

Query/400 Discovery Tool Usage Instructions



Updated January 2012



Contents

Overview	3
Installation	3
DSCQRYOBJ (Discover Query/400 Objects) command	4
Required parameters	4
Additional parameters	5
Sample commands.....	5
Tables generated by the tool.....	6
QRYDFNOUTF (QRYDFNOUTF).....	7
QD_SUMMARY (QD_SUMMARY).....	8
QD_SELECTED_FILES (QD_SLCFIL).....	9
QD_JOIN_TESTS (QD_JOINTST).....	10
QD_RESULT_FIELDS (QD_RSLFLD)	11
QD_SELECT_RECORD_TESTS (QD_SLCRCD)	12
QD_ORDER_SELECTED_FIELDS (QD_ORDFLD).....	13
QD_REPORT_COLUMN_FORMAT (QD_COLFMT)	14
QD_SPECIAL_CONDITIONS (QD_SPCCOND).....	15
QD_REPORT_BREAKS (QD_RPTBRK).....	15
QD_DATABASE_OUTPUT (QD_DBOUT)	15
QD_PRINTER_OUTPUT (QD_PRTOUT)	16
QD_PRINTER_COVER_PAGE_HEADING_FOOTING (QD_PRTCVR).....	16
Analyzing the results.....	17
Finding the libraries containing the queries	17
Finding the files used in the queries	17
Finding the selection criteria used in the queries.....	17
Finding the fields used for selection criteria in the queries	17
Finding the result fields used in the queries.....	18
Support	19



Overview

The Query/400 Discovery Tool is used to find all Query/400 Query Definition (*QRYDFN) objects on your system, retrieve their attributes, and sort them into a series of tables. The data in these tables can then be analyzed to find trends in your reporting environment.

It is important to note that this is only one part of an overall report modernization process. In addition to the report definition attributes, items such as most frequently used, last used date, last modified date, and several other items should be taken into account when planning any migration.

These trends can then be used to migrate your reporting to a more modern tool such as DB2 Web Query. If you need assistance with this migration or usage of tools such as DB2 Web Query, please contact the IBM STG Lab Based Services team at stgls@us.ibm.com for assistance.

This document contains the information for installing and using the Query/400 Discovery tool.

Installation

To install the tool, follow these steps:

1. Copy/move the qry400dsc.zip attachment from your email to your local PC. In this example we assume you copied the savf to C:\ (the root directory of your PC's hard drive).
2. Use an extraction utility to unzip the zip file to qry400dsc.savf
3. Sign on to 5250 session using a user profile with *ALLOBJ authority
4. From command line, issue the command: **CRTSAVF FILE(your_library/QRY400DSC)**
5. FTP the savefile to your IBM I system in binary mode. You can use If you have a graphical FTP client you can use that, a or take the following steps:
 - a. On your PC, click Start > Run> and type in "cmd". Click OK or press enter.
 - b. This will bring up a black DOS style command prompt. Change directories to where you saved the attachment to on your PC (ie: cd C:\).
 - c. Type "ftp your_system" where your_system is your IBM i's IP address or DNS name and press enter.
 - d. When prompted, type your IBM i user profile and press enter.
 - e. When prompted, type your IBM i password and press enter.
 - f. Type "bin" and press enter to switch to binary mode.
 - g. Type "put qry400dsc.savf your_library/QRY400DSC" where your_library is the library you specified in step 3 and press enter to upload the save file.
 - h. When done, type "quit" and enter to quit FTP
 - i. Type "exit" and enter to close the command prompt.
6. From command line, issue the command: **CRTLIB LIB(QRY400DSC)**
7. From command line, issue the command: **RSTOBJ OBJ(*ALL) SAVLIB(QRY400DSC) DEV(*SAVF) SAVF(mylib/QRY400DSC)**
8. From command line, issue the command: **CALL PGM(QRY400DSC/SETUP)**



