

# MVS/QuickRef – Beyond the Basics of QW

Session Number: **11583**

Date of Presentation: **August 6, 2012**

## **[PowerPoint Page 01 – Title Page]**

## **[PowerPoint Page 02 – Requesting Users with Experience]**

At a recent Technical Expo, I was called upon to answer a couple of questions from a person who was a current MVS/QuickRef client. He was asking me questions along the lines of “why can’t you do ....” in MVS/QuickRef. As it turns out, you can, in fact, do everything the client was asking for. I really did enjoy giving the client the good news.

So, this session was designed to help MVS/QuickRef users know what features are available and how to use them. This includes some of the basic features, but mainly features that are “beyond the basics”. I talked to my fellow product developers and came up with the Top 12 things you might not know about Q/R. I think knowing about these features will make you more productive.

## **[PowerPoint Page 03 – Presentation Topics 1 of 2]**

## **[PowerPoint Page 04 – Presentation Topics 2 of 2]**

For those of you who find themselves in the wrong session, or came in on purpose to find out what MVS/QuickRef is, or had no other session to attend and find yourself here, I want to start with an introduction about MVS/QuickRef. I’ll use a PowerPoint type of presentation for this session as this will be something you can later download and review; in addition, this allows me to be able to click to the next screen.

Everyone, at some time or another, has a need to lookup reference information pertaining to z/OS. This may be an ABEND code or a message you’ve not seen before. It can also be syntax for JCL or a programming language. It may be that you need the JCL for a function that you’ve not done in awhile...such as re-initialize a DASD volume. Maybe you need to look up a message for a software product that does not come with the operating system – for example, a message produced by SYNCSORT or some other independent software vendor.

## **[PowerPoint Page 05 – Ways to Find z/OS Reference Information]**

## MVS/QuickRef – Beyond The Basics of QW

Now, I'm sure we all have different ideas on how to find these messages and information. Some may simply ask someone else. Some may actually look for a printed manual (yes, those once existed, and sometimes an out-of-date copy is hidden in the systems programmers desk). Some may search for the information on the internet (Google, anyone?). And, some may visit IBM's z/OS library web site. But, if you are here for this presentation, you most likely know to look for this information in the MVS/QuickRef product from Chicago-Soft, Ltd.

Let's take a moment and get this out of the way. People come by the Chicago-Soft booth at the Technology Expo all the time and have for quite a number of years now. They ask 1) What is Chicago-Soft? and 2) What is MVS/QuickRef? For some reason, they appear to have never heard of our product. We've learned to say "Chicago-Soft makes the QW command". Okay, crisis solved; almost everyone has heard of the QW command.

**[PowerPoint Page 05 – (second click)]**

The typical way novice users invoke MVS/QuickRef is by typing the QW command on the command line and pressing ENTER. This gives the following screen which we refer to as the Main Menu.

**[PowerPoint Page 06 – 06 - Main Menu.jpg]**

- If you find yourself here in error, hit **PF3** or type END and press ENTER.
- If you want to know about the changes between this release of MVS/QuickRef and the previous release, type **?** and press ENTER.
- If you want to look for DASD Free Space, type **S** and press ENTER.
- If you want to list the vendors, products, and releases of information in MVS/QuickRef, type **L** and press ENTER.
- If you want to lookup information or, as we call them, items, type **R** and press ENTER.
- If you want to lookup categories of information, such as the TOOLBOX, type **C** and press ENTER. We will come back to this later.

**[PowerPoint Page 07 - Best Ways to Invoke MVS/Quick-Ref]**

## MVS/QuickRef – Beyond The Basics of QW

Of course, power users seldom, if ever, use the menus. Many will use what we refer to as fastpath commands. Others will use what we refer to as cursor-driven-invocation.

**[PowerPoint Page 07 – (second click)]**

First, let's look at fastpath.

**[PowerPoint Page 08 - 08 - Fast Path.JPG]**

Fastpath commands are entered on the command line from any ISPF screen.

**[PowerPoint Page 09 - Best Ways to Invoke MVS/Quick-Ref]**

Now, let's look briefly at cursor-driven-invocation.

