



Automated End-to-End Database Maintenance

**How to reduce your DB2 workload
and improve data availability using
DB2 Automation Tool v4.1 for z/OS**

*Jennifer Nelson
Product Specialist / Product Manager
Rocket Software, Inc.*

CONTENTS

List of Figures.....	iii
1 Today's data challenges	1
2 Automating database maintenance is the solution	1
2.1 Conditional Maintenance	2
2.2 Generate utility JCL automatically	4
2.3 Using the DB2 administrative task scheduler to automatically execute utility JCL	8
3 Evolve your strategy with an intelligent solution	12
References.....	xiii

LIST OF FIGURES

Figure 1: Exceptions enable you to find what objects to include.	2
Figure 2: Example - REORG exceptions.	3
Figure 3: Integrating custom exceptions.	4
Figure 4: Job Profiles are comprised of objects, exceptions and utilities.	5
Figure 5: Build your Job Profile in batch or online.	5
Figure 6: Automatically generating utility JCL.	6
Figure 7: Result of batch Job Profile.	6
Figure 8: Objects that match the exceptions are written to the TRIGGERS DD.	7
Figure 9: Generated utility JCL contains only objects that match the exceptions.	7
Figure 10: All utilities in the Utility Profile are generated into the utility JCL.	8
Figure 11: DB2 Administrative Task Scheduler accessible via DB2 Automation Tool v4.1.	9
Figure 12: Adding JCL job to the scheduler.	9
Figure 13: Generating JCL in DB2 Automation Tool and adding to Scheduler.	10
Figure 14: Execute JCL at a specific time or during an interval.	10
Figure 15: Utility JCL can also be added to the Scheduler.	11
Figure 16: Schedule the resultant utility JCL with different parameters.	12

1 Today's data challenges

According to industry experts, the amount of data is exploding in structured data, replicated data and unstructured data – all of which can be stored in a database such as DB2 for z/OS. The amount of data stored in DB2 for z/OS is more than doubling every two years.. At the same time, most DB2 for z/OS applications are global non-stop, requiring almost 100% accessibility. These circumstances place heavy demands on the amount of time required to execute critical maintenance that, if not done, affects the performance of your business critical applications and your bottom line.

Add to the challenge of exponential data growth, the current economic environment is shrinking the operational budgets of most companies. You need to justify each and every software purchase. Companies are often operating with reduced IT staff dealing with an increased workload. Maintaining optimal performance of your databases is as important to data application availability as ensuring a network remains online and available for non-stop business use. Database maintenance can be more time-consuming than you might think. Consider the following questions as they apply to your database maintenance:

- Is your database maintenance routine? Routine database maintenance doesn't mean that it is easy, quick or cost-effective.
- Do you run a third party application? Often times, packaged vendor JCL is not optimized or timely. For example, your vendor JCL could use syntax that does not follow best practices or routinely collect statistics, take an image copy or perform REORGs on objects whether they're needed or not.
- What if an object doesn't need specific maintenance, such as an image copy? Do you take one anyway just in case?
- What if an object needs maintenance, but is either being skipped for maintenance or has the wrong maintenance run? Do you run the utility now, or do you wait until later?
- Will you be able to fit all of your critical maintenance jobs within the batch window? What happens if objects don't have maintenance applied?
- Is your maintenance process automated? How much time and effort is required by your staff?
- Are there situations that require your DBA staff to interrupt other tasks to intervene and manage the maintenance window?

2 Automating database maintenance is the solution

Creating a strategy for running routine maintenance and automating that maintenance can help your DBA staff focus on more business critical tasks, while application data remains available for business. The IBM DB2 Automation Tool for z/OS provides the solution to your data maintenance challenges and moreover lets you have control of your objects,

data availability and applications. It determines which objects need what maintenance when according to your business specifications. Using exception criteria to detect which objects or group of objects need what maintenance is an efficient way to locate only those objects in need; you no longer need to run packaged vendor utility JCL. With the DB2 Automation Tool v4.1, exception jobs can be automatically run at intervals you determine by integrating with the DB2 Administrative Task Scheduler. By scheduling the exception job to execute automatically at regular intervals to evaluate your objects, you will always know what objects need which maintenance. And, you will be able to automate executing the utility jobs, eliminating any unnecessary DBA intervention. This paper will help anyone who is familiar with the DB2 Automation Tool learn how to use the DB2 Administrative Task Scheduler to fully automate database maintenance.