DSCQRYOBJ (Discover Query/400 Objects) command

Once the tool is installed, it can be invoked using the QRY400DSC/DSCQRYOBJ command from a CL command line. By pressing F4 you see the command parameters as shown below.

```

Discover Query/400 objects (DSCQRYOBJ)

Type choices, press Enter.

Query . . . . . Name, *ALL
Library . . . . . *ALL Name, *ALL
Output library . . . . . QZRDQRYSPF Name
Discovery stages to execute . . *ALL *ALL, *COLLECT, *PARSE

Additional Parameters

Output . . . . . *NONE *NONE, *PRINT
Replace or add records . . . . . *REPLACE *REPLACE, *ADD
Debug level . . . . . 0 0, 1, 2

```

Each of the parameters is described below.

Required parameters

- **QUERY** (Query name and Library)

Specify the name of the query and its library that you want to discover. Specify *ALL to discover all query objects in the specified library. For the library portion of this parameter, specify *ALL to discover query objects in all libraries on the system.
- **OUTLIB** (Output library)
 - Specify the name of the library (schema) that will contain the output tables generated by the tool. For more information on these tables, see the in section [Tables generated by the tool](#). If the output library (and tables) do not exist, the tool will automatically create them.
- **STAGE** (Discovery stages to execute)
 - ***COLLECT** – Executes only stage 1 of the process. This stage issues the PRINT_QUERY_DEFINITION stored procedure against all of the specified query objects. This information is collected and consolidated in a single “flat” file named QRYDFNOUTF and resides in the library specified in the OUTLIB parameter. The QRYDFNOUTF file can then be saved and sent to IBM for further analysis.
 - ***PARSE** – Executes only stage 2 of the discovery process. This stage reads the QRYDFNOUTF (in the library specified in the OUTLIB parameter) and parses all of the individual elements (pieces of information) of each query. Each element is then extracted and stored in columns within a series of tables (in a relational database model) where it can be more easily harvested and analyzed. This database model and each of the tables are described in section [Tables generated by the tool](#).
 - ***ALL** – Executes both stages 1 and 2 as described above. This is the default value.



Additional parameters

- **OUTPUT**
This parameter is no longer used and is only preserved for compatibility of previous versions of the tool.
- **DTAOPT** (Data option – replace or add records)
Indicates if you want to replace all of the records in the output tables or append new records to them. Can be used to incrementally add information about queries to existing tables.
 - ***REPLACE** – Clears (removes all records) the output tables before adding records.
 - ***ADD** – Preserves all records in the tables so that incremental additions can be made. If the specific query (queries) that you are discovering are already in the tables, they will be removed first.
- **DBGLVL** (Debug level)
Indicates the level of debugging that is to be logged in the spooled file. Useful for determining what transpired during the discovery process. Possible values are:
 - **0** – No debug messages are logged and no spooled file is generated.
 - **1** – Basic debugging messages are logged and a spooled file QSYSPRT with a userdata value of QRY400DSC is generated. This basic information include values of input parameters, errors that occurred, and lines of data in QRYDFNOUTF that could not be interpreted and extracted.
 - **2** – Detailed level of debugging messages are logged and a spooled file QSYSPRT with a userdata value of QRY400DSC is generated. Includes all information available in level 1 and additional information.

Sample commands

```
QRY400DSC/DSCQRYOBJ QRY(MY_LIB/MY_QUERY) OUTLIB(QRYDSCOUT)
```

```
QRY400DSC/DSCQRYOBJ QRY(MY_LIB/*ALL) OUTLIB(QRYDSCOUT)
```