**[PowerPoint Page 09 – (second click)]**

**[PowerPoint Page 10 - 10 - CDI.JPG]**

Cursor-driven-invocation is usually done when you are looking at some type of report – the JES2 Job Log using SDSF. The basic idea is to type QW on the command line, move the cursor to an item you wish to lookup in the database, and press ENTER.

As an alternative, most MVS/QuickRef power users set an ISPF Program Function Key (for our session today, I'll be using PF23) to QW.

**[PowerPoint Page 11 – Tip on Setting the PF Keys]**

Then, you put the cursor on the item you wish to lookup and press PF23. You can use the KEYS command to set your PF keys.

**[PowerPoint Page 12 - 12 KEYS.JPG]**

Okay, to summarize what we've seen thus far, we can use the QW menus to find items, use fast path, or use cursor-driven-invocation. Before we look at specific scenarios of how Q/R can be used, let's start with a Pop Quiz.

**[PowerPoint Page 13 - Quiz 1]**

# MVS/QuickRef – Beyond The Basics of QW

How many items are there in the MVS/QuickRef database (as distributed in Release 7.6)?

**[PowerPoint Page 13 (second click) Quiz 1 answers]**

Total number of items in data base: 4,428,037

Why do I bring this up? To show there is an enormous amount of reference data available. With this much information, you must be using the correct location techniques or you will never find what you are looking for.

Now, let's look at those specific scenarios of how Q/R can be used.

## 1 - Using Cursor-driven-invocation from within SDSF LOG function

**[PowerPoint Page 14 – Using Cursor-driven-invocation from within SDSF LOG function]**

When you think of how to use MVS/QuickRef, the thought that probably comes to mind is to look up a message.

**[PowerPoint Page 14 (second click)]**

An application programmer would likely get a call saying a batch-oriented production job failed. He or she would use SDSF to look at the JES2 Job Log to see the problem and discover a weird message they have never seen before.

- In this scenario, we'll see how cursor-driven-invocation can be used to lookup an ABEND code. We have a batch job to assemble and execute an assembler program for an assemble, link and Go process. The program is rather simple: we establish regular program entry code, GETMAIN a data area, do some processing, and then exit. However, when we submit the program, we see it ABENDs with an S80A ABEND code.

**[PowerPoint Page 15 – 15 - 80A.JPG]**

## MVS/QuickRef – Beyond The Basics of QW

- We press PF9 to swap to the other screen and enter SD,H to invoke SDSF to look at the job output. In the JES2 job log, we see the ABEND code 80A with a REASON CODE of 10.

### [PowerPoint Page 16 – 16 – Q/R searches for you]

- We start the lookup by placing the cursor on the abend code and press the Shift-PF11 (PF23) to execute the QW command to bring up MVS/QuickRef. Using cursor driven invocation, MVS/QuickRef locates the text the cursor is on and searches the MVS/QuickRef database for all items that match.

### [PowerPoint Page 17 – 17 - 80A Selection List.JPG]

From the selection list, let's choose the item from z/OS SYSTEM CODES.

- Notice that, in addition to the ABEND CODE, MVS/QuickRef sets up a FINDCODE subcommand (abbreviated FC) to issue a FIND command for the REASON CODE.

### [PowerPoint Page 18 – 18 – FINDCODE.JPG]

- All we need to do is press ENTER. We'll talk about this later.

### [PowerPoint Page 19 – 19 – RC 10.JPG]

Reason Code 10 says “There is not enough virtual private area storage available to satisfy the request.” Thus, a GETMAIN has tried to obtain storage but is requesting more virtual storage than is available. We press PF4 to exit out of MVS/QuickRef with a single keystroke. We are still in SDSF, so we press PF9 to return to the EDIT screen. Looking for a GETMAIN, we find one in the program. The program needs 480,000 bytes but the GETMAIN requested 4,800,000. We correct the assembler program and resubmit it again.

