample INCLUDE/EXCLUDE statements were modified as shown below:
then all items in release 2.2 of the product named 'ZACK MESSAGES' from vendor 'ALTAI' would be included as would all items in release 3.0 of product 'PL/I SYNTAX' from vendor 'IBM'. All other items in all other releases of all other products would be automatically excluded.

If the sample EXCLUDE statements were modified as shown below:

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Release</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>ALTAI</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>ALTAI</td>
<td>ZACK MESSAGES</td>
<td>2.2</td>
<td>*</td>
</tr>
<tr>
<td>IBM</td>
<td>C LANGUAGE SYNTAX</td>
<td>2.1</td>
<td>*</td>
</tr>
<tr>
<td>IBM</td>
<td>COBOL370 SYNTAX</td>
<td>1.1</td>
<td>*</td>
</tr>
<tr>
<td>IBM</td>
<td>PL/I SYNTAX</td>
<td>3.0</td>
<td>*</td>
</tr>
</tbody>
</table>

then all items in all releases of the product 'ZACK MESSAGES' from vendor 'ALTAI' would be excluded. All other items in all other releases of all other products would be automatically included.

If the sample EXCLUDE statements were modified as shown below:

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Release</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>ALTAI</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>ALTAI</td>
<td>*</td>
<td>2.2</td>
<td>*</td>
</tr>
<tr>
<td>IBM</td>
<td>C LANGUAGE SYNTAX</td>
<td>2.1</td>
<td>*</td>
</tr>
<tr>
<td>IBM</td>
<td>COBOL370 SYNTAX</td>
<td>1.1</td>
<td>*</td>
</tr>
<tr>
<td>IBM</td>
<td>PL/I SYNTAX</td>
<td>3.0</td>
<td>*</td>
</tr>
</tbody>
</table>

then all items in all releases of all products from vendor 'ALTAI' would be included. All other items in all other releases of all other products would be automatically excluded.

Remember, you should not mix INCLUDES and EXCLUDES within the QEXCLUDE member. If the first non-blank, non-asterisk character encountered in column one in the QEXCLUDE member is an 'E', then only EXCLUDE statements will be processed; if it is an 'I', then only INCLUDE statements will be processed.

You will have to decide which type of INCLUDE/EXCLUDE control statements to use. If you want to include more products than you want to exclude, then you will probably want to use EXCLUDE statements (since, in this case, there will be fewer 'E's required in column one compared to the number of 'I's required in column one to do INCLUDES). On the other hand, if you want to exclude more products than you want to include, then you will probably want to use INCLUDE statements (since, in this case, there will be fewer 'I's required in column one compared to the number of 'E's required in column one to do EXCLUDES).
Here is another factor to consider when trying to determine which type of INCLUDE/EXCLUDE control statements to use. If you are using EXCLUDE statements and fail to place an 'E' on a product you really wanted to exclude, the worst thing that happens is that you waste some disk space. If you are using INCLUDE statements and fail to place an 'I' on a product you really want to include, then that product will not be included (and the reference information provided by that product will not be available to your users). So, if you are using INCLUDE statements, make sure you place an 'I' in column one for each and every product you really want to include.

Selective Load Versus User-Directed Selection List Processing

There are two reasons for using the selective data base load facility:

1. to save DASD storage space by utilizing a smaller main data base

2. to eliminate user confusion, when two or more releases of a given product appear in the main data base, by eliminating all releases of that product other than the release currently being used

The user confusion mentioned in item number 2. above has to do with the fact that a request to display the reference text for a given item often results in a selection list showing that the requested item is carried in two or more releases of the same product. The user then has to know which release is appropriate in order to select the appropriate item from the list. Many users may not know which release is the correct one to choose.

If eliminating user confusion is your main reason for using the selective data base load facility, then you should also consider using user-directed selection list processing. User-directed selection list processing will not eliminate all the confusion that may result from having multiple releases of the same product in the data base, but it will alleviate the specific type of user confusion described above, and it is somewhat easier to use and more flexible than the selective data base load facility. (For a general description of user-directed selection list processing, see the section titled "User-Directed Selection List Processing" in Chapter Three.)

With the selective data base load facility, the only way to eliminate the type of user confusion described above is to eliminate every release of a given product except one. If you later install a new release of a given product, then you have to go back to your QEXCLUDE member, change it so that the new release is no longer excluded but the old release is, and then rerun the QWIKSLCT selective data base load utility. And, if you install a new release of the operating system, a lot of products may be involved in this process. Another disadvantage of this technique is that sometimes you really need to keep two or more releases of the same product in the data base. For example, you may have different groups of users using different releases of the same product.
User-directed selection list processing can solve many of these problems. For example, operating system level selection list processing, one of the three types of user-directed selection list processing, causes MVS/QuickRef to (for "operating system specific" products in the situation described above) automatically select the release which best matches the release of the operating system on which you are running. If you later install a new release of the operating system, operating system level selection list processing will automatically start selecting the appropriate product releases for the new operating system. And, while testing a new release of the operating system, when you may need to alternate running the old and new releases for a while, operating system level selection list processing will automatically adjust to the release being used each time you IPL the system.

When you use vendor/product specific selection list processing, another type of user-directed selection list processing, all you have to do to is change the preferred release of an ISV product when you install a new release of a that product. You simply change the appropriate entry in the Vendor/Product Specific Selection List Copy Member (QWIKVPSC) and then reassemble and relink the Vendor/Product Specific Selection List Table (QWIKVPST).

As indicated by the examples above, user-directed selection list processing does provide an easier to use, more flexible solution to the type of user confusion described above. However, unlike the selective data base load facility, it will not save DASD storage space.

You will have to develop your own philosophy governing the use of the selective data base load facility versus user-directed selection list processing. Here are some key points to keep in mind:

- if DASD storage space is in short supply, then you will definitely need to use the selective data base load facility
- if you use the selective data base load facility to eliminate all but one release of each product in the main data base, then you will not need to use user-directed selection list processing
- if DASD storage space is readily abundant, consider loading the full data base and relying on user-directed selection list processing to alleviate the type of user confusion described above
- you can use both techniques

As an example of using both techniques, suppose the main data base contains the following three releases of product CA-1: V5R1, V5R2, and V5R3. Now suppose you are currently running CA-1 V5R2 in production but expect to go to V5R3 in the near future. Then you could use the selective data base load facility to eliminate CA-1 V5R1 from the main data base. Since you are not likely to go back to a previous release of CA-1, this will save DASD storage that would be wasted by leaving CA-1 V5R1 in the data base. Then you could use user-directed selection list processing to make V5R2 the preferred release until such time as you install V5R3. When V53 is installed, you could make V5R3 the preferred release.
Correcting the Text of an MVS/QuickRef Item

The MVS/QuickRef development staff is constantly striving for product perfection; however, you may occasionally find a typographical error or a content error within an information item in the MVS/QuickRef data base. The steps that follow can be used to correct the error in your copy of the data base after you report the error to the MVS/QuickRef Support Group:

1. Invoke MVS/QuickRef and display a reference item similar in form to the one you want to add. If you are correcting a spelling or syntax error in an existing item, display the item in error.

2. When the item is displayed, type CUT ALL on the command line; the full text of the item will be cut for subsequent pasting.

3. Create and edit a new member that you want to use as the corrected override text and enter the !PASTE command on the command line while editing the member; the cut text will be pasted in. If you have renamed the PASTE command, to QWPASTE, type QWPASTE instead of !PASTE on the command line. Save the changed member.

4. Edit the member created in step three above and delete the TOP OF DATA, COPYRIGHT, and/or BOTTOM OF DATA lines. Correct the text error or alter the text to meet the criteria of the item you are adding.

5. Add an appropriate REPLACE statement to your QWPARM00 override parameter member. Parameters for the REPLACE statement are discussed in chapter six of this guide.

6. Allocate a QWPARMS DD statement whenever MVS/QuickRef is invoked, either via the LOGON procs, an ALLOCATE command, your LIBDEF CLIST for MVS/QuickRef, or the QWIKOPTS options table PARMDSN= option. For instructions on using the override parameter facility, refer to chapter six.

When MVS/QuickRef is now invoked, your corrected text will be displayed instead of the text from the main data base whenever the chosen item is selected for display by any MVS/QuickRef user. If the item did not previously exist in the data base, it will be treated as an “add” and will now be eligible for selection and display.

Note: If you find an item in error in the MVS/QuickRef main data base, please notify the MVS/QuickRef support group. If the item is used frequently, we can provide a corrected copy of the item in an update data base, post the data base to our web site, and you can then download the data base and use the QWIKMRGE utility to merge the update data base into your main data base, correcting the error.

Merging Data From a Previous Release

It may sometimes be desirable to merge data into your current main data base from a main data base associated with a previous release of MVS/QuickRef. As an example of this situation,
consider that the MVS/QuickRef main data base normally contains just the two most current releases of each product from an independent software vendor. If it is critical that your current main data base contain a release of a product from an independent software vendor that is older than the last two releases, you may want to merge that older release into your current main data base from the main data base associated with an older release of MVS/QuickRef.

Please note that this can be done only if there have been no data base structure changes between the MVS/QuickRef release containing the old data and the MVS/QuickRef release into which you will be merging the old data. You can generally tell if there have been data base structure changes by looking at the “Upgrade Information for Existing Users” at the beginning of Chapter Four in the User’s Guide for the current release of MVS/QuickRef.

You will also need to be familiar with the information presented in the sections on “Data Base Merge Facility” and “Selective Data Base Load Facility” presented earlier in this chapter.

The general technique to merge in old data is as follows:

1. Create a QEXCLUDE member which contains one INCLUDE statement for each release of each product you want to bring forward from the old release of MVS/QuickRef.

2. Run program QWIKSLCT using the QEXCLUDE member created in step 1. above against the MVS/QuickRef main data base containing the old data in order to create a “mini” data base that contains just the releases of the products you want to bring forward. You can customize the JCL in member QWIKSLCJ in the MVS/QuickRef JCL library to execute program QWIKSLCT in this manner.

3. Run program QWIKMRGE using the “mini” data base created in step 2. above as if it were an update data base and using the current MVS/QuickRef main data base as the input data base. You can customize the JCL in member QWIKMRGJ in the MVS/QuickRef JCL library to execute program QWIKMRGE in this manner.

This will create a new main data base containing everything in the current main data base as well as the data brought forward and merged in from the older main data base.

Since there can be no data base structure changes between the release of MVS/QuickRef containing the old data and the current release of MVS/QuickRef, you can run the QWIKSLCT and QWIKMRGE programs from the linklib associated with the current release of MVS/QuickRef.

When you run program QWIKMRGE taking in data from an older release, you will get a return code of 4 and a warning message shown in the Data Base Merge Report indicating that data from an update data base with an earlier time stamp is being merged in. As long as this is what you intended to do, this warning message can be ignored.
To better illustrate this process, let’s take a “real-life” example, using the following assumptions:

- the main data base associated with the current release of MVS/QuickRef contains release 2 and release 3 of product CA-1 from independent software vendor CA
- the main data base associated with the prior release of MVS/QuickRef contains release 1 of product CA-1 from independent software vendor CA
- you want to bring release 1 of product CA-1 forward from the prior release of MVS/QuickRef and merge it into the main data base associated with the current release of MVS/QuickRef
- the releases of CA-1 referred to above are not necessarily the correct or current releases of this product; they are intended for purposes of illustration only

With this set of assumptions, here are the steps to follow:

1. Create a QEXCLUDE member which contains one INCLUDE statement for release 1 of product CA-1 from vendor CA
2. Run program QWIKSLCT using the QEXCLUDE member created in step 1. above against the MVS/QuickRef main data base associated with the prior release of MVS/QuickRef. This will create a “mini” data base that contains just release 1 of product CA-1 from vendor CA.
3. Run program QWIKMRGE using the “mini” data base created in step 2. above as if it were an update data base and using the current MVS/QuickRef main data base as the input data base.
4. This will create a new main data base containing everything in the current main data base as well as release 1 of product CA-1 from vendor CA.

**Using MVS/QuickRef Under CA-ROSCOE**

CA-ROSCOE is a multiuser online program development system available from Computer Associates International. CA-ROSCOE has within it a facility known as “ETSO” which allows you to execute TSO commands under CA-ROSCOE. If you have CA-ROSCOE and the ETSO feature, you can execute MVS/QuickRef under CA-ROSCOE ETSO using a set of sample RPFs supplied with MVS/QuickRef.

Under CA-ROSCOE, none of the MVS/QuickRef subcommands are supported. These include CUT and PASTE, QPRINT, QINFO, or SEARCH.
Users of CA-ROSCOE 5.7 or Above

An interface for users of CA-ROSCOE Release 5.7 or later is available that invokes MVS/Quick-Ref directly under CA-ROSCOE ETSO as a called program, without the need for ISPF services. The CA-ROSCOE interface elements can be found in data set CSL.QUICKREF.CAROSCOE, stored as file nine (9) on the first MVS/QuickRef distribution tape. This file is a single tape copy of a PDS and can be restored to DASD using the ROSCOEDL member in the MVS/QuickRef JCL library. About 30 tracks of 3380 DASD space and 20 directory blocks are sufficient to contain this data set. After restoring it to DASD, review the $QWNOTES member in the data set to obtain installation instructions for this interface.

To prepare for execution of an ETSO program under ROSCOE, an ETSONL LIB DD must be added to the ROSCOE startup procedure, and the program being called must reside in the ETSONL LIB library. The program must also be added to the Eligible Program List (i.e., ETSOPGMS) which is usually located in the ROSCOE control signon. The parms for the Eligible Program List for MVS/QuickRef for ROSCOE ETSO 5.7 and 6.0 are listed in the table found in the next section.

Note: Each CA-ROSCOE user must have a sufficient number of AWSs for the user. Although 6 AWSs will support the full function of the interface, the recommended number is 7.

Users of CA-ROSCOE 5.6 or Below

If you have CA-ROSCOE 5.6 or earlier, then you must use member QWRPF in the MVS/QuickRef JCL library as the means of executing MVS/QuickRef under CA-ROSCOE ETSO. You will have to edit the RPF and change the data set names in it to match those at your installation. The RPF performs the necessary allocations and then invokes ISPF via the ISPSTART command, telling ISPSTART to immediately invoke MVS/QuickRef. When the user is finished with MVS/QuickRef, ISPF terminates and the user is back in CA-ROSCOE.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
<th>Col. Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Name</td>
<td>QWIKREF1</td>
<td>01 - 08</td>
</tr>
<tr>
<td>Max Concurrent Executions</td>
<td>110</td>
<td>10 - 12</td>
</tr>
<tr>
<td>Max Time Slices</td>
<td>5000</td>
<td>14 - 17</td>
</tr>
<tr>
<td>Max Total Getmain (K)</td>
<td>1024</td>
<td>19 - 24</td>
</tr>
<tr>
<td>Max Variable Getmain (K)</td>
<td>1024</td>
<td>26 - 31</td>
</tr>
<tr>
<td>Max Getmain above 16mb (K)</td>
<td>2048</td>
<td>33 - 38</td>
</tr>
<tr>
<td>Dump</td>
<td>N</td>
<td>47</td>
</tr>
<tr>
<td>Command Processor Flag</td>
<td>blanks</td>
<td>51 - 52</td>
</tr>
</tbody>
</table>

The EPL list addition in the chart above must be modified to use ISPSTART as the program name instead of QWIKREF1 when using the QWRPF RPF to invoke MVS/QuickRef under CA-ROSCOE 5.6. In addition, ROSCOE ETSO 5.6 users should retrofit the EPL values into the format of the EPL statement that is valid for CA-ROSCOE release 5.6.
CA-ROSCOE Considerations

When testing QWRPF under CA-ROSCOE 5.6 or earlier, you should ensure that your ISPF environment will work under your CA-ROSCOE system before using MVS/QuickRef under CA-ROSCOE under ISPF. If you experience problems using MVS/QuickRef under CA-ROSCOE, make sure that your CA-ROSCOE system has all necessary maintenance applied. If you are using QWRPF and you experience abends, try changing the CALL statement for ISPSTART at the bottom of the exec to a CALL to ISPF with no parameters on the CALL statement. This will make sure ISPF will execute without error under your CA-ROSCOE system. If you experience "timeout" situations, it may be necessary to increase the Max Time Slices field in the EPL entry for MVS/QuickRef.

Note: Remember that the program names in the EPL must be in alphabetical order.
Chapter 5 - Displaying Your Own Reference Information
Introduction

The information presented in this chapter will be helpful if you want to build and access your own MVS/QuickRef user data base, containing information you frequently need to refer to. The instructions in this chapter will guide you through the process of creating and installing your own MVS/QuickRef user data base.

Some of the instructions in this chapter may require that you modify one or more ISPF panel definitions in your ISPF panel library. If you are unfamiliar with the structure and usage of the ISPF panel definition language, please ask someone who is knowledgeable on this topic to assist you, or familiarize yourself with it before you proceed.

You should read this entire chapter before you build your own MVS/QuickRef data base. You should also review the information on "Data Base Structure" and "Access Methods" provided by the MVS/QuickRef on-line help facility. (See "Using The Help Facility" in Chapter Two if you are not familiar with the help facility.)

Overview

In addition to displaying its own reference information, MVS/QuickRef can also be used to display reference information that you have created, by allowing you to store it in an MVS/QuickRef user data base. Any reference information that you want to make available to all the ISPF users in your organization using the instant-display capabilities of MVS/QuickRef is ideal for inclusion in your user data base. Examples of this type of information are:

- Local job class or SYSOUT class standards
- Local JCL standards
- Tape handling procedures and instructions
- Employee phone number list
- “Hot” news items, such as down time schedules
- Hardware and software configuration information
- User abend code descriptions for local applications

The types of information presented above may not, at first glance, seem to fit into the structure used for MVS/QuickRef user data bases (where the information is divided into products - identified by a unique combination of vendor name, product name, and release number- and then further divided into discrete items of information which can be individually retrieved and displayed). However, it should be noted that you control the vendor name, product name, release number, and item name assigned to each piece of information and that you can use these "keys" in any way that will make sense to your users. You can also treat vendor name as "owner" and product name as "information topic". Release number can be a revision or publication number or, if it really does not apply, can be left blank.

Suppose, for example, that you want to put your company's "phone book" into a user data base and that you want to break the list of phone numbers out by office location. Then, if your
company's name is Acme, you could use 'ACME' as the vendor name and 'PHONELIST' as the product name. You could also use 'INTERNAL' as vendor name. Item names, corresponding to office locations, might be 'HOMEOFFICE', 'EASTCOAST', and 'WESTCOAST'. Release number would not seem to apply; however, you might want to use release number in this case to reflect the "effective date" of the information (for example, by specifying release number in the form 'AS OF 9/1/96'). Then, if and when your phone system is being "reconfigured", you could store and display the "current" phone list as will as the "future" phonelist and the date when the "future" list will become effective. In this case, then, the two phonelist "products" might show up on a product selection list as shown below:

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACME</td>
<td>PHONELIST</td>
<td>AS OF 9/1/96</td>
</tr>
<tr>
<td>ACME</td>
<td>PHONELIST</td>
<td>AS OF 1/1/98</td>
</tr>
</tbody>
</table>

When the user selects the first of these two products for display, he will see an item selection list showing 'V=ACME P=PHONELIST R=AS OF 9/1/96' at the top of the display along with the following item names: HOMEOFFICE, EASTCOAST, and WESTCOAST.

As this example illustrates, creative use of the vendor name, product name, release number, and item name should allow you to store most any type of information in a user data base and, at the same time, make the "keys" associated with that information meaningful to your users.

### Deciding how many user data bases you need

MVS/QuickRef supports definition and use of up to nine user data bases. Normally, one user data base is sufficient for storing all of the local user reference material your company plans to make available online for programmer and operator use. A single user data base will support as many products as you are likely to need and an unlimited number of items within each product.

However, there are situations that require that more than one user data base be created. If, for example, it is decided that two or more of the departments in your organization are to create and maintain their own individual user data bases, then you will have to implement multiple user data bases (one for each "owning" department).

### Building User Data Base(s)

The steps to follow to create your own user data base(s) are:

1. Allocate and load the user data base(s)
2. Define the name(s) of the user data base(s) to MVS/QuickRef
3. If you feel that it is needed, create or customize the appropriate user menu panel.
4. If you created a user menu panel, then add a user menu option to the MVS/QuickRef main menu

These steps are described in detail in the sections which follow.
Step 1 - Allocate and Load User Data Base(s)

You create and load a user data base by executing a batch job using the JCL found in Figure 25 below. This JCL can also be found in member QWUSER in the MVS/QuickRef JCL data set. A complete explanation of each element of the JCL follows the figure.