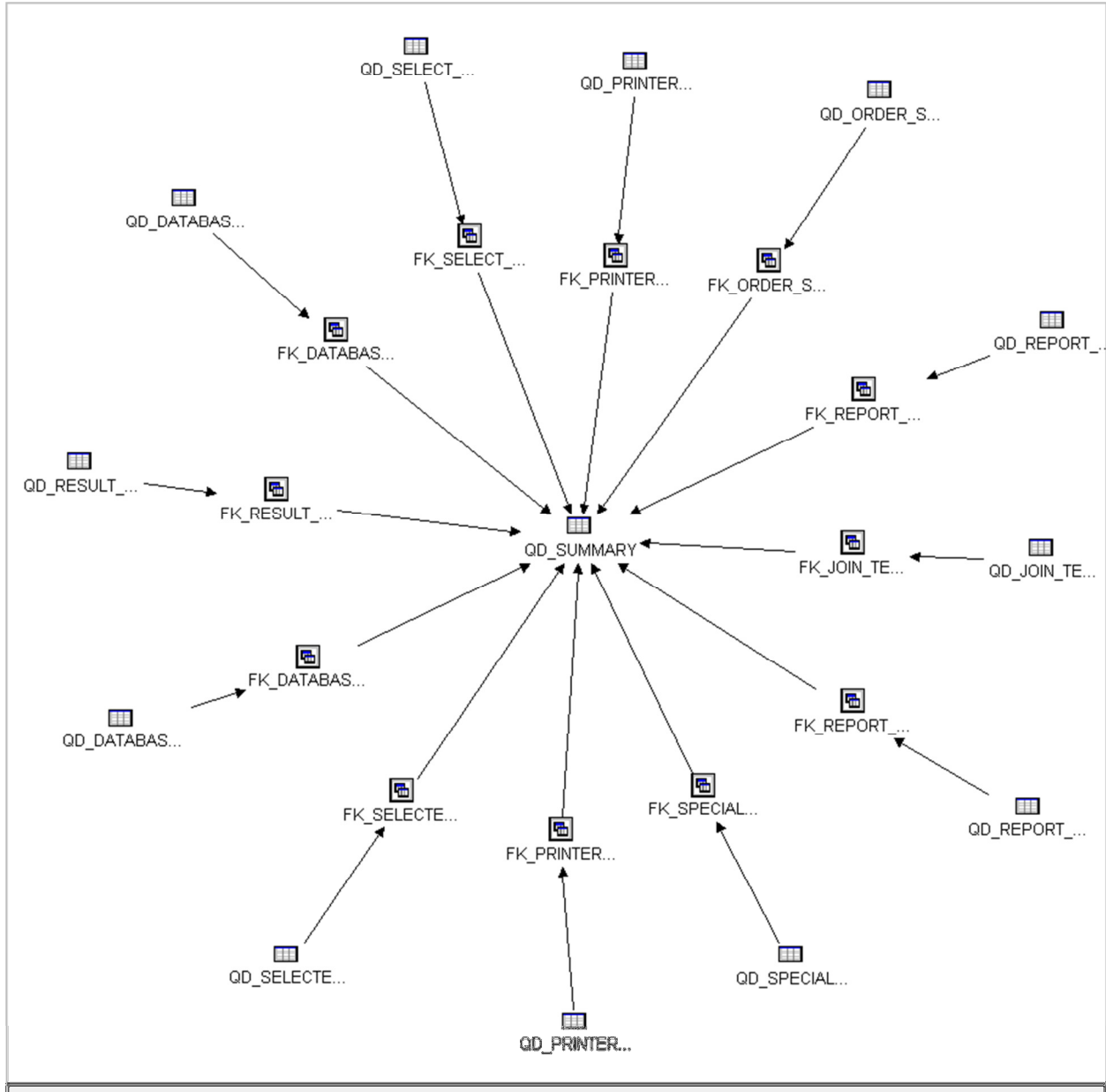
```
QRY400DSC/DSCQRYOBJ QRY(*ALL/*ALL) OUTLIB(QRYDSCOUT)
```



Tables generated by the tool

Once the command is run, the following tables are created. A brief description of each table's contents is also provided. The long/SQL name is listed first and the short/DDS name is in parenthesis.

Remember, these will be in the schema/library you provided on the second parameter of the command.