## 2 - Using Q/R from within EDIT to lookup JCL syntax

### [PowerPoint Page 20 – Using Q/R from within EDIT to lookup JCL syntax]

## MVS/QuickRef – Beyond The Basics of QW

- Here, we'll see how MVS/QuickRef can be used to assist us with our JCL. We have the QWLOAD Job which is used to install MVS/QuickRef. For this job, let's put a COND parameter on step 2 to prevent the step from executing unless the first step ends with a return code of 0.
- For me, saying “the COND parameter is a little tricky” is an understatement. I've been using MVS JCL for 34 years, and I still have trouble remembering how COND is coded. Let's look up the syntax to be sure we code it correctly. To do this, let's first type COND= and save the JCL.

### [PowerPoint Page 21 - 21 - QW in EDIT of QWLOAD.JPG]

- Now, let's put the cursor on the newly entered COND= parameter and hit Shift-PF11 (PF23) to invoke cursor driven invocation. MVS/QuickRef displays all COND items in the MVS/QuickRef database. Let's select the item for the JCL SYNTAX EXEC.

### [PowerPoint Page 22 - 22 - COND Selection List.JPG]

- The displayed text tells us that z/OS will execute the step if the COND= evaluates to FALSE. A TRUE condition causes the step to be skipped. Since we want to execute only if the step 1 return code is 0, we determine that COND=(0,NE) is the correct syntax.

### [PowerPoint Page 23 - 23 – Facts on COND.JPG]

- We press PF4 to exit MVS/QuickRef, finish typing the COND parameter.

## 3 - Using DASD Free Space report

### [PowerPoint Page 24 – Using DASD Free Space report]

## MVS/QuickRef – Beyond The Basics of QW

- In this example, we'll see how MVS/QuickRef can again be used to assist us with our JCL, but this time it will be in finding available DASD Free Space. We have the QWUPDATE Job which is used during an MVS/QuickRef database-only install. When we look at the JCL, we see that the R7.6 database requires 28,946 tracks.

### [PowerPoint Page 25 - 08 - Fast Path.JPG]

- We want to load the database onto any DASD volume that begins with TSO\* (such as TSO001, TSO002, etc.). We jump to the EDIT command line, and enter QW S=TSO\*. This will show all volumes that begin with the wildcard TSO\*.

### [PowerPoint Page 26 - 26 – The Unsorted TSOonn Volumes.JPG]

- The volume list, however, is sorted by volser. We want to resort the list in decreasing sequence by the available tracks. So we enter “SORT 5 D” to sort column 5 in descending order.

### [PowerPoint Page 27 - 27 - The Sorted TSOonn Volumes.JPG]

- We choose TSO011 and press PF4 to exit MVS/QuickRef and return to ISPF edit. To specify the volume, we must change the DVOL parameter.

## 4 - Using Cursor-driven invocation from within EDIT on COBOL source

### [PowerPoint Page 28 – Using Cursor-driven invocation from within EDIT on COBOL source]

- In this example, we'll see how MVS/QuickRef can be used by a COBOL programmer to lookup the COBOL syntax. We have a COBOL program which is not working correctly. After analyzing the program, we determine the PERFORM verb is executing a single procedure when it should in fact be executing multiple procedures. Putting the cursor on the PERFORM

# MVS/QuickRef – Beyond The Basics of QW

verb, we press Shift-PF11 (PF23) to launch MVS/QuickRef and bring up a selection list of all the PERFORM items in the database.

## [PowerPoint Page 29 - 29 - Perform Selection List.JPG]

- We choose the item from the “COBOL Z/OS SYNTAX” product. MVS/QuickRef is used to verify the correct syntax for "PERFORM procedure-1 THRU procedure-2". Press PF4 to exit MVS/QuickRef and to return to ISPF EDIT. The COBOL program is then modified with the "enhanced" PERFORM statement.

## [PowerPoint Page 30 - 30 - Perform Text.JPG]

## 5 - Using Cut & paste from the TOOLBOX

### [PowerPoint Page 31 – Using Cut & paste from the TOOLBOX]

- In this example, we’ll see how the MVS/QuickRef TOOLBOX can be used to find the proper JCL for the job. We need to compress a partitioned dataset. When we look at the JCL, we determine the program executed is IEBGENER instead of IEBCOPY. That is, someone has clobbered the COMPRESS JCL.