NOTE: A user data base cannot be “added-to” once it is built; it must be deleted and rebuilt from scratch whenever you want to update it or add new information to it. You can, however, use the override parameter facility described in chapter six of this guide to overcome this limitation. You can also define your user data base as a Generation Data Group (GDG) data set, then catalog the plus one (+1) generation when you rebuild it, and have MVS/QuickRef always read and use the zero generation (0).

Figure 25 - User Data Base Creation JCL

MVS/QuickRef supports parameters specified via the JCL PARM field. The parameters are TEXTLEFT=, TEXTRIGHT=, TITLES=, NOTRANS=, and OLDKEYS=. The syntax and usage instructions for these parameters are described below:

TEXTLEFT=n - specifies the column number in the input data records where your reference text begins. The first column in the input records is column 1. The default for this parameter is column 2. (For a variable-length input file, column 1 is the first column following the 4-byte RDW.)

TEXTRIGHT=n - specifies the column number in the input data records where your reference text ends. If this parm is not specified, your reference text will be considered to run up through the last column in each input record.
TITLES={Y | N} (N is the default) - this parameter indicates whether or not the key indicator records that you have defined in the input data for creating your user data base contain titles. If this value is N, no user titles are supported. If this value is Y, then user titles up to 48 characters long are extracted from all of the key indicator records found in the input stream. The titles will appear on screen with the items stored in your user data base when a selection list is displayed. See the discussion of the format of key indicator records for more information.

OLDKEYS={Y | N} (N is the default) - this parameter indicates whether or not the input data contains one or more key indicator records that are still in Release 4.3 format. If this value is N, then all key indicator records are assumed to be in Release 5.0 (or later) format. If this value is Y, then the input data may contain one or more key indicator records in Release 4.3 format. In this case, when the data base build program (QWIKREF2) encounters a key indicator record in Release 4.3 format, it will place the old Release 4.3 topic indicator in the product name field and will set the vendor name and release number fields to blanks. The purpose of this parm is to allow for already existing user data bases to be rebuilt and used in Release 5.0 without having to convert the key indicator records to Release 5.0 format. For more details on the format required for Release 5.0 key indicator records, see the section on "Key Indicator Records" below.

NOTRANS={Y|N} (N is the default) - this parameter indicates whether or not unprintable characters in the input file are to be translated to blanks. Use NOTRANS=Y if your input file contains printer control characters or other non-printable characters that you do not want translated to blanks.

User Data Base JCL DD Statements

STEPLIB DD: The STEPLIB DD statement is used to point to the program library into which the MVS/QuickRef programs were copied when the product was installed. This DD statement is not needed if the MVS/QuickRef programs were copied to a program library in the system link list.

SYSPRINT DD: A "File Create Report" and any warning or error messages issued during the data base build process are written to the SYSPRINT DD. The DCB characteristics for this file are RECFM=FBA, LRECL=80, BLKSIZE=6160. You should always check the File Create Report for warning or error messages and to make sure the data base information and file create statistics contained in the report are correct.

QWREFDD DD: The ‘QWREFDD’ DD statement allocates space for your user data base. The RLSE parameter is coded in the JCL SPACE= keyword for the QWREFDD DD statement so that unused DASD tracks will be freed at the end of the job. The block size used for this file must be 6160 and the RECFM used must be F.
The data base build program (QWIKREF2) stores the input reference text in a compressed format as it builds the output user data base. In addition to the input reference text, it will also store certain control blocks in the output user data base. In most cases, the amount of disk storage space saved by compressing the input reference text will provide more than enough disk space for the control blocks which are added. So, in most cases, the output user data base will require no more disk space than the input file(s) used to create it. This provides an easy method of estimating the amount of disk space required for your user data base. Simply determine (or estimate) the amount of disk space required for your input file(s), and then use that amount for your user data base's primary allocation. As a "fudge" factor, set the secondary allocation at 50% of the primary allocation. In any case, remember that unused space will be released when the user data base is actually created, so unless you are faced with an extreme shortage of disk space (i.e., you have problems finding space for the initial allocation of even a minimal size file), it is better to estimate high.

**DATAIN DD:** This DD statement specifies the input reference data that will be compressed and stored in your user data base. This input file can be fixed or variable-length and can take the form of a sequential data set (on tape or disk), a member or a set of concatenated members of a partitioned data set or a PDSE data set, an entire partitioned data set or PDSE data set, or a set of concatenated partitioned data sets or PDSE data sets. The RECFM may be F, V, FB, or VB. The LRECL can be any size up to 32,767. The BLKSIZE can be any value supported by the operating system.

Note: When setting up your input file, please remember that you cannot concatenate a sequential data set with a partitioned data set unless the dd statement for the partitioned data set references a single member of that partitioned data set.

The ‘DATAIN’ input data stream contains six types of records: key indicator records, alias indicator records, copyright indicator records, category code indicator records, start access based on content indicator records, and text records. Key indicator records identify the vendor name, product name, release number, and item name associated with a given set of text records. Alias indicator records specify optional item names for the item identified on the immediately preceding key indicator record(s). The text records for a particular reference item follow the key indicator record(s), or, if present, the optional alias indicator record(s), and contain the reference text to be displayed for the item identified in the preceding key indicator record(s). Copyright indicator records indicate when copyright statements are to be displayed and specify the wording to be used for each copyright statement. Category code indicator records define product category codes. Start access based on content indicator records dictate which products are to have getnext based on content processing applied to them. The different type of records which can appear in the input file are described in the sections which follow.
DATAIN2 DD: This DD statement provides for a second, optional input file. It has the same characteristics as the DATAIN input file and can be either fixed or variable-length. If the DATAIN2 DD statement is not provided, then the second input file is ignored. The main purpose for providing this second input file is so that, if needed, one input file can be fixed-length and the other can be variable-length. If all your input is either fixed or variable-length, then you may choose not to use the second input file.

SYSIN: This DD statement provides for optional member selection statements. It must be a fixed-length file with an LRECL of at least 72. It is only needed if you are using a PDS for the DATAIN or DATAIN2 input file and want to specify exactly which members are used as input.

SORTIN, SORTOUT, SORTWKnn, SYSOUT: These are required SORT work files. They should not require any modification unless the disk space allocations should turn out to be too small.

QWIKREF2 Return Codes

QWIKREF2, the data base build program, issues the following return codes:

- 0 - successful execution; no error or warning messages were issued
- 4 - warning messages were issued
- 8 - error messages were issued

For return codes 4 and 8, check the File Create Report for warning and error messages. Be sure to correct all errors; otherwise, you user data base may not perform as expected.

Key Indicator Records

Key indicator records identify the vendor name, product name, release number, and item name under which each set of reference text records is to be stored. There must be one or more key indicator records for each item for which reference text is to be displayed by MVS/QuickRef. The key indicator record(s) must precede the reference text that is to be stored under that key.

Each key indicator record is formatted as shown below:

\[ K V='\text{vendor name}' \ P='\text{product name}' \ R='\text{release number}' \ I='\text{itemname} T='\text{title}' \]

where

- a capital (upper case) letter 'K' must appear in column 1
♦ for a variable-length input file, column 1 is the first column following the 4-byte RDW
♦ V= specifies the associated vendor name (maximum length of 15 characters)
♦ P= specifies the associated product name (maximum length of 20 characters)
♦ R= specifies the associated release number (maximum length of 15 characters)
♦ I= specifies the associated item name (maximum length of 16 characters)
♦ T= specifies the optional item title (maximum length of 48 characters)
♦ the operands must be separated from the K in column 1 and from one another by one or more spaces
♦ the operands can be coded in any order and can appear anywhere in the input record after column 2
♦ all operands other than T= must be specified in capital (upper case) letters; the item title can be specified in upper and/or lower case letters
♦ any operand which contains blanks must be enclosed in single quote marks; the quote marks are optional if the operand contains no spaces
♦ a single quote mark contained in an operand enclosed in single quote marks must be represented as two consecutive single quote marks
♦ as many consecutive key indicator records as needed may be used to specify the key for a given item
♦ no operand can be continued into the next record
♦ the V=, P=, R=, and I= operands are required
♦ the item name cannot be blank and cannot contain blanks
♦ all item names in the same product must be unique; in other words, there can be no duplicate item names within a given vendor name, product name, and release number combination
♦ if the vendor name, product name, or release number is to be set equal to blanks, then the corresponding V=, P=, or R= operand should be coded as one or more blanks enclosed in single quote marks (for example, R=' ')
♦ the T= operand should be specified only if the TITLES=Y parm is specified
♦ even when the TITLES=Y parm is specified, the T= operand is optional; if omitted, the title assigned to the corresponding item is blank

Here are two examples with all the operands specified in a single key indicator record:
K V=ACME P=PHONELIST R=' ' I=HOMEOFFICE T='Homeoffice phone numbers'
...reference text line 1....
...reference text line 2....
Here are two examples with the operands spread over several key indicator records:

K V=ACME
K P=PHONELIST
K R=''
K I=HOMEOFFICE
K T='Homeoffice phone numbers'

...reference text line 1....
...reference text line 2....

As shown in the examples above, the reference text for a given item must appear in the input records immediately following the key indicator record(s) which define the key for that item. The only exception to this rule has to do with alias indicator records; one or more optional alias indicator records can appear between the key indicator record(s) and the associated reference text records.

**Alias Indicator Records**

You can supply up to 500 alias names for each item you store in your user data base. The aliases allow you to use more than one name to retrieve the reference text for an item. For example, you might want to store your local JOB statement coding rules under the two names “JOB” and “JOBCARD”, or important data center news under the names “HOT_NEWS”, “FLASHES”, and “READ_THIS!”.

To specify an alias for an item in your user data base, include an alias indicator record immediately after the key indicator record(s) for which the alias(es) are to be assigned. The alias indicator record contains a capital (upper-case) letter 'K' in column one; the second character on the alias indicator record must be an asterisk. (For a variable-length input file, column 1 is the
first column following the 4-byte RDW.) The 'K*' characters are then followed by as many alias item names as you would like to assign to the current item (or as many as will fit in a single input record).

The alias item names must:
- be separated from one another and from the 'K*' characters by one or more spaces
- contain no more than 16 characters
- be specified in capital (upper-case) letters
- contain no embedded spaces
- not be enclosed in quote marks
- be unique within a given product

If the number of alias names that you want to assign to a given item will not fit into a single alias indicator record, then you can specify multiple alias indicator records. The only requirement is that all alias indicator records associated with a given item must appear as consecutive records and must appear immediately after the key indicator record(s) for that item.

There is no limit on the number of alias indicator records specified for a given item. However, there is a limit on the total number of aliases which may be specified for a given item (i.e., 500).

As an example of an alias indicator record, suppose our data center has published revision number 2 of a manual containing data center standards and that one of the topics in this manual covers JCL standards. If we decide to store this information in a user data base, then we might want to use 'DATA CENTER' for vendor name, 'STANDARDS' for product name, '2' for release number, and 'JCL' for the item name of the item covering JCL standards. Now suppose we decide to provide the following aliases for the JCL item: JOBCARD, DDCARD, and EXECCARD. Here is an example of the required key indicator and alias indicator records:

```
K V='DATA CENTER' P='STANDARDS' R=2 I=JCL
K* JOBCARD EXECCARD DDCARD
...reference text line 1....
...reference text line 2....
```

Here is the same example showing multiple alias indicator records:

```
K V='DATA CENTER' P='STANDARDS' R=2 I=JCL
K* JOBCARD
K* EXECCARD
K* DDCARD
...reference text line 1....
...reference text line 2....
```
As shown in the examples above, the alias indicator record(s) must appear immediately after the associated key indicator records and immediately before the associated reference text input records. With this setup, information on JCL standards could be retrieved from the user data base by requesting the display of item JCL, JOBCARD, EXECCARD, and/or DDCARD. An item selection list for release '2' of product 'STANDARDS' associated with vendor 'DATA CENTER' would include all 4 item names: JCL, JOBCARD, EXECCARD, and DDCARD. Selection of any one of these items from such an item selection list would result in a display of the reference information for item JCL.

Please note that all columns of an alias indicator record are scanned by MVS/QuickRef. If you use PDF edit to create your alias indicator records, remember to blank out the sequence numbers in columns 73 through 80 so that they will not be treated as an alias name by MVS/QuickRef.

Text Records

Text records follow the associated key indicator record(s), or the optional alias indicator record(s), and contain the reference text for the item identified by the associated key indicator record(s). All text records that follow the key indicator record(s) are stored as reference data for that key until the next key indicator record is encountered, or until end-of-file is reached on the input file.

The data in the first text record for a given item will be displayed on the first line of the display for that item; the data in the second text record for that item will be displayed on the second line of the display for that item, etc.

The user data base build program (QWIKREF2) will pick up and store the data from each input text record based on the TEXTLEFT and TEXTRIGHT JCL parms. If TEXTLEFT and TEXTRIGHT are not specified, then each line of stored text will start with column 2 of the input text record and will continue to the end of the input record. (For a variable-length input file, column 1 is the first column following the 4-byte RDW.) If the amount of text stored for a given line exceeds 80 characters, then, when that item is displayed by MVS/QuickRef, the user will have to scroll to the right to see the rest of the reference text for that line.

As an example of how the TEXTLEFT and TEXTRIGHT parms might be used, suppose your input file contains fixed-length 80-byte records with a sequence number in columns 73 through 80. Then, to keep the sequence number from being stored (and displayed) as part of the reference text for each item, you could specify TEXTRIGHT=72. As another example, suppose your input file contains fixed-length 90-byte records, the reference text actually appears in columns 6 through 85, and that columns 1 through 5 and 86 through 90 contain blanks. In this case, you might want to specify TEXTLEFT=6 and TEXTRIGHT=85 just so each text line will be fully displayed on the MVS/QuickRef display panel and your users will not have to scroll to the right to see the right-hand end of each text line.
The user data base build program translates "obvious" non-printable or non-display characters in the input text records to blanks. However, in order to ensure that the data in your user data base will be displayed and printed properly, you should make sure that your input text records do not contain any characters which will not display or print correctly in your particular operating environment.

In order to minimize the use of virtual storage, the amount of virtual storage allocated for the purpose of storing a given user data base item (once it is retrieved from the data base) is limited to 500K bytes. However, this number can be increased, if necessary, with the USERSTR parameter of the QWIKOPTS table. See the section on "Setting MVS/QuickRef Global Installation Options" in Chapter Four for details.

When storing reference information in a user data base, MVS/QuickRef will truncate the reference information for a given item at 2,147,483,647 bytes if its total length exceeds 2,147,483,647 bytes after compression. A warning message is issued when this truncation occurs.

Assuming a 50% compression ratio and an average text line length of 78 characters, this allows for about 55 million text records per item.

Comment Records

Comment records can be placed anywhere in your user data base input data, and may be useful for adding pertinent information that is intended for your own use and that you do not wish to display to users of your user data base. You might use comment records to indicate the last date that your user data base was modified, or to add a note which might help you to add to your user data base correctly in the future. Comment records are formatted as shown below:

K** user data base comment text…

where a capital (upper case) letter 'K' must appear in column 1, followed by two consecutive '*' characters, followed by at least one blank space.

Comment records can contain, after the initial 'K** ' characters, any characters you wish, and may extend for the full record length of your input data set. You may use as many comment records in your user data base input as you wish, since there is no limit set by MVS/QuickRef.

Copyright Indicator Records

Copyright indicator records indicate when copyright statements are to be displayed and specify the wording to be used for each copyright statement.

Copyright indicator records are formatted as shown below:

KCPY V='vendor name' P='product name' R='release number' l=itemname C='copyright text'

where
capital (upper case) letters 'KCPY' must appear in columns 1 through 4

for a variable-length input file, column 1 is the first column following the 4-byte RDW

V= specifies the associated vendor name (maximum length of 15 characters)
P= specifies the associated product name (maximum length of 20 characters)
R= specifies the associated release number (maximum length of 15 characters)
I= specifies the associated item name (maximum length of 16 characters)
C= specifies the copyright statement text (maximum of 70 characters) to be used for the associated vendor name, product name, release number, and/or item name

the operands must be separated from the KCPY characters in columns 1 through 4 and from one another by one or more spaces
the operands can be coded in any order and can appear anywhere in the input record after column 5
all operands other than C= must be specified in capital (upper case) letters; the copyright statement can be specified in upper and/or lower case letters
any operand which contains blanks must be enclosed in single quote marks; the quote marks are optional if the operand contains no spaces
a single quote mark contained in an operand enclosed in single quote marks must be represented as two consecutive single quote marks
as many consecutive copyright indicator records as needed may be used for a single copyright statement definition
no operand can be continued into the next record
the V= (vendor name) operand is required and marks the start of a new copyright definition
the C= (copyright text) operand is also required and marks the end of a copyright definition
the P=, R=, and I= operands are optional and may be coded in any order so long as they follow the V= operand and precede the C= operand

If more than one record is required for a given copyright definition, then no other type of record can appear between the record containing the V= operand and the record containing the C= operand. In other words, once a copyright definition starts (with the record containing the V= operand) it must end (with the C=operand) before any regular key indicator records, any alias indicator records, or any text records are encountered.
The records making up a given copyright definition can appear anywhere in the input file. This means that all the copyright indicators can be gathered together at the beginning of the file or in a special PDS member. Alternatively, they can be spread out into the file and placed near the definition of the items that they control.

The \texttt{V=} P=, R= and \texttt{I=} (i.e., the vendor, product, release, and item) operands can be specified as a full key or as a generic key. If any one of the optional operands (i.e., product, release, or item) is unspecified, then that operand is treated as if coded as a single asterisk (so that it will match on all key values for that particular key component). For example:

\begin{verbatim}
KCPY V=IBM P=C* C=...
\end{verbatim}
is equivalent to

\begin{verbatim}
KCPY V=IBM P=C* R=* I=* C=...
\end{verbatim}

and would match for all items in all releases of all products whose names start with the character \texttt{C} for vendor IBM.

The \texttt{C=} copyright text operand provides all the copyright text to be shown for the associated vendor, product, release, and/or item. The text making up this copyright statement will be centered in the middle of the display line and will be surrounded on the left and right by asterisks. This copyright statement will then be displayed as the first line of text to be shown for each item which the associated copyright definition "covers".

The \texttt{C=} copyright text operand can also be specified as \texttt{C=NO} to indicate that no copyright is to be shown for the associated vendor, product, release, and/or item. If \texttt{C=NO} is specified for a given item, then, instead of a copyright statement, the first line of displayed text for that item will be a "TOP OF DATA" line.

If no copyright definition exists for a given item, then a copyright line will be automatically generated as shown below:

\begin{verbatim}
*****************  Text Below Copyright (c) YYYY, VVVVVVVVV *******************
\end{verbatim}

where

\begin{verbatim}
YYYY will be replaced with the current year (for example, 1997)
VVVVVVVVVV will be replaced with the vendor name in the associated data base key
\end{verbatim}

Note: This means that a copyright statement \textit{WILL ALWAYS BE} automatically displayed unless a copyright definition is provided with \texttt{C=NO} specified.

A given item is covered first by any copyright definition that specifies its associated vendor, product, release, and item name. Any item which is not covered by a copyright definition which
specifies its associated vendor, product, release, and item name is covered by any copyright definition that specifies its associated vendor, product, and release. If not covered by a copyright definition specifying its associated vendor, product, and release, then an item is covered by any copyright definition specifying its associated vendor and product. Finally, if not covered by a copyright definition specifying its associated vendor and product, then an item is covered by any copyright definition specifying its associated vendor.