QRYDFNOUTF (QRYDFNOUTF)

This table is created and populated during the collection stage and is used to store the consolidated output of the PRINT_QUERY_DEFINITION stored procedure. It is then read during the stage 2 parsing/extraction process. This file should NOT be used for analysis as it is merely the holding table that is consumed during stage 2. The only reason this temporary table is not removed after the completion of stage 2 is for debugging purposes. If problems occur during the discovery process, this table can be saved and sent to IBM where it can be used in a debugging process.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
LINE	CHAR	380	LINE



QD_SUMMARY (QD_SUMMARY)

This table contains one row per query definition. It provides high level information such as the query's library and name as well as the output type. This table also contains the generated QUERY_ID value which is the link from this master table to all other tables below (always join on A.QUERY_ID=B.QUERY_ID).

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID. Use this to join to all other tables.
QRY_LIBRARY	CHAR	10	Query Library
QRY_NAME	CHAR	10	Query Name
QRY_TEXT	CHAR	50	Query Text
QRY_CCSID	INTEGER	4	Query CCSID
QRY_LANGUAGE_ID	CHAR	3	Query Language ID
QRY_COUNTRY	CHAR	2	Query country or region id
QRY_DATE_FORMAT	CHAR	8	Query Date Format
QRY_DATE_SEPARATOR	CHAR	1	Query Date Separator
QRY_TIME_FORMAT	CHAR	8	Query Time Format
QRY_TIME_SEPARATOR	CHAR	1	Query Time Separator
QRY_DECIMAL_SEPARATOR	CHAR	1	Query Decimal Separa
QRY_COLLATING_SEQUENCE	VARCHAR	32	QRY_COLLATING_SEQUENCE
QRY_SORT_SEQUENCE_CCSID	INTEGER	4	Query Sort Sequence CCSID
QRY_SORT_SEQUENCE_WEIGHTS	CHAR	10	Query Sort Sequence Weights
QRY_SORT_SEQUENCE_SUB_CHAR	CHAR	3	Query Sort Sequence Substitution Charac
QRY_ROUND	CHAR	3	Use Rounding
QRY_IGNORE_DEC_ERRORS	CHAR	3	Ignore decimal data errors
QRY_IGNORE_SUB_WARNINGS	CHAR	3	Ignore substitution warnings
QRY_USE_COL_SEQ_FOR_COMPARES	CHAR	3	Use collating for all compares
QRY_OUTPUT_TYPE	VARCHAR	16	Query Output Type
QRY_OUTPUT_FORM	VARCHAR	16	Query Output Form
QRY_LINEWRAP	CHAR	3	Line Wrapping
QRY_WRAP_WIDTH	SMALLINT	2	Wrapping Width
QRY_RECORD_ONE_PAGE	CHAR	3	Record on one page
QRY_PRINT_DEF	CHAR	3	Print Query Definition
QRY_DESCRIPTION_TEXT	CHAR	50	QRY_DESCRIPTION_TEXT
QRY_OWNER	CHAR	10	Query Owner
QRY_PRIMARY_GROUP	CHAR	10	QRY_PRIMARY_GROUP
QRY_CREATION_DATE	TIMESTAMP	10	QRY_CREATION_DATE
QRY_CREATOR	CHAR	10	QRY_CREATOR
QRY_CHANGE_DATE	TIMESTAMP	10	QRY_CHANGE_DATE
QRY_LAST_USED_DATE	TIMESTAMP	10	QRY_LAST_USED_DATE
QRY_DAYS_USED	INTEGER	4	QRY_DAYS_USED
QRY_RESET_DATE	TIMESTAMP	10	QRY_RESET_DATE
QRY_SIZE	BIGINT	8	QRY_SIZE
QRY_SAVE_DATE	TIMESTAMP	10	QRY_SAVE_DATE
QRY_RESTORE_DATE	TIMESTAMP	10	QRY_RESTORE_DATE



QD_SELECTED_FILES (QD_SLCFIL)

