

YOU ARE HERE:	HOME	PROGRAMMING	SQL	QUERY MANAGEMENT FORMS	
---------------	------	-------------	-----	------------------------	--

# Query Management Forms

GUEST.VISITOR / 28 FEBRUARY 1995

## TOOLS

 PRINT

 EMAIL

Have you ever wished you could design a custom report without having to write a program? Have there ever been times when Query/400 just wouldn't give you the report format you really needed? If so, you need to get acquainted with Query Management (QM) forms.

Query Management/400 far outshines Query/400 in forms control. Here are some advantages that make QM reporting a far more dynamic tool than plain Query/400 reporting.

- o You can use variables from the report columns in your page headings, text printed at control breaks (break text), and text associated with final totals. The variables used in the break text don't necessarily have to be break fields.
- o You can explicitly design and place page heading and footing text, break heading and footing text, and final text lines. For example, final summary values can all be put on the last page in any order, instead of being spread throughout the report in separate columns.
- o Forms and queries are separate objects, so each can be changed independently.
- o Query Management/400 uses SAA date, time, and numeric edits.
- o Query Management/400 queries and forms control specifications are supported on other IBM operating systems, such as MVS, VM, and OS/2.
- o Query Management/400 reports can be run in batch mode even if they require variables to be passed to them at run time. This can be done using the SETVAR parameter of the Start Query Management Query (STRQMQRy) command.

QM is an OS/400 tool that is meant for data processing professionals-it is not an end-user tool. It gives programmers the ability to create queries that can be used within CL and HLL programs. With QM, a programmer creates an AS/400 object that retrieves information from the OS/400 database. This object is known as a QM query (object type QMQRy).

You can create a QMQRy object from an SQL statement entered into a source member. You can also create a QMQRy object by retrieving the SQL statement from a Query/400 query definition.

The output from a QMQRy presents a basic, generic report format. You get column headings, but that's about it. However, you can create your own custom format with another object known as a QM form (object type QMFORM). This article will show you how to code the specifications required to create a QMFORM object. (For more information on QM, see "Query Management Magic," MC, November 1994.)

Creating QM Forms with Other AS/400 Products

QM forms can be created in several ways. Two of the methods-SQL/400 and Query/400-require additional AS/400 licensed programs. SQL/400 Query Manager is a free functions of OS/400 and it makes designing a form much easier and faster by taking you through a series of prompted displays in setting up your report.

To create a QM form from a Query/400 query definition, first run the Analyze Query (ANZQRY) command on the query definition. You'll get messages suggesting ways to adjust the query definition before converting it to QM/400 objects. Use the Work with Queries (WRKQRY) command to make these adjustments. Then create a source file member for the QM form. (QQMFORMSRC is the IBM-suggested source file.) Run the Retrieve Query Management Form (RTVQMFORM) command to extract the form definition. Edit the definition as needed to make any desired changes to it. (Editing the form will be covered in the remainder of this article.) Finally, use the Create Query Management Form (CRTQMFORM) command to create the QM form object.

#### Creating QM Forms From Scratch

If you don't have SQL/400 Query Manager or Query/400, don't despair. It's still possible to code the source member for a QM form yourself in a source physical file and then use CRTQMFORM to create the form object from this source member.

It isn't even necessary to create a form; a QM/Query will run without a form being specified for it. It will simply default to the SAA forms control standards. For most people, though, this default form is a bit sparse.

Let's assume you want to design your own report form, assuming the SQL statement for retrieving the information is

```
SELECT REFDCS, REFNO, JRNRNO, ACCTNO, ACCTDS, GLAMT FROM GLTRAN
```

This statement may have been retrieved from an existing Query/400 query or keyed from scratch.

The SAA forms control standard is for a report 80 columns wide. You probably want a wider report-say, 132 columns. One of the first things to do is use the Change Printer File (CHGPRTF) command to change the QM print file QPQXPRTF in library QSYS to a width of 132 positions per line.

#### QM Source Record Types

A QM form, like an RPG program, is made up of several record types. Each record type controls specific information for the report. The basic QM record types are listed here and summarized in 1.

A QM form, like an RPG program, is made up of several record types. Each record type controls specific information for the report. The basic QM record types are listed here and summarized in Figure 1.

- o Header (H) record
- o Value (V) record
- o Table description (T) record
- o Table row (R) record
- o End of object (E) record

The Header Record: The H record must be the first record in the file. It gives information about the object-what kind of object it is, its characteristics, and its format. Don't let this one scare you; much of the information in it is not needed on input, so those positions can be left blank. The format of the H record is given in 1, section A.