### [PowerPoint Page 32 - 32 - Clobbered Compress JCL.JPG]

- We put the cursor on a blank section of the screen so we won’t launch cursor driven invocation and hit the QW key. The main menu of MVS/QuickRef is displayed.

### [PowerPoint Page 33 - 33 - Request Categories.JPG]

- We type “C” to see the categories of the information in the database.

## MVS/QuickRef – Beyond The Basics of QW

### [PowerPoint Page 34 - 34 – List of Categories.JPG]

- From this list, we select the TOOLBOX.

### [PowerPoint Page 35 - 35 – TOOLBOX table of contents.JPG]

- We then look at the .TBL-OF-CONTENTS item to locate the IEBCOPY-COMPRESS item which we then select.

### [PowerPoint Page 36 - 36 – FIND COMPRESS.JPG]

- From here, we use cursor-driven-invocation to see the TOOLBOX information.

### [PowerPoint Page 37 - 37 – CUT ALL.JPG]

- On the command line, we enter the MVS/QuickRef subcommand CUT ALL. This copies the JCL to the scratch pad.

### [PowerPoint Page 38 - 38 – !PASTE.JPG]

- We press PF4 to exit MVS/QuickRef and to return to ISPF EDIT. On the EDIT command line, we specify an MVS/QuickRef distributed edit macro, !PASTE. This subcommand copies the contents of the scratch pad into the member.

### [PowerPoint Page 39 - 39 - PASTED RESULTS.JPG]

- The JCL is tailored and submitted to perform the compress.

# MVS/QuickRef – Beyond The Basics of QW

## 6 - Using Dataset List report

### [PowerPoint Page 40 - Using Dataset List report]

Every once in awhile, probably in non-SMS environments, you have a situation where you want to see the data sets that are on a specific volume serial number. I recently was running a job to create a dataset but got a “NOT CATLG 2” message – if you’ve been around a long time, you’ve probably seen that one too. It looks like this:

### [PowerPoint Page 41 - 41 - NOT CATLGD.JPG]

In a nutshell, it says, I tried to catalog this dataset but it was already cataloged to a different volume. Usually, this triggers a search to see all the volumes where that dataset may reside. You can use MVS/QuickRef to help search. This time, we’ll use the DASD Free Space function and the Dataset List function.

- In this example, we’ll see how MVS/QuickRef can be used to assist us with locating datasets.
- I will assume the datasets can be found anywhere on the twelve volumes TSO900-TSO911 and will use fastpath to launch the DASD Free Space report using:

```
QW  S=TSO9*
```

### [PowerPoint Page 42 - 42 - DFS DISPLAY.JPG]

- To list the datasets on a volume, place the cursor on the volume and press ENTER.

### [PowerPoint Page 43 - 43 - DSN List Display.JPG]

### [PowerPoint Page 44 – List of DSN List Fields – 1 of 2]

- Features of the Dataset Name List include the following fields:

## MVS/QuickRef – Beyond The Basics of QW

- o dataset name (DSN)
- o allocated tracks (ALLOC TRACKS)
- o free tracks (FREE TRACKS)
- o number of extents (EXT)
- o allocation amount of first extent (1ST EX ALLOCN)
- o allocation amount of successive extents (SECOND ALLOCN)
- o allocation units (ALC UNT)
- o dataset organization (DSORG)
- o indicator if Hierarchical File System (HFS) or Partitioned Data Set Extended (POE)

### [PowerPoint Page 45 – List of DSN List Fields – 2 of 2]

- o dataset recording format (RECFM or RFM)
  - o dataset logical record length (LRECL)
  - o dataset block size (BLKSIZE or BLKSI)
  - o dataset creation date (CR. DATE)
  - o dataset Last Referenced Date (LAST REF)
  - o For EAV-only, EAV Attributes (EATTR)
  - o For EAV-only, jobname of last updating job (JOBNAME)
  - o For EAV-only, stepname of last updating step (STEPNAME)
  - o For EAV-only, creation time (CR. TIME)
- To return to the DASD Free Space list, press PF3.
  - Select another volume and continue.
  - If you want to look at the datasets on a single volume, say TSO911, you would use fastpath the launch the Dataset List function:

QW N=TSO911

### [PowerPoint Page 45 – 2<sup>nd</sup> click]

- For additional information about Dataset Name List, see Q/R item

# MVS/QuickRef – Beyond The Basics of QW

DSN-LIST-PANEL

## 7 - Using QWIKREXX function to capture Q/R output

[PowerPoint Page 46 - Using QWIKREXX function to capture Q/R output]

MVS/QuickRef is designed as an ISPF application. As such, its primary mode is to read/display panels and use internal ISPF facilities. I would like to tell you about two slightly different flavors of Q/R.