2.1 Conditional Maintenance

With over 180 exceptions, DB2 Automation Tool can help you find just those objects that need maintenance. Do you need to trigger an image copy only when certain conditions occur? Or do you need to trigger a reorg with a combination of conditions? The DB2 Automation Tool Exceptions Profiles can help. Exception Profiles are simply a re-usable collection of exceptions, or conditions, against which to evaluate objects. They can be used with any set of objects, called Object Profiles, to generate any kind of utility JCL.

For example, the real time stats exceptions CLUSTERSENS and SCANACCESS can be used to evaluate when a REORG should take place. In the screen shot below, the column S contains an 'A' for the SCANACCESS exception, which is an AND condition. This means that both the CLUSTERSENS condition and the SCANACCESS condition must be met before the object will be included in the utility.

```

AUTOTOOL V4R1 ---- Update Exceptions Profile Display --- 2011/10/03 11:09:26
Option ==> Scroll ==> CSR
Line Commands: A - And O - Or S - Select D - Deselect R - Repeat
CONDITIONS: LT|<|LE|<=|EQ|=|GT|>|GE|>=|NE|,=|<> "*" indicates a DAT stat
-----
Creator: CSJENN Profile: REORG AVOIDANCE EXCP User: CSJENN
-----
Share Option U (U - Update, V - View, N - No)
Description Scroll Right for Column Help
Use Stats From: C (R - Repository, Update Runstats Options: N (Yes/No)
                  C - Catalog, Save Triggers in Repository: N (Yes/No)
                  U - Runstats, WTD number of triggered Objects: N (Yes/No)
                  S - Shadow, Combine IX/TS Exceptions when
                  H - History) evaluating an IX triggering a TS: N (Yes/No)
----- Row 73 of 206 -->
S Statistics Type--- *Column----- Cond -----Exception Value-----
Q REALTIME REORG TS CLUSTERSENS >= 500
- HASHACCESS
A SCANACCESS >= 1000
- REALTIME REORG IX REAL TIME STAT
- DAYS SINCE_LAST
- INS_DEL

```

Figure 1: Exceptions enable you to find what objects to include.

Scrolling right will display an explanation of the exceptions, as shown below. With a complete description of the exception conditions, you can make a comprehensive decision on which exceptions to set to trigger just the objects you need.

```

AUTOTOOL V4R1 ---- Update Exceptions Profile Display --- 2011/10/03 11:17:48
Option ==>
Line Commands: A - And O - Or S - Select D - Deselect R - Repeat
CONDITIONS: LT|<|LE|<=|EQ|=|GT|>|GE|>=|NE|,=|<> "*" indicates a DAT stat
-----
Creator: CSJENN Profile: REORG AVOIDANCE EXCP User: CSJENN
-----
Share Option U (U - Update, V - View, N - No)
Description
Use Stats From: C (R - Repository, Update Runstats Options: N (Yes/No)
C - Catalog, Save Triggers in Repository: N (Yes/No)
U - Runstats, WTD number of triggered Objects: N (Yes/No)
S - Shadow, Combine IX/TS Exceptions when
H - History) evaluating an IX triggering a TS: N (Yes/No)
-----
S Statistics Type-- *Column----- Column Description-----
Q REALTIME REORG TS CLUSTERSENS No. of times data read by SQL sensitiv
HASHACCESS No. of times data is accessed using ha
SCANACCESS No. of times data is accessed for SELE
A REALTIME REORG IX REAL TIME STAT Reorg Index related Thresholds...since
DAYS SINCE_LAST No. of days since the last REORG
INS_DEL No. of Inserts and Deletes
-----

```

Figure 2: Example - REORG exceptions.