This table contains one row per file that is part of the query definition. It provides information such as library, file, and member names.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID (use to join to QD_SUMMARY)
QRY_SELECTED_FILE_ID	CHAR	6	QRY_SELECTED_FILE_ID
QRY_SELECTED_FILE_NAME	CHAR	10	QRY_SELECTED_FILE_NAME
QRY_SELECTED_FILE_LIBRARY	CHAR	10	QRY_SELECTED_FILE_LIBRARY
QRY_SELECTED_FILE_MEMBER	CHAR	10	QRY_SELECTED_FILE_MEMBER
QRY_SELECTED_FILE_RECORD_FORMAT	CHAR	10	QRY_SELECTED_FILE_RECORD_FORMAT



QD_JOIN_TESTS (QD_JOINTST)

This table contains one row per join test that is part of the query definition. It provides information such as the fields and the test operator.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID (use to join to QD_SUMMARY)
QRY_JOIN_TYPE	CHAR	50	QRY_JOIN_TYPE
QRY_JOIN_FROM_FIELD	CHAR	20	QRY_JOIN_FROM_FIELD
QRY_JOIN_TEST	CHAR	10	QRY_JOIN_TEST
QRY_JOIN_TO_FIELD	CHAR	20	QRY_JOIN_TO_FIELD



QD_RESULT_FIELDS (QD_RSLFLD)

This table contains one row per result (derived) field that is part of the query definition. It provides information about the result fields define in each Query/400 definition.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID (use to join to QD_SUMMARY)
QRY_RESULT_FIELD_NAME	CHAR	10	QRY_RESULT_FIELD_NAME
QRY_RESULT_FIELD_EXPRESSION	VARCHAR	256	QRY_RESULT_FIELD_EXPRESSION
QRY_RESULT_FIELD_COLUMN_HEADING	VARCHAR	20	QRY_RESULT_FIELD_COLUMN_HEADING
QRY_RESULT_FIELD_LENGTH	INTEGER	4	QRY_RESULT_FIELD_LENGTH
QRY_RESULT_FIELD_DECIMAL_POSITIONS	INTEGER	4	QRY_RESULT_FIELD_DECIMAL_POSITIONS



QD_SELECT_RECORD_TESTS (QD_SLCRCD)

This table contains one row per record selection test that is part of the query definition. It provides information such as the field name, test operator, and test value.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID (use to join to QD_SUMMARY)
QRY_SELECTED_RECORD_AND_OR	CHAR	3	QRY_SELECTED_RECORD_AND_OR
QRY_SELECTED_RECORD_FIELD_NAME	CHAR	20	QRY_SELECTED_RECORD_FIELD_NAME
QRY_SELECTED_RECORD_FIELD_TEST	CHAR	10	QRY_SELECTED_RECORD_FIELD_TEST
QRY_SELECTED_RECORD_FIELD_VALUE	VARCHAR	4096	QRY_SELECTED_RECORD_FIELD_VALUE



QD_ORDER_SELECTED_FIELDS (QD_ORDFLD)

This table contains one row per ordered or selected field that is part of the query definition. It provides information such as the sequence, field name, and ordering criteria.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID (use to join to QD_SUMMARY)
QRY_FIELD_SEQUENCE_NUMBER	INTEGER	4	QRY_FIELD_SEQUENCE_NUMBER
QRY_FIELD_NAME	CHAR	20	QRY_FIELD_NAME
QRY_FIELD_SORT_PRIORITY	INTEGER	4	QRY_FIELD_SORT_PRIORITY
QRY_FIELD_ORDER	CHAR	1	QRY_FIELD_ORDER
QRY_FIELD_BREAK_LEVEL	CHAR	4	QRY_FIELD_BREAK_LEVEL
QRY_FIELD_TEXT	VARCHAR	128	QRY_FIELD_TEXT



QD_REPORT_COLUMN_FORMAT (QD_COLFMT)