[PowerPoint Page 47 - 47 - QWBATCH.JPG]

**BATCH MODE** - In the JCL dataset, member QWBATCH can be tailored and used to submit MVS/QuickRef to run in batch. When MVS/QuickRef does not detect the presence of ISPF, it assumes it is running in batch mode. Many installations use this facility on a daily basis to obtain the DASD Free Space information. By using the output from DASD Free Space (written to a dataset), installations can programmatically determine DASD storage constraints and requirements as well as keep up with the daily changes.

[PowerPoint Page 48 - 48 - QWIKREXF.JPG]

**ON-LINE** - However, many clients wish to execute the batch flavor of MVS/QuickRef at their TSO/E session. In other words, they wish to execute MVS/QuickRef and have the output captured to a dataset. That should be simple, right? However, MVS/QuickRef will detect ISPF, and it won't cooperate.

If you use the TSOEXEC command to invoke a program such as a REXX Exec, MVS/QuickRef will be tricked into running in batch mode while under TSO/E. In fact, in your JCL distribution dataset, you will find member QWIKREXF is a sample REXX Exec that illustrates this. QWIKREXF consists of both documentation and a sample program that calls MVS/QuickRef as a program.

[PowerPoint Page 49 - 49 – Steps to follow]

# MVS/QuickRef – Beyond The Basics of QW

Here's what you need to do:

- Edit member QWIKREXF located in the JCL distribution set.
- Find the following line:

```
x = QWIKREXX('S=*')
```

and modify it to issue an MVS/QuickRef subcommand. In this case, the parameter string requests DASD Free Space to return information on all DASD volumes.

- Make sure the following MVS/QuickRef modules are available either by using a STEPLIB DD statement in the TSO logon procedure OR by putting them into the Linked List:  
QWIKREF1 – the Q/R load module  
QWIKREXX – the program called by the QWIKREXF Rexx Exec
- Issue

```
TSOEXEC QWIKREXF
```

**[PowerPoint Page 50 - 50 - QWIKREXF Output.JPG]**

The output will be passed back to the QWIKREXF Rexx Exec and displayed by the “say” instructions. In addition, the data will be stored into a Rexx stem variable for later use.

## 8 - Using recursive invocation of Q/R

**[PowerPoint Page 51 - Using recursive invocation of Q/R]**

MVS/QuickRef can be recursively invoked "on top of itself". What this means is that, once MVS/QuickRef is initially invoked, you can invoke a second copy of MVS/QuickRef without canceling the current invocation. When you are through working with the second invocation of MVS/QuickRef, the RETURN command or, from the main menu, the END command, will return you to the MVS/QuickRef invocation and panel that was originally displayed.

# MVS/QuickRef – Beyond The Basics of QW

MVS/QuickRef can be invoked recursively with the QW command (with or without a fast-path string) as well as with cursor-driven invocation. In addition, on some screens, such as the DASD Free Space screen, you can simply enter another volser

## [PowerPoint Page 52 - 52 - Recursive 1.JPG]

```
Vol ==> tso901          *MVS/QuickRef 7.6 - Free Space*          Line 1 of 1
Command ==>                                         Scroll ==> CSR
You may scroll the information below UP, DOWN, LEFT, and/or RIGHT as needed

  1 volumes matched TSO900  - % Free:  69.2 - Free Tracks: 104,071
                    (cursor select any volume below for dataset display)
Volume  Mount  Device   Free   Free   Free  Lrg Free  LgFree DAS
Serial  Attr   Type     Exts   Tracks Cyls  #Tracks  #Cyls Nmb
  (01)  (02)   (03)     (04)   (05)  (06)   (07)    (08) (09)
-----
TSO900 STORAGE 3390          10    104071    6930    60079    4005 OAC0
***** BOTTOM OF LIST *****
```