Here are some examples of copyright indicators specified in a single input record:

KCPY V=SEA C='Text Below Copyright (c) 2005 Software Engineering America'
KCPY V=ISOGON C=NO

Here are some examples spread over several input records:

KCPY V='CA'
KCPY P=CA-ROSCOE
KCPY R=6.1
KCPY C='Text Below Copyright (c) 2005, Computer Associates'

KCPY V='BMC'
KCPY P=DBTOOLS
KCPY R=3.3
KCPY C='Text Below Copyright (c) 2000-2005, BMC Software'

KCPY V=IBM P=C*
KCPY C=NO

If these were all the copyright indicators included in the input file, then:

♦ all items for all releases of all products associated with vendor SEA would show the following copyright line:
  ****** Text Below Copyright (c) 1995, Software Engineering America ******

♦ no copyright would be shown for any item associated with vendor ISOGON; the "TOP OF DATA" line would be shown for these items

♦ items belonging to release 6.1 of Computer Associates’ CA-ROSCOE product would show the following copyright line:
  ****** Text Below Copyright (c) 2005, Computer Associates ******

♦ items belonging to release 3.3 of BMC's DBTOOLS product would show the following copyright line: ****** Text Below Copyright (c) 2000-2005, BMC Software ****

♦ assuming the current year is 2005, all other items associated with vendor Computer Associates would have the following generated copyright line shown:
  ****** Text Below Copyright (c) 2005, Computer Associates ******
no copyright would be shown for any item associated with product names starting with
the character C associated with vendor IBM; the "TOP OF DATA" line would be shown
for these items

assuming the current year is 2005, all other items associated with vendor IBM would have
the following generated copyright line shown:
****** Text Below Copyright (c) 2005, IBM ******

assuming the current year is 2005, all other items associated with all other vendors other
than those specifically described above would have the following generated copyright line
shown:
****** Text Below Copyright (c) 2005, VVVVVV ******
   where VVVVVV is the vendor name carried in the data base for each such item

It should be noted that, if the following copyright indicator appears in the input file:

   KCPY V=* C=NO

then NO copyrights would be automatically generated. This can used to "turn off" all copyrights
for a given user data base.

Category Code Indicator Records

As an aid in making the information in your user data base(s) more accessible to your user
community, you can use category code indicator records to relate a particular product name or
group of product names to a 'category'. Category names can be up to eight characters long, and
are displayed on the MVS/QuickRef category selection panel, along with a short descriptive title
you can also associate with that category.

Using categories, you can group the products in your data base according to the type of
information they contain. This facility allows novice or infrequent users of MVS/QuickRef to
quickly find what they need in your defined user data base(s), without knowing precisely what
item name or product name to use.

If you elect to use category code definitions, you will:

- allow for definition of eight-character product category codes within you user data base(s)
- allow for selected products to be associated with a given defined product category code
- allow for product category codes to be displayed and, when selected,
- allow for a product list to be displayed showing all products associated with the selected
  product category code
As an example, you will notice that one of the product category codes defined for the MVS/QuickRef main data base is category code 'PROGLANG', which is defined as the "Programming Languages Syntax Descriptions" category. When a user requests a display of available product category codes, category code 'PROGLANG' is included in the list. If the user selects this product category code from the list, a product list containing all "programming language syntax" products is produced.

Category codes are completely optional for user data bases, but they can provide your users with a more structured view of the information in your user data base(s).

Two types of key indicator records are used to define category codes for you user data base(s):

- KCAD is used to define an up to eight-character product category code
- KCAP is used to associate a given product with a defined product category code

The KCAD record is formatted as follows:

```
KCAD C=xxxxxxxx D='category description'
```

where
- KCAD starts in column 1
- xxxxxxxx is the (up to) eight character product category code being defined (any non-blank characters can be used)
- the category description can be up to 48 characters and must be enclosed in single quotes
- each of the components must be separated from the KCAD and from each other by one or more spaces

The KCAP record is formatted as follows:

```
KCAP V=vvv P=ppp R=rrr C=xxxxxxxx
```

where
- KCAP starts in column 1
- vvv is a non-generic vendor name
- ppp is a non-generic product name
- rrr is release number, and may be specified as generic, non-generic, or as a single asterisk
- xxxxxxxx is the (up to) eight-character category code that this product is to be associated with
- if rrr is specified as a generic release number, then all MATCHING releases of the specified product are associated with this product category code
- if rrr is specified as a single asterisk, then ALL releases of the specified product are associated with this product category code
- each of the components must be separated from the KCAP and from each other by one or more spaces
− the KCAP components can be spread across multiple records; however, the V= component must be the first component and the C= component must be the last component

Here are two examples of KCAD records (always define starting in column one):

```plaintext
KCAD C=PROGLANG D='Programming Language Syntax'
KCAD C=UTILS D='Utility Program Descriptions'
```

Here are some examples of KCAP records (always define starting in column one):

```plaintext
KCAP V=IBM P='ASSEMBLER SYNTAX' R=* C=PROGLANG
KCAP V=IBM P='C LANGUAGE SYNTAX' R=* C=PROGLANG
KCAP V=IBM P='IEBCOPY UTILITY' R=* C=UTILS
KCAP V=IBM P='IEBGENER UTILITY' R=* C=UTILS
```

Note that, with the KCAP records above, if there are two releases of product ASSEMBLER SYNTAX, then both releases of this product will be associated with product category code PROGLANG.

Syntax rules for the category code indicator records are as follows:
− each product category code can be up to eight characters in length
− a given product category code is defined by one and only one KCAD record
− each product category code specified on a KCAP record must be defined by a KCAD record
− each product category code defined by a KCAD record is specified on at least one KCAP record (i.e., that each category is associated with at least one product)
− each product specified on a KCAP record must actually exist in your user data base
− both KCAP and KCAD records can appear anywhere in the input stream, but should be kept together in one member for ease of maintenance

KCAD and KCAP records can appear anywhere in the data base input source file. For purposes of control and ease of maintenance, it is best to keep all of your KCAD and KCAP records together in the input file. For a sequential input file, consider putting all your KCAD and KCAP records near the beginning of the file; for a partitioned input file, consider putting these records in a separate PDS member.
Start Access Based On Content Indicator Records

Start access based on content indicator records dictate which products are to have getnext based on content processing applied to them. (You can refresh your understanding of getnext based on content processing by reviewing item GETNEXT-CONTENT in the on-line help facility.)

The start access based on content indicator record is formatted as shown below:

KSAC

where

♦ capital (upper case) letters 'KSAC' must appear in columns 1 through 4
♦ for a variable-length input file, column 1 is the first column following the 4-byte RDW

As shown above, the KSAC indicator record is specified without any operands.

KSAC indicator records are completely optional. They only have to be used when you want to apply getnext based on content processing to one or more products in your user data base.

Each KSAC indicator record specifies that the product defined by the immediately following key indicator record(s) is to have getnext based on content processing applied to it. For this reason, each KSAC indicator record must be immediately followed by one or more key indicator records.

For example:

KSAC
K V=ACME P='PROGRAMMING LANGUAGE' R=1.0 I=INTRODUCTION

would specify that getnext based on content processing is to be applied to release 1.0 of the product named 'PROGRAMMING LANGUAGE' from the vendor named 'ACME'.

When a KSAC indicator record is encountered, the data base build program adds "next item name" and "previous item name" to the information stored in the data base for each item in the specified product. "Next item name" is the name of the item which follows the current item in the input file. "Previous item name" is the name of the previous item in the input file.

For example, if a portion of the input file appears as shown below:

KSAC
K V=ACME P='PROGRAMMING LANGUAGE' R=1.0 I=INTRODUCTION
  The ACME programming language is designed to .................

would indicate that the ACME programming language is designed to function in the following way:

...
The ADD verb allows you to

The ADD verb uses the following operands

The ADD verb has the following special considerations

then getnext based on content processing would be applied to release 1.0 of the product named 'PROGRAMMING LANGUAGE' from vendor ACME.

If the user was viewing the reference information for item ADD within this product, then the GETNEXT command would bring up a display of the reference information for item ADD-OPERANDS within this same product. Another GETNEXT command would then bring up a display of the reference information for item ADD-CONSIDERATNS within this same product.

If the user was viewing the reference information for item ADD-CONSIDERATNS within this product, then the GETPREV command would bring up a display of the reference information for item ADD-OPERANDS within this same product. Another GETPREV command would then bring up a display of the reference information for item ADD within this same product.

As you can see from this example, getnext based on content processing should only be applied to products containing items that the user may want to review in the order in which they appear in the input file. Or, to put this another way, to products where you can arrange the item definitions in the input file into "content" sequence. For example, a product containing information that represents a "reference work" may be a good candidate for getnext based on content processing.

Since there is some storage overhead associated with getnext based on content processing, it is probably not a good idea to apply getnext based on content processing to every product in your user data base. Products containing items which have no "logical relationship", like a set of more or less independent abend codes, for example, are probably not a good candidate for getnext based on content processing.

A KSAC indicator record applies only to the product defined by the key indicator record(s) which immediately follow it. The application of getnext based on content processing will automatically terminate whenever another product is encountered. It will also terminate (and then restart for the next product) whenever another KSAC indicator record is encountered.

It should be noted that a product is defined by MVS/QuickRef as a unique combination of vendor name, product name, and release number. So if getnext based on content processing is to be provided for two releases of the same product, a KSAC indicator record will have to be
placed in front of the first item for the first release of the product and also in front of the first item for the second release of that product. Otherwise, the first item with a different release number will constitute the start of a new product and getnext based on content processing will terminate for the first release and will not start for the second release without another KSAC indicator record.

For example, if a portion of the input file appears as shown below:

```
KSAC
K V=ACME P='PROGRAMMING LANGUAGE'  R=1.0 I=INTRODUCTION
   The ACME programming language is designed to ..................
    .
K V=ACME P='PROGRAMMING LANGUAGE'  R=1.0 I=ADD
   The ADD verb allows you to ..................
    .
    .
K V=ACME P='PROGRAMMING LANGUAGE'  R=2.0 I=INTRODUCTION
   The ACME programming language is designed to ..................
    .
K V=ACME P='PROGRAMMING LANGUAGE'  R=2.0 I=ADD
   The ADD verb allows you to ..................
    .
```

then getnext based on content processing **would not** apply to release 2.0 of the ACME PROGRAMMING LANGUAGE product. If you wanted getnext based on content processing to also apply to this release, then you would have to insert another KSAC indicator record, as shown below:

```
KSAC
K V=ACME P='PROGRAMMING LANGUAGE'  R=2.0 I=INTRODUCTION
   The ACME programming language is designed to ..................
    .
K V=ACME P='PROGRAMMING LANGUAGE'  R=2.0 I=ADD
   The ADD verb allows you to ..................
    .
```

**Automatic Lookup Based On Pointers**

Automatic lookup is a predisplay analysis feature which causes additional relevant items to be automatically read in and displayed following the reference text for the current item. There are two types of automatic lookup:
Automatic lookup based on pointers, which are placed in the reference text by the person preparing the data for inclusion in the data base.

Automatic lookup based on text content, in which MVS/QuickRef determines which additional items to bring in by analyzing the content of the reference text about to be displayed.

For more information on automatic lookup, see item AUTOMATIC-LOOKUP in the MVS/QuickRef on-line help facility.

For those types of items to which it applies, automatic lookup based on text context is automatically provided. However, if you want automatic lookup based on pointers applied to particular items in your user data base, then you must place the required pointer in the appropriate position in the reference text associated with each such item.

Each pointer, which is referred to as a %> pointer, is formatted as follows:

%>iii

where iii is the item name of the additional item to be automatically read in.

Each pointer must appear in the last line of the reference text to which it applies. Each pointer must appear no further to the left than the first column which contains your normal reference text. From one to ten pointers can be specified for a single item; they must all appear in the last text line for that item and be separated by one or more spaces.

As an example of automatic lookup based on pointers, suppose the text which appears between the lines of equal signs below is part of the input used to create a user data base:

```
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
K V='ACME CORP’ P='ACME MESSAGES’ R='V1R1’ I='ABC101’
   ABC101 A SEVERE ERROR HAS OCCURRED

   This message indicates a severe error has occurred. See item SEVERE-ERROR to
determine how to respond.
K V=............
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

Now suppose you want the item named SEVERE-ERROR to be automatically added to the reference text for message ABC101. Then you could add a %> pointer to the text for message ABC101 as shown below:

```
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```
K V=’ACME CORP’ P=’ACME MESSAGES’ R=’V1R1’ I=’ABC101’
ABC101 A SEVERE ERROR HAS OCCURRED

This message indicates a severe error has occurred. See item SEVERE-ERROR to
determine how to respond.

%>SEVERE-ERROR
K V=..............

As another example, suppose you want the item named SEVERE-ERROR and the item named
PHONE-LIST to be automatically added to the reference text for message ABC101. Then you
could add two %> pointers, as shown below:

K V=’ACME CORP’ P=’ACME MESSAGES’ R=’V1R1’ I=’ABC101’
ABC101 A SEVERE ERROR HAS OCCURRED

This message indicates a severe error has occurred. See item SEVERE-ERROR to
determine how to respond.

%>SEVERE-ERROR %>PHONE-LIST
K V=..............

Input File Processing Considerations

When the DATAIN or DATAIN2 input file is allocated to a PDS or set of concatenated PDS's,
MVS/QuickRef will read and process every member in the allocated PDS(s) with a unique
member name. If two or more PDS's which contain the same member name are concatenated to
either DATAIN or DATAIN2, then the member with the duplicate member name will be pulled
only from the first concatenated PDS which contains this member name.

If you want to limit the PDS members which are selected for processing, you can use the SYSIN
DD statement to pass member selection control statements to MVS/QuickRef. The syntax for the
SYSIN member selection control statements is described below in the section on “User Data
Base Member Control Statement Syntax”.

User Data Base Member Selection Control Statement Syntax

Member selection control statements are passed to MVS/QuickRef via the SYSIN DD statement.
The member selection control statements identify the names of the members to be selected.
Member names on member selection control statements may appear anywhere in columns 1
through 72. The member names must be separated by blanks or commas. A statement with an
asterisk in column 1 is treated as a comment statement. Comments can also appear to the right of
the member names on a member selection statement if the comments are preceded by the character string ‘/*’.

If you are using the DATAIN2 input file, then the member selection statements to be applied to the DATAIN2 file must be preceded by a record with the characters '!DATAIN2 ' in columns 1 through 9. The member selection statements to be applied to the DATAIN file may also be preceded by a special record, in this case with the characters '!DATAIN ' in columns 1 through 9. However, if the first non-comment record in the SYSIN file is not a '!DATAIN ' or '!DATAIN2 ' statement, then that and all following member selection statements (up to the appearance of the !DATAIN2 record, if there is one) are considered to be applied to the DATAIN file. Any data following the !DATAIN or !DATAIN2 identifier in the !DATAIN or !DATAIN2 statement is ignored.

You must be careful with member selection statements being used to select members from a set of concatenated PDS's. If you are using a set of concatenated PDS's for either the DATAIN or DATAIN2 input file and you use member selection statements to select a member name that appears in two or more of the PDS's concatenated to the same input file, then that member will be extracted and processed only from the PDS containing the first occurrence of that member name. The appearance of that member in the subsequent PDS's concatenated to that input file will be ignored.

Figure 26 which follows is an example of some valid member selection statements for MVS/QuickRef.

```
//SYSIN DD *
* comment statement: following selection statements apply to DATAIN
JCLGUIDE,TAPEINFO
JOBCLASS,OUTCLASS, FONENUMB /* comment on selection statement
MEMBER1 /* 1 member name per statement,
MEMBER2 WORKSCHD /* or more than 1 name per statement
A123 B992, F252, G788 /* blanks or commas separate names
!DATAIN2
MEMBERA
MEMBERB
!DATAIN2
MEMBERA
MEMBERB
```

Figure 26 - Sample Member Selection Control Statements

In the example above, MEMBERA and MEMBERB would be extracted from the DATAIN2 file; all of the other specified members would be extracted from the DATAIN file.

**User Data Base JCL Examples**

Figure 27 is an example of a job to create a user data base by selecting 3 members from a single PDS allocated to the DATAIN input file. Key indicator records (and alias indicator and copyright indicator records, if any) are imbedded within the members themselves. The text in the
members is in column 2 through the end of each input record, so the `TEXTLEFT=` and
`TEXTRIGHT=` JCL parms are not needed.

Figure 27 - Sample User Data Base Creation Job Number 1

The JCL in Figure 28 creates a user data base using all of the members in the single PDS
allocated to the DATAIN input file. Key, alias, and copyright indicator records are imbedded
within the members themselves. The actual text is found in columns 1 through 72 of the PDS
members, so the `TEXTLEFT=1` and `TEXTRIGHT=72` JCL parms are specified. All members
from the DATAIN PDS are selected since no member selection statements are specified by a
SYSIN DD statement.

```jcl
//JS10 EXEC PGM=QWIKREF2,PARM='TEXTLEFT=1,TEXTRIGHT=72'
//SYSPRINT DD SYSOUT=* 
//DATAIN DD DISP=SHR,DSN=SYS3.MIS.STANDARDS
//QWREFDD DD DSN=SYS2.QWIKUSER,DISP=(,CATLG,DELETE),
//        UNIT=SYSDA,
//        DCB=(RECFM=F,BLKSIZE=6160,LRECL=6160),
//        SPACE=(TRK,(75),RLSE)
//SYSIN DD *
//SORTIN DD UNIT=3380,SPACE=...
//SORTOUT DD UNIT=3380,SPACE=...
//SORTWK01 DD UNIT=3380,SPACE=...
//SORTWK02 DD UNIT=3380,SPACE=...
//SORTWK03 DD UNIT=3380,SPACE=...
//SYSOUT DD SYSOUT=* 
```

Figure 28 - Sample User Data Base Creation Job Number 2

The example JCL in Figure 29 will create a user data base from several concatenated PDS
members. Columns 2 through 75 of the PDS members contain the actual reference text, so the `TEXTRIGHT=75` JCL parm is used and the `TEXTLEFT=` parm is left to its default value
(column 2). Since the DATAIN file specifies sequential input, no member selection control

```jcl
//JS10 EXEC PGM=QWIKREF2,REGION=0K
//SYSPRINT DD SYSOUT=* 
//DATAIN DD DISP=SHR,DSN=PROD.JCL.STANDARD.DOC
//QWREFDD DD DSN=SYS2.QWIKUSER,DISP=(,CATLG,DELETE),
//        UNIT=SYSDA,
//        DCB=(RECFM=F,BLKSIZE=6160,LRECL=6160),
//        SPACE=(TRK,(10),RLSE)
//SYSIN DD *
//JOBCLASS /* Local job classes and standards
//OUTCLASS /* SYSOUT classes and forms
//UNITNAME /* valid local JCL UNIT= unit names
//*
//SORTIN DD UNIT=3380,SPACE=...
//SORTOUT DD UNIT=3380,SPACE=...
//SORTWK01 DD UNIT=3380,SPACE=...
//SORTWK02 DD UNIT=3380,SPACE=...
//SORTWK03 DD UNIT=3380,SPACE=...
//SYSOUT DD SYSOUT=* 
```
statements or SYSIN DD statement are needed. Key, alias, copyright, and category indicator records are NOT contained in the members themselves, so the indicator records for each member are concatenated above it as in-stream data. Category code indicator records appear last in this example but may appear anywhere in the input stream.

The technique used in Figure 29 allows you to process existing reference text without modifying it by adding key indicator records (and if needed, alias indicator and copyright indicator records) as in-stream data. The TITLES=Y parameter is specified so that the title that appears in each key indicator record will be extracted and kept with the item in the data base.

```
//JS10  EXEC PGM=QWIKREF2,REGION=0K,
//       PARM='TITLES=Y,TEXTRIGHT=75'
//SYSPRINT DD SYSOUT=*
//DATAIN  DD *
KCPY V=*,C=NO
K V='DATA CENTER' P='MISCELLANEOUS INFO' R=' ' I=JCLINFO
K T='JCL Coding Standards'
   DD DSN=SYS2.HELP(MEM1),DISP=SHR
   DD *
K V='DATA CENTER' P='MISCELLANEOUS INFO' R=' ' I=DASDINFO
K T='Rules for DASD Space Allocation'
   DD DSN=SYS2.HELP(MEM2),DISP=SHR
   DD *
K V='DATA CENTER' P='MISCELLANEOUS INFO' R=' ' I=TAPINFO
K T='Job Classes for Tape Jobs'
   DD DSN=SYS2.USRGUIDE.TEXT(TAPES),DISP=SHR
   DD *
K V='DATA CENTER' P='MESSAGES & ABENDS' R=1.2 I=PRODMSGS
K T='List of Messages Issued by Production Jobs'
   DD DSN=PROD001.CNTL(PRODMSGS),DISP=SHR
   DD *
K V='DATA CENTER' P='MESSAGES & ABENDS' R=1.2 I=USRABNDS
K T='List of User Abends Issued by Production Jobs'
   DD DSN=PROD001.CNTL(USRABEND),DISP=SHR
   DD *
KCAD C=MISCINFO D='Miscellaneous Job Information'
KCAP C=MISCINFO V='DATA CENTER' P='MISCELLANEOUS INFO' R='*
KCAD C=MSG_ABND D='Messages and Abends for Production Jobs'
KCAP C=MSG_ABND V='DATA CENTER' P='MESSAGES & ABENDS' R='*
/**//QWREFDD DD DSN=PROD.QWIKREF.DATA,DISP=(,CATLG,DELETE),
   UNIT=3380,
   DBC=(RECFM=P,BLKSIZE=6160,LRECL=6160),
   SPACE=(TRK,(30),RLSE)
//SORTIN  DD UNIT=3380,SPACE=...
//SORTOUT  DD UNIT=3380,SPACE=...
//SORTWK01  DD UNIT=3380,SPACE=...
//SORTWK02  DD UNIT=3380,SPACE=...
//SORTWK03  DD UNIT=3380,SPACE=...
//SYSOUT  DD SYSOUT=*  
```

**Figure 29 - Sample User Data Base Creation Job Number 3**

It should be noted that, in the example above, we are assuming that the reference text for item JCLINFO is stored in member MEM1 of the PDS named SYS2.HELP, reference text for DASDINFO is stored in member MEM2 of SYS2.HELP, reference text for TAPEINFO is stored in member TAPES in SYS2.USRGUIDE.TEXT, etc. Notice the copyright indicator record
immediately following the DATAIN DD statement, which effectively eliminates automatic generation of any copyright statements.