Select any combination of DB2 catalog statistics, SYSCOPY information, MVS catalog statistics, DB2 display status information or real time exception conditions to trigger just the objects that meet those criteria. Optionally, you can have the DB2 Automation Tool run a RUNSTATS utility to evaluate your object using current statistics. You can elect to store the new statistics by setting UPDATE to ALL, ACCESSPATH, or SPACE. You can also set UPDATE to NONE to enable only the Exception Profile to evaluate current statistics but not store those values in the catalog.

The Automation Tool is also flexible enough to allow you to supply input that is not defined in the product. This can be a user-supplied formula, user-written REXX exec, assembled load module or stored procedure with your customized exceptions to evaluate against your objects. You provide the load module name, REXX exec name, or stored procedure name you want to execute as shown in the example below. The load module or REXX exec is loaded into memory once and called for each object within the associated Object Profile. The stored procedure can be called in three different ways, depending on what you want it to do and how you want it to process your objects. You may elect to process your stored procedure either before the DB2 Automation Tool exception criteria evaluation occurs, as the criteria and objects are being evaluated, or after the evaluation has completed. Enter your stored procedure name in the appropriate field where you want the DB2 Automation Tool to run the stored procedure.

```

AUTOTOOL V4R1 ---- Update Exceptions Profile Display --- 2011/10/03 12:02:11
Option ==>
Line Commands: A - And O - Or S - Select D - Deselect R - Repeat
CONDITIONS: LT|<LE|<=|EQ|=|GT|>|GE|>=|NE|,=|<> "*" indicates a DAT stat
-----
Creator: CSJENN Profile: BETA EXCEPTIONS User: CSJENN
-----
Share Option U (U - Update, V - View, N - No)
Description
Use Stats From: R (R - Repository, Update Runstats Options: N (Yes/No)
C - Catalog, Save Triggers in Repository: N (Yes/No)
U - Runstats, WTD number of triggered Objects: N (Yes/No)
S - Shadow, Combine IX/TS Exceptions when
H - History) evaluating an IX triggering a TS: N (Yes/No)
-----
S Statistics Type *Column Cond Exception Value
- USER EXIT LOAD_MODULE
- CLIST_REXX_EXEC
- STORED_PROC_PRE
O STORED_PROCEDURE EQ SCHEMA.GOTESTPROC27
- STORED_PROC_POST
-----
Row 29 of 206 -->

```

Figure 3: Integrating custom exceptions.

Combine your own exceptions in a load module, REXX exec or stored procedure with any of the exceptions provided by DB2 Automation Tool to meet your needs. This level of scalability and flexibility lets you tailor the DB2 Automation Tool to meet your company's needs.

2.2 Generate utility JCL automatically

The DB2 Automation Tool can help with the frequency of when your objects are evaluated whether it's once a week or daily for your business-critical objects. The DB2 Automation Tool can be executed as needed or placed in a scheduler to be executed at regular intervals with no intervention from the DBA.

To generate utility JCL, a Job Profile must contain at least one Object Profile, which is a reusable list of objects, and one Utility Profile, which is a list of utilities for which you want to generate JCL. This enables you to generate a utility for all objects within an Object Profile. But to conditionally generate utility JCL for only objects that meet or exceed exception criteria, an Exception Profile must be added to the Job Profile. Using this method, you can easily generate utility JCL for only those objects that need maintenance, enabling you to save on CPU consumption and reducing the amount of elapsed time required to run maintenance, thus improving data availability.

If, for example, you must take an image copy each week for just those objects that need it, you can re-run the Job Profile that contains the exception criterion for running an image copy and see which objects are generated into the utility JCL. That utility JCL can then be added to a job scheduler, such as the DB2 administrative task scheduler, to run during the next maintenance window.