The Header Record: The H record must be the first record in the file. It gives information about the object-what kind of object it is, its characteristics, and its format. Don't let this one scare you; much of the information in it is not needed on input, so those positions can be left blank. The format of the H record is given in Figure 1, section A.

For example, H QM4 03 F 01 E V W E R 01 03 92/06/30 19:12 means a QM4 FORM file is at object level 1, in encoded format, with a valid status (no errors or warnings), containing code for a whole object, in English, usable for complete replacement, with one byte of control area and three bytes for integer lengths, written on June 30, 1992, at 7:12 p.m.

The Value Record: The V record describes a single form attribute (such as detail line spacing or column head separators). 1, section B, contains the format of the V record. It includes a unique field number for each attribute. For example, 1201 is the "blank lines before page heading" attribute. It also includes the value length value for that attribute.

The Value Record: The V record describes a single form attribute (such as detail line spacing or column head separators). Figure 1, section B, contains the format of the V record. It includes a unique field number for each attribute. For example, 1201 is the "blank lines before page heading" attribute. It also includes the value length value for that attribute.

For example, V 1201 003 001 is the value record for the "blank lines before page heading" (field number 1201) attribute. The data value has a length of 3, and the data value is 1.

The Table Description Record: The T record describes the content and format of the information that follows in the R records. The T record identifies which table is being described (by its unique table number), which columns are included (by their unique field numbers), what order they come in, and the lengths of the values in these columns. 1, section C, contains the format of the T record.

The Table Description Record: The T record describes the content and format of the information that follows in the R records. The T record identifies which table is being described (by its unique table number), which columns are included (by their unique field numbers), what order they come in, and the lengths of the values in these columns. Figure 1, section C, contains the format of the T record.

For example, T 1210 001 003 1212 003 1213 006 1214 055 is form table 1210 (page heading text table), containing one row and three columns, with column 1212 (page heading line number) of length 3, column 1213 (page heading align) of length 6, and column 1214 (page heading text) of length 55. 2 contains a chart of table numbers, field numbers, and their descriptions:

For example, T 1210 001 003 1212 003 1213 006 1214 055 is form table 1210 (page heading text table), containing one row and three columns, with column 1212 (page heading line number) of length 3, column 1213 (page heading align) of length 6, and column 1214 (page heading text) of length 55. Figure 2 contains a chart of table numbers, field numbers, and their descriptions:

The Table Row Record: The R record gives a set of values for a single row in the current table. It contains an ordered list of values that are described by the corresponding T record. An R record must exactly match the positions and lengths described in the T record. 1, section D, contains the format of the R record.

The Table Row Record: The R record gives a set of values for a single row in the current table. It contains an ordered list of values that are described by the corresponding T record. An R record must exactly match the positions and lengths described in the T record. Figure 1, section D, contains the format of the R record.

The End-of-Object Record: The E record is used to mark the end of the object. It simply consists of an E in the first position. It should be the last record in the file.

Coding a Sample QM Form Source

To code a QM form, you will need to create a source physical file called QQMFORMSRC in the library of your choice, with a record length of 162. You will probably want to create a template form source member in this file, as follows.

First, set up a default form source member called DEFAULT, source type QMFORM. You will need an H record and at least one T, R, and E record, as illustrated in 3. This is a one-column form.

First, set up a default form source member called DEFAULT, source type QMFORM. You will need an H record and at least one T, R, and E record, as illustrated in Figure 3. This is a one-column form.

Run CRTQMFORM to create the form object, and then run RTVQMFORM to retrieve a template source member from it, with all the information filled in for a one-column report with all default values. Here are examples of the commands:

```
CRTQMFORM QMFORM(MYLIB/TMPLAT)+ SRCFILE(MYLIB/QQMFORMSRC)+ SRCMBR(DEFAULT) RTVQMFORM QMFORM(MYLIB/TMPLAT)+ SRCFILE(MYLIB/QQMFORMSRC)+ SRCMBR(DEFAULT)
```

The resulting template source member (edited for space limitations) is listed in 4. Now all you have to do is copy and edit TEMPLT, changing field values and adding more columns.

The resulting template source member (edited for space limitations) is listed in Figure 4. Now all you have to do is copy and edit TEMPLT, changing field values and adding more columns.

#### Defining The Report Columns

This example uses six columns in the report, in this sequence:

- o REFDC, character, 21 bytes wide.
- o REFNO, character, 20 bytes wide.
- o JOURNAL\_REF NO, character, 12 bytes wide.
- o ACCOUNT NUMBER, numeric, 15 bytes wide, zero decimal places, showing leading zeros, no negative sign, no thousands separator, no currency symbol, and no percent sign.
- o DESCRIPTION, character, 15 bytes wide.
- o G/L AMT, numeric, 21 bytes wide (the actual field is 13 bytes long, but it allows extra positions for editing and totals), two decimal places, no leading zeros, negative sign to print, thousands separator to print, no currency symbol, and no percent sign.