This table contains one row per selected field that is part of the query definition. It provides information such as the field's format and edit words.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID (use to join to QD_SUMMARY)
QRY_CF_FIELD_NAME	CHAR	20	QRY_CF_FIELD_NAME
QRY_CF_SUMMARY_FUNCTIONS	CHAR	10	QRY_CF_SUMMARY_FUNCTIONS
QRY_CF_COLUMN_SPACING	SMALLINT	2	QRY_CF_COLUMN_SPACING
QRY_CF_COLUMN_HEADINGS	VARCHAR	64	QRY_CF_COLUMN_HEADINGS
QRY_CF_COLUMN_LENGTH	SMALLINT	2	QRY_CF_COLUMN_LENGTH
QRY_CF_COLUMN_DECIMAL_POSITION	CHAR	4	QRY_CF_COLUMN_DECIMAL_POSITION
QRY_CF_NULL_CAPABLE	CHAR	1	QRY_CF_NULL_CAPABLE
QRY_CF_OVERRIDE_COLUMN_LENGTH	SMALLINT	2	QRY_CF_OVERRIDE_COLUMN_LENGTH
QRY_CF_OVERRIDE_COLUMN_DECIMAL_POSITION	SMALLINT	2	QRY_CF_OVERRIDE_COLUMN_DECIMAL_POSITION
QRY_CF_NUMERIC_EDIT	CHAR	3	QRY_CF_NUMERIC_EDIT
QRY_CF_OVERRIDE_COLUMN_HEADING	CHAR	3	QRY_CF_OVERRIDE_COLUMN_HEADING
QRY_CF_OVERRIDE_COLUMN_SPACING	CHAR	3	QRY_CF_OVERRIDE_COLUMN_SPACING
QRY_CF_OVERRIDE_LENGTH_DECIMAL_POSITION	CHAR	3	QRY_CF_OVERRIDE_LENGTH_DECIMAL_POSITION
QRY_CF_EDIT_OPTION	CHAR	3	QRY_CF_EDIT_OPTION
QRY_CF_DECIMAL_POINT_CHARACTER	CHAR	3	QRY_CF_DECIMAL_POINT_CHARACTER
QRY_CF_THOUSANDS_SEPARATOR	CHAR	3	QRY_CF_THOUSANDS_SEPARATOR
QRY_CF_SHOW_NEGATIVE_SIGN	CHAR	3	QRY_CF_SHOW_NEGATIVE_SIGN
QRY_CF_LEFT_NEGATIVE_SIGN	CHAR	3	QRY_CF_LEFT_NEGATIVE_SIGN
QRY_CF_RIGHT_NEGATIVE_SIGN	CHAR	3	QRY_CF_RIGHT_NEGATIVE_SIGN
QRY_CF_SHOW_CURRENCY_SYMBOL	CHAR	3	QRY_CF_SHOW_CURRENCY_SYMBOL
QRY_CF_LEFT_CURRENCY_SYMBOL	CHAR	4	QRY_CF_LEFT_CURRENCY_SYMBOL
QRY_CF_RIGHT_CURRENCY_SYMBOL	CHAR	4	QRY_CF_RIGHT_CURRENCY_SYMBOL
QRY_CF_PRINT_ZERO_VALUE	CHAR	3	QRY_CF_PRINT_ZERO_VALUE
QRY_CF_REPLACE_LEADING_ZERO	CHAR	3	QRY_CF_REPLACE_LEADING_ZERO
QRY_CF_REPLACE_LEADING_ZERO_WITH	CHAR	3	QRY_CF_REPLACE_LEADING_ZERO_WITH
QRY_CF_SINGLE_LEADING_ZERO	CHAR	3	QRY_CF_SINGLE_LEADING_ZERO
QRY_CF_DATE_TIME_SEPARATOR	CHAR	1	QRY_CF_DATE_TIME_SEPARATOR
QRY_CF_EDIT_CODE	CHAR	1	QRY_CF_EDIT_CODE
QRY_CF_OPTIONAL_EDIT_CODE_MODIFIER	CHAR	1	QRY_CF_OPTIONAL_EDIT_CODE_MODIFIER
QRY_CF_EDIT_WORD	VARCHAR	50	QRY_CF_EDIT_WORD
QRY_CF_EDIT_WORD_SUMMARY	VARCHAR	100	QRY_CF_EDIT_WORD_SUMMARY



QD_SPECIAL_CONDITIONS (QD_SPCOND)

This table contains any special conditions that exist with the query at the time the tool was run.

