IBM i Services for the Developer

—

Sue Romano – slromano@us.ibm.com
Db2 for i development

2020
iSight

SEMIUG

© 2020 IBM

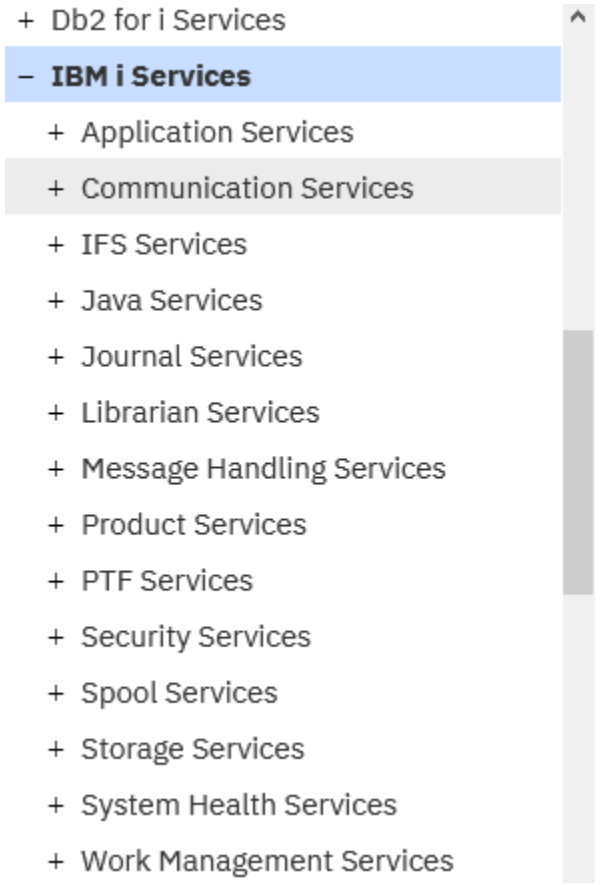
What are IBM i Services

- SQL views, functions, and procedures
- Provide SQL access to system information
  - Similar to what CL commands provide
  - Similar to what APIs provide
- Use the power of the SQL query engine to access and manipulate this data

© 2020 IBM Corporation

Documentation for IBM i Services

- Access directly in Knowledge Center



There are many system services that can be accessed through system-provided SQL v
provide an SQL interface to access, transform, order, and subset the information withc

- **Application Services**
  These procedures, functions, and views provide interface information that can be
- **Communication Services**
  These services provide communication information.
- **IFS Services**
  These table functions provide information about the integrated file system (IFS).
- **Java Services**
  This view and procedure provide Java information and JVM management options.
- **Journal Services**
  This function and view provide journal information.

© 2020 IBM Corporation

Documentation for IBM i Services

- Access through the Db2 for i Technology Updates wiki
  - http://ibm.biz/Db2foriServices

IBM i Service	Type of Service	IBM i 7.4	IBM i 7.3	IBM i 7.2
Application Services				
QSYS2.BOUND_MODULE_INFO	View	SF99704 Level 4	SF99703 Level 16	Not Supported
QSYS2.BOUND_SRVPGM_INFO	View	SF99704 Level 4	SF99703 Level 16	Not Supported
QSYS2.CLEAR_DATA_QUEUE()	Procedure	SF99704 Level 4	SF99703 Level 16	Not Supported
QSYS2.DATA_AREA_INFO()	Table function	Base	SF99703 Level 15	Not Supported

© 2020 IBM Corporation

Program information

© 2020 IBM Corporation

Examine *PGM and *SRVPGM objects

- **DSPPGM and DSPSRVPGM return information for a single object**

- **Details returned by four views**
  - PROGRAM_INFO: *BASIC, *SIZE, and *SIGNATURE
  - BOUND_MODULE_INFO: *MODULE
  - BOUND_SRVPGM_INFO: *SRVPGM
  - PROGRAM_EXPORT_IMPORT_INFO: *PROCEXP, *DTAEXP, *ACTGRPEXP, and *ACTGRPIMP

© 2020 IBM Corporation

PROGRAM_INFO

- One row for every *PGM and *SRVPGM
- ILE and OPM programs

```
select activation_group, count(*) as activation_group_count
 from qsys2.program_info
 where program_library = 'APPLIB' and program_type = 'ILE'
 group by activation_group
 order by 2 desc;
```

ACTIVATION_GROUP	ACTIVATION_GROUP_COUNT
*CALLER	68
QILE	7
*NEW	4

© 2020 IBM Corporation

BOUND_MODULE_INFO

- One row for each module bound into a program or service program

```
select optimization_level,
       count(*) as optimization_level_count
  from qsys2.bound_module_info
  where program_library = 'APPLIB'
  group by optimization_level order by 2 desc;
```

OPTIMIZATION_LEVEL	OPTIMIZATION_LEVEL_COUNT
10	79
40	66

© 2020 IBM Corporation

Compare bound module information


```
with tst as (select program_name, bound_module,
                    optimization_level, debug_data
               from qsys2.bound_module_info
               where program_library =  'TESTLIB'),
  prd as (select program_name, bound_module,
                  optimization_level, debug_data
               from qsys2.bound_module_info
               where program_library = 'PRODLIB')
select * from tst, prd
  where tst.program_name = prd.program_name
    and tst.bound_module = prd.bound_module
    and (tst.optimization_level <> prd.optimization_level or
          tst.debug_data <> prd.debug_data);
```

PROGRAM_NAME	BOUND_MODULE	OPTIMIZATION_LEVEL	DEBUG_DATA	OPTIMIZATION_LEVEL	DEBUG_DATA
DATEPROC	DATEPROC	10	*NO	10	*YES

© 2020 IBM Corporation

BOUND_SRVPGM_INFO

- One row for every service program bound into a program or service program

```
select *
 from qsys2.bound_srvpgm_info
   where program_library = 'APPLIB' and
         program_name like '%RPG%';
```

PROGRAM_NAME	OBJECT_TYPE	BOUND_SERVICE_PROGRAM_LIBRARY	BOUND_SERVICE_PROGRAM	BOUND_SERVICE_PROGRAM_ACTIVATION
BADRPGINS	*PGM	QSYS	QRNXIE	*IMMED
BADRPGINS	*PGM	QSYS	QRNXIO	*IMMED
BADRPGINS	*PGM	QSYS	QRNXUTIL	*IMMED
BADRPGINS	*PGM	QSYS	QLEAWI	*IMMED
RPGINSERT	*PGM	QSYS	QRNXIE	*IMMED
RPGINSERT	*PGM	QSYS	QRNXIO	*IMMED

© 2020 IBM Corporation

PROGRAM_EXPORT_IMPORT_INFO

- One row for each data or procedure export and import

```
select *
  from qsys2.program_export_import_info
  where program_library = 'QSYS' and
        symbol_name = 'Qp0zLprintf';
```

PROGRAM_LIBRARY	PROGRAM_NAME	OBJECT_TYPE	SYMBOL_NAME	SYMBOL_USAGE	Argument Optimization
QSYS	QP0ZCPA	*SRVPGM	Qp0zLprintf	*PROCEXP	*NO

© 2020 IBM Corporation

SYSPROGRAMSTAT

- One row for each *PGM, *SRVPGM, and *MODULE that contains SQL statements
- More complete SQL details than BOUND_MODULE_INFO

```
select *
  from qsys2.sysprogramstat
  where program_schema = 'SLR' and
        number_external_routines > 0;
```

SYSTEM_PROGRAM_NAME	PROGRAM_TYPE	NUMBER_EXTERNAL_ROUTINES
EXTPROC_C	*PGM	32
E1	*PGM	1
EF03CO02S	*SRVPGM	4
EF03CO03S	*SRVPGM	1
EPROC1	*SRVPGM	12

© 2020 IBM Corporation

SYSPROGRAMSTMTSTAT

- One row for each embedded SQL statement in a *PGM, *SRVPGM, and *MODULE
- Includes statement text and other statement-related details

```
select *
  from qsys2.sysprogramstmtstat
  where program_schema = 'TOYSTORE' and
        statement_text like 'UPDATE%' and
        statement_text like '%EMPLOYEE %';
```

PROGRAM_NAME	PROGRAM_TYPE	MODULE_NAME	NUMBER_TIMES_EXECUTED
GTTUPD	*PGM	GTTUPD	2
PACKAGES_AI_HB_ONH_CALC_FIELDS	*PGM	PACKA00001	31
REWRITEM	*PGM	REWRITEM	0
UPDATE_PROC1	*PGM	UPDAT00001	0
UPDATE_PROC1	*PGM	UPDAT00001	0

© 2020 IBM Corporation

System interfaces

© 2020 IBM Corporation

QCMDEXC

- SQL procedure that executes a CL command

```
call qsys2.qcmdexc('CHGJOB CCSID(37)');
```

- Use expressions to build the CL command string

```
call qsys2.qcmdexc('ADDLIBLE ' CONCAT libname);
```


© 2020 IBM Corporation

LPRINTF

- Print to the joblog using SQL

```
call systools.lprintf('Hello There! Have a great ' concat
                                    dayname(current date));
```

928971/QUSER/QZDASOINIT Job Log - sync3.rch.stglabs.ibm.com

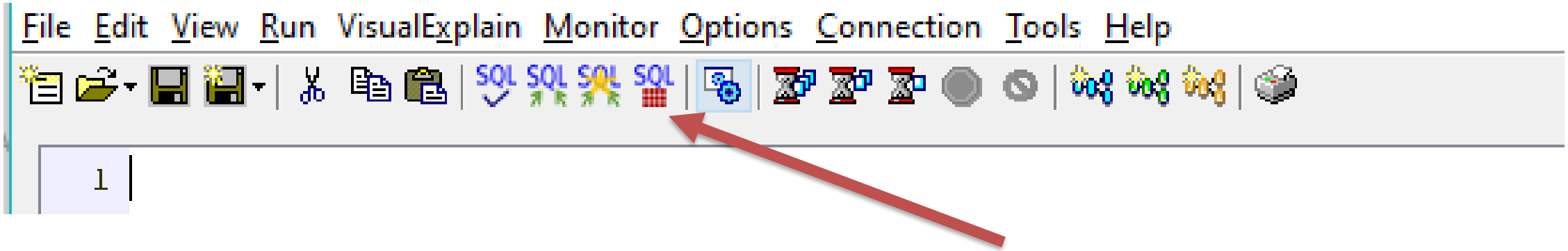
File View

Type	Message ID	Severity	Message
INFORMATIONAL	-	0	Hello There! Have a great Friday
INFORMATIONAL	SQL799C	10	The following special registers have been set: CLIENT_ACCTNC
COMPLETION	CPC1129	0	Job 928971/QUSER/QZDASOINIT changed by SCOTTF.