This example has one blank space between the columns and a level break on the field REFDC. Total the G/L AMT field. 5, section A contains the T record that defines this report columns table.

This example has one blank space between the columns and a level break on the field REFDC. Total the G/L AMT field. Figure 5, section A contains the T record that defines this report columns table.

The field number 1110 identifies this table as the columns table. The 006 means that there are six R records following this T record. The 007 means that there are seven field number/field length pairs in the T record. These length pairs represent information about various characteristics of the report columns. The information is given in the R records following this T record. You can have any number of field number/field length pairs you desire, as long as the information in the R records matches the field number/field length pairs in value and length. Any fields (attributes) you don't specify will be given their default values at run time.

5, section B contains the R records for this columns table. Each record contains the information about column usage, column indent, column width, column data type, column edit, column sequence, and column heading, just as specified in the field number/field length pairs in the T record preceding them. Each field number/field length pair in the R record must be separated by a blank.

Figure 5, section B contains the R records for this columns table. Each record contains the information about column usage, column indent, column width, column data type, column edit, column sequence, and column heading, just as specified in the field number/field length pairs in the T record preceding them. Each field number/field length pair in the R record must be separated by a blank.

The field number/field length pairs in the T record, which appear in the following order, control specific aspects of the report's form:

- o 1114 stands for column usage and has a length of 6. The first R record has "BREAK1" in the first field. This means that the REFDSC field is going to be the first level break field for this report. The last R record has "SUM" in the first field. This means that the G/L\_AMT field will be accumulated and totaled. The other R records have blanks in the first field, meaning that they don't have a special usage value.

- o 1115 stands for column indent and has a length of 2. All of the R records have a column indent of 1.

- o 1116 stands for column width and has a length of 3. The REFDSC column is 21 bytes wide; the REFNO column is 20, the JOURNAL\_REF\_NO column is 12, the ACCOUNT\_NUMBER column is 15, and the G/L\_AMT column is 21.

- o 1112 stands for column data type (character or numeric) and has a length of 9. ACCOUNT\_NUMBER and G/L\_AMT are numeric, and the other four columns are character fields.

- o 1117 stands for column edit and has a length of 5. 6 contains a chart of the edit codes used by QM to format numeric data. The numbers after some of the codes tell how many decimal positions are in the field.

- o 1117 stands for column edit and has a length of 5. Figure 6 contains a chart of the edit codes used by QM to format numeric data. The numbers after some of the codes tell how many decimal positions are in the field.

You can also use the edit codes field to control the format of character fields. C causes no change in the display of a character value. CW allows it to be wrapped if it can't fit on one line; and CT wraps it according to the text in the column (so it is broken at a blank rather than in the middle of a word).

- o 1118 stands for column sequence and has a length of 3. Notice that the columns are numbered in multiples of ten, so a column may be inserted between two other columns later.

- o 1113 stands for column heading and has a length of 62. The underline character in JOURNAL\_REF NO indicates a new line for a multiple line heading.

#### Setting Up The Page

This sample page heading reads, "TRY TO BALANCE G/L." It should be centered, with one blank line before the heading and two blank lines after it. Here are the records that define the page heading text table (these records can also be found in 5, section C):

This sample page heading reads, "TRY TO BALANCE G/L." It should be centered, with one blank line before the heading and two blank lines after it. Here are the records that define the page heading text table (these records can also be found in Figure 5, section C):

V 1201 003 001

The 1201 field number is the "blank lines before page heading" attribute. It has a value length of 3 and a value of 1.

V 1202 003 002

The 1202 field number is the "blank lines after page heading" attribute. It has a value length of 3 and a value of 2.

The next T record is the page heading text table:

```
T 1210 001 003 1212 003 1213 006 1214 055 R 001 CENTER TRY TO BALANCE G/L
```

This T record shows that the fields in the R record are 1212 (line number) with a length of 3, 1213 (alignment) with a length of 6, and 1214 (text) with a length of 55, respectively. The R record shows the heading is on line number 1, centered, with a text of "TRY TO BALANCE G/L."

5, section D, contains the records that define the page footing. These records create two blank lines before the page footing and no blank lines after it. The page footing is left aligned, with a text of "GLREFSQL."

Figure 5, section D, contains the records that define the page footing. These records create two blank lines before the page footing and no blank lines after it. The page footing is left aligned, with a text of "GLREFSQL."