Step 2 - Define Name(s) of User Data Base(s)

If you are planning to use a single user data base, then the name of the user data base can be defined to MVS/QuickRef in several ways.

First of all, you can preallocate it in your TSO logon procs with a DD name of QWREFDDU, as shown in the example below:

```
//QWREFDDU  DD   DSN=SYS2.QWIKUSER,DISP=SHR
```

If you choose to preallocate the user data base, MVS/QuickRef will always use the preallocated version and will assume that only this one, single user data base exists. If you choose to preallocate the user data base, then you can disregard the rest of this step and skip to Step 3, "Customize User Menu Panel".

If you decide not to preallocate your user data base, another option is to specify the name of the user data base in the LOCLDB1= parameter of the QWIKOPTS table. See the section on "Setting MVS/QuickRef Global Installation Options" in Chapter 4 for more details. If you use the LOCLDB1= parameter in the QWIKOPTS table, the user data base data set name must be cataloged (since the specified data set name is used to dynamically allocate the data set). Generation Data Group (GDG) data sets are supported, so you can make your user data set an element of a GDG and specify its name, including the relative generation number, if you prefer.

If you are running with a single user data base and are using the QW LIBDEF CLIST, there is one more option. You can uncomment the ALLOC statement for the QWREFDDU file in the QW CLIST and fill in the data set name.

If you are running with multiple user data bases, then you must use the LOCLDBn= parameters of the QWIKOPTS table to define the names of your user data bases. The LOCLDBn= parameters in the QWIKOPTS table are described on page 105 in the section titled "Setting MVS/QuickRef Global Installation Options" in Chapter Four of this guide.

If you are running with a single user data base, then, no matter how the name of that user data base is defined to MVS/QuickRef, that user data base is considered to be user data base number 1. If you are running with multiple user data bases, then the user data base defined by the LOCLDB1 QWIKOPTS parameter is considered to be user data base number 1; the user data base defined by the LOCLDB2 QWIKOPTS parameter is considered to be user data base number 2; etc. The user data base number assigned in this fashion is very important because this is the number that is used to indicate the user data base to be accessed when using fast-path invocation and the display request panels. It is also the user data base number used with the override parameter facility UDBN= parm.
Determine If User Menu Panel Is Needed

It should be noted, before you go on to Step 3: Customize User Menu Panel, that you **may not need** to provide a user menu panel. There are several reasons for this. In the first place, once you define your user data base(s) to MVS/QuickRef, a cross-data base search will automatically be done wherever a specific data base is not requested. Prior to Release 5.0, you had to specifically request a cross-data base search (for example, by specifying QW ?=itemname). For this reason, prior to Release 5.0, your users needed to know that one or more user data bases had been defined so they could know to specifically request a cross-data base search. With Release 5.0 and higher, unless a specific data base is requested, a cross-data base search will be done automatically whenever one or more user data bases are defined (whether or not your users know that user data bases exist). This is true for fast-path invocation (like "QW itemname") and for cursor-driven invocation.

Another reason that you may not need to provide a user menu panel is that, once one or more user data bases are defined to MVS/QuickRef, the Request Reference Information and List Vendors/Products/Releases panels will automatically be adjusted to show the following additional entry field:

Data Base to Search ==> (M for MVS/QuickRef Main data base, 1-9 for User data base, * or blank for all data bases)

The '1-9 for User data base' portion will show exactly how many user data bases are defined. For example, if you have 3 user data bases defined, then this new entry field will be shown as:

Data Base to Search ==> (M for MVS/QuickRef Main data base, 1-3 for User data base, * or blank for all data bases)

So, once one or more user data bases are defined, the Request Reference Information and List Vendors/Products/Releases panels will be automatically adjusted to show your users exactly how many user data bases have been defined.

As another consideration, remember that, from the List Vendors/Products/Releases panel or via fast-path invocation, your users can get a list of all products in a specific user data base as well as a list of all vendors in a specific user data base. For example, QW D=1 L=P will provide a product selection list showing all products (in product name sequence) in user data base number 1. QW D=1 L=V will provide a product selection list showing all products in user data base number 1 in vendor name sequence. QW D=1 L=O will provide a vendor selection list showing vendors only from user data base number 1. They can select a given vendor from the vendor selection list in order to get a list of products belonging to that vendor. They can select a given product from the product selection list to get a list of the items carried in that product. So, if the vendor name, product name, and release number are meaningful, this should provide sufficient tools for your users to "navigate" to the information they need - without the need for a user menu.
As a last consideration, note that, when a new vendor or product is added to your user data base, it will automatically be added to the vendor and product selection lists described above. So, no user menu maintenance will be required when you add or remove vendors or products.

Now, if you still feel that you need to provide a user menu panel, please go on to Step 3: Customize User Menu Panel. If you do not need to provide a user menu panel, then you can skip both Step 3 and Step 4: Add Option U to MVS/QuickRef Menu.

**Step 3 - Customize User Menu Panel**

MVS/QuickRef provides two sample user menu panels (QWIKREFU and QWIKREFM). These panels are intended to function as the primary menu panel for extraction of information from the user data base(s). Both of these panels are stored in the MVS/QuickRef panel library and both require custom modifications, to match the requirements of your particular user data base(s), before they are used. QWIKREFU, as delivered, is designed to work with a single user data base. QWIKREFM is designed to provide an example of the type of menu processing required when multiple user data bases are being processed.

Let's start with panel QWIKREFU. Look at the panel definition for QWIKREFU in Figure 30 on page 168, which shows the panel as it is distributed with MVS/QuickRef. The )BODY of the panel shows a numbered list of products in the form:

1 - User Product 1     User Product 1 item descriptions
2 - User Product 2     User Product 2 item descriptions

You will need to modify this section of the panel so it contains a numbered list of the products contained in your own user data base.

The )BODY section of the panel also contains an Option field where the user can enter the selection number corresponding to the product which he would like to see displayed. There is also an optional Item field where the user can enter a full or generic item name.

The )INIT section of the panel contains a set of variable definitions in the form:

```
&V1=''
&P1=''
&R1=''
&V2=''
&P2=''
&R2=''
```

You will need to modify these variable definitions so that they match the vendor names, product names, and release numbers corresponding to the products listed in the body of the panel.
The )PROC section of the QWIKREFU panel contains a VER statement that ensures that the product selection number entered by the user matches one of the products listed in the body of the panel. You will need to modify this VER statement so that it only allows product selection numbers which actually match one of the numbered products listed on the panel.

The )PROC section also contains an IF statement that checks to see if the user entered a valid product selection number. You will need to modify this IF statement so that it matches the number of products listed in the body of the panel.

Finally, there is a TRANS statement at the bottom of the proc section that translates the ZSEL variable into the proper call to the user menu help panel (QWIKREFN) or to the MVS/QuickRef display program (QWIKREF1). Note that, when QWIKREF1 is called, it is passed a fast-path invocation string in the form:

```
D=1 V='&Vn'' P='&Pn'' R='&Rn'' I=&KEY
```

The D=1 specifies that user data base number one (the only user data base we have) is to be accessed. The V=, P=, and R= fast-path string components will resolve to the vendor name, product name, and release number of the product selected by the user based on the product selection number he entered in the Option field. I= will be set equal to the item name the user entered in the Item field or, if the Item field was left blank, a single asterisk. So, if the user left the Item field blank, he will get an item selection list showing all items in the selected product. If the user entered a generic item name, then he will get an item selection list showing all matching items in the selected product. Finally, if he entered a full item name, then he will get a display of the reference information for that specific item within the selected product.

If you want to, you can modify panel QWIKREFU so that for a given option, it invokes another panel instead of program QWIKREF1. The secondary panel can be used to display additional information or additional products, allowing for virtually unlimited customization at the panel level. The secondary panel can invoke yet another panel, but to actually display a reference item or item list, you must eventually invoke program QWIKREF1 with the parameter list format used on panel QWIKREFU. Program QWIKREF1 will not execute properly unless a valid parameter is passed to it.

Given these facts, you can now modify the QWIKREFU panel to describe the information that your user data base contains. Figure 31 on page 170 shows the QWIKREFU panel after sample modifications have been made for a hypothetical installation called “HyperDyne Systems Corporation”. Figure 29 on page 158 shows the JCL that created the user data base that is used with the modified panel in Figure 31 on page 170.

First of all, look at the top of the )BODY section of the modified panel in Figure 31 on page 170. You will notice that the panel title has been changed to:

```
* HyperDyne Systems User Menu *
```

Now look at the body of the modified panel where it previously showed:
Note that this section has been changed to:

| %1+-- JCL Information/Requirements | Enter JCLINFO, DASDINFO, TAPEINFO, or * for list |
| %2+-- Production Messages/Abends    | Enter PRODMSGS, USRABNDS, or * for list |

so that it matches the two products that we placed in our user data base with the JCL shown in Figure 29 on page 158.

Now look at the top of the )INIT section of the modified panel. The variable definitions previously shown as:

```plaintext
&V1=' '
&P1=' '
&R1=' '
&V2=' '
&P2=' '
&R2=' '
```

have been changed as shown below:

```plaintext
&V1='DATA CENTER'
&P1='MISCELLANEOUS INFO'
&R1=' '
&V2='DATA CENTER'
&P2='MESSAGES & ABENDS'
&R2='1.2'
```

so that they match the vendor names, product names, and release numbers of the two products we placed into our user data base with the JCL shown in Figure 29 on page 158.

Now look at the )PROC section of the modified panel. Note that the VER, IF, and TRANS statements have been changed as shown below:

```plaintext
VER(&ZSEL,LIST,HELP,1,2,*)
IF (&ZSEL = 1,2,*)
&ZSEL=TRANS( TRUNC(&ZSEL,')')
HELP,'PANEL(QWIKREFN)'  
  1,'PGM(QWIKREF1) PARM(D=1 V=''&V1'' P=''&P1'' R=''&R1'' I=&KEY)'  
  2,'PGM(QWIKREF1) PARM(D=1 V=''&V2'' P=''&P2'' R=''&R2'' I=&KEY)'  
  *,'PGM(QWIKREF1) PARM(D=1 &KEY)'
```

so that they match the number of products actually shown on this modified panel.
Notice that, if the user enters '1' in the Option field and 'DASDINFO' in the Item field, then program QWIKREF1 will be invoked and will be passed the following fast-path invocation string:

\[
D=1 \ V='DATA CENTER' \ P='MISCELLANEOUS INFO' \ R=' ' \ I=DASDINFO
\]

This will cause MVS/QuickRef to display the reference information for item 'DASDINFO' associated with product 'MISCELLANEOUS INFO' belonging to vendor 'DATA CENTER' in user data base number 1.

Option * is included in the VER, IF, and TRANS statements so that the user can request a cross-product search. For example, if the user enters 'JCL*' in the Item field and '*' in the Option field, then program QWIKREF1 will be invoked and will be passed the following fast-path invocation string:

\[
D=1 \ JCL*
\]

This will cause MVS/QuickRef to display an item selection list showing all item names in user data base number 1 which start with the characters 'JCL' - no matter what product they are in.

Now let's take a look at panel QWIKREFM as shown in Figure 36 on page 182. This panel handles the same type of processing described above for QWIKREFU in terms of entering, verifying, and processing product selection numbers and item names. However, panel QWIKREFM is also setup to show a numbered list and a description of each of the multiple user data bases that have been defined and to allow the user to specify the user data base to be processed. So, in addition to the types of changes described above for panel QWIKREFU, you will need to make additional changes to QWIKREFM.

First of all, notice that panel QWIKREFM has an extra entry field, labeled as 'Database'. In this field, the user will enter a data base selection number corresponding to the user data base which he would like to access.

To make room in the body of the QWIKREFM panel for the list of defined user data bases, the numbered list of products has been "compressed" as shown below:

%1+- User Product 1  %6+- User Product 6
%2+- User Product 2  %7+- User Product 7
%3+- User Product 3  %8+- User Product 8
%4+- User Product 4  %9+- User Product 9
%5+- User Product 5

You will need to change this section of QWIKREFM to provide a numbered list of the products actually stored in the various user data bases which have been defined.

You will also need to change the numbered list of user data bases, as shown below:
so that it provides a numbered list and a description of the user data bases which have actually been defined.

You will also need to change the following VER statement:

```
VER(&DOPT, LIST, 1, 2, 3, 4, 5, 6, 7, 8, 9, *)
```

so that it matches the number of user data bases actually listed on the panel.

Finally, notice that the TRANS statement for QWIKREFM has the D= component of the fast-string passed to QWIKREF1 formatted as: D=&DOPT. &DOPT is set based on the data base selection number entered into the Database field. So, if the user enters data base selection number 3 in the Database field, the fast-path string passed to QWIKREF1 will specify D=3 and user data base number 3 will be accessed.

QWIKREFM is intended to provide just one example of how, when multiple user data bases are being utilized, the particular user data base to be processed can be determined. The actual technique you will want to use depends on the content and purpose of your user data bases and upon the existence of some available criteria that can be used to make this determination. For example, you may be able to make this decision "behind the scenes" (i.e., by manipulating the &DOPT variable in the PROC section of the user menu panel) based upon the product the user chooses, the execution environment (i.e., test or production), the user's TSO userid, or any other available criteria that makes sense based on the intended usage and content of your user data bases.

Letting the user make the determination, as with QWIKREFM, is, depending upon the circumstances, not likely to be the optimum solution to this problem. But QWIKREFM does show how this can be done if no "behind the scenes criteria" exists, and, no matter what technique is finally utilized, it provides an example of how the &DOPT variable must be manipulated when multiple user data bases are being utilized. No matter what type of panel processing is involved, only one user data base can be open at a time, and the data set specified in the fast-path string (as indicated by the &DOPT variable) is the one that will be opened and processed by program QWIKREF1 in order to satisfy a user data base display request.

MVS/QuickRef normally assumes that the name of the user data selection panel is QWIKREFU. So, if multiple user data bases are to be processed utilizing panel QWIKREFM, then, once it has been customized, steps must be taken so that it will be used in place of panel QWIKREFU. One obvious way to accomplish this is to delete QWIKREFU from your MVS/QuickRef panel library (or rename it or move it to another library) and then rename QWIKREFM to QWIKREFU. Another option is to use the QWIKOPTS options table to change the default name of the user data base menu panel. As explained in the section on "Setting MVS/QuickRef Global
Installation Options" in Chapter 4, this means that you would use the QWIKOPTS USERMNU= parm to identify the name of your user data base menu panel. This option in QWIKOPTS would be coded as shown below:

```
USERNMU=QWIKREFM,
```

Once this change is made and QWIKOPTS is reassembled and linked (as outlined in the section on "Setting MVS/QuickRef Global Installation Options"), the customized version of QWIKREFM will be established as the user data base menu panel.

**Step 4 - Add Option U to MVS/QuickRef Menu**

This step is necessary only if you decided to provide a customized user menu panel. So if you decided not to provide a customized user menu panel, this step can be skipped.

If you did provide a customized user menu panel, then you must make the customized menu panel accessible by “hooking” it to the MVS/QuickRef main menu, panel QWIKREFA.

The QWIKREFA panel resides in your MVS/QuickRef panel library. At the bottom of this panel, following the )END statement, you will find the text for a user menu panel option. Move this line somewhere into the )BODY section of the panel so that it is a valid selectable option on the panel.

Before you save the changed panel, make sure that the body section of the panel is not larger than the smallest terminal screen size in use for ISPF at your installation (normally 24 lines). (In other words, before you add this line to the body of the panel, delete one of the blank lines in the body of the panel to make room for it.) You will also need to change the VER statement in the )PROC section as indicated at the bottom of the panel so that 'U' will now be a valid option.

Once the modifications outlined above are completed, ISPF users will be able to access the information in your user data base(s) using your customized user menu panel. When option ‘U’ is entered on the MVS/QuickRef main menu, MVS/QuickRef will display your customized user menu panel and, once the user enters his choices on the user menu panel, MVS/QuickRef will read and display information from the appropriate user data base.
/* This panel is designed to provide an example of how user menu
panels may be defined. It displays a list of the products in
a single user data base and allows the user to select the
product to be displayed.

It is suggested that you use the QWIKREFM panel instead of this
one if you have multiple user data bases.

You will need to modify the section of the panel where it says:

User Product 1    User Product 1 item descriptions
User Product 2    User Product 2 item descriptions

so that it describes your own user data base products and items.

You will need to modify the section of the panel where it says:

% TYPE(TEXT)  INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT)  INTENS(LOW)  SKIP(ON)
_ TYPE(INPUT) INTENS(HIGH) JUST(LEFT) PAD(' ')

%                         * MVS/QuickRef User Menu *
Option ==> _ZSEL                                                             +

Enter the Option field with a selection number corresponding to one of
the products listed below. Enter the optional Item field with a full
item name or a generic item name.

%Item ==> _KEY             +

%1++- User Product 1         User Product 1 item descriptions
%2++- User Product 2         User Product 2 item descriptions
%3++- User Product 3         User Product 3 item descriptions
%4++- User Product 4         User Product 4 item descriptions
%5++- User Product 5         User Product 5 item descriptions
%6++- User Product 6         User Product 6 item descriptions
%7++- User Product 7         User Product 7 item descriptions
%8++- User Product 8         User Product 8 item descriptions
%9++- User Product 9         User Product 9 item descriptions

&V1=' '
&P1=' '
&P2=' '
&P9=' '
&R1=' '
&R2=' '
&R9=' '

PROC

&P9=' '
&P2=' '
&R9=' '

&ZSEL = TRUNC(&ZSEL, ' ')
/* Modify the VER statement below so that it matches the number of
/* products displayed on the panel
VER(&ZSEL, LIST, HELP, 1, 2, 3, 4, 5, 6, 7, 8, 9, *)
/* Modify the IF statement below so that it matches the number of
/* products displayed on the panel
IF (&ZSEL = 1, 2, 3, 4, 5, 6, 7, 8, 9, *)
&POPT = &ZSEL
IF (&KEY = &Z)
&KEY = *
IF (.RESP = ENTER)
&QWUM = Y
VPUT (QWUM) SHARED
IF (.RESP = END)
&POPT = ' '
&KEY = ' '
/* Modify the statement below so that it matches the number of
/* products displayed on the panel
/* The statement below could also be modified to call other user menu
/* panels, and if needed, you can have as many levels of user menu
/* panels as you need. No matter how many levels of user menus you
/* provide, in order for user data base data to be displayed, you
/* must ultimately invoke module QWIKREF1 with a fast-path string
/* like those shown in the statement below.
&ZSEL = TRANS( TRUNC(&ZSEL, '.')
HELP, 'PANEL(QWIKREF1)
    1, 'PGM(QWIKREF1) PARM(D=1 V=''&V1'' P=''&P1'' R=''&R1'' I=&KEY)'
2, 'PGM(QWIKREF1) PARM(D=1 V=''&V2'' P=''&P2'' R=''&R2'' I=&KEY)'
3, 'PGM(QWIKREF1) PARM(D=1 V=''&V3'' P=''&P3'' R=''&R3'' I=&KEY)'
4, 'PGM(QWIKREF1) PARM(D=1 V=''&V4'' P=''&P4'' R=''&R4'' I=&KEY)'
5, 'PGM(QWIKREF1) PARM(D=1 V=''&V5'' P=''&P5'' R=''&R5'' I=&KEY)'
6, 'PGM(QWIKREF1) PARM(D=1 V=''&V6'' P=''&P6'' R=''&R6'' I=&KEY)'
7, 'PGM(QWIKREF1) PARM(D=1 V=''&V7'' P=''&P7'' R=''&R7'' I=&KEY)'
8, 'PGM(QWIKREF1) PARM(D=1 V=''&V8'' P=''&P8'' R=''&R8'' I=&KEY)'
9, 'PGM(QWIKREF1) PARM(D=1 V=''&V9'' P=''&P9'' R=''&R9'' I=&KEY)'
*, 'PGM(QWIKREF1) PARM(D=1 &KEY)')
)END

Figure 30 - QWIKREFU Panel as Distributed
/* This panel is designed to provide an example of how user menu 
panels may be defined. It displays a list of the products in 
a single user data base and allows the user to select the 
product to be displayed.