© 2020 IBM Corporation

LPRINTF – get source for procedure

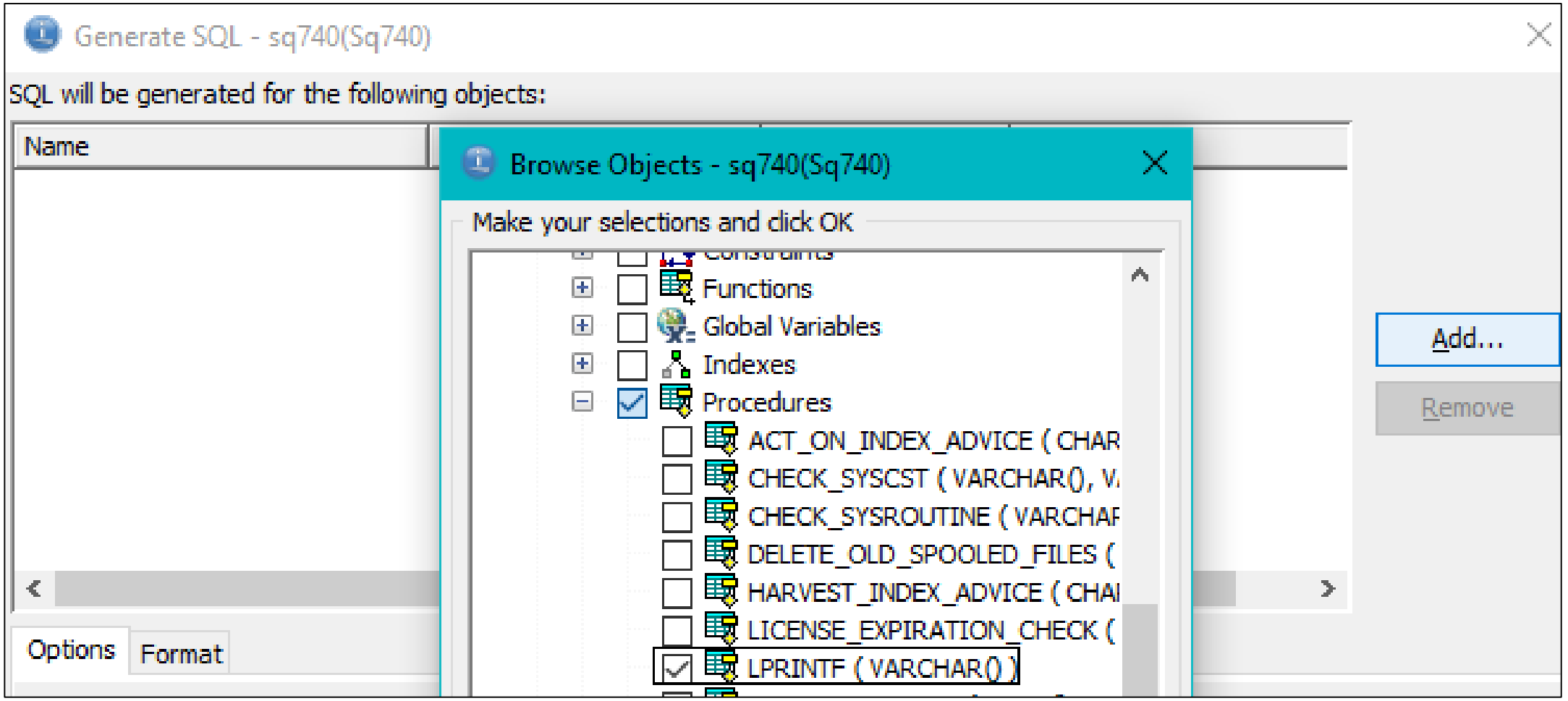
- Use ACS to extract the procedure source using Insert Generated SQL



© 2020 IBM Corporation

LPRINTF – get source for procedure

- Select SYSTOOLS procedure LPRINTF



© 2020 IBM Corporation

LPRINTF – get source for procedure

2020 iSight

```
CREATE PROCEDURE SYSTOOLS.LPRINTF (IN PRINT_STRING VARCHAR(1000) CCSID 37 )
  SET OPTION BINDOPT = 'BNDSRVPGM(QSYS/QP0ZCPA)'
BEGIN
-- Additional setup steps are required before creating this procedure.
-- Create a C source member that contains the code for LPRINTF.
-- The procedure will embed this with the INCLUDE statement and bind to the
appropriate service program using BINDOPT.
--CL:CRTSRCPF FILE(QTEMP/QCSRC) RCDLEN(160) MBR(*NONE);
--CL:ADDPFM FILE(QTEMP/QCSRC) MBR(LPRINTF);
--INSERT INTO QTEMP.QCSRC(SRCSEQ,SRCDTA) VALUES
-- (1,'{'),
-- (2,'extern int Qp0zLprintf (char *format, ...);'),
-- (3,'Qp0zLprintf("%.*s\n",LPRINTF.PRINT_STRING.LEN,LPRINTF.PRINT_STRING.DAT);'),
-- (4,'}');
IF PRINT_STRING IS NOT NULL THEN
INCLUDE QTEMP / QCSRC ( LPRINTF ) ;
END IF ;
END  ;
```


© 2020 IBM Corporation

Finding messages

© 2020 IBM Corporation

MESSAGE_FILE_DATA

- Returns one row for each message in a message file
- SQL alternative to the Display Message Description (DSPMSGD) CL command

```
select * from qsys2.message_file_data
  where message_file_library = 'QSYS' and
        message_file         = 'QSQLMSG' and
        (message_text like '%JSON%' or
         message_second_level_text like '%JSON%');
```

MESSAGE_ID	MESSAGE_TEXT	MESSAGE_SECOND_LEVEL_TEXT
SQL0097	Use of data type not valid.	Cause . . . . . : A data type specified in the s
SQL0107	&1 too long. Maximum &2 characters.	Cause . . . . . : The name or string beginning w
SQL0189	Coded Character Set Identifier &1 not valid.	Cause . . . . . : Coded Character Set Identifier
SQL0255	Function not supported for query.	Cause . . . . . : The reason code is &1: 1 -- Sc
SQL0612	&1 is a duplicate column name.	Cause . . . . . : One of the following errors ha
SQL0683	Clause not valid for specified type.	Cause . . . . . : One of the following has occur
SQ16400	Duplicate key name &1.	Cause . . . . . : The generation of JSON text re
SQ16402	JSON data is not valid.	Cause . . . . . : The data supplied by the JSON-

© 2020 IBM Corporation

Finding objects

© 2020 IBM Corporation

OBJECT_STATISTICS

- List details about objects in libraries, like DSPOBJD
- 69 columns of information

- Return all objects in TOYSTORE created at least 2 years ago

```
select *
   from table(qsys2.object_statistics('TOYSTORE', '*ALL'))
   where objcreated < current date - 2 years;
```


© 2020 IBM Corporation

OBJECT_STATISTICS

- Return only file names, no extended details, for all files in TOYSTORE

```
select *
  from table(qsys2.object_statistics('TOYSTORE','*FILE',
                '*ALLSIMPLE'));
```

- Return a list of all libraries with no extended details

```
select *
  from table(qsys2.object_statistics('*ALLSIMPLE','*LIB'));
```

- Return all the information for a specific file in TOYSTORE

```
select *
  from table(qsys2.object_statistics('TOYSTORE', '*FILE',
                'EMP'));
```


© 2020 IBM Corporation

OBJECT_STATISTICS

- *ALLUSR, *ALLAVL, *ALLUSRAVL, *ALL library options

```
select '*ALLUSR' search_style, count(*) lib_count
  from table (qsys2.object_statistics('*ALLUSR', '*LIB')) union
select '*ALLAVL', count(*)
  from table (qsys2.object_statistics('*ALLAVL', '*LIB')) union
select '*ALLUSRAVL', count(*)
  from table (qsys2.object_statistics('*ALLUSRAVL', '*LIB')) union
select '*ALL', count(*)
  from table (qsys2.object_statistics('*ALL', '*LIB'))
order by 2 desc;
```

SEARCH_STYLE	LIB_COUNT
*ALLAVL	2421
*ALL	2369
*ALLUSRAVL	2188
*ALLUSR	2140

- ❑ *ALL – All libraries
- ❑ *ALLAVL – All libraries in all available ASP groups
- ❑ *ALLUSRAVL – All user libraries in all available ASP groups
- ❑ *ALLUSR – All user libraries in *SYSBAS and the current thread's ASP group

OBJECT_STATISTICS

- Which IBM CL commands have had their command parameter defaults changed using CHGCMDDFT
- Identify commands that should have CHGCMDDFT added to QSTRUP