#### Defining The Final Summary

There is no new page for the final summary. It reads "FINAL TOTAL:" and is left aligned and one line is skipped before it. 5, section E, contains the records that define the final summary.

There is no new page for the final summary. It reads "FINAL TOTAL:" and is left aligned and one line is skipped before it. Figure 5, section E, contains the records that define the final summary.

#### Defining Options

The report formatting options for this example are defined in 5, section F. In this case, the detail lines are single spaced and break columns are outlined, so the break value is only printed when it changes at a level break. There are asterisks at each break level.

The report formatting options for this example are defined in Figure 5, section F. In this case, the detail lines are single spaced and break columns are outlined, so the break value is only printed when it changes at a level break. There are asterisks at each break level.

Other options defined here include column wrapped lines being kept on the same page, column heading separators and break summary separators appearing as dash lines in the report, and the final summary separators appearing as equal signs.

#### Defining Break Levels

This example contains no new pages for the break heading and the break footing. Column headings are not repeated at break headings, and there are no blank lines before or after.

One blank line appears before and after the break footing, which reads "REFDSC &1 TOTAL:" and is left aligned. (The "&1" is a reference to a QM/400 variable; in this case, the value of column 1 of the last record before the break footing.) 5, section G, contains the records defining the break level options selected.

One blank line appears before and after the break footing, which reads "REFDSC &1 TOTAL:" and is left aligned. (The "&1" is a reference to a QM/400 variable; in this case, the value of column 1 of the last record before the break footing.) Figure 5, section G, contains the records defining the break level options selected.

At this point, the CRTQMFORM command is used to create the form object. Once the form object is created, you can run the QM query with the STRQMQRy command specifying the QM form object name in the QMFORM parameter.

#### Arm Yourself with Knowledge

This article is by no means intended to be exhaustive, but it should give you some idea of the raw power and flexibility of QM forms. For more information, consult Query Management/400: Programmer's Guide and Reference. Armed with this knowledge, you'll be well on your way to designing your own QM report forms.

Sharon Cannon is a programmer/analyst with Lenoir Mirror Company. She can be contacted through MC-BBS.

#### REFERENCES

V2: Query Management/400: Programmer's Guide and Reference (SC41-0090, CD-ROM QBKA7K01).

V3: DB2/400 Query Management Programming (SC41-3703, CD-ROM QBKAUE00).

---

#### Query Management Forms

### Figure 1 Record Formats

Section A: Header Record Format H-PPP-RR-T-00-F-U-S-N-A-CC-II-YY/MM/DD-HH:MM- PPP is the proc

Query Management Forms

Figure 2 Table Record Chart



## Query Management Forms

### Figure 3 A Default QM form Source Member

## Query Management Forms

### Figure 4 Source Template Created From Default Source

## Query Management Forms

### Figure 5 The Complete Form Source for Sample Rej

## Query Management Forms

### Figure 6 QM Edit Codes

Edit Lead Negative Thousands Currency Percent Display of Code Zeros Sign Se

ALSO ON MC PRESS ONLINE

**SQL 101: Using ILE  
RPG Programs and ...**

a year ago • 1 comment

How do you create a stored procedure? Almost any existing program or ...

**RPG Academy:  
Database ...**

a year ago • 1 comment

An industry expert explains how an I/O Server fits into database modernization ...


**RPG Academy:  
Database ...**

a year ago • 2 comments

I showed you one way to generate automatic unique keys for your records. ...

**New S  
Comb**

3 month

SQL's L  
makes  
values0 Comments MC Press Online  Disqus' Privacy Policy Login ▾ Recommend Tweet Share

Sort by Newest ▾



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name

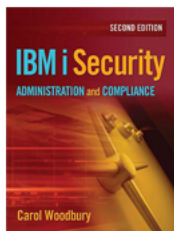
Be the first to comment.

 Subscribe  Add Disqus to your siteAdd DisqusAdd  Do Not Sell My Data

BLOG COMMENTS POWERED BY DISQUS

**MC-STORE.COM**

SECURITY &gt;&gt;



IBM i security in V7R2 and V7R3

RPG &gt;&gt;



The one guide you need to learn how to be successful with all aspects of ILE RPG

SYSTEM ADMIN &gt;&gt;

**THE  
ULTIMATE  
GUIDE TO  
IBM i**[Click to Learn More >>](#)

PROGRAMMING: CL &gt;&gt;

**Your Guide  
to Control  
Language  
Excellence**[Click to Learn More >>](#)

RPG &gt;&gt;



Learn the Java language and core Java-supplied functionality

DATABASE &gt;&gt;



Build ETL solutions for business applications