Examples of this include the following:

- This output database file definition is not defined due to the query definition being incomplete
- All records selected by default
- A file with this format cannot be taken to a release prior to V2R1M1
- All records selected by default

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID (use to join to QD_SUMMARY)
QRY_SPECIAL_COND	VARCHAR	256	QRY_SPECIAL_COND

QD_REPORT_BREAKS (QD_RPTBRK)

This table contains one row per report break that is part of the query definition. It provides information such as the break level and text.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID (use to join to QD_SUMMARY)
QRY_BREAK_LEVEL	SMALLINT	2	QRY_BREAK_LEVEL
QRY_BREAK_NEW_PAGE	CHAR	3	QRY_BREAK_NEW_PAGE
QRY_BREAK_SUPPRESS_SUMMARIES	CHAR	3	QRY_BREAK_SUPPRESS_SUMMARIES
QRY_BREAK_TEXT	VARCHAR	128	QRY_BREAK_TEXT

QD_DATABASE_OUTPUT (QD_DBOUT)

If the output type for the query is option 3 (database file), this table contains information such as the name and library of the output file that is to hold the results when the query is executed.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	Query ID (use to join to QD_SUMMARY)
QRY_DATABASE_OUTPUT_FILE	CHAR	10	QRY_DATABASE_OUTPUT_FILE
QRY_DATABASE_OUTPUT_LIBRARY	CHAR	10	QRY_DATABASE_OUTPUT_LIBRARY
QRY_DATABASE_OUTPUT_MEMBER	CHAR	10	QRY_DATABASE_OUTPUT_MEMBER
QRY_DATABASE_DATA_IN_FILE	VARCHAR	16	QRY_DATABASE_DATA_IN_FILE
QRY_DATABASE_NEW_FILE_AUTH	CHAR	10	QRY_DATABASE_NEW_FILE_AUTH
QRY_DATABASE_NEW_FILE_TEXT	CHAR	50	QRY_DATABASE_NEW_FILE_TEXT
QRY_DATABASE_OUTPUT_RCD_LEN	INTEGER	4	QRY_DATABASE_OUTPUT_RCD_LEN
QRY_DATABASE_OUTPUT_CCSD	INTEGER	4	QRY_DATABASE_OUTPUT_CCSD



QD_PRINTER_OUTPUT (QD_PRTOUT)

If the output type for the query is option 2 (printer file), this table contains information such as the printer device, length and width of the report, and the form type.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	QRY_ID
QRY_PRINTER_DEVICE	CHAR	10	QRY_PRINTER_DEVICE
QRY_PRINTER_REPORT_LENGTH	CHAR	20	QRY_PRINTER_REPORT_LENGTH
QRY_PRINTER_REPORT_WIDTH	CHAR	20	QRY_PRINTER_REPORT_WIDTH
QRY_PRINTER_REPORT_START_LINE	CHAR	20	QRY_PRINTER_REPORT_START_LINE
QRY_PRINTER_REPORT_END_LINE	CHAR	20	QRY_PRINTER_REPORT_END_LINE
QRY_PRINTER_REPORT_SPACING	CHAR	20	QRY_PRINTER_REPORT_SPACING
QRY_PRINTER_PRINT_DEFINITION	CHAR	3	QRY_PRINTER_PRINT_DEFINITION
QRY_PRINTER_SPOOL_OUTPUT	CHAR	50	QRY_PRINTER_SPOOL_OUTPUT
QRY_PRINTER_FORM_TYPE	CHAR	50	QRY_PRINTER_FORM_TYPE
QRY_PRINTER_COPIES	SMALLINT	2	QRY_PRINTER_COPIES
QRY_PRINTER_HOLD_REPORT	CHAR	50	QRY_PRINTER_HOLD_REPORT
QRY_PRINTER_PRINT_COVER_PAGE	CHAR	20	QRY_PRINTER_PRINT_COVER_PAGE
QRY_PRINTER_PRINT_STANDARD_PAGE_HEADING	CHAR	20	QRY_PRINTER_PRINT_STANDARD_PAGE_HEADING

QD_PRINTER_COVER_PAGE_HEADING_FOOTING (QD_PRTCVR)

If the output type for the query is option 2 (printer file), this table contains information such as the cover page title, and report page headings and footings.