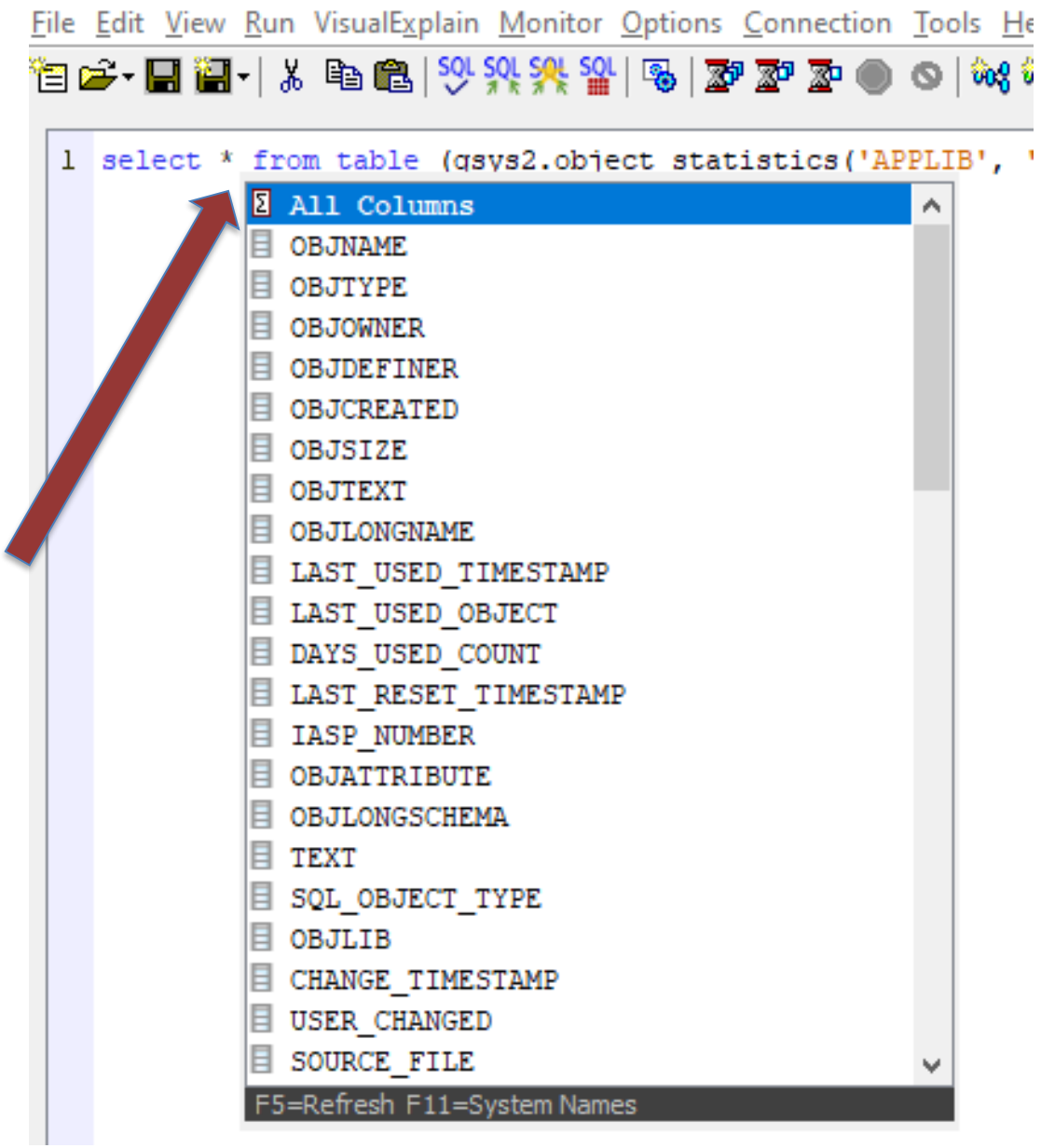
```
select *
  from table(qsys2.object_statistics('QSYS', '*CMD'))
  where APAR_ID = 'CHGDFT';
```

OBJNAME	OBJTYPE	OBJOWNER	OBJTEXT	CHANGE_TIMESTAMP
CRTUSRPRF	*CMD	QSYS	Create User Profile	2019-09-13 02:30:13.000000
STRDBG	*CMD	QSYS	Start Debug	2019-09-13 03:22:15.000000

© 2020 IBM Corporation

ACS prompting

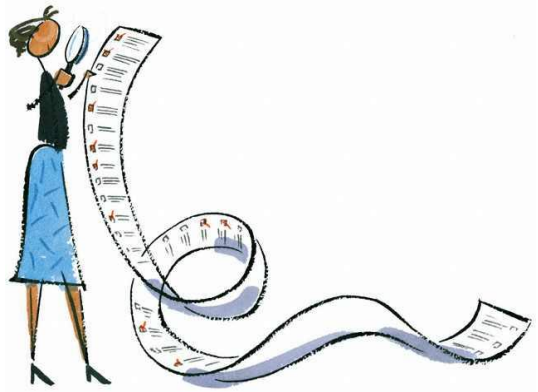
- Position cursor at location in valid SQL statement
  - Control+Space
  - OR –
  - F4



© 2020 IBM Corporation

Additional object list views

- USER_INFO
  - List of user profiles (WRKUSRPRF)
- JOB_QUEUE_INFO
  - List of job queues (WRKJOBQ)
- SUBSYSTEM_INFO
  - List of subsystems (WRKSBS)
- JOB_DESCRIPTION_INFO
  - List of job descriptions (WRKJOBD)



© 2020 IBM Corporation

IFS_OBJECT_STATISTICS

- SQL Alternative to RTVDIRINF and WRKLNK
- 87 columns of data returned

- Return object identified by a path name
  - OR –
- Return all objects that can be accessed starting with a path name

© 2020 IBM Corporation

Search an IFS tree

- Find objects based on a partial name

```
select * from table(qsys2.ifs_object_statistics('/usr'))
  where path_name like '%.txt'
  order by path_name;
```

PATH_NAME	OBJECT_TYPE	CREATE_TIMESTAMP
/usr/install_out.txt	*STMF	2016-08-30 13:46:45
/usr/scottf/dlyjob CL source.txt	*STMF	2018-03-01 08:17:30
/usr/scottf/dlyjobCLsource.txt	*STMF	2018-03-01 08:35:08
/usr/local/guardium/install_out.txt	*STMF	2017-05-30 16:31:07
/usr/local/guardium/guard_itap.stderr.1.txt	*STMF	2017-05-30 16:31:27
/usr/local/guardium/guard_itap.stderr.2.txt	*STMF	2017-06-07 20:40:19

© 2020 IBM Corporation

Integrated File System

© 2020 IBM Corporation

Read from the IFS

- IFS_READ
  - character data in job CCSID
- IFS_READ_UTF8
  - data converted to UTF-8
- IFS_READ_BINARY
  - no conversion

- Three parameters
  - Path name – either absolute or relative
  - Line length – maximum length for a returned line
  - End of line – carriage return and line feed combinations, or none

© 2020 IBM Corporation

Read from the IFS

- Search the IFS

```
select o.path_name, i.*
  from table (qsys2.ifs_object_statistics(
            start_path_name => '/home/SCOTTF',
            subtree_directories => 'YES',
            object_type_list => '*ALLSTMF')) o,
       table (qsys2.ifs_read(
        path_name => path_name, end_of_line => 'ANY')) i
 where upper(line) like '%IFS%'
 order by 1,2;
```


© 2020 IBM Corporation

Read from the IFS

- Search the IFS

PATH_NAME	LINE_NUMBER	LINE
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	75	<a href="#9.12">9.12 Integrat
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	76	<a href="#9.12.1">9.12.1 IFS,
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	77	<a href="#9.12.2">9.12.2 IFS
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	78	<a href="#9.12.3">9.12.3 IFS
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	79	<a href="#9.12.4">9.12.4 IFS
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	80	<a href="#9.12.5">9.12.5 IFS
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	391	Extract the latest product zip archiv
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	1229	ifs Integrated File Syste
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	1270	ifs ifs
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	1373	9.1.33 IFS
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	1375	/PLUGIN=ifs /SYSTEM=<system>
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	1388	See section <a href="#9.12">9.12 Integrat
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	1795	To do this, you can extract the product z
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	1837	ifs Integrated File Syste
/home/SCOTTF/IBM/iAccessClient_Help/Base/en/GettingStarted_en.html	1877	ifs ifs

© 2020 IBM Corporation

Write to the IFS

- IFS_WRITE
  - data written is in job CCSID
- IFS_WRITE_UTF8
  - data written is UTF-8
- IFS_WRITE_BINARY
  - no conversion
- Five parameters
  - Path name – either absolute or relative
  - Line – the data to write
  - File CCSID, if creating a new file
  - Replace or Append
  - End of line – carriage return and line feed combinations, or none

© 2020 IBM Corporation

Write to the IFS

- Write a list of program names to the IFS

```
begin
  call qsys2.ifs_write(path_name => '/usr/pgms', line => '',
                end_of_line => 'NONE', overwrite => 'REPLACE');
  for select objname, objtype
       from table (qsys2.object_statistics(
                     'SLR','SRVPGM PGM','*ALLSIMPLE'))
       order by objname
    do
      call qsys2.ifs_write(path_name => '/usr/pgms',
             line => cast(objtype as char(7)) concat ' '
                          concat objname,
             overwrite => 'APPEND', end_of_line => 'CRLF');
  end for;
end;
```


© 2020 IBM Corporation

Write to the IFS

- What did we write?

```
select * from table (qsys2.ifs_read( 'usr/pgms'));
```

LINE_NUMBER	LINE
30	*PGM CBLCALL
31	*SRVPGM CHARVAR
32	*SRVPGM CHAR37
33	*SRVPGM CHAR65535
34	*PGM CHECK00001
35	*PGM CHECK00002
36	*PGM CHGPARM
37	*PGM CONNHV
38	*PGM CONNLIT
39	*PGM CONNREFS
40	*PGM CONSUMER
41	*PGM CPROCRPG
42	*PGM CPROCRPG2
43	*PGM CREATEREFS

© 2020 IBM Corporation

Data areas

© 2020 IBM Corporation

DATA_AREA_INFO view

- Alternative to the Retrieve Data Area (RTVDTAARA) command

```
select * from qsys2.data_area_info
  where data_area_library = 'GONZO333';
```

DATA_AREA_LIBRARY	DATA_AREA_NAME	DATA_AREA_TYPE	LENGTH	DECIMAL_POSITIONS	DATA_AREA_VALUE
GONZO333	MYBOOLEAN	*LGL	1	-1	
GONZO333	MYPLACE	*CHAR	100	-	Place holder : 555

© 2020 IBM Corporation

DATA_AREA_INFO table function

- QTEMP, *CURLIB, and *LIBL special cases require the UDTF
- Can access *LDA, *PDA, and *GDA data areas

```
select * from
 table(qsys2.data_area_info(DATA_AREA_LIBRARY => 'QTEMP',
                            DATA_AREA_NAME    => 'SECRET'));
```

DATA_AREA_LIBRARY	DATA_AREA_NAME	DATA_AREA_TYPE	LENGTH	DECIMAL_POSITIONS	DATA_AREA_VALUE
QTEMP	SECRET	*CHAR	100		-The answer is 42