Press ENTER and you'll see TSO901

## [PowerPoint Page 53 - 53 - Recursive 2.JPG]

```
Vol ==>                *MVS/QuickRef 7.6 - Free Space*          Line 1 of 1
Command ==>                                         Scroll ==> CSR
You may scroll the information below UP, DOWN, LEFT, and/or RIGHT as needed

  1 volumes matched TSO901  - % Free:  44.4 - Free Tracks: 66,810
                    (cursor select any volume below for dataset display)
Volume  Mount  Device   Free   Free   Free  Lrg Free  LgFree DAS
Serial  Attr   Type     Exts   Tracks Cyls  #Tracks  #Cyls Nmb
  (01)  (02)   (03)     (04)   (05)  (06)   (07)    (08) (09)
-----
TSO901 STORAGE 3390          9     66810    4449    66510    4434 OAC1
***** BOTTOM OF LIST *****
```

## [PowerPoint Page 54 - 54 - Recursive 3.JPG]

Press PF3 (END) and you'll return to TSO900. Again, and you'll be out of QW.

# MVS/QuickRef – Beyond The Basics of QW

## 9 - Using automated message lookup / inclusion for ABENDs embedded in a message

**[PowerPoint Page 55 - Using automated message lookup / inclusion for ABENDs embedded in a message]**

All of us have to work in a similar environment: we are asked to do more with fewer resources. Over the years, we have learned tricks and techniques that help us do more such as write macros, subroutines, or Clists that perform often used commands.

With today's internet access, we try to search for information in a sea of clutter. As you approach my age, you realize that your memory seems to be a limiting factor. My wife and I often laugh about what we've forgotten today. At work, I can go chasing after rabbits and, when I find the rabbit, can't remember what I needed it for to start with.

For example, think back to our first scenario, and, if you've forgotten, let me mention it again.

When you think of how to use MVS/QuickRef, the thought that probably comes to mind is to look up a message. An application programmer would likely get a call saying a batch-oriented production job failed. He / She would use SDSF to look at the JES2 Job Log to see the problem and discover a weird message they have never seen before.

Are you with me? Well, suppose the analyst sees an ABEND code, in this case, SD37. And, they start chasing the D37 rabbit.

**[PowerPoint Page 56 - 56 – D37 in Output.JPG]**

**[PowerPoint Page 57 - 57 – D37 Selection List.JPG]**

On the third line of the Explanation, the text refers you to message IEC031I. So, they start chasing the IEC031I rabbit. But, oops, the IEC031I rabbit requires you to know the return code.

# MVS/QuickRef – Beyond The Basics of QW

**[PowerPoint Page 58 - 58 - IEC031I Reference.JPG]**

So, you flip back to the original JES2 Job Log and look for the return code. Where am I going with all of this?

When you lookup the ABEND code using MVS/QuickRef, the pre-display analysis detects the SD37 ABEND refers to IEC031I. So, MVS/QuickRef looks up that message and appends its description to the bottom of the SD37 description. MVS/QuickRef does that for all messages, but in the case of an ABEND description, we save you from looking in two places. As far as I know, this feature is unique to Q/R. I don't believe any IBM z/OS doc or Google search does it.

**[PowerPoint Page 59 - 59 - D37-04.JPG]**

Now, let's take this example a little further. We are in time-saving mode here. Whatever we need to do to get to the answer quicker and more correctly is what we are striving for. Let's examine the FINDCODE function.

And now, it is time for another pop Quiz...

## Pop Quiz II

**[PowerPoint Page 60 - for Quiz II Question]**

**What are the top 3 largest items in the Q/R database?**

1. ?
2. ?
3. 3

**[PowerPoint Page 60 – second click]**

## Answers

1. **-RSN-CD-BY-VALUE UNIX(R) Reason Codes (Errnojrs) Listed by Value 10804 lines**

# MVS/QuickRef – Beyond The Basics of QW

## 2. IEC161I – 10804 lines

## 3. IDC3009I – 10802 lines

Each of these items is at the maximum. They are very close to filling up the dynamic storage area from which we display a screen. We have to remove the blank lines to get them to fit.