In this example below, the Job Profile contains three pieces of information: the Object Profile called "Reorg Avoidance Objs", a set of utilities and corresponding options in a Utility Profile called "Reorg Avoidance Util", and a set of exception criteria in an Exception Profile called "Reorg Avoidance Excp".


```

AUTOTOOL V4R1 ----- Update Jobs Profile Display ----- 2011/10/03 17:16:35
Option ==> _____ Scroll ==> CSR_
-----
Line Commands: V - View A - Add D - Delete U - Update
-----
Creator: CSJENN      Profile: BATCH EVALUATION      User: CSJENN
-----
Share Option U (U - Update,      Description _____
                V - View,
                N - No)
Update Job Generation Options N (Yes/No) Row 1 of 3 >
-----
Cmd  Type  Order  Name                Creator  Userid
-----
 1  OBJS   1      REORG AVOIDANCE OBJS  CSJENN  CSJENN
 2  UTIL   1      REORG AVOIDANCE UTIL  CSJENN  CSJENN
 3  EXCP   1      REORG AVOIDANCE EXCP  CSJENN  CSJENN
***** Bottom of Data *****

```

Figure 4: Job Profiles are comprised of objects, exceptions and utilities.

In this example, the Job Profile “Batch Evaluation” is being generated to run in batch mode. This means that each time this Job Profile is “built”, a batch job will be created. This batch job, called REORGB, will evaluate the objects in the Object Profile against the exception criteria in the Exception Profile. Any object that matches the exceptions will be generated into the utility JCL, which is also created when running this batch job.

```

----- Build Job for CSJENN.BATCH EVALUATION -----
Build Online or Batch. . .  (O - Online, B - Batch)
Edit Generated Job . . . Y (Yes/No)
Schedule Job . . . . . N (Yes/No) Update options . . N (Yes/No)
Build job in Dataset . . CSJENN.HAA410.JCL
Member . . REORGB_
Job Cards:
=> //REORG      JOB CSJENN,CLASS=A,NOTIFY=&SYSUID,MSGLEVEL=(1,1)
=> //*
=> //* EVALUATE OBJECTS TO INCLUDE IN A REORG
=> //*

```

Figure 5: Build your Job Profile in batch or online.

The second pop-up window enables you to specify the PDS data set and member name into which the resultant utility JCL will be generated.

```

Build Job for CSJENN.BATCH EVALUATION

You have selected your job to be built in a batch mode. The
batch generation JCL will be stored in dataset
CSJENN.HAA410.JCL(REORGB).
Please specify in the dataset and member below where you want the
JCL built by the batch module to be placed.

Schedule Job . . . . N (Yes/No)  Update options . . N (Yes/No)

Build job in Dataset. . CSJENN.HAA410.JCL
Member . . REO#_____

Jobcard data to be used on the generated job:
==> //REORGJCL JOB CSJENN,CLASS=A,NOTIFY=&SYSUID
==> /*
==> /* THIS IS THE ACTUAL UTILITY JCL
==> /*

```

Figure 6: Automatically generating utility JCL.

The JCL to evaluate the objects is presented in an ISPF edit panel to review. It is this JCL that you can add to a scheduler to execute at regular intervals. By running this batch JCL job regularly, you can be sure that only those objects that need a REORG run will be included.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT CSJENN.HAA410.JCL(REORGB) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR_
***** Top of Data *****
000001 //REORG JOB CSJENN,CLASS=A,NOTIFY=&SYSUID,MSGLEVEL=(1,1)
000002 /*
000003 /* EVALUATE OBJECTS TO INCLUDE IN A REORG
000004 /*
000005 /*
000006 /** * * * *
000007 /*
000008 /* Job Generated by IBM DB2 Automation Tool V4R1.01
000009 /*
000010 /* DB2 SSID: DA1A
000011 /* SQLID: CSJENN
000012 /* Profile: CSJENN.BATCH EVALUATION
000013 /* Desc:
000014 /* User: CSJENN
000015 /* Date: Monday October 03, 2011
000016 /* Time: 17:42:26.13
000017 /*
000018 /** * * * *

```

Figure 7: Result of batch Job Profile.

When the JCL is executed, DB2 Automation Tool evaluates the objects against the exceptions defined in the Exception Profile. In the screen shot below, the TRIGGERS DD in the job output lists the exceptions defined in the Exception Profile, and lists those objects that met or exceeded the defined exception. In this example, the tablespace AUOCOPY met both the CLUSTERSENS exception and the SCANACCESS exception and will therefore be included in the utility JCL.