© 2020 IBM Corporation

Change data area

- SQL procedure to implement CHGDTAARA CL command

```
create or replace procedure coolstuff.chgdtaara (
  data_area_library varchar(10),
  data_area_name varchar(10),
  new_value varchar(2000))
begin
  declare cmd_string varchar(2500);
  set cmd_string = 'CHGDTAARA ' concat data_area_library
    concat '/' concat data_area_name concat ' VALUE('''
    concat new_value concat ''') ';
--call systools.lprintf (cmd_string); -- use for debug
  call qsys2.qcmdexc(cmd_string);
end;
```


© 2020 IBM Corporation

Data queues

© 2020 IBM Corporation

Data Queues – SQL Style

SQL alternatives to Data Queue APIs –

- Receive Data Queue Description (QMHQRDQD)
- Send Data Queue (QSNDDTAQ)
- Receive Data Queue (QRCVDTAQ)
- Clear Data Queue (QCLRDTAQ)

Data Queue – IBM i Services:

- **DATA_QUEUE_INFO** – Review the attributes
- **DATA_QUEUE_ENTRIES** – Look at all the data
- **SEND_DATA_QUEUE** – Put character data
- **SEND_DATA_QUEUE_UTF8** – Put Unicode (JSON) data
- **SEND_DATA_QUEUE_BINARY** – Put binary data
- **RECEIVE_DATA_QUEUE** – Extract data
- **CLEAR_DATA_QUEUE** – Remove data

© 2020 IBM Corporation

DATA_QUEUE_INFO view

- Details for data queues
  - FIFO/LIFO/KEYED
  - Key length
  - Local or remote data queue
  - Number of messages currently on queue
  - Maximum messages on queue

© 2020 IBM Corporation

Data Queues and SQL

- Create a data queue, and send character data to the queue

```
cl:CRTDTAQ DTAQ(SCOTTF/SQLCANDOIT) MAXLEN(64000)
SENDERID(*YES) SIZE(*MAX2GB) TEXT('DQueue Time');

--
-- Send character data
--
call qsys2.send_data_queue(
        data_queue_library => 'SCOTTF',
        data_queue   => 'SQLCANDOIT',
        message_data => 'Ready for the ' concat 'Big time');
```


© 2020 IBM Corporation

Data Queues and SQL

- Retrieve and remove an entry

```
select * from table(qsys2.receive_data_queue(
        data_queue_library => 'SCOTTF',
        data_queue         => 'SQLCANDOIT'));
```

MESSAGE_DATA	MESSAGE_DATA_BINARY	KEY_DATA	SENDER_JOB_NAME	SENDER_CURRENT_USER
Ready for the Big time	D9858184A8408696994...	-	990035/QUSER/QZDASOINIT	SCOTTF

© 2020 IBM Corporation

Data Queues and SQL

- Create a remote data queue, send data to the remote IBM i

```
cl:CRTDTAQ DTAQ(SCOTTF/RMT_SYNC3) TYPE(*DDM)
RMTDTAQ(SCOTTF/FROM_SQ740) RMTLOCNAME(*RDB) RDB(SYNC3)
TEXT('sq740 to sync3');

--
-- IBM i to IBM i
--
call qsys2.send_data_queue(
        data_queue_library => 'SCOTTF',
        data_queue         => 'RMT_SYNC3',
        message_data       => 'Sending out an SOS');
```


© 2020 IBM Corporation

Data Queues and RPG

- Send data using an ILE RPG program with Embedded SQL

```
DCL-DS Order;
    Ordno   int(10);
    Customer_number int(10);
    Order_date timestamp;
    Ship_date timestamp;
END-DS;
DCL-S Order_data char(60) ccsid(65535) based(P_Order);
DCL-S P_Order pointer inz(%ADDR(Order));

Exec sql call qsys2.send_data_queue_binary(
              data_queue_library => 'SCOTTF',
              data_queue         => 'SQLCANDOIT',
              message_data       => :Order_data );
```


© 2020 IBM Corporation

Data Queues and RPG

- Receive the data into a host variable

```
DCL-DS Order;
    Ordno   int(10);
    Customer_number int(10);
    Order_date timestamp;
    Ship_date timestamp;
END-DS;
DCL-S Order_data char(60) ccsid(65535) based(P_Order);
DCL-S P_Order pointer inz(%ADDR(Order));

Exec sql select message_data into :Order_data
            from table (qsys2.receive_data_queue_binary (
              data_queue_library => 'SCOTTF',
              data_queue         => 'SQLCANDOIT');
```


© 2020 IBM Corporation

DATA_QUEUE_ENTRIES table function

- Show all or a subset of a data queue's content

```
select * from table (
      qsys2.data_queue_entries('TESTDQ', 'SLR'));
```

ORDINAL_POSITION	DATA_QUEUE_LIBRARY	DATA_QUEUE	MESSAGE_DATA	KEY_DATA
1	SLR	TESTDQ	message 1	1
2	SLR	TESTDQ	Next message	1
3	SLR	TESTDQ	Sunday	1
4	SLR	TESTDQ	message 2	2
5	SLR	TESTDQ	message 3	3

© 2020 IBM Corporation

DATA_QUEUE_ENTRIES table function

- Use key selection to subset what is returned

```
select * from table (
      qsys2.data_queue_entries('TESTDQ', 'SLR',
            selection_type => 'KEY',
            key_data => '  1', key_order => 'GT'));
```

ORDINAL_POSITION	DATA_QUEUE_LIBRARY	DATA_QUEUE	MESSAGE_DATA	KEY_DATA
1	SLR	TESTDQ	message 2	2
2	SLR	TESTDQ	message 3	3

© 2020 IBM Corporation

CLEAR_DATA_QUEUE

- Clear all messages from a data queue

```
call qsys2.clear_data_queue(
        data_queue_library => 'SCOTTF',
        data_queue         => 'SQLCANDOIT');
```

- Clear messages according to a key value

```
call qsys2.clear_data_queue(
      data_queue_library => 'SLR',
      data_queue         => 'DQKEYED',
      key_data           => '0002',
      key_order          => 'EQ');
```


© 2020 IBM Corporation

Spooled output

© 2020 IBM Corporation

Work with spooled output

- OUTPUT_QUEUE_INFO
  - One row for every output queue
- OUTPUT_QUEUE_ENTRIES & OUTPUT_QUEUE_ENTRIES_BASIC
  - One row for every entry in an output queue
- SPOOLED_FILE_DATA
  - Returns rows for all the data in a single spooled file
- DELETE_OLD_SPOOLED_FILES
  - Procedure to remove spooled files

© 2020 IBM Corporation

OUTPUT_QUEUE_INFO


```
select * from qsys2.output_queue_info
  where output_queue_library_name = 'QUSRSYS';
```

OUTPUT_QUEUE_NAME	OUTPUT_QUEUE_STATUS	NUMBER_OF_FILES
JAVAPRINT	RELEASED	0
PRT01	RELEASED	176168
QEZDEBUG	RELEASED	312
QEZJOBLOG	RELEASED	0
QFQOUTQ	RELEASED	0
QTPPPOUTQ	RELEASED	8
QVPN01IBM2	RELEASED	0
SCOTTPRINT	RELEASED	0

© 2020 IBM Corporation

OUTPUT_QUEUE_ENTRIES_BASIC


```
select * from qsys2.output_queue_entries_basic
  where output_queue_library_name = 'QUSRSYS' and
        output_queue_name = 'PRT01'
order by size desc
limit 5;
```

CREATE_TIMESTAMP	SPOOLED_FILE_NAME	USER_NAME	SIZE	TOTAL_PAGES	USER_DATA
2020-02-26 11:39:44.627389	QPSRVTRCJ	SBURRICH	10308	1711	T000000001
2020-02-26 11:35:47.354857	QPSRVTRCJ	SBURRICH	8260	1454	T000000001
2020-02-29 13:17:35.401442	QSQSYSIBM	SLROMANO	4164	566	CRTCMOD
2020-03-04 16:57:20.195250	QPSRVTRCJ	POIRIER	836	138	T000000002
2020-03-05 15:00:29.200888	QPSRVTRCJ	SBURRICH	836	121	T000000002

© 2020 IBM Corporation

SPOOLED_FILE_DATA

2020 iSight

```
select * from table (systools.spooled_file_data(
  '424630/SLROMANO/QPADEV000G', 'RPGDTAQ'))
  order by ordinal_position;
```

ORDINAL_POSITION	SPOOLED_DATA
1	5770WDS V7R4M0 190419 RN IBM ILE RPG SLR/RPGDTAQ SQ740 03/07/2
2	Command . . . . . : CRTBNDRPG
3	Issued by . . . . . : SLROMANO
4	Program . . . . . : RPGDTAQ
5	Library . . . . . : SLR
6	Text 'description' . . . . . : *SRCMBRTXT
7	Source Member . . . . . : RPGDTAQ
8	Source File . . . . . : QSQLTEMP1
9	Library . . . . . : QTEMP
10	CCSID . . . . . : 37
11	Text 'description' . . . . . : SQL data queue - send
12	Last Change . . . . . : 03/07/20 08:55:00

© 2020 IBM Corporation

DELETE_OLD_SPOOLED_FILES

- Manage spooled files by age
- Older than the specified timestamp?
  The spooled file is deleted…
- Wow, this is easy

```
-- Preview removal of any spooled file older than 30 days
call systools.delete_old_spooled_files(
    delete_older_than => current date - 30 days,
    preview => 'YES');
```


© 2020 IBM Corporation

DELETE_OLD_SPOOLED_FILES

2020 iSight

Filter by:

- Age
- Output queue
- User

Preview before removing

Parameters:

Number	Mode	Name	Data Type	Length	CCSID	Locator	Default Value
1	IN	DELETE_OLDER_THAN	TIMESTAMP	6			(CURRENT_TIMESTAMP - 3 MONTHS)
2	IN	P_OUTPUT_QUEUE_LIBRARY_NAME	VARCHAR	10			'*ALL'
3	IN	P_OUTPUT_QUEUE_NAME	VARCHAR	10			'*ALL'
4	IN	P_USER_NAME	VARCHAR	10			'*ALL'
5	IN	PREVIEW	VARCHAR	3			'NO'

© 2020 IBM Corporation

Debug assistance

© 2020 IBM Corporation

Gathering additional debug information

- Useful tools for difficult debug situations
- Gather information at point of failure to retain state information for later analysis
  - JOBLOG_INFO
  - STACK_INFO
  - LIBRARY_LIST_INFO
  - OBJECT_LOCK_INFO
  - RECORD_LOCK_INFO
  - JOB_LOCK_INFO
  - GET_JOB_INFO
  - ACTIVE_JOB_INFO

© 2020 IBM Corporation

JOBLOG_INFO

- Returns all the messages in a joblog for the current job or a specific job

```
select * from table(qsys2.joblog_info('*'));
```

MESSAGE_TIMESTAMP	MESSAGE_ID	MESSAGE_TYPE	MESSAGE_TEXT
2020-03-01 14:19:37.787915	-	INFORMATIONAL	CHGDTAARA SLR/CHARDTAARAVALUE('New value')
2020-03-01 14:19:37.811674	CPD0012	DIAGNOSTIC	Characters in qualifier beginning 'CHARDTAARA' not valid.
2020-03-01 14:19:37.811700	CPF0001	ESCAPE	Error found on *N command.
2020-03-01 14:19:37.811869	CPF0006	ESCAPE	Errors occurred in command.
2020-03-01 14:19:37.815003	SQL0443	DIAGNOSTIC	Trigger program or external routine detected an error.