It is suggested that you use the QWIKREFM panel instead of this 
one if you have multiple user data bases.

You will need to modify the section of the panel where it says:

User Product 1         User Product 1 item descriptions
User Product 2         User Product 2 item descriptions

so that it describes your own user data base products and items.

You will need to modify the section of the panel where it says:

% TYPE(TEXT)  INTENS(HIGH)  SKIP(ON)
+ TYPE(TEXT)  INTENS(LOW)  SKIP(ON)
_ TYPE(INPUT) INTENS(HIGH)  JUST(LEFT)  PAD(' ')

%                  * HyperDyne Systems User Menu *
Option ==>_ZSEL

Enter the Option field with a selection number corresponding to one of 
the products listed below. Enter the optional Item field with a full 
item name or a generic item name.

%Item ==>_KEY

%1+-  JCL Information/Requirements  Enter JCLINFO, DASDINFO, TAPEINFO, 
or * for list
%2+-  Production Messages/Abends    Enter PRODMSGS, USRABNDS, 
or * for list

&ZSEL = &POPT
.CURSOR = ZSEL
/* Use the variables below to set:
/* - the appropriate vendor names (&V1, &V2, etc.)
/* - the appropriate product names (&P1, &P2, etc.)
/* - the appropriate release numbers (&R1, &R2, etc.)
/* &V1 is the vendor name of the first product listed on the panel
/* &P1 is the product name of the first product listed on the panel
/* &R1 is the release number of the first product listed on the panel
/* &V2 is the vendor name of the second product listed on the panel
/* &P2 is the product name of the second product listed on the panel
/* &R2 is the release number of the second product listed on the panel
/* etc.
/* These are the vendor names, product names, and release numbers
/* actually carried in the corresponding user data base.
/* If a given vendor name, product name, or release number was left 
/* blank in the data base, then leave it blank in the settings below.
&V1='DATA CENTER'
&P1='MISCELLANEOUS INFO'
&R1=''
&V2='DATA CENTER'
&P2='MESSAGES & ABENDS'
&R2='1.2'
)PROC
 &ZSEL = TRUNC(&ZSEL, ' ')
/* Modify the VER statement below so that it matches the number of
 products displayed on the panel
 VER(&ZSEL, LIST, HELP, 1, 2, *)
/* Modify the IF statement below so that it matches the number of
 products displayed on the panel
 IF (&ZSEL = 1, 2, *)
 &POPT = &ZSEL
 IF (&KEY = &Z)
 &KEY = *
 IF (.RESP = ENTER)
 &QWUM = Y
 VPUT (QWUM) SHARED
 IF (.RESP = END)
 &POPT = ' '
 &KEY = ' '
/* Modify the statement below so that it matches the number of
 products displayed on the panel
 /* The statement below could also be modified to call other user menu
 panels, and if needed, you can have as many levels of user menu
 panels as you need. No matter how many levels of user menus you
 provide, in order for user data base data to be displayed, you
 /* must ultimately invoke module QWIKREF1 with a fast-path string
 /* like those shown in the statement below.
 &ZSEL=TRANS( TRUNC(&ZSEL, '.')
    HELP,'PANEL(QWIKREFN)'
    1,'PGM(QWIKREF1) PARM(D=1 V=''&V1'' P=''&P1'' R=''&R1'' I=&KEY)'
 2,'PGM(QWIKREF1) PARM(D=1 V=''&V2'' P=''&P2'' R=''&R2'' I=&KEY)'
 *,'PGM(QWIKREF1) PARM(D=1 &KEY)')
)END

---

Figure 31 - QWIKREFU Panel After Modification
Examples Of Creating User Data Bases

This section contains two complete examples of how MVS/QuickRef user data bases might be created. The first example is for a single user data base; the second example involves multiple user data bases.

Example 1: Single User Data Base

Let's assume that we want to create a single user data base which will show the following types of data:

♦ Internal JCL coding standards for the following JCL elements: TIME, CLASS, JOBNAME

♦ A description of the following production application user abends: U4001 and U4002

♦ Mailing addresses for the sales and accounting departments

♦ Fax phone numbers for the sales and accounting departments

Now let's assume that our company's name is Acme and that we want to organize the data in the MVS/QuickRef user data base as follows:

♦ Internal JCL coding standards will be placed in a product with vendor name 'ACME', product name 'JCL CODING STANDARDS', and release number 'Revision 1.2'. This product will contain the following item names: 'TIME', 'CLASS', and 'JOBNAME'.

♦ User abends will be placed in a product with vendor name 'ACME', product name 'USER ABENDS', and release number 'V3'. This product will contain the following item names: 'U4001' and 'U4002'.

♦ Mailing addresses will be placed in a product with vendor name 'INTERNAL', product name 'MAILING ADDRESSES', and release number 'AS OF 12/1/96'. This product will contain the following item names: 'SALES' and 'ACCOUNTING'.

♦ Fax phone numbers will be placed in a product with vendor name 'INTERNAL', product name 'FAX PHONE NUMBERS', and release number 'AS OF 12/1/96'. This product will contain the following item names: 'SALES' and 'ACCOUNTING'.

Based on this setup, information on JCL coding standards for the TIME parameter will be stored in (and retrieved from) item 'TIME' within product 'JCL CODING STANDARDS' associated with vendor 'ACME'. Item 'U4001' within product 'USER ABENDS' associated with vendor 'ACME' will describe user abend U4001; item 'SALES' within product 'MAILING ADDRESSES' associated with vendor 'INTERNAL' will contain the mailing address for the sales department; item 'SALES' within product 'FAX PHONE NUMBERS' associated with vendor 'INTERNAL' will contain the fax phone number for the sales department, etc.

For purposes of this example, let's assume that:
we want a copyright statement to appear at the top of the reference information for all items in product 'JCL CODING STANDARDS' and that we want that copyright statement to automatically reflect the current year

we want a copyright statement to appear with the items in product 'USER ABENDS' but we want the year in this copyright statement to appear as 1995

no copyright statement is needed or should be shown for the other two products.

Figure 32 shows a sample of the JCL that could be used to create this user data base.

Note: Please keep in mind that this is a "simplified" example. A "real" user data base would probably contain many more items in each product and the associated reference text would be more "realistic".

```
//JS10  EXEC PGM=QWIKREF2,REGION=0K,PARM='TITLES=N'
//STEPLIB DD DSN=quickref.program.linklib,DISP=SHR
//SYSPRINT DD SYSOUT=*  
//DATAIN DD *  
K V=ACME P='JCL CODING STANDARDS' R='REVISION 1.2' I=TIME
*** TIME PARAMETER ***

The TIME parameter should be coded on each job step and should not exceed the time allowed by the CLASS parameter.

K V=ACME P='JCL CODING STANDARDS' R='REVISION 1.2' I=CLASS
*** CLASS PARAMETER ***

The CLASS parameter should be coded as A (for 10 CPU seconds), B (for 30 CPU seconds), and C (for 2 minutes of CPU time).

K V=ACME P='JCL CODING STANDARDS' R='REVISION 1.2' I=JOBNAME
*** JOBNAME PARAMETER ***

The JOBNAME must consist of your 5-character TSO userid followed by a single alphanumeric character.

KCPY V=ACME P='USER ABENDS' R=V3
KCPY C='Text Below Copyright (c) 1995, Acme Inc.'

K V=ACME P='USER ABENDS' R=V3 I=U4001
User abend U4001 is a serious problem. Call programmer Smith.

K V=ACME P='USER ABENDS' R=V3 I=U4002
User abend U4002 is a critical problem. Call programmer Jones.

KCPY V=INTERNAL C=NO
K V=INTERNAL P='MAILING ADDRESSES' R='AS OF 12/1/96' I=SALES
Sales Dept.  PO Box 123  Little Rock, ARK. 45923
K V=INTERNAL P='MAILING ADDRESSES' R='AS OF 12/1/96' I=ACCOUNTING
Accounting Dept.  PO Box 200  Little Rock, ARK. 45923
K V=INTERNAL P='FAX PHONE NUMBERS' R='AS OF 12/1/96' I=SALES
Fax: 571-449-4512
K V=INTERNAL P='FAX PHONE NUMBERS' R='AS OF 12/1/96' I=ACCOUNTING
Fax: 571-449-2281

//QWREFDD DD DSN=SYS3.QUICKREF.USER.DATABASE,DISP=(,CATLG,DELETE),
// DCFM=F,BLKSIZE=6160,LRECL=6160)
//SORTIN DD UNIT=3380,SPACE=...  
//SORTWK01 DD UNIT=3380,SPACE=...
//SORTWK02 DD UNIT=3380,SPACE=...
//SORTWK03 DD UNIT=3380,SPACE=...
//SYSOUT DD SYSOUT=*  
```

Figure 32 - JCL for User Data Base Example 1

In this example, the DATAIN file takes the form of an in-stream sequential file. The first record:
**K V=ACME P='JCL CODING STANDARDS' R='REVISION 1.2' I=TIME**

is a key indicator record because it has a K in column 1. The vendor name is 'ACME', the product name is 'JCL CODING STANDARDS', the release number is 'REVISION 1.2', and the item name is 'TIME'. The records which follow (up to the next key indicator record):

*** TIME PARAMETER ***

The TIME parameter should be coded on each job step and should not exceed the time allowed by the CLASS parameter.

provide the reference text to be stored in the data base for item TIME in this product. With this setup, a fast-path request like **QW D=1 TIME** (or **QW TIME** for a cross-data base search) would cause this item to be retrieved from the user data base. The resulting display is shown in Figure 2 below:

---

![Figure 33 - User Data Base Item TIME](image)

---

The rest of the records in the DATAIN file provide the key indicator records, copyright indicator records, and text records needed for the rest of the items in our user data base.

We did not use the TEXTLEFT= and TEXTRIGHT= parameters because our input reference text falls within the default columns (i.e., TEXTLEFT=2, TEXTRIGHT=79).

We used the TITLES=N parm so no user data base titles were specified in our input file.

Notice the copyright statement that appears just above the reference text for item 'TIME' in Figure 2. Because we did not specify a copyright indicator record for product 'JCL CODING STANDARDS', this copyright statement is automatically generated and will always show the current calendar year (which shows up as 1997 in this example). If this item were displayed in 1998, this date would automatically show up as 1998. Since it is automatically generated, the vendor name in this copyright statement is taken from the vendor name in the associated key indicator record (ACME).
We specified the following copyright indicator record for product 'USER ABENDS':

KCPY V=ACME P='USER ABENDS' R=V3
KCPY C='Text Below Copyright (c) 1995, Acme Inc.'

For this reason, items 'U4001' and 'U4002', which belong to this product, would be displayed with the following copyright statement:

*************** Text Below Copyright (c) 1995, Acme Inc. ***************
This copyright statement (and the date it contains) will continue to be shown for these items until the associated copyright statement is updated or changed.

We also specified the following copyright indicator for vendor 'INTERNAL':

KCPY V=INTERNAL C=NO

For this reason, all items in all products belonging to vendor 'INTERNAL' (like items 'SALES' and 'ACCOUNTING' in product 'MAILING ADDRESSES') would have no copyright statement shown. Instead of a copyright statement, these items would be preceded, whenever they are displayed, by the "TOP OF DATA" line.

With a single user data base, you can request a product selection list showing all products in that data base (in vendor sequence) with the following fast-path string: QW D=1 L=V. With the single user data base setup shown in Figure 1, this would produce the product selection list shown in Figure 32 below:

<table>
<thead>
<tr>
<th>Item =&gt;</th>
<th>MVS/QuickRef</th>
<th>Col 1 Line 1 of 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command =&gt;</td>
<td>Scroll ==&gt; PAGE</td>
<td></td>
</tr>
<tr>
<td>Select desired product for item list and/or enter desired item at top left</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Vendor</td>
<td>Product</td>
<td>Release</td>
</tr>
<tr>
<td>_ ACME</td>
<td>JCL CODING STANDARDS</td>
<td>REVISION 1.2</td>
</tr>
<tr>
<td>_ ACME</td>
<td>USER ABENDS</td>
<td>V3</td>
</tr>
<tr>
<td>_ INTERNAL</td>
<td>FAX PHONE NUMBERS</td>
<td>AS OF 12/1/96</td>
</tr>
<tr>
<td>_ INTERNAL</td>
<td>MAILING ADDRESSES</td>
<td>AS OF 12/1/96</td>
</tr>
</tbody>
</table>
| ******************************************* BOTTOM OF LIST *******************************************

Figure 34 - User Data Base Product Selection List

You could then select product 'JCL CODING STANDARDS' from this product selection list in order to produce the generic item selection list shown in Figure 33 below.
You could then select item 'TIME' from this generic item selection list in order to display the reference text for item 'TIME' as shown in Figure 2 above.

You will notice that we only specified 10 tracks for our user data base in this example. This is because our input text is relatively small and 10 tracks should be more than adequate to hold this limited amount of data. Depending upon the amount of reference text to be stored, you may have to increase this for a "real" user data base.

Once the JCL in Figure 1 is submitted and successfully executes, we will have completed the first of the 4 steps required to create and process a user data base (i.e., Step 1 - Allocate and load your user data base).

The next step is: Step 2 - Define name of your user data base. The name we assigned to our user data base in this example is SYS3.QUICKREF.USER.DATABASE. Let's assume that, rather than preallocating our user data base, we decide to define the name using the QWIKOPTS table.

As explained in the section on "Setting MVS/QuickRef Global Installation Options", this means that, since we have a single user data base, we would use the QWIKOPTS LOCLDB1= parm to identify the name of our user data base and it would be coded as shown below:

| LOCLDB1=SYS3.QUICKREF.USER.DATABASE, |

Once this change is made and QWIKOPTS is reassembled and linked (as outlined in the section on "Setting MVS/QuickRef Global Installation Options"), the name of our user data base will be defined to MVS/QuickRef. Once this is done, our user data base will be ready for processing and the Request Reference Information and List Vendors/Products/Releases panels will automatically be adjusted to show the following additional entry field:

Data Base to Search ==> (M for MVS/QuickRef Main data base, 1 for User data base, * or blank for all data bases)

The next step is: Step 3 - Customize user menu panel. Of course, before we do this, we must first determine if we really need to provide a user menu panel. For a discussion of the considerations involved in making this decision, see the section titled "Determine If User Menu Panel Is Needed" in this chapter. For the purposes of this example, let's assume that we want to provide a user menu panel. Figure 30 shows panel QWIKREFU as distributed on the install tape. Figure 3 below shows how QWIKREFU might be modified for our current example.
If you look in Figure 3 at the comments in the QWIKREFU panel, which describe the modifications that must be made to this panel in order to use it, you will see where changes were made to QWIKREFU for our current example. Notice that, if the user enters '1' in the Option field and 'TIME' in the Item field, then, based on the way the TRANS statement is coded at the bottom of the panel, program QWIKREF1 would be invoked and would be passed the following fast-path invocation string:

D=1 V='ACME' P='JCL CODING STANDARDS' R='REVISION 1.2' I=TIME

This would cause the reference text for item 'TIME' in product 'JCL CODING STANDARDS' to be displayed, as shown in Figure 2.

The last step in setting up our user data base involves adding option U to the MVS/QuickRef main menu. This will allow our users to enter a U on the main menu in order to bring up the user menu we customized in the step above. This is described in the section above titled "Step 4 - Add Option U to MVS/QuickRef Menu".
/* This panel is designed to provide an example of how user menu panels may be defined. It displays a list of the products in a single user data base and allows the user to select the product to be displayed.

It is suggested that you use the QWIKREFM panel instead of this one if you have multiple user data bases.

You will need to modify the section of the panel where it says:

User Product 1 User Product 1 item descriptions
User Product 2 User Product 2 item descriptions

so that it describes your own user data base products and items.

You will need to modify the section of the panel where it says:

% TYPE(TEXT) INTENS(HIGH) SKIP(ON)
+ TYPE(TEXT) INTENS(LOW) SKIP(ON)
_ TYPE(INPUT) INTENS(HIGH) JUST(LEFT) PAD(' ')

Use the variables below to set:
- the appropriate vendor names (&V1, &V2, etc.)
- the appropriate product names (&P1, &P2, etc.)
- the appropriate release numbers (&R1, &R2, etc.)

&V1 is the vendor name of the first product listed on the panel
&P1 is the product name of the first product listed on the panel
&R1 is the release number of the first product listed on the panel
&V2 is the vendor name of the second product listed on the panel
&P2 is the product name of the second product listed on the panel
&R2 is the release number of the second product listed on the panel

These are the vendor names, product names, and release numbers actually carried in the corresponding user data base.
If a given vendor name, product name, or release number was left blank in the data base, then leave it blank in the settings below.

&V1=ACME
&P1='JCL CODING STANDARDS'
&R1='REVISION 1.2'
&V2='ACME'
&P2='USER ABENDS'
&R2=V3

Enter the Option field with a selection number corresponding to one of the products listed below. Enter the optional Item field with a full item name or a generic item name.
&V3=INTERNAL
&P3='MAILING ADDRESSES'
&R3='AS OF 12/1/96'
&V4=INTERNAL
&P4='FAX PHONE NUMBERS'
&R4='AS OF 12/1/96'
PROC
&ZSEL = TRUNC(&ZSEL,' ')
/* Modify the VER statement below so that it matches the number of
/* products displayed on the panel
VER(&ZSEL,LIST,HELP,1,2,3,4,*)
/* Modify the IF statement below so that it matches the number of
/* products displayed on the panel
IF (&ZSEL = 1,2,3,4,*)
 &POPT = &ZSEL
 IF (&KEY = &Z)
   &KEY = *
 IF (.RESP = ENTER)
   &QWUM = Y
   VPUT (QWUM) SHARED
 IF (.RESP = END)
 &POPT = ' '
 &KEY = ' '
/* Modify the statement below so that it matches the number of
/* products displayed on the panel
/* The statement below could also be modified to call other user menu
/* panels, and if needed, you can have as many levels of user menu
/* panels as you need. No matter how many levels of user menus you
/* provide, in order for user data base data to be displayed, you
/* must ultimately invoke module QWIKREF1 with a fast-path string
/* like those shown in the statement below.
&ZSEL=TRANS( TRUNC(&ZSEL,'.'))
HELP,'PANEL(QWIKREFN)'
 1,'PGM(QWIKREF1) PARM(D=1 V=''&V1'' P=''&P1'' R=''&R1'' I=&KEY)'
 2,'PGM(QWIKREF1) PARM(D=1 V=''&V2'' P=''&P2'' R=''&R2'' I=&KEY)'
 3,'PGM(QWIKREF1) PARM(D=1 V=''&V3'' P=''&P3'' R=''&R3'' I=&KEY)'
 4,'PGM(QWIKREF1) PARM(D=1 V=''&V4'' P=''&P4'' R=''&R4'' I=&KEY)'
 *,'PGM(QWIKREF1) PARM(D=1 &KEY)'
)END

Figure 35 - QWIKREFU Panel Modified For User Data Base Example 1
Example 2: Multiple User Data Bases

Let's assume that, in example 1 above, the information on JCL coding standards and user abends is managed by the data processing department and that the information on mailing addresses and fax phone numbers is controlled by the personnel department and, furthermore, that these two departments have problems coordinating user data base updates. With all the information in one user data base, then, when the user data base is being rebuilt (as may be required from time to time to incorporate changes or additions), they must both make sure all updates are in place in the input file. (In other words, the data processing department cannot rebuild the data base while the personnel department is working on updates.)