```

Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY REORGJCL J0438727 DSID 107 LINE 0 COLUMNS 02- 133
COMMAND INPUT ==> SCROLL ==> CSR
***** TOP OF DATA *****
IBM Shared Profile Support -- Print Exception Triggers -- V04.10 Run Date 2011/10/04 Run Time 11:54:25

16 Triggers created...

Statistics Type--- Column----- Type- Cond -----Exception Value-----
REALTIME REORG TS CLUSTERSENS BGINT > 500

DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index---- TOWNER- IXCRTR-- PART# -----TRIGGER Value-----
AUOVRDB AUOCOPY 0 13738

Statistics Type--- Column----- Type- Cond -----Exception Value-----
REALTIME REORG TS SCANACCESS BGINT > 1000

DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index---- TOWNER- IXCRTR-- PART# -----TRIGGER Value-----
AUOVRDB AUOCOPY 0 13738

Statistics Type--- Column----- Type- Cond -----Exception Value-----
REALTIME REORG TS CLUSTERSENS BGINT > 500

DBNAME-- TSNAME-- TRIGGER Column---- TRIGGER Index---- TOWNER- IXCRTR-- PART# -----TRIGGER Value-----
AUOVRDB AUODBASE 0 4531325

```

Figure 8: Objects that match the exceptions are written to the TRIGGERS DD.

As a result, the utility JCL is built with the tablespace AUOCOPY included in the utility JCL. First, a RUNSTATS utility is generated, as shown in the screen shot below.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
EDIT CSJENN.HAA410.JCL(REQ#) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
000022 //* Step: RUN00101 *
000023 //* * *
000024 //* Desc: This step will invoke the IBM Runstats Utility *
----- 18 Line(s) not Displayed -----
000043 //SYSIN DD *
000044 LISTDEF RUNT1001
000045 INCLUDE TABLESPACE AUOVRDB.AUOCOPY
000046 INCLUDE TABLESPACE AUOVRDB.AUODBASE
000047 INCLUDE TABLESPACE AUOVRDB.AUOOBJ
000048 INCLUDE TABLESPACE AUOVRDB.AUOPKAGE
000049 INCLUDE TABLESPACE AUOVRDB.AUOPLAN
000050 INCLUDE TABLESPACE AUOVRDB.AUOSEQ
000051 INCLUDE TABLESPACE AUOVRDB.AUOUSER
000052 INCLUDE TABLESPACE AUOVRDB.AUOIEWS
000053
000054 RUNSTATS TABLESPACE LIST RUNT1001
000055 TABLE (ALL)
000056 SAMPLE 25
000057 INDEX (ALL)

```

Figure 9: Generated utility JCL contains only objects that match the exceptions.

Further down in the utility JCL, the REORG TABLESPACE utility is generated and shows tablespace AUOCOPY included in the REORG TABLESPACE utility as well. The utilities RUNSTATS and REORG TABLESPACE were included in the Utility Profile. Any combination of the available DB2 utilities can be generated.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT          CSJENN.HAA410.JCL(REO#) - 01.00          Columns 0001 0072
Command ==>
000068 /** Step:      RTS00102
000069 /**
000070 /** Desc:       This step will invoke the IBM Reorg Tablespace Utility
000071 /**
000072 /** ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
----- - 26 Line(s) not Displayed
000099 //SYSIN DD *
000100 LISTDEF REO11002
000101 INCLUDE TABLESPACE AUDVRDB.AUOCOPY
000102 INCLUDE TABLESPACE AUDVRDB.AUOBASE
000103 INCLUDE TABLESPACE AUDVRDB.AUOPKAGE
000104 INCLUDE TABLESPACE AUDVRDB.AUOPLAN
000105 INCLUDE TABLESPACE AUDVRDB.AUOSEQ
000106 INCLUDE TABLESPACE AUDVRDB.AUOUSER
000107
000108 REORG TABLESPACE LIST REO11002
000109 SCOPE ALL
000110 LOG YES
000111 SORTDATA NO

```

Figure 10: All utilities in the Utility Profile are generated into the utility JCL.

The resultant utility JCL can then be executed on demand, or added to a job scheduler to run during a specific batch window. Job Profiles can be created to build utility JCL for taking image copies, for updating statistics, for running REORGs or for any other routine maintenance your company requires.

By running a Job Profile at regular intervals to look for objects that need maintenance and to generate utility JCL, routine database maintenance can be much less manual and error-prone and allow your DBA staff to pursue other tasks.

2.3 Using the DB2 administrative task scheduler to automatically execute utility JCL

Do you have a job scheduler to run the Job Profiles or the resultant utility JCL in batch? Or do you manually run the utility JCL as needed? The generated utility jobs created by DB2 Automation Tool can be manually added to any job scheduler by the user. But if you are running on DB2 v8 or higher, you can now integrate DB2 Automation Tool v4.1 with the DB2 Administrative Task Scheduler and automatically execute *any* JCL. By selecting option 12 from the main DB2 Automation Tool menu, you can invoke the Db2 administrative task scheduler to add a job to the scheduler.