© 2020 IBM Corporation

STACK_INFO

- Dump the job stack for the current job or a specific job
- Either all threads or a specific thread

```
create table coolstuff.stack_dump as
 (select * from table (qsys2.stack_info('*'))
  where entry_type IN ('ILE', 'OPM')) with data;
```


© 2020 IBM Corporation

LIBRARY_LIST_INFO

- Returns the library list for the current job

```
select * from qsys2.library_list_info;
```

ORDINAL_POSITION	SCHEMA_NAME	TYPE	TEXT_DESCRIPTION
1	QSYS	SYSTEM	System Library
2	QSYS2	SYSTEM	System Library for CPI's
3	QHLPSYS	SYSTEM	-
4	QUSRSYS	SYSTEM	System Library for Users
5	QIWS	PRODUCT	-
6	TOYSTORE	CURRENT	COLLECTION - created by SQL
7	QGPL	USER	General Purpose Library
8	QTEMP	USER	-

© 2020 IBM Corporation

LIBRARY_LIST_INFO

- Write the current library list to the joblog.

```
call systools.lprintf('Library list: ' concat
 (select listagg(schema_name, ', ')
                 within group (order by ordinal_position)
  from qsys2.library_list_info));
```

Sent	Message ID	Message
2020-11-15 15:55:54.010698	-	Library list: QSYS, QSYS2, QHLPSYS, QUSRSYS, QIWS, QGPL, QTEMP,

© 2020 IBM Corporation

GET_JOB_INFO

- Returns information about a single job
- 22 columns of information

```
select * from table
(qsys2.get_job_info('399133/QUSER/QSQSRVR'));
```

V_JOB_STATUS	V_ACTIVE_JOB_STATUS	V_ACTIVE_JOB_TYPE	V_AUTHORIZATION_NAME	V_SBS_NAME	V_CPU_USED
*ACTIVE	CNDW	PJ	JAVA	QSYSWRK	116572

© 2020 IBM Corporation

ACTIVE_JOB_INFO

- Returns information about active jobs (WRKACTJOB)
- Filtering criteria by job name, subsystem, and user
- 117 columns of information

```
select * from table
   (qsys2.active_job_info(job_name_filter=>'QSQSRVR',
                          detailed_info=>'ALL'));
```

JOB_NAME	JOB_ACTIVE_TIME	SERVER_MODE_CONNECTING_JOB
399050/QUSER/QSQSRVR	2020-03-06 04:18:22	399005/QDIRSRV/QUSRDIR
399133/QUSER/QSQSRVR	2020-03-06 04:20:40	399418/JAVA/QP0ZSPWP
399420/QUSER/QSQSRVR	2020-03-06 05:00:03	399459/JAVA/QJVAEXEC
399991/QUSER/QSQSRVR	2020-03-06 06:00:08	399418/JAVA/QP0ZSPWP
399992/QUSER/QSQSRVR	2020-03-06 06:00:10	-

© 2020 IBM Corporation

OBJECT_LOCK_INFO

- Returns one row for every lock on every object
- Use a WHERE clause to limit the result set

```
select * from qsys2.object_lock_info
where object_schema = 'TOYSTORE' and object_name ='SALES';
```

MEMBER_LOCK_TYPE	LOCK_STATE	LOCK_STATUS	JOB_NAME
-	*SHRRD	HELD	509329/SLROMANO/QPADEV000X
MEMBER	*SHRRD	HELD	509329/SLROMANO/QPADEV000X
DATA	*EXCL	HELD	509329/SLROMANO/QPADEV000X

© 2020 IBM Corporation

RECORD_LOCK_INFO

- Returns one row for every record lock on every file
- Use a WHERE clause to limit the result set

```
select lock_state, rrn, job_name from qsys2.record_lock_info
  where table_schema = 'TOYSTORE' and table_name = 'SALES'
```

LOCK_STATE	RRN	JOB_NAME
UPDATE	1	509329/SLROMANO/QPADEV000X
UPDATE	2	509329/SLROMANO/QPADEV000X
UPDATE	3	509329/SLROMANO/QPADEV000X
UPDATE	4	509329/SLROMANO/QPADEV000X
UPDATE	5	509329/SLROMANO/QPADEV000X
UPDATE	6	509329/SLROMANO/QPADEV000X
UPDATE	7	509329/SLROMANO/QPADEV000X
UPDATE	8	509329/SLROMANO/QPADEV000X

© 2020 IBM Corporation

JOB_LOCK_INFO

- Returns a list of locks held by a job

```
select *
  from table(qsys2.job_lock_info('*'))
  order by object_type;
```

OBJECT_LIBRARY	OBJECT_NAME	OBJECT_TYPE	OBJECT_ATTRIBUTE	LOCK_STATE	LOCK_STATUS	LOCK_COUNT	LOCK_SCOP
QSYS	QIWS	*LIB	PROD	*SHRRD	HELD	1	JOB
QSYS	QSYS	*LIB	PROD	*SHRRD	HELD	1	JOB
QSYS	QSYS2	*LIB	PROD	*SHRRD	HELD	1	JOB
QSYS	QUSRSYS	*LIB	PROD	*SHRRD	HELD	1	JOB

© 2020 IBM Corporation

RECORD_LOCK_INFO + GET_JOB_INFO

- Combine services for more insight

```
with lock_conflict_table as (
  select distinct lock_state, job_name
  from qsys2.record_lock_info
  where table_schema = 'TOYSTORE' and table_name = 'SALES')
select v_sql_statement_text, v_client_ip_address
from lock_conflict_table c,
   table(qsys2.get_job_info(c.job_name));
```

V_SQL_STATEMENT_TEXT	V_CLIENT_IP_ADDRESS
insert into toystore2.sales select * from toystore.sales ...	9.10.110.53

© 2020 IBM Corporation

SQL PL – Capture details for lock problems


```
declare continue handler for sqlstate '57033'
begin /* Message SQL0913 object in use */
  declare msg_schema_name varchar(128);
  declare msg_table_name varchar(128);
  declare dot_location integer;
  declare msg_token varchar(1000);
  get diagnostics condition 1 msg_token = db2_ordinal_token_1;
  set dot_location = locate_in_string(msg_token, '.');
  set msg_schema_name = substr(msg_token, 1, dot_location - 1);
  set msg_table_name = substr(msg_token, dot_location + 1,
                              length(msg_token) - dot_location);
  insert into applib.hard_to_debug_problems
    select * from table (qsys2.joblog_info('*'))
            order by ordinal_position desc limit 5;
  insert into applib.hard_to_debug_lock_problems
    select * from qsys2.object_lock_info
         where object_schema = msg_schema_name and
               object_name = msg_table_name;
end;
```


© 2020 IBM Corporation

ILE RPG – Capture details for lock problems


```
dcl-s msg_token char(100);
dcl-s msg_schema_name char(128);
dcl-s msg_table_name char(128);
dcl-s dot_location int(5);
if SQLCODE = -913;                    /* Message SQL0913 object in use */
  exec sql get diagnostics condition 1 :msg_token = db2_ordinal_token_1;
  exec sql set :dot_location = locate_in_string(:msg_token, '.');
  exec sql set :msg_schema_name = substr(:msg_token, 1, :dot_location - 1);
  exec sql set :msg_table_name = substr(:msg_token, :dot_location + 1,
                                 length(:msg_token) - :dot_location);
  exec sql
    insert into hard_to_debug_problems
      select * from table (qsys2.joblog_info('*'))
               order by ordinal_position desc limit 5;
  exec sql
    insert into hard_to_debug_lock_problems
      select * from qsys2.object_lock_info
             where object_schema = :msg_schema_name and
                   object_name = :msg_table_name;
endif;
```


© 2020 IBM Corporation

Continue handler to print SQL failure information

```
declare v_stmt_text clob(50k) ccsid 37;
declare continue handler for sqlexception
  begin
    declare local_sqlcode  integer;
    declare local_sqlstate char(5);
    declare v_message_text varchar(200) ccsid 37;
    get diagnostics condition 1
        local_sqlcode  = db2_returned_sqlcode,
        local_sqlstate = returned_sqlstate,
        v_message_text = message_text;
    call systools.lprintf('APP1 failed with SQLCODE=' concat local_sqlcode
        concat ' SQLSTATE=' concat local_sqlstate
        concat ' MESSAGE= '  concat v_message_text
        concat ' STMT='     concat v_stmt_text);
  end;
```


© 2020 IBM Corporation

Additional topics

© 2020 IBM Corporation

SQL Tutor

- Aggregation of all Gist Examples
- Aggregation of all iSee video blogs
- Different perspectives to easily find what you want

You are in: IBM i Tutorials, Demos, and SQL examples

List of all of the Db2 for i SQL Gists by Scott Forstie

List of all of the IBM i COMMON Tutorials by Scott Forstie & Tim Rowe

Categories
Access Client Solutions (ACS)
Database Engineering Topics
Db2 for i Services
IBM i Services