## 10 - Using FC function

### [PowerPoint Page 61 – Using the FC Function]

As you know, many messages have alternative endings depending upon a return code or a reason code. For example, message IEC161I has over 255 sub-codes. Messages IDC3009I also has over 255 sub-codes. And, as we've just seen, these 255 sub-codes are spread over 10,800+ lines.

Remember Cursor Driven Invocation? When you look up messages that have standard format return/reason codes using MVS/QuickRef, the pre-display analysis detects this and attempts to assist you by providing the FINDCODE function.

As an example of automatic FINDCODE processing, suppose you invoke MVS/QuickRef using cursor-driven invocation after placing the cursor under the IEC141I message id in the SYSLOG segment shown below:

### [PowerPoint Page 62 - 59 - IEC141I 013-18.JPG]

```
IEC141I 013-18, IGG0191B,RSC004A,C,SYSIN,VIO,SYS00322,  
T165050.RA00.RSC004A.TEMP.HO1  
IEA995I SYSTEM DUMP OUTPUT
```

## MVS/QuickRef – Beyond The Basics of QW

It should be noted that numeric code **013** is the system ABEND code associated with this message and does not appear in a code list in the reference text for the IEC141I message.

Numeric code **18**, however, is the return code associated with the S013 ABEND and it does appear in the list of return codes described in the reference text for the IEC141I message.

So, when the reference text for the IEC141I message is displayed, the following FINDCODE command would be shown in the command line:

FC 18

and the following message would be displayed in the message line:

**'Press ENTER to FIND indicated code; else, blank out the command line'**

**[PowerPoint Page 63 - 63 - FC 18.JPG]**

By pressing ENTER, MVS/QuickRef will display the following:

- 18 An OPEN macro instruction was issued for a partitioned data set.  
The DSNAMES parameter specified a member that could not be found.

**[PowerPoint Page 64 - 64 - FINDCODE Results.JPG]**

MVS/QuickRef does that for all cursor-driven invocation requests for messages that meet the format criteria. As far as I know, this feature is unique to Q/R. I don't believe any IBM z/OS doc or Google search does it.

### 11 - Using Textmarks

**[PowerPoint Page 65 - Using Textmarks]**

- Textmarks are bookmarks for MVS/QuickRef. When you find yourself looking up the same items, you simply bookmark them for later.

## MVS/QuickRef – Beyond The Basics of QW

- What would be a good item to bookmark? It depends on what you do. For me, I often have to lookup dynamic allocation. It is useful to have the dynamic allocation return codes and reason codes available.
- Those of you that often look up items in MVS/QuickRef know that you will sometimes get a selection list. For example, if I were to look up item DYNALLOC using the fastpath method, I would get the following selection list:

**[PowerPoint Page 66 - 66 - DYNALLOC Selection List.JPG]**

- I then select this DYNALLOC item that I want.
- Once the correct item is displayed, I can use the SETMARK (alias is SM) subcommand to tell Q/R to set this to the next textmark number as shown here

**[PowerPoint Page 67 - 67 - SETMARK.JPG]**

SETMARK

Or designate the number (1 through 9) for the specific textmark number as shown here

SM 1

SM is an alias of SETMARK.

**[PowerPoint Page 68 - 68 - LISTMARK.JPG]**

LISTMARK

To obtain the complete list of your Textmarks, you issue the LISTMARK subcommand. LM is an alias of LISTMARK.

**[PowerPoint Page 69 - 69 - GETMARK.JPG]**

GETMARK n

To display the Textmark information, you issue the GETMARK subcommand. GM is an alias of GETMARK.

# MVS/QuickRef – Beyond The Basics of QW

**[PowerPoint Page 70 – 70 - Advanced Textmark Ideas]**

**[PowerPoint Page 71 – 71 - Fastpath Textmark.JPG]**

When using fastpath, a textmark can be referenced using the T parameter. In this example, you want to go directly to textmark #2. Here's how you do it:

```
QW T=2
```

**[PowerPoint Page 72 – 72 Scrolling Textmark.JPG]**

The previous example I've given you is of textmarking an item. However, a selection list can also be textmarked.