COLUMN_NAME	DATA_TYPE	LENGTH	COLUMN_HEADING
QRY_ID	INTEGER	4	QRY_ID
QRY_PRTCVR_RECORD_TYPE	CHAR	8	QRY_PRTCVR_RECORD_TYPE
QRY_PRTCVR_SEQ_NUM	SMALLINT	2	QRY_PRTCVR_SEQ_NUM
QRY_PRTCVR_DATA	CHAR	80	QRY_PRTCVR_DATA



Analyzing the results

Once the command is run and the tables are populated, we can start to review this for trends. This part is more art than science. The queries below are just a few examples to help you get started in looking for information about your reporting environment. These sample queries assume you directed the output to library QGPL.

This is **not** an all-inclusive list and should not be your only points of data in developing your reporting modernization strategy. If you need assistance with this migration or usage of tools such as DB2 Web Query, please contact the IBM STG Lab Based Services team at stgls@us.ibm.com for assistance.

Finding the libraries containing the queries

```
SELECT COUNT(*) AS COUNT, QRY_LIBRARY
FROM QGPL.QD_SUMMARY
GROUP BY QRY_LIBRARY
ORDER BY 1 DESC;
```

Finding the files used in the queries

```
SELECT COUNT(*) AS COUNT, QRY_SELECTED_FILE_LIBRARY, QRY_SELECTED_FILE_NAME,
QRY_SELECTED_FILE_MEMBER
FROM QGPL.QD_SELECTED_FILES
GROUP BY QRY_SELECTED_FILE_LIBRARY, QRY_SELECTED_FILE_NAME, QRY_SELECTED_FILE_MEMBER
ORDER BY 1 DESC;
```

Finding the selection criteria used in the queries

```
SELECT COUNT(*) AS COUNT, QRY_SELECTED_RECORD_FIELD_NAME,
QRY_SELECTED_RECORD_FIELD_TEST,
QRY_SELECTED_RECORD_FIELD_VALUE
FROM QGPL.QD_SELECT_RECORD_TESTS
GROUP BY QRY_SELECTED_RECORD_FIELD_NAME,
QRY_SELECTED_RECORD_FIELD_TEST,
QRY_SELECTED_RECORD_FIELD_VALUE
ORDER BY 1 DESC;
```

Finding the fields used for selection criteria in the queries

```
SELECT COUNT(*) AS COUNT,
SUBSTRING(QRY_SELECTED_RECORD_FIELD_NAME FROM (POSITION('.') IN
QRY_SELECTED_RECORD_FIELD_NAME)+1)) AS FIELD_NAME
FROM QGPL.QD_SELECT_RECORD_TESTS
GROUP BY SUBSTRING(QRY_SELECTED_RECORD_FIELD_NAME FROM (POSITION('.') IN
QRY_SELECTED_RECORD_FIELD_NAME)+1))
ORDER BY 1 DESC;
```



Finding the result fields used in the queries

```
SELECT COUNT(*) AS COUNT, QRY_RESULT_FIELD_NAME, QRY_RESULT_FIELD_EXPRESSION  
FROM QGPL.QD_RESULT_FIELDS  
GROUP BY QRY_RESULT_FIELD_NAME, QRY_RESULT_FIELD_EXPRESSION  
ORDER BY 1 DESC;
```



Support

This tool is provided “as is” and considered a Type II Material. Type II Materials are those, created by IBM, in which IBM or third parties have all right, title, and interest (including ownership of copyright). IBM grants you an irrevocable, nonexclusive, worldwide, license to use, execute, display, perform, and distribute, within your Enterprise only, copies of Type II Materials. You can submit informal requests or problems to qu2@us.ibm.com but there is no guarantee of response and/or corrective code.