Security Services	Github Gists
QSYS2.OBJECT_OWNERSHIP	Who owns the most objects.sql
	TopN user storage report.sql
	Object ownership by user - total report.sql

ibm.biz/Db2foriSQLTutor

© 2020 IBM Corporation

Db2 Web Query and IBM i Services

IBM i Administration Samples folder

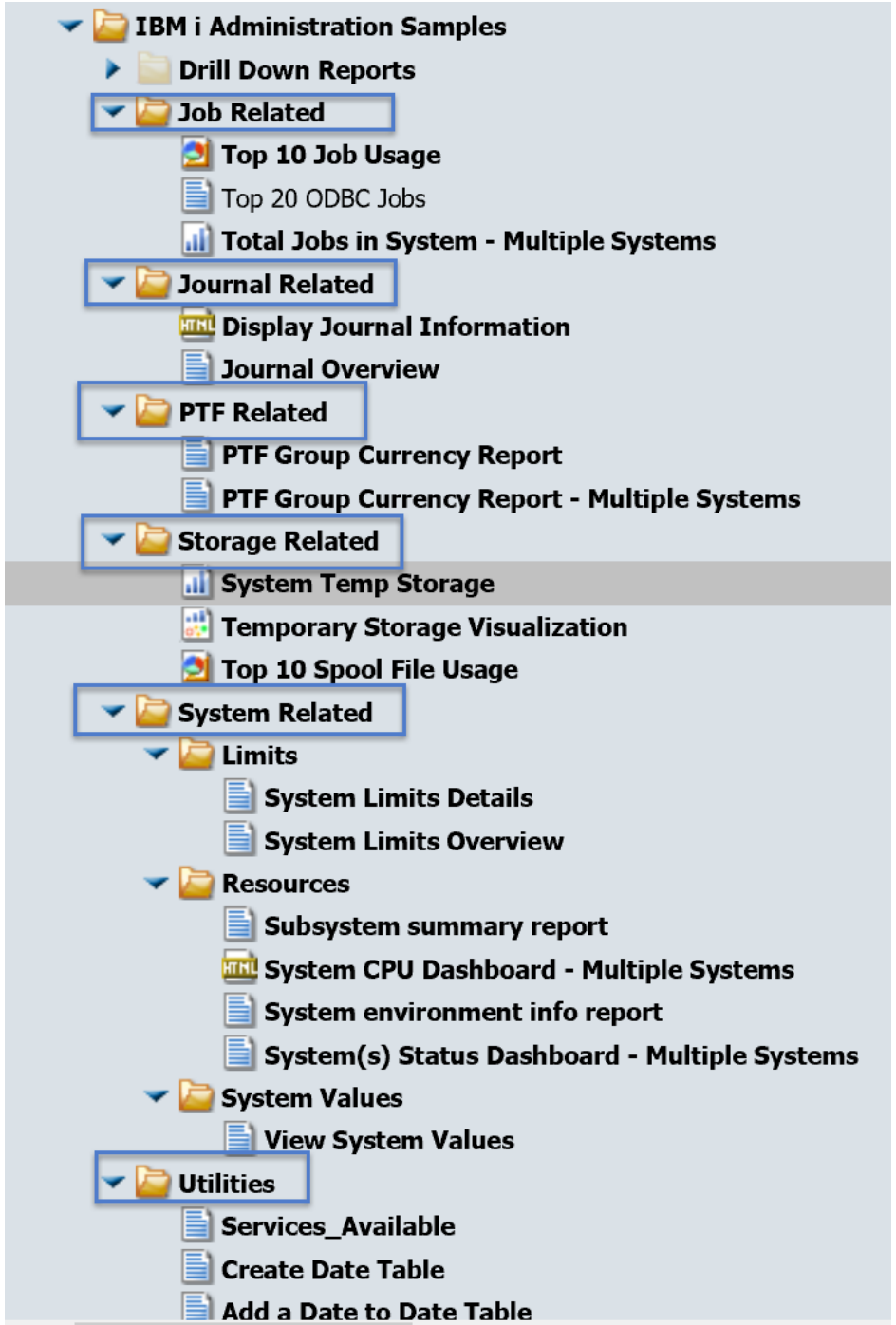
- Delivered as part of the **Db2 Web Query EZ-Install** package
- Sample Reports provided across multiple subject areas
- Learn how you can extend or build new ones

There are other things of interest in EZ-Install

- Query/400 Discovery Tool
- Create Date Dimension Table
- **EZ-Report** ← NEW
  - Great for using “insert from examples” for **FAST** creation of a report over an IBM i Service

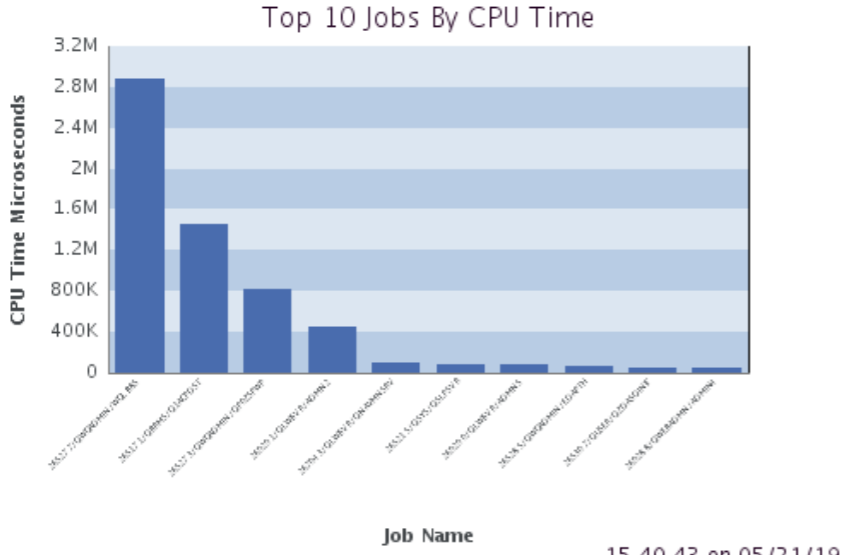
Request EZ-Install (or standalone sample administration reports if you already have Db2 Web Query) by sending an email to QU2@us.ibm.com