In addition, when you textmark an item or a list, QuickRef will note the scroll location and position the display as it was when the textmark was made.

This example shows

## 12 - Handling the license key file

**[PowerPoint Page 73 - Handling the license key file]**

Although the license key file is not a straight-forward QuickRef user function, I have been asked to mention it.

MVS/QuickRef prior to Release 7.0, used a zap to activate the license for the product. Starting with R7.0, Chicago-Soft changed to a license key file. Because of its importance, I was asked to walk you through handling the license key file.

The license key file is a great improvement over the zap system. We've eliminated (for the most part), human intervention. You access the Chicago-Soft Customer Service portal system and request a new license key file. This is true for a Disaster Recovery license key file as well.

## MVS/QuickRef – Beyond The Basics of QW

After requesting, say a new DR license, you will receive an e-mail with the license key file as an attachment. By default, the attachment comes with a .QWL extension although you can request the attachment be changed to a .TXT extension.

Note: The attachment does contain binary information. For this reason, you should never open the attachment on the PC as NotePad has been reported to change x'00' to x'20' without any warning.

- Before uploading the license key file, you should preallocate the license key file on the mainframe.

The SPACE of the license key file is negligible – 221 bytes for a single processor. Although the size of the file varies depending upon the number of processors in the key, it is rather small. One track would be sufficient for this file.

You must use the following DCB attributes:

### [PowerPoint Page 74 - DCB attributes]

DSORG – PS  
RECFM – VB  
LRECL – 10440  
BLKSIZE – 10444

- Upload the attachment containing the license key file to the PC using your file transfer program of choice; however, you must transfer the file using the BINARY data transfer option. If you upload it as a TEXT file, you will scramble the contents.
- After the upload is complete, you need to make the license key file available to MVS/QuickRef. If you are using LIBDEF to access MVS/QuickRef, then locate the QW Clist and tailor the following statement:

### [PowerPoint Page 75 – 75 - QW clist]

## MVS/QuickRef – Beyond The Basics of QW

```
ALLOC F(QWLICKEY) +
```

```
DA('hlq.LICENSE.KEY.FILE') SHR REUS
```

If you are not using LIBDEF, then you will probably want to modify the QWIKOPTS parameter list to specify the license key file. In this case, tailor the following statement:

### [PowerPoint Page 76 70 – 76 - QWIKOPTS]

```
QDB=,          MAIN Q/R DATA BASE DSN, NO QUOTES   X
```

```
LICDSN=,       LICENSE KEY DSN, NO QUOTES QP0710B 7.0X
```

Then, reassemble the QWIKOPTS file using member QWOPTASM in the JCL dataset.

To make the new file available to MVS/QuickRef, get out of ISPF and get back in. Bring up QW. If the main menu comes up, then you were successful.

### [PowerPoint Page 77 – Useful Links]

For the 7.2 release of MVS/QuickRef, I created five videos consisting of the first five scenarios in this presentation...

1. Lookup ABEND Code
2. Lookup JCL Syntax
3. Find DASD Free Space
4. COBOL PERFORM Verb
5. The TOOLBOX

**View these videos at:**

[https://www.quickref.com/newQuickRef/helpdocs/video/video\\_help](https://www.quickref.com/newQuickRef/helpdocs/video/video_help)

### [PowerPoint Page 78 – Notes & Warnings]

These last few slides contain notes and warning that you may find useful:

- Command Not Found?
- Invalid License Key?

## MVS/QuickRef – Beyond The Basics of QW

### [PowerPoint Page 79 – Command Not Found?]

NOTE: If typing QW and pressing ENTER gives ”**command not found**”, then MVS/QuickRef is not installed correctly at your site ...specifically, the ISPF Command Table update is not correctly installed.

### [PowerPoint Page 80 – Invalid License Key?]

Note: The QuickRef license key file is sent to installations via e-mail with an attachment.

The attachment contains both text and binary information.

For this reason, you should never open the attachment on the PC because NotePad has been reported to change x'00' to x'20' without warning.