One solution to this problem is to provide a separate user data base for each of the two departments. This can be done, in example 1 above, by splitting the DATAIN file shown in Figure 1 into two separate input files: one containing the information controlled by the data processing department and the other containing the information controlled by the personnel department. Once this is done, we can run the JCL in Figure 1 twice; once with the first input file and once with the second input file, being sure to change the name of the user data base in the QWREFDD DD statement each time. Let's assume we do this and thereby create two user data bases: one for the data processing department named QREF.DP.USER.DATABASE and one for the personnel department named QREF.PER.USER.DATABASE. Once this is done, we will have completed the first of the 4 steps required to create and process user data bases (i.e., Step 1 - Allocate and load your user data bases).

The next step is: Step 2 - Define name(s) of your user data base(s). As explained in the section on "Setting MVS/QuickRef Global Installation Options" in Chapter 4, this means that we would use the QWIKOPTS LOCLDB1 and LOCLDB2 parms to identify the names of our user data bases and they would be coded as shown below:

```
LOCLDB1=QREF.DP.USER.DATABASE,
LOCLDB2=QREF.PER.USER.DATABASE,
```

Once this change is made and QWIKOPTS is reassembled and linked (as outlined in the section on "Setting MVS/QuickRef Global Installation Options"), the names of our two user data bases will be defined to MVS/QuickRef. Once this is done, our user data bases will be ready for processing and the Request Reference Information and List Vendors/Products/Releases panels will automatically be adjusted to show the following additional entry field:

```
Data Base to Search ==> (M for MVS/QuickRef Main data base, 1-2 for User data base, * or blank for all data bases)
```

It should be noted that, with this setup, the data processing department's user data base (QREF.DP.USER.DATABASE) would be treated as user data base #1; the personnel department's user data base (QREF.PER.USER.DATABASE) would be user data base #2. This means that item 'TIME' could be retrieved from the data processing department's user data base with the following fast-path invocation string: **QW D=1 TIME** or, for a cross-data base...
search, **QW TIME**. You could get a list of all products in the user data base belonging to the personnel department (in vendor name sequence), by specifying **QW D=2 L=V**.

The next step is: Step 3 - Customize user menu panel. As explained in example 1 above, before we do this, we must first determine if we really need to provide a user menu panel. For a discussion of the considerations involved in making this decision, see "Determine If User Menu Panel Is Needed". For the purposes of this example, let's assume that we want to provide a user menu panel.

We cannot use QWIKREFU for our user data base menu panel, as in example 1 above, because we need to identify the user data base as well as product and item to be displayed. One solution to this problem is to customize panel QWIKREFM and use it in place of QWIKREFU.

Figure 4 below shows QWIKREFM as it is distributed with MVS/QuickRef (and stored in your MVS/QuickRef panel library). Figure 5 below shows how QWIKREFM might be customized for our current example.

If you look in Figure 5 at the comments in the QWIKREFM panel, which describe the modifications that must be made to this panel in order to use it, you will see where changes were made to QWIKREFM for our current example. Notice that, if the user enters '1' in the Option field, 'TIME' in the Item field, and '1' in the Database field, then, based on the way the TRANS statement is coded at the bottom of the panel, program QWIKREF1 would be invoked and would be passed the following fast-path invocation string:

```
D=1 V='ACME' P='JCL CODING STANDARDS' R='REVISION 1.2' I=TIME
```

This would cause the reference text for item 'TIME' in product 'JCL CODING STANDARDS' to be displayed, as shown in Figure 2.

Once panel QWIKREFM has been customized, steps must be taken so that it will be used in place of panel QWIKREFU. Let's assume that we decide to use the QWIKOPTS table to change the default name of the user data base menu panel. As explained in the section on "Setting MVS/QuickRef Global Installation Options", this means that we would use the QWIKOPTS USERMNU= parm to identify the name of our user data base menu panel and it would be coded as shown below:

```
USERMNU=QWIKREFM,
```

Once this change is made and QWIKOPTS is reassembled and linked (as outlined in the section on "Setting MVS/QuickRef Global Installation Options"), the customized version of QWIKREFM will be established as the user data base menu panel.

The last step in setting up our user data bases involves adding option U to the MVS/QuickRef main menu. This will allow our users to enter a U on the main menu in order to bring up the user menu we customized in the step above. This is described in the section titled "Step 4 - Add Option U to MVS/QuickRef Menu" in this chapter.
This panel is designed to provide an example of how user menu panels may be defined. It displays a list of the products in two or more user data bases and allows the user to select a product to be displayed as well as the user data base to be accessed. You will need to modify the section of the panel where it says:

```plaintext
/* User Product 1              User Product 6
   User Product 2              User Product 7
   . . . .
   so that it describes your own user data base products.

You will need to modify the section of the panel where it says:

/* User Database 1             User Database 6
   User Database 2             User Database 7
   . . . .
   so that it describes your own user data bases.
```

The panel provides the following fields:

- **Option**: Enter the selection number corresponding to one of the products listed below:
  - User Product 1
  - User Product 2
  - User Product 3
  - User Product 4
  - User Product 5
  - User Product 6
  - User Product 7
  - User Product 8
  - User Product 9

- **Item**: Enter an optional full item name or a generic item name.

- **Database**: Enter the selection number corresponding to one of the data bases listed below:
  - User Database 1
  - User Database 2
  - User Database 3
  - User Database 4
  - User Database 5
  - User Database 6
  - User Database 7
  - User Database 8
  - User Database 9

You will need to modify the variables below to set:

- The appropriate vendor names (&V1, &V2, etc.)
- The appropriate product names (&P1, &P2, etc.)
- The appropriate release numbers (&R1, &R2, etc.)

These are the vendor names, product names, and release numbers actually carried in the corresponding user data base. If a given vendor name, product name, or release number was left
/* blank in the database, then leave it blank in the settings below.*/
&V1=''
&P1=''
&R1=''


&V9=''
&P9=''
&R9=''
)

PROC
&ZSEL = TRUNC(&ZSEL,' ')
/* Modify the VER statement below so that it matches the number of*/
/* products displayed on the panel*/
VER(&ZSEL,NB,LIST,HELP,1,2,3,4,5,6,7,8,9,*
/* Modify the VER statement below so that it matches the number of*/
/* user databases displayed on the panel*/
VER(&DOPT,LIST,1,2,3,4,5,6,7,8,9,*
/* Modify the IF statement below so that it matches the number of*/
/* products displayed on the panel*/
IF ( &ZSEL = 1,2,3,4,5,6,7,8,9,*
   &POPT = &ZSEL
   IF ( &KEY = &Z )
      &KEY = *
   IF ( &DOPT = &Z )
      &DOPT = *
   IF (.RESP = ENTER)
      &QWUM = Y
      VPUT (QWUM) SHARED
   IF (.RESP = END)
      &POPT = '
      &KEY = '
      &DOPT = '
/* Modify the statement below so that it matches the number of*/
/* products displayed on the panel*/
/* The statement below could also be modified to call other user menu*/
/* panels, and if needed, you can have as many levels of user menu*/
/* panels as you need. No matter how many levels of user menus you*/
/* provide, in order for user database data to be displayed, you*/
/* must ultimately invoke module QWIKREF1 with a fast-path string*/
/* like those shown in the statement below.*/
&ZSEL = TRANS( TRUNC(&ZSEL,'.')
HELP,'PANEL(QWIKREFN)'  
   1,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V1'' P=''&P1'' R=''&R1'' I=&KEY)'
   2,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V2'' P=''&P2'' R=''&R2'' I=&KEY)'
   3,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V3'' P=''&P3'' R=''&R3'' I=&KEY)'
   4,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V4'' P=''&P4'' R=''&R4'' I=&KEY)'
   5,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V5'' P=''&P5'' R=''&R5'' I=&KEY)'
   6,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V6'' P=''&P6'' R=''&R6'' I=&KEY)'
   7,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V7'' P=''&P7'' R=''&R7'' I=&KEY)'
   8,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V8'' P=''&P8'' R=''&R8'' I=&KEY)'
   9,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V9'' P=''&P9'' R=''&R9'' I=&KEY)'
*/,'PGM(QWIKREF1) PARM(I=&KEY)'
)

END

---

Figure 36 - QWIKREFM Panel As Distributed
This panel is designed to provide an example of how user menu panels may be defined. It displays a list of the products in two or more user data bases and allows the user to select a product to be displayed as well as the user data base to be accessed.

You will need to modify the section of the panel where it says:

```
   User Product 1              User Product 6
   User Product 2              User Product 7
```

so that it describes your own user data base products.

You will need to modify the section of the panel where it says:

```
   User Database 1             User Database 6
   User Database 2             User Database 7
```

so that it describes your own user data bases.

```
% TYPE(TEXT)  INTENS(HIGH) SKIP(ON)
+% TYPE(TEXT)  INTENS(LOW)  SKIP(ON)
$_ TYPE(INPUT) INTENS(HIGH) JUST(LEFT) PAD(' ')
$ TYPE(OUTPUT) INTENS(LOW) JUST(LEFT) PAD(' ') CAPS(ON)
)BODY */ MVS/QUICKREF, COPYRIGHT 1989, 1996 CHICAGO-SOFT, LTD.
% * MVS/QuickRef User Data Base Selection Panel *
Option ==> _ZSEL
Enter the Option field with a selection number corresponding to one of the products listed below:

- %1+- JCL Coding Standards (Choose user data base 1)
- %2+- User Abends
- %3+- Mailing Addresses (Choose user data base 2)
- %4+- FAX Phone Numbers

%Item ==> _KEY
Enter the optional Item field with a full item name or a generic item name.

%Database ==> _Z+
Enter the optional Database field with a selection number corresponding to one of the data bases listed below:

- %1+- Data Processing Data Base
- %2+- Personnel Data Base

)INIT
.ZVARS='(DOPT)'
&ZSEL = &POPT
.CURSOR = ZSEL
/* Use the variables below to set:
   - the appropriate vendor names (&V1, &V2, etc.)
   - the appropriate product names (&P1, &P2, etc.)
   - the appropriate release numbers (&R1, &R2, etc.)
   &V1 is the vendor name of the first product listed on the panel
   &P1 is the product name of the first product listed on the panel
   &R1 is the release number of the first product listed on the panel
   &V2 is the vendor name of the second product listed on the panel
   &P2 is the product name of the second product listed on the panel
   &R2 is the release number of the second product listed on the panel
   etc.
   These are the vendor names, product names, and release numbers actually carried in the corresponding user data base.
   If a given vendor name, product name, or release number was left blank in the data base, then leave it blank in the settings below.
   &V1=ACME
   &P1='JCL CODING STANDARDS'
&R1='REVISION 1.2'
&V2='ACME'
&P2='USER ABENDS'
&R2=V3
&V3=INTERNAL
&P3='MAILING ADDRESSES'
&R3='AS OF 12/1/96'
&P4='FAX PHONE NUMBERS'
&R4='AS OF 12/1/96'

PROC
&ZSEL = TRUNC(&ZSEL,' ')
/* Modify the VER statement below so that it matches the number of */
/* products displayed on the panel */
VER(&ZSEL,NB,LIST,HELP,1,2,3,4,*)
/* Modify the VER statement below so that it matches the number of */
/* user data bases displayed on the panel */
VER(&DOPT,LIST,1,2,*)
/* Modify the IF statement below so that it matches the number of */
/* products displayed on the panel */
IF (&ZSEL = 1,2,3,4,*)
&POPT = &ZSEL
IF (&KEY = &Z)
&KEY = *
IF (&DOPT = &Z)
&DOPT = *
IF (.RESP = ENTER)
&QWUM = Y
VPUT (QWUM) SHARED
IF (.RESP = END)
&POPT = ''
&KEY = ''
&DOPT = '''
/* Modify the statement below so that it matches the number of */
/* products displayed on the panel */
/* The statement below could also be modified to call other user menu */
/* panels, and if needed, you can have as many levels of user menu */
/* panels as you need. No matter how many levels of user menus you */
/* provide, in order for user data base data to be displayed, you */
/* must ultimately invoke module QWIKREF1 with a fast-path string */
/* like those shown in the statement below. */
&ZSEL = TRANS( TRUNC(&ZSEL,'.')
HELP,'PANEL(QWIKREF1)
 1,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V1'' P=''&P1'' R=''&R1'' I=&KEY)'
 2,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V2'' P=''&P2'' R=''&R2'' I=&KEY)'
 3,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V3'' P=''&P3'' R=''&R3'' I=&KEY)'
 4,'PGM(QWIKREF1) PARM(D=&DOPT V=''&V4'' P=''&P4'' R=''&R4'' I=&KEY)'
*,,'PGM(QWIKREF1) PARM(I=&KEY))
)END

______________________________________________________________________________

Figure 37 - QWIKREFM Panel Modified For User Data Base Example 2
Chapter 6 - Using the Override Parameter Feature
Introduction

The MVS/QuickRef override parameter feature allows you to augment, replace, or prevent access to the reference information presented by MVS/QuickRef. This feature of MVS/QuickRef is optional and need not be used unless you want to alter the way MVS/QuickRef normally processes and presents reference information.

Overview

In chapter five of this guide, you learned how to create your own “user” reference data base for use at your installation. Locally produced reference information is stored in your user data base when you create it; if you want to change the stored information, you must delete the user data base and rebuild it from scratch. Although the rebuild process is very quick, there may be times when you want to include frequently changing information in your user data base, making the delete-recreate process inconvenient.

The MVS/QuickRef override parameter feature allows you to identify additional information to be used to augment or replace any item in your user data base, or in the MVS/QuickRef main data base. You identify the additional reference information by creating a set of override parameters in a parameter Partitioned Data Set (PDS). The override parameters tell MVS/QuickRef which PDS members contain the augmenting or replacing text. To help you understand when and how to use the override parameter feature, let’s look at an example that uses the feature.

Override Parameter Usage Example

Suppose for the purpose of our example that you have created your own local user data base with MVS/QuickRef as outlined in chapter five of this guide. The data base contains relevant reference information on JCL coding conventions, tape handling procedures, and user abend codes for the nightly production batch cycle. You have decided to add “hot news” information to your user data base, but you know that the “hot news” information will probably change very frequently, possibly daily. To support this volatile new reference information, you decide to use the MVS/QuickRef override parameter feature. Here are the steps you follow to implement your “hot news” information; these steps can also be used as a checklist for your use later in defining override parameters.

1. If you created or customized one or more user menu panels, add the new “hot news” information to the user menu panel(s).

2. Place the text for your “hot news” entry in one or more PDS members. The PDS containing the "hot news" text may be allocated with RECFM F, V, FB, or VB. The LRECL and BLKSIZE can be any value supported by the operating system.
3. Create a PDS member with the name QWPARM00. The PDS containing this member must contain blocked or unblocked fixed-length records. The LRECL and BLKSIZE can be any value supported by the operating system. The QWPARM00 member will contain the override parameters that will tell MVS/QuickRef the format and location of your “hot news” text. Edit the QWPARM00 member and add the override parameters, following the syntax rules and instructions outlined in the “Override Parameter Syntax” section in this chapter.

4. Define the name of the PDS containing the QWPARM00 member to MVS/QuickRef following the instructions in the section on “Defining the Parameter Data Set Name” later in this chapter.

When MVS/QuickRef is next invoked by any user, it will extract the parameter data set name, dynamically allocate it (if it is not preallocated), and read in and process the parameters in the QWPARM00 member. The parameters will tell MVS/QuickRef where the “hot news” text resides. If a MVS/QuickRef requests that the “hot news” information be displayed, the “hot news” PDS member will be dynamically allocated and its contents presented to the user, in the same format that the user data base contents are presented. If you edit and change the “hot news” text, MVS/QuickRef will immediately pick up the changed text when it is next invoked.

Using the steps in the above example as a guide, you can now create your own override parameter set to meet the needs of your installation. Possible other uses for this feature include:

- Adding special instructions to the MVS/QuickRef MVS abend descriptions
- Adding local JCL usage hints to the MVS/QuickRef MVS JCL element descriptions
- Adding new information to your user data base or correcting errors in it

The “Override Parameter Syntax” section in this chapter contains a detailed description of the format for the override parameters you will use. Member QWPARM00 in the MVS/QuickRef JCL library contains sample parameters that you can use as a starting point.

Override Parameter Syntax

This section describes the format of the override parameters that you can create to establish your overrides. Parameters that are optional will be enclosed in brackets like these ‘{}’ in the syntax descriptions below; parameters that are required will not be enclosed in brackets. When one of a list of parameters must be chosen, the parameters will be separated by a vertical bar ‘|’.

The parameters can reside in any PDS containing fixed-length records but they must reside in a member with the name QWPARM00. The parameters are read in and processed each time MVS/QuickRef is invoked, so parameter changes are picked up immediately.
If there are any syntax errors in the override parameters, the override parameters will NOT be used. Syntax errors encountered during parameter parsing will NOT be reported unless MVS/QuickRef is invoked via the QWTEST CLIST, as described later in the section entitled “Validating Your Override Parameters”.

Override parameters can be specified anywhere in a parameter record, from column 1 to the end of the record. Multiple parameters can appear on the same record, if they are separated by a comma. Continuation onto the next record is indicated by following the last parameter on a record with a comma and at least one blank. Continued parameters can start in any column on the continuation record. Comments may appear on the same record as a parameter, if they are separated from the parameter by at least one blank. An asterisk in column one of a record indicates that the entire record is a comment, so the record is ignored. Blank lines cannot be included in the QWPARM00 member; you should use a comment line that begins with an asterisk instead of a blank line.

The types of override statements that can be specified are:

- **DATAPDS**, used to establish the name of the PDS containing the actual new override reference text to use - this statement must appear first in the QWPARM00 member
- **AUGMENT**, which adds to an existing reference item - this statement can appear anywhere in the QWPARM00 member
- **REPLACE**, to replace an existing item or define a new item - this statement can appear anywhere in the QWPARM00 member
- **ALLOW**, used to allow access to an item based upon the TSO user ID or LOGON PROC name of the user - this statement can appear anywhere in the QWPARM00 member and is opposite in function to the PREVENT statement described below
- **PREVENT**, which is used to deny user access to an item based upon the TSO user ID or LOGON PROC name of the user - this statement can appear anywhere in the QWPARM00 member

The sections that follow describe the syntax for each type of override statement you can specify.

**DATAPDS Statement**

The DATAPDS statement is required and must appear first in the QWPARM00 member. The DATAPDS statement specifies the name of the PDS in which the actual reference text resides that is to be read in and used for an AUGMENT or REPLACE override. The PDS named on the DATAPDS statement will be searched for the indicated reference text member unless another PDS name is specified by the DSN= parameter on the AUGMENT or REPLACE statement. If another PDS name is specified by the DSN= parameter on the AUGMENT or REPLACE
statement, then that PDS will be searched for the indicated text member instead of the PDS specified on the DATAPDS statement.

The format of the DATAPDS statement is:

    DATAPDS DSN=datasetname

For more details on the DSN= parameter, see the section on "Common Override Parameters" below.

Here are three examples of valid DATAPDS statements:

    DATAPDS DSN=SYS2.SYSIN
    DATAPDS DSN=PROD.B321.USERS_GUIDE
    DATAPDS DSN=ALPHA.PRODLIB.EXTRACT

AUGMENT Statement

The AUGMENT statement specifies an item to be augmented (added-to) and indicates where the extra or "augmenting" text is to be found. The extra or "augmenting" text will be automatically read in and placed AFTER the data base text for the indicated item. The format of the AUGMENT statement is:

    AUGMENT {VENDOR='vendor name',}
             {PRODUCT='product name',}
             {RELEASE='release number',}
             ITEM=itemname,
             MEMBER=membername,
             {DSN=datasetname,}
             {TEXTLEFT=n,}
             {TEXTRIGHT=n,}
             {UDBN=n}

For a description of the various AUGMENT statement parameters, see the section on "Common Override Parameters" below.

Here are some examples of valid AUGMENT statements:

    AUGMENT VENDOR=CA,
             PRODUCT='CA-PDSMAN MESSAGES',
             RELEASE=7.1,
             ITEM=FCO100I,
             MEMBER=AUGF100,
             TEXTLEFT=1,
             TEXTRIGHT=72
             Vendor to augment
             Product to augment
             Release to augment
             Item to augment
             Member where text resides
             Columns where augment text resides
In this case, the reference text for item 'FCO100I' in release '7.1' of product 'CA-PDSMAN MESSAGES' belonging to vendor 'CA' in the MVS/QuickRef main data base would be augmented by the reference text in columns 1 through 72 of member AUGF100 within the PDS specified by the DATAPDS statement.

AUGMENT VENDOR=INTERNAL,
        PRODUCT=JCL*,
        ITEM=*,
        MEMBER=JCLAUG,
        UDBN=1,                     User data base number
        DSN=JCL.AUG.TEXT            DSN for member

In this case, the reference text for all items in any release of any product with a product name starting with the characters 'JCL' belonging to vendor 'INTERNAL' in user data base number 1 would be augmented by the reference text starting in column 2 and running to the end of each record in member JCLAUG in PDS JCL.AUG.TEXT.

It should be noted that the first AUGMENT statement which applies to a given item is used. So, if more than one AUGMENT could be applied to the same item(s), then the least inclusive statement should be specified first. For example, with the AUGMENT statements below:

AUGMENT VENDOR=IBM,
        PRODUCT=C*,
        ITEM=ADD,
        MEMBER=ADD1

AUGMENT VENDOR=IBM,
        PRODUCT=COBOL*,
        ITEM=ADD,
        MEMBER=ADD2

the second override would never be used because the first is more inclusive. Item 'ADD' in any product starting with the characters 'COBOL*' belonging to vendor 'IBM' would never be augmented by member ADD2 because the first override above would also apply to this item and the first override which applies is used.

If these statements were specified in reverse order, as shown below:

AUGMENT VENDOR=IBM,
        PRODUCT=COBOL*,
        ITEM=ADD,
        MEMBER=ADD2

AUGMENT VENDOR=IBM,
        PRODUCT=C*,
        ITEM=ADD,
        MEMBER=ADD1

then item 'ADD' in any product starting with the characters 'COBOL' belonging to vendor 'IBM' would be augmented by member ADD2 while item 'ADD' in any other product starting with the character 'C' belonging to vendor 'IBM' would be augmented by member ADD1.
REPLACE Statement

The REPLACE statement specifies an item to be replaced (or added, if it does not exist) and indicates where the "replacement" text is to be found. The "replacement" text will be automatically read in and used in place of the data base text for the indicated item. If the indicated item does not exist, then the REPLACE statement functions as an "add". The format of the REPLACE statement is:

```
REPLACE {VENDOR='vendor name',}
{PRODUCT='product name',}
{RELEASE='release number',}
ITEM=itemname,
MEMBER=membername,
{DSN=datasetname,}
{TEXTLEFT=n,}
{TEXTRIGHT=n,}
{UDBN=n,}
{TITLE='up-to-48-character-title'}
```

For a description of the REPLACE statement parameters other than the TITLE= parm, see the section on "Common Override Parameters" below.

It should be noted that, in order for a REPLACE override to function as an "add", the VENDOR=, PRODUCT=, RELEASE=, and ITEM= parameters must all be specified as full (i.e., non-generic) key values.

The TITLE= parameter is optional and should be specified only used when the REPLACE statement is being used to add a given item to a user data base and that user data base carries titles (i.e., only when the UDBN= parm is also specified and only when the TITLES=Y JCL parm was specified when the indicated user data base was created). In this case, the specified title functions like the titles (T=) operand on a key indicator record. (In other words, the specified title will be shown on any item selection list containing the associated item.) The specified title can be up to 48 characters long and must be enclosed in single quote marks; imbedded single quote marks must be represented as two consecutive single quote marks.

Here are some examples of valid REPLACE statements:

```
REPLACE VENDOR=CA,
PRODUCT='CA-PDSMAN MESSAGES',
RELEASE=7.1,
ITEM=FCO100I,
MEMBER=AUGF100,
TEXTLEFT=1,
TEXTRIGHT=72
```

In this case, the reference text for item 'FCO100I' in release '7.1' of product 'CA-PDSMAN MESSAGES' belonging to vendor 'CA' in the MVS/QuickRef main data base would be replaced by the reference text in columns 1 through 72 of member AUGF100 within the PDS specified by
the DATAPDS statement. If item 'FCO100I' does not already exist in the specified product, then, since the VENDOR=, PRODUCT=, RELEASE=, and ITEM= parameters were all specified as full (non-generic) key values, this REPLACE statement would be treated as an "add".

REPLACE VENDOR=IBM,
PRODUCT=IMS*,
ITEM=*,
MEMBER=IMSREP

In this case, the reference text for all items in all releases of all products with product names starting with the characters 'IMS' belonging to vendor 'IBM' in the MVS/QuickRef main data base would be replaced by the reference text in starting in column 2 and running to the end of each record in member IMSREP within the PDS specified by the DATAPDS statement.

It should be noted that the first REPLACE statement which applies to a given item is used. So, if more than one REPLACE could be applied to the same item(s), then the least inclusive statement should be specified first. For example, with the REPLACE statements below:

REPLACE VENDOR=IBM,
PRODUCT=C*,
ITEM=ADD,
MEMBER=ADD1,

REPLACE VENDOR=IBM,
PRODUCT=COBOL*,
ITEM=ADD,
MEMBER=ADD2

the second override would never be used because the first is more inclusive. Item 'ADD' in any product starting with the characters 'COBOL*' belonging to vendor 'IBM' would never be replaced by member ADD2 because the first override above would also apply to this item and the first override which applies is used.

If these statements were specified in reverse order, as shown below:

REPLACE VENDOR=IBM,
PRODUCT=COBOL*,
ITEM=ADD,
MEMBER=ADD2

REPLACE VENDOR=IBM,
PRODUCT=C*,
ITEM=ADD,
MEMBER=ADD1

then item 'ADD' in any product starting with the characters 'COBOL' belonging to vendor 'IBM' would be replaced by member ADD2 while item 'ADD' in any other product starting with the character 'C' belonging to vendor 'IBM' would be replaced by member ADD1.
PREVENT Statement

The PREVENT statement\(^3\) is used to identify an item which is NOT to be displayed if selected by a particular TSO user. The TSO user for which access is to be denied is identified by his TSO user ID or by his LOGON procedure name. PREVENT keywords can be used to prevent one or more TSO users from gaining access to information considered to be “sensitive” or not for wide distribution. PREVENT statements can also be used to keep one or more TSO users from obtaining DASD free space information for DASD volumes named by the VOLSER= parameter.

If an MVS/QuickRef user tries to view an item or volume serial to which a PREVENT statement applies, an “access denied” message is displayed on the screen and no information for the PREVENTed item or volume serial is displayed.

NOTE: THE PREVENT STATEMENT IS NOT SUPPORTED UNDER CA-ROSCOE OR IN BATCH.

The format of the PREVENT statement is:

PREVENT {VENDOR='vendor name',} {PRODUCT='product name',} {RELEASE='release number',} {ITEM=itemname | VOLSER=volser,} {UDBN=n,} {USERID=tsouserid | LOGPROC=logonprocname}

For a description of the various PREVENT statement parameters, see the section on "Common Override Parameters" below.

Here are some examples of the PREVENT statement:

<table>
<thead>
<tr>
<th>PREVENT</th>
<th>VENDOR=CA,</th>
<th>Product to prevent</th>
</tr>
</thead>
<tbody>
<tr>
<td>VENDOR='CA',</td>
<td>PRODUCT='CA-PDSMAN MESSAGES',</td>
<td>Release to prevent</td>
</tr>
<tr>
<td>PRODUCT=</td>
<td>RELEASE=7.1,</td>
<td>Item to prevent</td>
</tr>
<tr>
<td>CA-PDSMAN</td>
<td>ITEM=FCO100I,</td>
<td>TSO Userid to be prevented</td>
</tr>
<tr>
<td>MESSAGES,</td>
<td>VOLSER=FCO100I,</td>
<td></td>
</tr>
<tr>
<td>RELEASE=</td>
<td>USERID=ACT*</td>
<td></td>
</tr>
<tr>
<td>7.1,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM=FCO100I,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOLSER=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCO100I,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USERID=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACT*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For the PREVENT statement above, all TSO users with user id's starting with the characters 'ACT' would be prevented from viewing the reference information for item 'FCO100I' in release '7.1' of product 'CA-PDSMAN MESSAGES' belonging to vendor 'CA' in the MVS/QuickRef main data base.

PREVENT     VENDOR=CA, |
| PRODUCT=CA-PDSMAN* , | |
| RELEASE=*, | |
| ITEM=*, | |
| VOLSER=RSC* | |
| USERID=RSC* | |

\(^3\)Note: MVS/QuickRef supports an optional security exit that can also be used to permit or deny access to the information presented by MVS/QuickRef. See chapter three for more information on the user security exit.
For the PREVENT statement above, all TSO users with user id's starting with the characters 'RSC' would be prevented from viewing all items in all releases of any product with a product name starting with the characters 'CA-PDSMAN' from vendor 'CA' in the MVS/QuickRef main data base.

```
PREVENT VOLSER=PRI*,  Volume serial to prevent
  USERID=RS*            TSO Userid to be prevented
```

For the PREVENT statement above, all TSO users with user id's starting with the characters 'RS' would be prevented from viewing DASD free space information for DASD volume serials starting with the characters 'PRI'.

If you want to PREVENT an entire product, code the item name as a single asterisk. For example:

```
PREVENT  VENDOR=CA,
    PRODUCT='CA-PDSMAN MESSAGES',
    RELEASE=7.1,
    ITEM=*,
    USERID=ACT*
```

would PREVENT all TSO users with user id's starting with the characters 'ACT' from viewing the any item in release '7.1' of product 'CA-PDSMAN MESSAGES' belonging to vendor 'CA' in the MVS/QuickRef main data base. This effectively PREVENT's the entire product.

If you want to PREVENT an entire vendor, code the product name, release number, and item name as single asterisks. For example:

```
PREVENT  VENDOR=CA,
    PRODUCT=*,
    RELEASE=*,
    ITEM=*,USERID=C24*
```

would PREVENT all TSO users with user id's starting with the characters 'C24' from viewing any item in any release of any product belonging to vendor 'CA' in the MVS/QuickRef main data base. This effectively PREVENT's the entire vendor.

It should be noted that a PREVENT statement takes precedence over an ALLOW statement. (See section on the "ALLOW Statement" which follows for details).

**ALLOW Statement**

The ALLOW statement is used to identify an item that is to be displayed ONLY if it is requested by a particular TSO user. The TSO user for which access is to be allowed is identified by his TSO user ID or LOGON procedure name. ALLOW statements can also be used to allow one or more TSO users to obtain DASD free space information for DASD volumes named by the

---

4Note: MVS/QuickRef supports an optional security exit that can also be used to permit or deny access to the information presented by MVS/QuickRef. See chapter three for more information on the user security exit.
VOLSER= parameter. If an MVS/QuickRef user tries to view an item or volume serial information to which an ALLOW statement applies (and he is not one of the TSO users who is ALLOWed), a short “access denied” message is displayed on the screen and no information for the ALLOWed item or volume serial is displayed.

ALLOW keywords can be used to prevent many TSO users from gaining access to information considered to be “sensitive” or not for wide distribution. The ALLOW statement is the opposite of a PREVENT statement and is provided for cases where a small group of TSO users should be allowed access to certain information but most should not be.

**NOTE: THE ALLOW STATEMENT IS NOT SUPPORTED UNDER CA-ROSCOE OR IN BATCH.**

The format of the ALLOW statement is:

```
ALLOW {VENDOR='vendor name',} 
{PRODUCT='product name',} 
{RELEASE='release number',} 
{ITEM=itemname | VOLSER=volser,} 
{UDBN=n,} 
{USERID=tsouserid | LOGPROC=logonprocname}
```

For a description of the various ALLOW statement parameters, see the section on "Common Override Parameters" below.

The ALLOW entry below will permit all TSO users with a user id starting with the characters 'RSC' to view DASD free space information for DASD volumes whose volser start with the characters ‘USR’. However, no other TSO users will be able to review DASD free space information for these volumes.

```
ALLOW VOLSER=USR*,        Volume serial to allow
USERID=RSC*         TSO userid to be allowed
```

The ALLOW statements which follow let TSO users whose user id’s begin with the characters ‘ACC’ look at reference information for items FCO100I and FCO101I within release 7.1 of product 'CA-PDSMAN MESSAGES' from vendor 'CA' in the MVS/QuickRef main data base. However, no other TSO users will be able to view these items.

```
ALLOW VENDOR=CA,                  Vendor to allow
PRODUCT='CA-PDSMAN MESSAGES',  Product to allow
RELEASE=7.1,                Release to allow
ITEM=FCO100I,               Item to allow
USERID=ACC*                 TSO userid to be allowed
```

```
ALLOW VENDOR=CA,                  Vendor to allow
PRODUCT='CA-PDSMAN MESSAGES',  Product to allow
RELEASE=7.1,                Release to allow
ITEM=FCO101,               Item to allow
USERID=ACC*                 TSO userid to be allowed
```
It should be noted that a PREVENT statement takes precedence over an ALLOW statement. For example, given the PREVENT and ALLOW statements which follow:

```
PREVENT VENDOR=CA,
  PRODUCT='CA-PDSMAN MESSAGES',
  RELEASE=7.1,
  ITEM=FCO100I,
  USERID=U2401
ALLOW VENDOR=CA,
  PRODUCT='CA-PDSMAN MESSAGES',
  RELEASE=7.1,
  ITEM=FCO100I,
  USERID=U2401
```

userid U2401 would not be able to access item FCO100I in the specified product because the PREVENT statement takes precedence over the ALLOW statement.

It should also be noted that one ALLOW statement does not take precedence over another ALLOW statement. For example, with the ALLOW statements below:

```
ALLOW VENDOR=CA,
  PRODUCT='CA-PDSMAN MESSAGES',
  RELEASE=7.1,
  ITEM=FCO100I,
  USERID=U2401
ALLOW VENDOR=CA,
  PRODUCT='CA-PDSMAN MESSAGES',
  RELEASE=7.1,
  ITEM=FCO100I,
  USERID=U2402
```

both userid U2401 and userid U2402 would be allowed to access item FCO100I in the indicated product.

If you want to ALLOW an entire product, code the item name as a single asterisk. For example:

```
ALLOW VENDOR=CA,
  PRODUCT='CA-PDSMAN MESSAGES',
  RELEASE=7.1,
  ITEM=*,
  USERID=ACT*
```

would ALLOW only TSO users with user id's starting with the characters 'ACT' to view any item in release '7.1' of product 'CA-PDSMAN MESSAGES' belonging to vendor 'CA' in the MVS/QuickRef main data base. This effectively ALLOW's the entire product.

If you want to ALLOW an entire vendor, code the product name, release number, and item name as single asterisks. For example:

```
ALLOW VENDOR=CA,
  PRODUCT=*,
  RELEASE=*,
  ITEM=*,
  USERID=C24*
```
would ALLOW only TSO users with user id's starting with the characters 'C24' to view any item in any release of any product belonging to vendor 'CA' in the MVS/QuickRef main data base. This effectively ALLOW's the entire vendor.

Common Override Parameters

The various types of override statements share a number of common parameters. These include the ITEM=, VENDOR=, PRODUCT=, RELEASE=, and UDBN= parameters, which can be specified on any override statement other than the DATAPDS statement; the MEMBER=, TEXTLEFT=, and TEXTRIGHT= parameters, which can be specified on both the AUGMENT and REPLACE statements; the DSN= parameter, which can be specified on the DATAPDS, AUGMENT, and REPLACE statements; and the VOLSER=, USERID= and LOGPROC= parameters, which can be specified on both the PREVENT and ALLOW statements.

The ITEM= parameter specifies the item name of the item to be overridden. It may be specified as a full item name or a generic item name. For example, if specified as ITEM=FCO100I, the associated override would apply to item 'FCO100I'. If specified as ITEM=FCO*, the associated override would apply to all item names which begin with the characters 'FCO'. If specified as ITEM=*, the associated override would apply to all items within the specified product. The ITEM= parameter is required on the AUGMENT and REPLACE statements. Unless the VOLSER= parameter is specified, it is also required on the PREVENT and ALLOW statements.

The VENDOR=, PRODUCT=, and RELEASE= parameters specify the vendor name, product name, and release number of the item to be overridden. They should be used only in conjunction with the ITEM= parameter. These parameters must be enclosed in single quote marks if they contain embedded spaces and, if enclosed in single quote marks, then a single quote mark must be specified as two consecutive quote marks. Each of these parameters may be specified as a full name (or release number) or as a generic name (or release number). If, for example, the VENDOR= parameter is specified as VENDOR=IBM, the associated override would apply to vendor 'IBM'. If specified as VENDOR=I*, the associated override would apply to all vendor names which begin with the character 'I'. If specified as VENDOR=*, the associated override would apply to all vendors. If one of these parameters is omitted, then, if the ITEM= parameter is specified, it is treated as if specified as a single asterisk. For example, if ITEM= is specified but RELEASE= is omitted, then the RELEASE= parameter is treated as if specified as RELEASE=*.

Note: If you are using a REPLACE statement to logically add an item to the data base, then you must specify all four of the parameters VENDOR=, PRODUCT=, ITEM= and RELEASE=, so that MVS/QuickRef can correctly file the item in the data base index structure. Furthermore, all values specified must be non-generic, i.e., they must not end in an asterisk.

The UDBN= parameter specifies the user data base number of the user data base to be overridden. This parameter is described in the section which follows.
The **MEMBER=** parameter is a required parameter for both the AUGMENT and REPLACE statements. It specifies the member name of the PDS member which contains the augment or replacement text.

The **TEXTLEFT=** parameter is an optional parameter on the AUGMENT and REPLACE statements. If specified, it indicates the column number within each record of the specified PDS member where the augment or replacement text starts. If omitted, it defaults to column 2. For a fixed-length PDS, the first column of each record is column 1. For a variable-length PDS, the first column following the 4-byte RDW is column 1.

The **TEXTRIGHT=** parameter is an optional parameter on the AUGMENT and REPLACE statements. If specified, it indicates the column number within each record of the specified PDS member where the augment or replacement text ends. If omitted, the augment or replacement text will be considered to run up through the last column of each record. For a fixed-length PDS, the first column of each record is column 1. For a variable-length PDS, the first column following the 4-byte RDW is column 1.

The **DSN=** parameter is required on the DATAPDS statement and is optional on the AUGMENT and REPLACE statements. It specifies the data set name of the PDS where the member containing the augment or replacement text is to be found. The specified PDS data set name must be cataloged. The specified PDS may have RECFM F, V, FB, or VB, with any LRECL and BLKSIZE supported by the operating system. It may NOT be a Generation Data Group (GDG) data set, nor should the data set name include a member name specification. If the **DSN=** parameter is specified on an AUGMENT or REPLACE statement, then the PDS specified by that **DSN=** parameter will be searched for the member indicated by the **MEMBER=** parameter on that statement. If the **DSN=** parm is not specified on an AUGMENT or REPLACE statement, then the PDS specified by the **DSN=** parameter on the DATAPDS statement will be searched for the member indicated by the **MEMBER=** parameter on that statement.

The **VOLSER=** parameter can be specified on both the PREVENT and ALLOW statements. It specifies the volume serial of a DASD volume to be PREVENTed or ALLOWed when showing DASD free space information. It should not be used in conjunction with the **ITEM=, VENDOR=, PRODUCT=, or RELEASE=** parameters. It can be specified as a full volume serial or as a generic volume serial. For example, if specified as **VOLSER=PUB831**, then DASD volume serial 'PUB831' would be PREVENTed or ALLOWed. If specified as **VOLSER=PUB***, then all DASD volume serials starting with the characters 'PUB' would be PREVENTed or ALLOWed. The **VOLSER=** parameter can also be specified as **VOLSER=STORAGE, VOLSER=PUBLIC, or VOLSER=PRIVATE to PREVENT or ALLOW all DASD volumes mounted with the STORAGE, PUBLIC, or PRIVATE attributes. If you specify **VOLSER=*** on a PREVENT statement, no DASD volumes will be displayed for the user(s) that match the other criteria on that PREVENT statement.

The **USERID=** parameter can be specified on both the PREVENT and ALLOW statements. It specifies the TSO user id of the TSO user to be PREVENTed or ALLOWed. It is required if the **LOGPROC=** parameter (with which it is mutually exclusive) is not specified. It may be
specified as a full TSO user id or as a generic TSO user id. For example, if specified as USERID=RSC002, then TSO user id 'RSC002' would be PREVENTed or ALLOWed. If specified as USERID=RSC*, then all TSO user id's starting with the characters 'RSC' would be PREVENTed or ALLOWed.

The LOGPROC= parameter can be specified on both the PREVENT and ALLOW statements. It specifies the STEPNAME coded on the EXEC statement of the TSO LOGON procedure used by those TSO users to be PREVENTed or ALLOWed. It is required if the USERID= parameter (with which it is mutually exclusive) is not specified. It must be specified as the full (non-generic) stepname. For example, if specified as LOGPROC=RSC, then all TSO users who use a TSO LOGON proc with a STEPNAME of 'RSC' would be PREVENTed or ALLOWed.

**UDBN= Parameter**

The UDBN= (USER DATA BASE NUMBER) parameter is used to indicate that a given override applies to a user data base and to specify the user data base to which that override applies. It should be used only in conjunction with the ITEM= parameter. It should not be used if the override applies to an item in the MVS/QuickRef main data base.

The UDBN= parameter must specify a single numeric digit from 1 through 9 which indicates the user data base to which the override applies. If you are using a single user data base, then the UDBN parameter is always specified as:

```
UDBN=1
```

If you are using multiple user data bases, then the UDBN= parameter must be specified as a single numeric digit which corresponds to the QWIKOPTS LOCLDBn parameter which specifies the name of the user data base to which the override applies.

As an example, suppose the three QWIKOPTS LOCLDBn parameters shown below are used to define three user data bases:

```
LOCLDB1=USERDATA.BASEX,
LOCLDB2=USERDATA.BASEY,
LOCLDB3=USERDATA.BASEZ,
```

Then, to specify an override to be applied to USERDATA.BASEY, the following UDBN= parameter would be used:

```
UDBN=2
```

As another example of the UDBN= parameter, consider the following:

```
PREVENT VENDOR=RAMAC,
      PRODUCT=MACROS,
      RELEASE=5,
      ITEM=*,
```
This PREVENT statement would prevent users with TSO user id's starting with the characters 'RSC' from accessing any item in release '5' of product 'MACROS' belonging to vendor 'RAMAC' in user data base number 3.

As a reminder, if the PREVENT statement shown above were specified without the UDBN= parameter, i.e.,

```
PREVENT VENDOR=RAMAC,
    PRODUCT=MACROS,
    RELEASE=5,
    ITEM=*,
    USERID=RSC*
```

then the PREVENT statement would apply to the MVS/QuickRef main data base instead of a user data base.

**Validating Your Override Parameters**

After creating a set of override parameters for use at your installation, you should request MVS/QuickRef to syntax check them to ensure that they are valid and will take affect. A CLIST is provided for you to use in performing a syntax check of your override parameter set. The CLIST is named QWTEST and is located in the MVS/QuickRef JCL library. The CLIST must be edited before use; if you are using the LIBDEF facility to invoke MVS/QuickRef, be sure to change the data set names used on the LIBDEF statements in the QWTEST CLIST to match those used for the MVS/QuickRef data sets at your site.

Now invoke the QWTEST CLIST. QWTEST will prompt for a single parameter, the name of the PDS in which the QWPARM00 member containing the override parameters resides.

The PDS data set name needed by QWTEST can also be passed as a single positional parameter when the CLIST is invoked, as follows:

```
EX 'mvs.quickref.jcl(QWTEST)' 'LOCAL.PARMS.PDS'
```

When the QWTEST CLIST executes, it will cause MVS/QuickRef to syntax check the parameters in the QWPARM00 member, but no further processing will be performed. Messages indicating the success or failure of the parameter checking process will be displayed for your review.

Once your override parameters have been validated, define your override PDS data set name to MVS/QuickRef, using the instructions outlined in the next section.
Defining the Parameter Data Set Name

Once your override parameters have been syntax checked and found to be valid, they should be placed into production. To do this, you must define the name of the PDS containing the QWPARM00 member to MVS/QuickRef.

You can do this by preallocating the PDS in your TSO logon procs, using a DD name of QWPARMS, as shown below:

```
//QWPARMS DD DSN=SYS2.QUICKREF.PARMS,DISP=SHR
```

If you prefer not to preallocate it, you can define the data set name of the PDS containing the QWPARM00 member using the MVS/QuickRef options table, QWIKOPTS. To do this, you must edit the QWIKOPTS member in the MVS/QuickRef JCL library and use the PARMDSN= keyword to specify the name of the override parameter PDS. Examples and syntax for the PARMDSN= keyword of the QWIKOPTS table can be found on page 99 of this guide. Once this change is made, you can use member QWOPTASM in the MVS/QuickRef JCL library to assemble and link the QWIKOPTS table.

If you are using the LIBDEF facility to invoke MVS/QuickRef, then you can also define the name of the override parameter PDS by uncommenting and customizing the ALLOC and FREE statements for the QWPARMS DD statement in the QW LIBDEF CLIST. See the QW LIBDEF CLIST referred to in the “Alternative MVS/QuickRef Invocation Methods” section of chapter four for an example.
MVS/QuickRef Message Descriptions

MVS/QuickRef issues a number of error or warning messages. These messages are generally formatted as QWIKMnnn, where the ‘nnn’ represents a sequence number.

All MVS/QuickRef error and warning messages are described in the MVS/QuickRef main data base. The easiest way to look up the description of an MVS/QuickRef error or warning message is to use MVS/QuickRef itself (just like you would for any other type of error message). For example, to look up MVS/QuickRef message QWIKM060, just type 'QW QWIKM060' on the command line of any ISPF panel.

Of course, there may be times when you need to look up an MVS/QuickRef message description and, for whatever reason, MVS/QuickRef is not working correctly or is not available. In this case, browse member QWIKMSGS in the MVS/QuickRef JCL library. This member is the MVS/QuickRef main data base input source member for the messages issued by MVS/QuickRef. This means that it contains the same MVS/QuickRef message descriptions that appear in the data base (along with the key indicator records required by the data base build program). Once you are browsing this member, you can use the FIND command to find any message description you may need.
Appendix B - MVS/QuickRef User Abends
MVS/QuickRef User Abend U820

Certain MVS/QuickRef programs issue a user abend (U820) when conditions are detected that prevent further processing. When this user abend is issued, a message describing the error will be written to the failing job’s JES job log and/or the SYSPRINT DD statement. Be sure to look at the messages that may appear in these areas for assistance in resolving the problem.

If you cannot resolve the problem, have the dump that was produced available when you contact the MVS/QuickRef Support Group. If no dump was produced, execute the job again with a SYSUDUMP DD statement present so that a dump will be produced.
User Feedback Form
User Feedback Form

The MVS/QuickRef development team is committed to making the product as useful and powerful as possible. Please use this form to provide us with your suggestions on how to improve or extend the product, or to report an error or product deficiency. You can FAX the completed form to (773) 282-9036 or call us at (773) 282-4777 to speak to a member of the product support group. User feedback can also be sent via e-mail to support@quickref.com.

<table>
<thead>
<tr>
<th>Name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Voice telephone number:</td>
</tr>
<tr>
<td>Fax telephone number:</td>
</tr>
<tr>
<td>Email address:</td>
</tr>
<tr>
<td>Comment/Problem/Suggested enhancement:</td>
</tr>
</tbody>
</table>
Index
%> pointer · 154

A

accessing MVS/QuickRef · 22–30
alias indicator records · 141
aliases · 141
allocating MVS/QuickRef data sets, SMP/E install method · 75
allocating SMP/E data sets, SMP/E install method · 76
ALLOW statement · 188, 194
alternative invocation methods · 95
AMASPZAP · 91, 109
assembler language reference data · 14
AUGMENT statement · 188, 189
AUTOFC= option · 104
AUTOLUP= option · 104
automatic FINDCODE · 47
   turning on/off via QWIKOPTS · 104
automatic lookup · 47
   based on pointers · 47
   based on text content · 47
   implementing in user data bases · 153
   turning on/off via QWIKOPTS · 104

B

batch commands · 59
   errors · 62
   examples · 61
   return codes · 62
batch execution · 57
   examples · 58
   return codes · 59
BATPDSRD= option · 105

C

C language syntax · 18
CA-OPS/MVS II · 95
CA-ROSCOE · 69, 130
   considerations · 132
ETSO · 130
  interface for MVS/QuickRef · 131
CA-SYSVIEW/E · 95
category code · 22
category code indicator records · 148
category codes
definition syntax · 149
description · 148
examples · 150
CDIXLAT · 103
CHRXLAT · 102
CHRXTAB · 102
CICS reference information · 15
COBOL language syntax · 18
commands
  END · 30
  FIND · 32
  GETNEXT · 32
  GETPREV · 32
  HELP · 30
  invocation · 24
  QINFO · 32
  QW · 24
  QWS · 24
  RETURN · 30
  SEARCH · 32
  SORT · 32
common override parameters · 197
console invocation · 48
  examples · 48
contents of data base · 12
copyright indicator records · 144
correcting main data base contents · 128
cursor-driven invocation · 28
troubleshooting · 113
customizing
data base · 48
  MVS/QuickRef · 96
cut and paste facility · 32

D

DASD free space display · 26
  fast-path invocation · 28
DASD space requirements · 65
DASDFRE= · 101
data base
  allocation sequence · 99
  contents · 12
  correcting text of items within · 128
  customization · 48
  item · 22
  merge facility · 116
  merge facility JCL · 119
  number DASD extents · 65
  number DASD volumes · 65
  product · 22
  refreshing · 116
  selective load · 120
data base name
  defining · 78, 86
  defining via non SMP/E method · 86
DATAIN · 138
DATAIN DD statement · 138
DATAIN2 · 139
DATAIN2 DD statement · 139
DATAPDS statement · 188
DB2 messages · 18
defining main data base name · 78, 86
  via non-SMP/E method · 86
direct program call · 48
  considerations · 52
  example · 54
  parmlist · 49
  parmlist for releases prior to 5.3 · 54
  return codes · 53
display request panel · 25
DSN= parameter on override statement · 198

E
END command · 30
error messages · 204
ETSO, executing QuickRef under · 130
EXCLUDE statement format · 123
exiting MVS/QuickRef · 30
expiration date extension zap · 109
  applying via non-SMP/E method · 91
expiration date zap
  applying · 83
  overview · 109
  troubleshooting · 112
fast-path
  for DASD free space · 28
  for item · 26
  for product · 27
  for product category code · 27
  for vendor · 27
  invocation · 26
  string · 26
features
  of MVS/QuickRef · 32
  on-line · 32
FIND command · 32
FINDCODE command · 32
FINDCODE processing · 47
FMIDs for ISPF · 81
FOLD= option · 104
full key · 22

GDG · 136, 159
generation data group · See GDG
generic key · 22
GETNEXT
  implementing in user data base · 151
  troubleshooting · 113
GETNEXT command · 32, 61
GETPREV command · 32, 61

HELP command · 30, 61
  troubleshooting · 111
help facility · 30
HELPDD= · 101
hot key invocation · See cursor-driven invocation

IMS reference information · 16
INCLUDE statement format · 123
INCLUDE/EXCLUDE control statements · 122
independent software vendor product messages · 19
information storage and retrieval overview · 22
initializing SMP/E data sets, SMP/E install method · 76
installation assistance dialog · See QWIAD
installation checklist, QWIAD · 71
installation steps · 70
  Method 1 - QWIAD · 71
  Method 2 - manual SMP/E · 75
  Method 3 - manual non-SMP/E · 85
  Method 4 - data base only · 92
installing MVS/QuickRef · 65
  background material · 69
  initial tape unload JCL · 70
invocation
  alternative methods · 95
  batch · 57
  commands · 24
  console · 48
  cursor-driven · 24
  fast-path · 26
  menu-driven · 25
  recursive · 32
  techniques · 24
invoking QWIAD · 74
ISPF command table
  description · 94
  modifying · 66, 79
  modifying, non-SMP/E method · 87
  overview · 94
ISPF FMIDs · 82
ISV
  list · 19
  product messages · 19
item
  in data base · 22
  name · 22
item name
  on override statement · 197

J

JCL reference data · 13
JCL, unloading · 70
JES2 JECL · 13
JES2CHR= · 103
JES3 JECL · 13

K

KCAD record syntax · 149
KCAP record syntax · 149
key
  full · 22
  generic · 22
  matching · 23
key indicator records · 139

L

LIBDEF · 95
  troubleshooting · 113
  using · 96
List Vendors/Products/Releases panel · 25
LOCLDB1=
  in QWIKOPTS options table · 105
LOCLDB2=
  in QWIKOPTS options table · 105
LOCLDB3=
  in QWIKOPTS options table · 105
LOCLDB4=
  in QWIKOPTS options table · 105
LOCLDB5=
  in QWIKOPTS options table · 105
LOCLDB6=
  in QWIKOPTS options table · 105
LOCLDB7=
  in QWIKOPTS options table · 105
LOCLDB8=
  in QWIKOPTS options table · 105
LOCLDB9=
  in QWIKOPTS options table · 105
LOGPROC= parameter on override statement · 199
lookup pointer · 154

M

main data base
  updating · 115
main menu · 25
  setting default name of, via QWIKOPTS · 100
MAINMNU= · 100
member selection statements
  for user data base · 155
MEMBER= parameter
  on override statement · 198
menu-driven invocation · 25
merge facility · 116
merging data from previous release · 128
messages · 204
multiple user data bases · 135
MVS
  messages and codes · 12
  utilities · 14

N

NOTRANS= parameter · 137

O

OLDKEYS= parameter · 137
ONLPSRD= option · 106
operating system level selection list processing · 36
  setting up via QWIKOPTS · 103
OPS/MVS · 95
options table · 97
OS/390 messages and codes · 12
OSLSD= option · 103
OUTCLAS= · 99
OUTDEST= · 99
OUTDISP= · 100
override parameters · 186
  ALLOW statement · 194
  AUGMENT statement · 189
  common · 197
  DATAPDS statement · 188
  defining data set name · 201
  overview · 186
  PREVENT statement · 193
  REPLACE statement · 191
  syntax · 187
  usage example · 186
  validation · 200
overview
  of MVS/QuickRef · 10
of this guide · 9

P

PARMDSN= · 99
PF key troubleshooting · 113
PF3= · 101
philosophy behind MVS/QuickRef · 11
PL/1 language syntax · 18
preallocating main data base via QWREFDD DD statement · 86
predisplay analysis · 47
preprocessed panels · 92
PREVENT statement · 188, 193
printing capabilities · 32
PRODCOPY · 91
product
  category code selection list · 25
  in data base · 22
product category code · 22, See Also category code
product name
  on override statement · 197
production implementation · 114
program call · 48

Q

QDB= · 98
QEXCLUDE member · 78, 86, 124
QINFO command · 32, 61
QPRINT
  setting default output class via QWIKOPTS · 99
  setting default output destination via QWIKOPTS · 99
  setting output default HOLD disposition via QWIKOPTS · 100
QPRINT command · 61
QW CLIST · 96
QW command · 26
  alternative to use of · 95
  troubleshooting · 112
QWC started task · 48
QWCMDS CLIST · 79, 87, 94
QWDC1EYE · 49
QWDC1REL · 49
QWDC2FPA · 49
QWDC2ITM · 51
QWDC2LL · 50
QWDC2PRD · 51
QWDC2RAL · 50
QWDC2REL · 51
QWDC2SAA · 50
QWDC2TIR · 51
QWDC2VEN · 51
QWIAD · 71
  installation checklist · 71
  invoking · 74
QWIEXPDT · 84
QWIISPFS · 81
QWIKCMDS · 94, 96, 111
QWIKDPC1 copy member · 49
QWIKDPC2 copy member · 49
QWIKEEXIT · 56
QWIKMRGE · 119, 129
QWIKMRGJ · 119, 129
QWIKOPTS · 97
  member in JCL library · 98
QWIKREF2 return codes · 139
QWIKREFA
  as setting for MAINMNU= option in QWIKOPTS · 100
  possible errors · 114
QWIKREFB panel · 97
QWIKREFC panel · 92
QWIKREFE panel · 92
QWIKREFM panel · 164
QWIKREFU
  as setting for USERMNU= option in QWIKOPTS · 100
  panel · 161
QWIKSLCT
  control statement format · 122
  examples of use · 124
  JCL for · 121
  when refreshing main data base · 116
QWIKVPSA · 107
QWIKVPSC · 107
QWIKVPST · 39
  customizing · 106
QWINUOPT · 98
QWLOAD · 85
QWOPTASM · 87, 98
QWPARM00 · 187, 201
QWPARMS · 201
QWPASTE CLIST · 96
QWREFDD DD statement · 86, 137
QWS command · 24
  troubleshooting · 112
QWTEST · 200
QWUPDATE · 93, 116
  for data base refresh · 116
  SMP/E install method · 77
QWUSER · 136

R

recursive invocation · 32
release number
  on override statement · 197
  specifying via user-directed selection list processing · 33
REP statement · 109
REPLACE statement · 188, 191
  used to correct data base items · 128
Request DASD Free Space Information panel · 26
Request Reference Information panel · 25
restoring previous display · 30
return codes
  batch command · 62
  direct program call · 53
RETURN command · 30
REXX language syntax · 18
ROSCOE · See CA-ROSCOE
ROUTCDE= · 100

S

SEARCH command · 32, 61
security exit · 55
security system considerations · 95
selection list · 23
Selection List Profile panel · 41
selective data base load facility
  control statement format · 122
  example of use · 124
  JCL · 121
  overview · 56
  using · 120
  vs user-directed selection list processing · 126
site command table
  modifying · 79
  modifying non SMP/E method · 87
SLP command · 41
SLPROFILE command · 41
SMP/E ACCEPT function, SMP/E install method · 77
SMP/E APPLY function, SMP/E install method · 77
SMP/E RECEIVE function, SMP/E install method · 76
SMPALLOC · 75
SMPAPPLY · 76
SMPCSI · 76
SMPDDDEF · 76
SMPRAP2 · 82
SMPRAP4 · 84
SMPRECV · 76
SORT command · 32, 61
space requirements · 65
split-screen invocation · 24
SQL return codes · 18
start access based on content indicator records · 151
SYSHELP
  DD name · 101
  security system considerations · 95
SYSIN DD statement · 139
system abends · 12
system messages · 12
SYSVIEW/E · 95

T

tech support phone numbers · 208
terminating MVS/QuickRef · 30
testing MVS/QuickRef · 110
  preparation for · 110
text stored in user data base · 143, 144
TEXTLEFT= parameter · 136
  on override statement · 198
textmark facility · 32
TEXTRIGHT= parameter · 136
  on override statement · 198
TEXTSTR= · 100
time stamp · 117
TITLE= parameter
  on REPLACE statement · 191
TITLES= parameter · 137
translation of special characters, setting via QWIKOPTS · 102
troubleshooting · 111
tutorial for MVS/QuickRef · 22
typographical errors in a data base item · 127
U

U820 · 206
UDBN= parameter on override statement · 199
UISLOPT= option · 102
unload MVS/Quickref JCL · 70
update data bases · 116
  dummy info product · 118
  special processing information · 118
  time stamp · 117
updating MVS/QuickRef data base · 115
  UPDATE DATABASE INFO dummy product · 118
upgrading · 67
usage examples for MVS/QuickRef · 11
user abends · 205
user data base
  allocating and loading · 136
  creation example 1- single · 171
  creation example 2- multiple · 179
  defining name(s) · 159
  implementation steps · 135
  input file · 138
  input file considerations · 155
  JCL · 136
  JCL examples · 156
  JCL parms · 136
  limit on item size · 144
  member selection statements · 155
  overview · 134
user data bases
  assigning data base number · 159
  deciding how many · 135
user documentation · 114
user feedback form · 208
user menu panel
  adding option U to main menu · 166
  customizing · 161
  determine need for · 160
  setting default name of, via QWIKOPTS · 100
user specific selection list processing · 41
user-directed selection list processing · 33
  example of · 45
  general rules · 34
  order of precedence · 44
  setting up via QWIKOPTS · 103
  types of · 36
vs. selective load · 126
USERID= parameter on override statement · 198
USERMNU= · 100
USERSTR= · 100
using the help facility · 30
utilities reference data · 14

V

vendor name
  on override statement · 197
Vendor/Product Specific Selection List Copy Member · 107
vendor/product specific selection list processing · 39
    setting up via QWIKOPTS · 104
Vendor/Product Specific Selection List Table · 39
    customizing · 106
VER statement · 109
virtual storage for retrieved text, setting size via QWIKOPTS · 100
VOLSER= parameter on override statement · 198
volume serial
    change entry requirement · 97
    for DASD free space display · 28
    on override statement · 198
VPSSLO= option · 39, 104

W

what's new for this release · 9
WTO
  ROUTCDE used · 48
    setting default routing code via QWIKOPTS · 100

Y

YSHELP DD · 52

Z

z/OS messages and codes · 12
zapping MVS/QuickRef · 109