- Include name, email ID, serial number of system

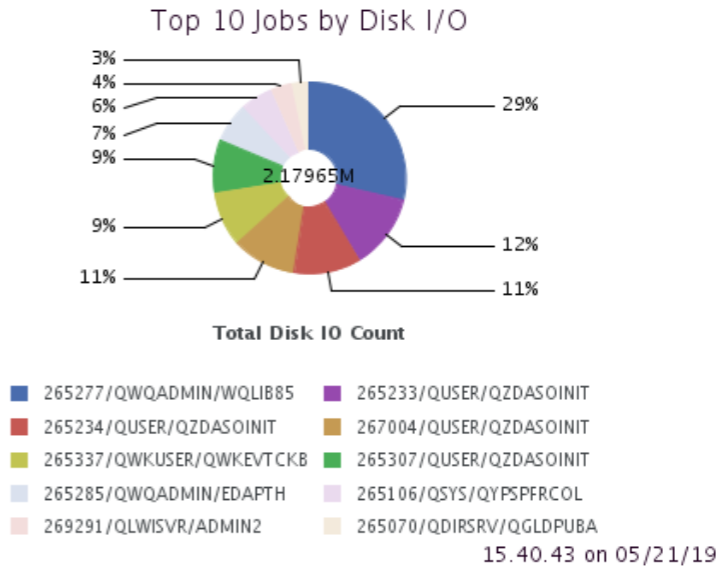


Job Related Samples

Top 10 Job Usage

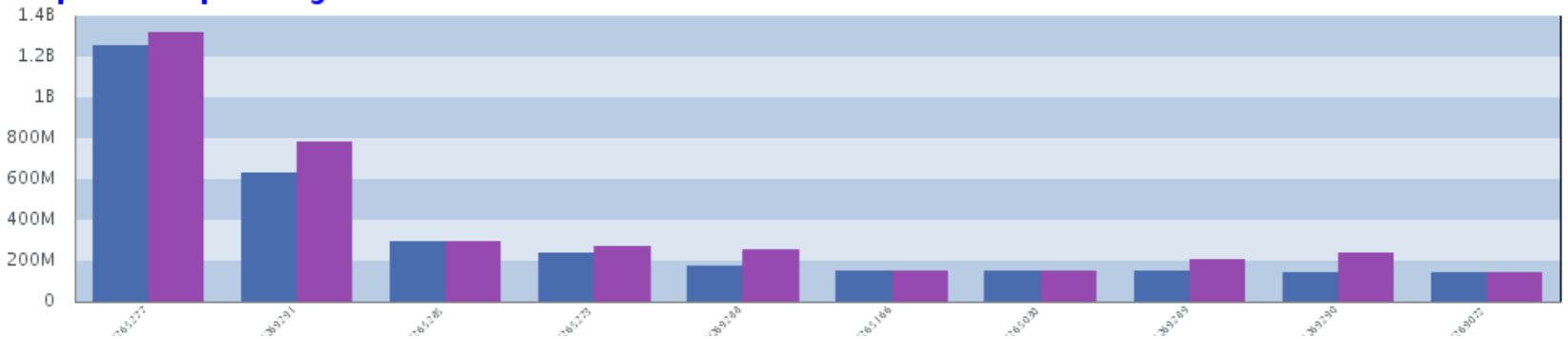


15.40.43 on 05/21/19



15.40.43 on 05/21/19

Top 10 Temp Storage Abusers



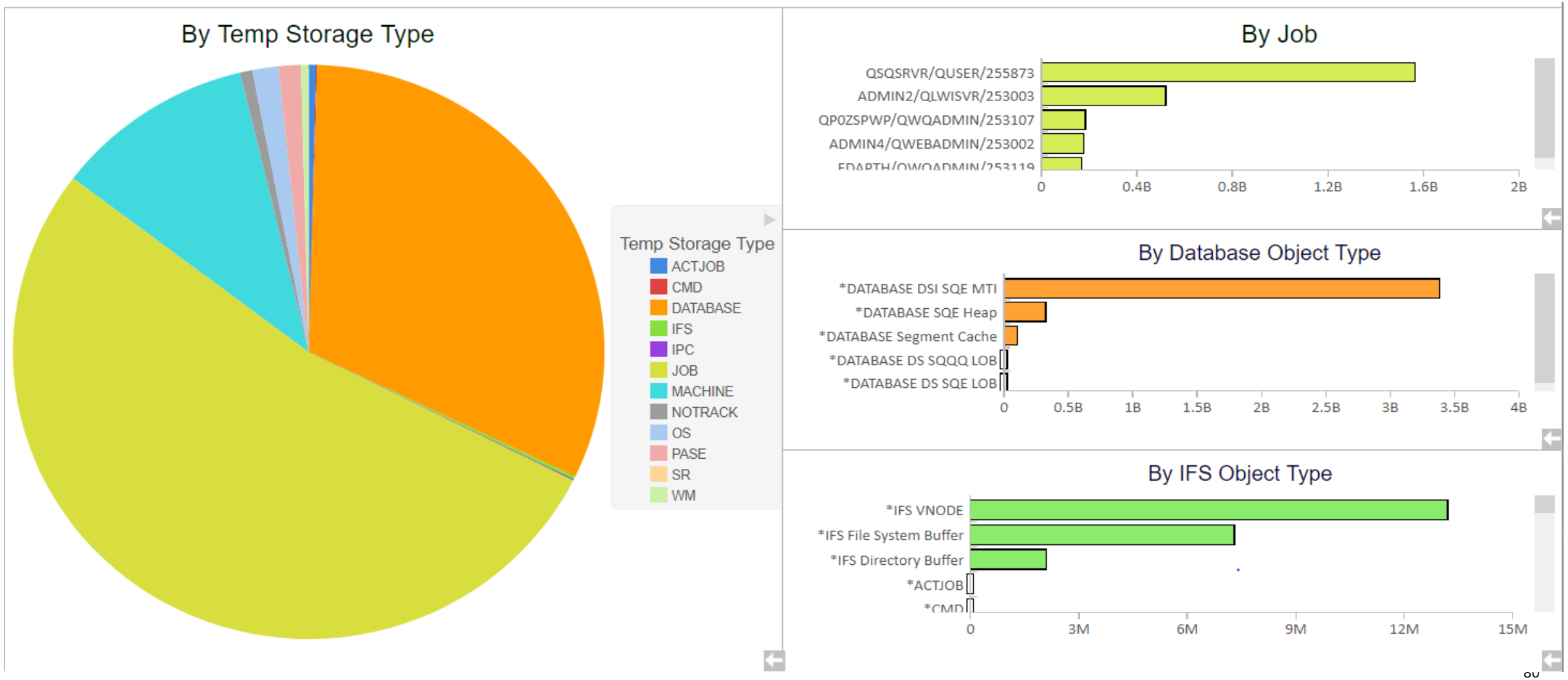
PTF Related Reports

PTF Group currency… for **multiple systems**

System	OS Release	Date Information Gathered	Status	PTF Group ID	PTF Name	Installed Level	Level Available
DB2ICOE3	V7R2M0	2019/04/18	INSTALLED LEVEL IS CURRENT	SF99720	Current Cumulative PTF Media Documentation	18249	18249
				SF99721	All PTF Groups except Cumulative PTF Package	10	10
				SF99747	DB2 Web Query for i V2.1.0	17	17
				SF99766	Print PTFs	3	3
				SF99767	720 TCP/IP PTF	4	4
				SF99769	IBM i integration with BladeCenter and System	1	1
				SF99775	Hardware and Related PTFs	33	33
				SF99776	High Availability for IBM i	10	10
			UPDATE AVAILABLE	SF99702	DB2 for IBM i	24	25
				SF99718	Group Security	72	73
				SF99719	Group Hiper	144	145
				SF99759	IBM MQ for IBM i - v7.1.0/v8.0.0/v9.0.0/v9.1.0	7	22
DB2ICOE4	V7R3M0	2019/04/18	INSTALLED LEVEL IS CURRENT	SF99225	IBM Open Source Solutions for i	6	6
				SF99252	Content Manager OnDemand for i - 5770-RD1 7.3	10	10
				SF99433	Db2 Web Query for i V2.2.1	5	5
				SF99723	Performance Tools	4	4
				SF99727	Technology Refresh	5	5
				SF99730	Current Cumulative PTF Media Documentation	18242	18242
				SF99731	All PTF Groups except Cumulative PTF Package	11	11
				SF99867	730 TCP/IP PTF	1	1
				SF99876	High Availability for IBM i	7	7
			RELATED GROUP	SF99333	DB2 Web Query for i V2.2.0	5	5
			UPDATE AVAILABLE	SF99581	WebSphere App Server V8.5	6	7
				SF99703	DB2 for IBM i	12	14

Storage Related Sample

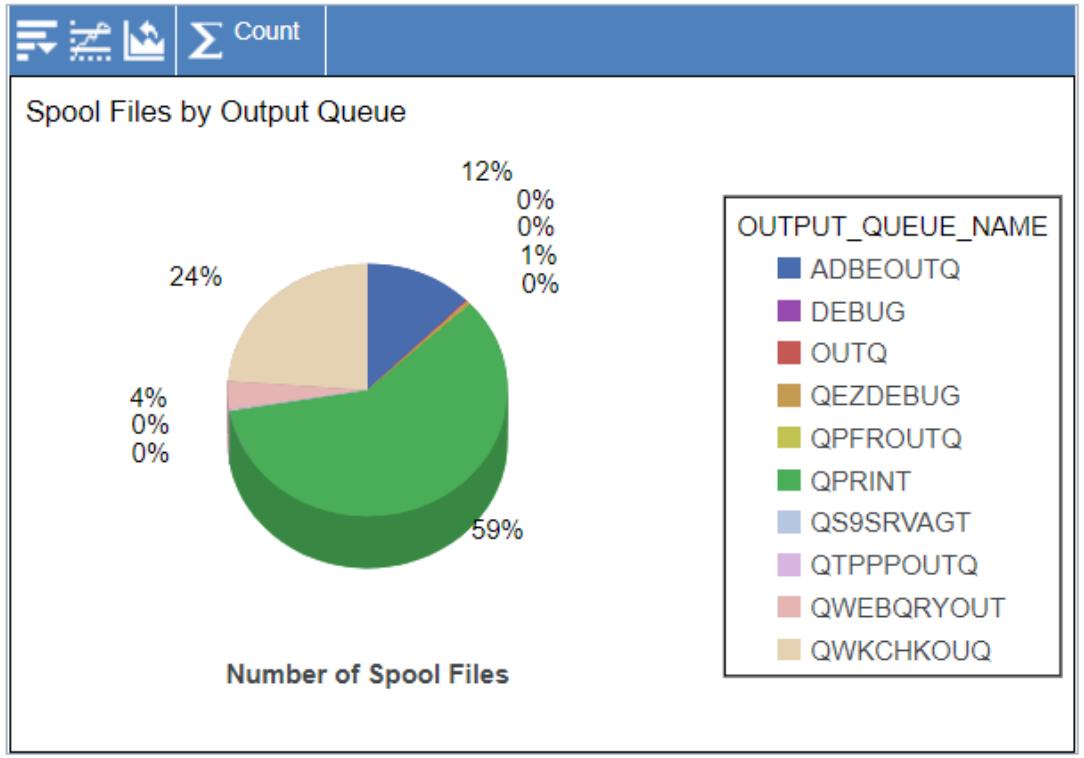
Temporary storage usage visualization

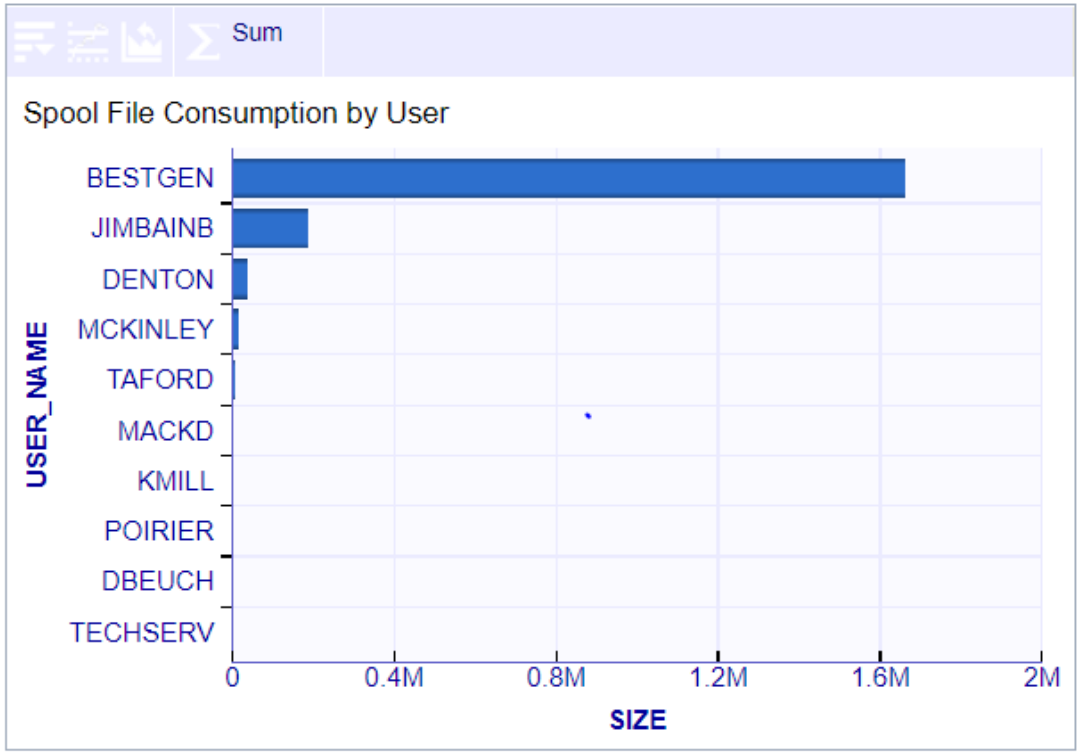


Spool Files

Spool file usage dashboard

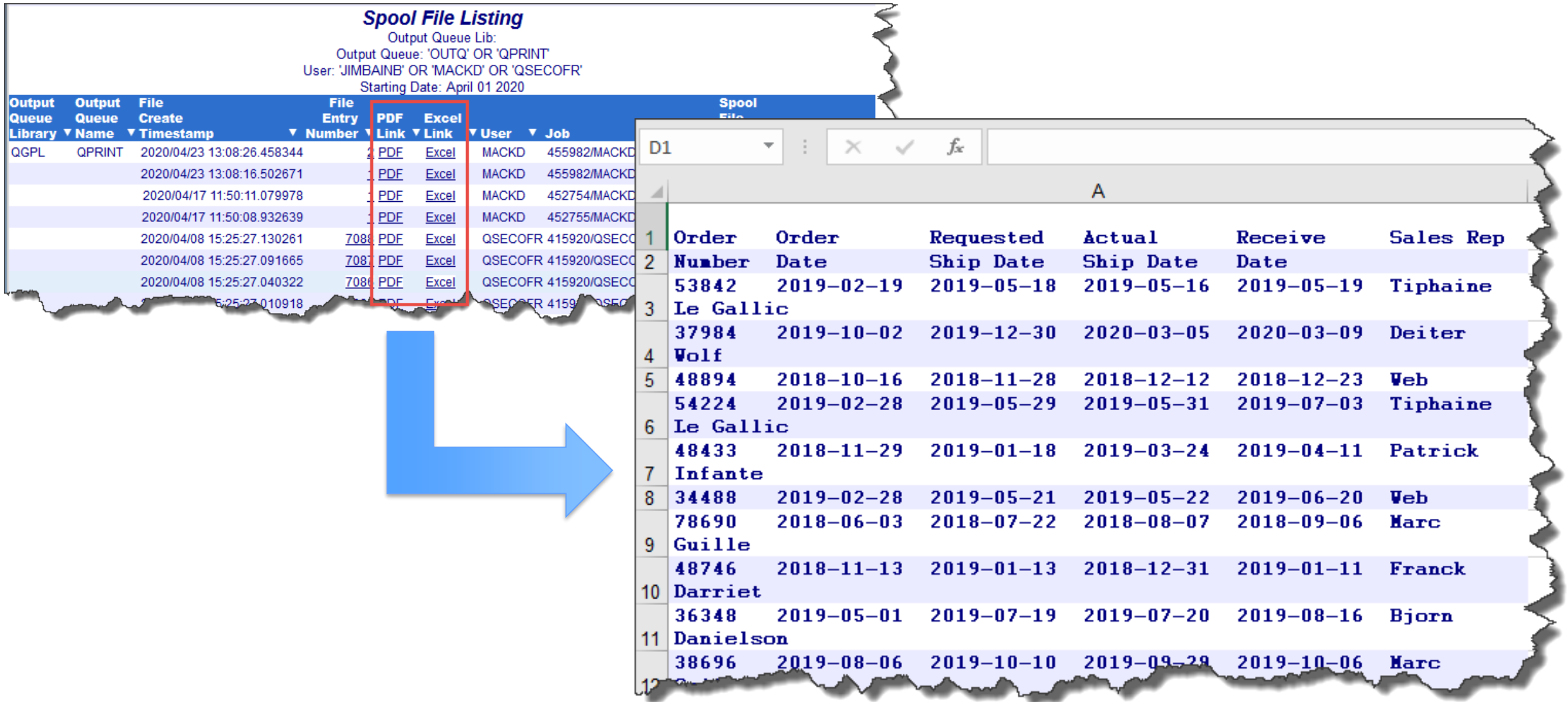
Output Queue Top 10 at 10.13.34 on April 18, 2019





Create Spreadsheet or PDF from Spoolfile Data

Spool File Information and PDF Generation



Spool File Listing

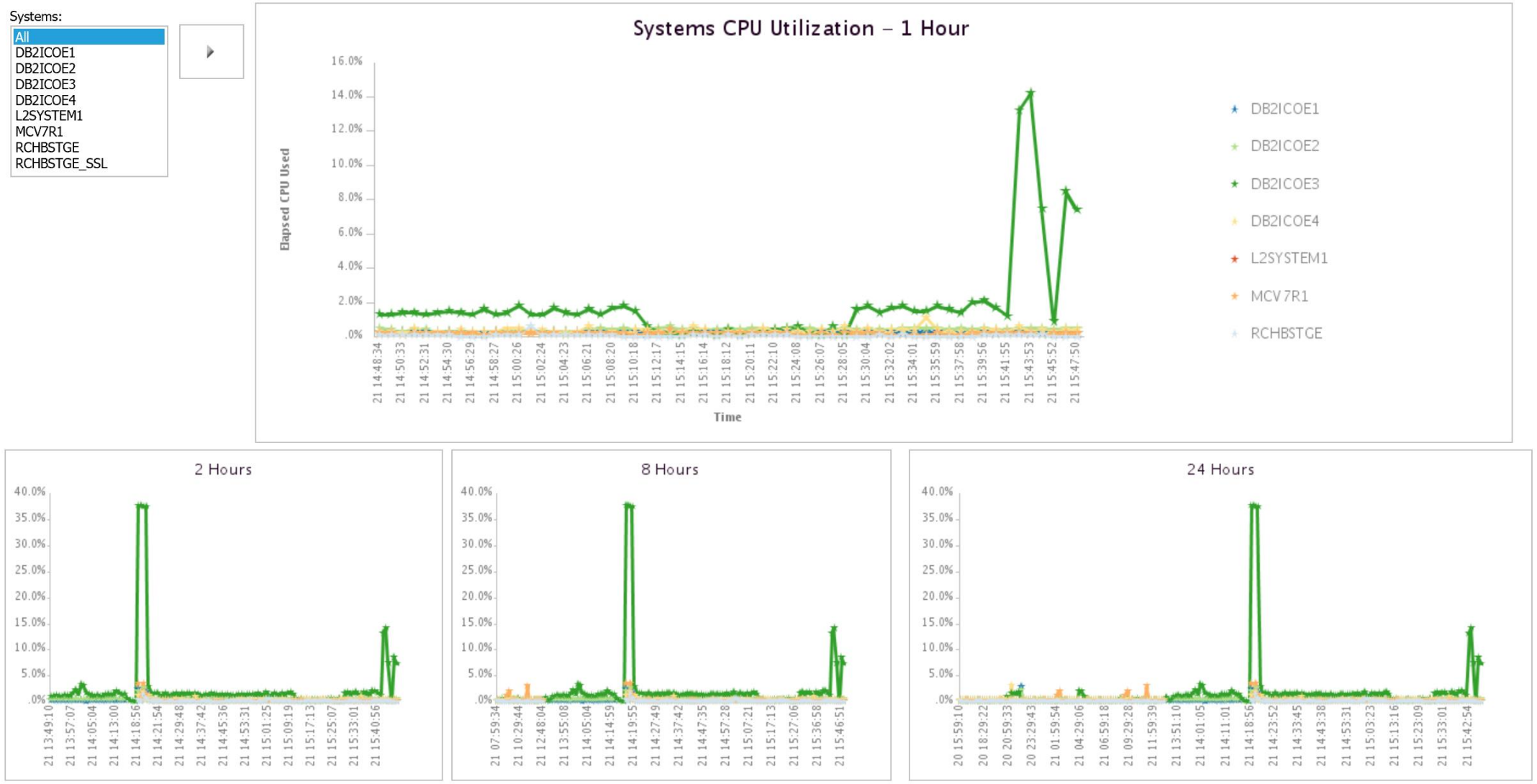
Output Queue Lib:
Output Queue: 'OUTQ' OR 'QPRINT'
User: 'JIMBAINB' OR 'MACKD' OR 'QSECOFR'
Starting Date: April 01 2020

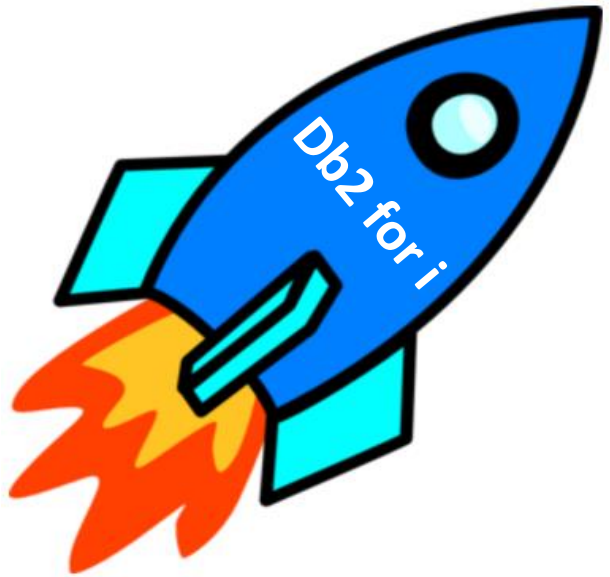
Output Queue Library	Output Queue Name	File Create Timestamp	File Entry Number	PDF Link	Excel Link	User	Job	Spool File
QGPL	QPRINT	2020/04/23 13:08:26.458344	2	PDF	Excel	MACKD	455982/MACKD	
		2020/04/23 13:08:16.502671	1	PDF	Excel	MACKD	455982/MACKD	
		2020/04/17 11:50:11.079978	1	PDF	Excel	MACKD	452754/MACKD	
		2020/04/17 11:50:08.932639	1	PDF	Excel	MACKD	452755/MACKD	
		2020/04/08 15:25:27.130261	7088	PDF	Excel	QSECOFR	415920/QSECO	
		2020/04/08 15:25:27.091665	7087	PDF	Excel	QSECOFR	415920/QSECO	
		2020/04/08 15:25:27.040322	7086	PDF	Excel	QSECOFR	415920/QSECO	
		5:25:27.010918		PDF	Excel	QSECOFR	4159	

D1

	A					
1	Order Number	Order Date	Requested Ship Date	Actual Ship Date	Receive Date	Sales Rep
2	53842	2019-02-19	2019-05-18	2019-05-16	2019-05-19	Tiphaine Le Gallic
3	37984	2019-10-02	2019-12-30	2020-03-05	2020-03-09	Deiter Wolf
4	48894	2018-10-16	2018-11-28	2018-12-12	2018-12-23	Web
5	54224	2019-02-28	2019-05-29	2019-05-31	2019-07-03	Tiphaine Le Gallic
6	48433	2018-11-29	2019-01-18	2019-03-24	2019-04-11	Patrick Infante
7	34488	2019-02-28	2019-05-21	2019-05-22	2019-06-20	Web
8	78690	2018-06-03	2018-07-22	2018-08-07	2018-09-06	Marc Guille
9	48746	2018-11-13	2019-01-13	2018-12-31	2019-01-11	Franck Darriet
10	36348	2019-05-01	2019-07-19	2019-07-20	2019-08-16	Bjorn Danielson
11	38696	2019-08-06	2019-10-10	2019-09-29	2019-10-06	Marc

System Related – Self Refreshing CPU Utilization Dashboard





© 2020 IBM Corporation