```

HAA$MAIN V4R1 --- IBM DB2 Automation Tool for z/OS --- 2011/10/04 12:25:31
Option ==> 12
-----
Options: 0 - Setup                      8 - Dataset Manager
         1 - Object Profiles             9 - Data Page Display
         2 - Utility Profiles            10 - Disaster Recovery
         3 - Exception Profiles          11 - Stand Alone Utilities
         4 - Job Profiles                12 - DB2 Admin Scheduler
         5 - Quick Build                 13 - DB2 Autonomic Statistics
         6 - Execution Reports           X - Exit
         7 - DB2 Command Processor
-----
DB2 Subsystem ID: DA1A (1-4 Character Subsystem ID or ? for list)
Current SQLID: CSJENN__ User: CSJENN - Q41
-----

```

Figure 11: DB2 Administrative Task Scheduler accessible via DB2 Automation Tool v4.1.

From the DB2 Admin Task Scheduler panel, you can create a new batch job to execute during a specific time interval, view the results of a task that has already run, update a task, or delete a task.

```

AUTOTOOL V4R1 ----- DB2 Admin Task Scheduler ----- 2011/10/04 12:38:56
Option ==> _____ Scroll ==> CSR_
-----
Line Commands: C - Create D - Delete V - View S - Status U - Update
-----
Task Name Like * _____ DB2 Subsystem: DA1A
Task Creator Like CSJENN__ Row 1 of 13 +>
-----
Task Description Last Modified
Cmd Name _____
- AUTOTOOL STATMONITOR: TESTING 2011-08-26-15.22
- BETA JOB PROFILE: JCLMSGU 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSGV 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSGW 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSGX 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSGY 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSGZ 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSG0 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSG1 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSG2 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSG3 2011-07-15-16.36
- BETA JOB PROFILE: JCLMSG4 2011-07-15-16.36

```

Figure 12: Adding JCL job to the scheduler.

The benefit of integrating with the DB2 administrative task scheduler is when you build your Job Profile as shown in the screen shot below.

```

Build Job for CSJENN.BATCH EVALUATION

Build Online or Batch . . b (O - Online, B - Batch)

Edit Generated Job . . . Y (Yes/No)
Schedule Job . . . . . y (Yes/No)   Update options . . y (Yes/No)

Build job in Dataset . . CSJENN.HAA410.JCL
Member . . . REORGB

Job Cards:
==> //REORGJCL JOB CSJENN CLASS=A NOTIFY=&SYSUID MSGLEVEL=(1,1)
==> //          MSGCLASS=X
==> // * EVALUATE OBJECTS TO INCLUDE IN A REORG
==> // *

```

Figure 13: Generating JCL in DB2 Automation Tool and adding to Scheduler.

When building your Job Profile that evaluates exceptions and generates maintenance utility JCL, the options ‘Schedule Job’ and ‘Update Options’ enables you to add your Job Profile to the DB2 administrative task scheduler so that it will run at regular intervals, or just once; which ever method you choose.

In the example below, the Job Profile “Batch Evaluation” is being added to the DB2 administrative task scheduler. The options presented on this panel are specific to the DB2 administrative task scheduler, and their descriptions can be found in the DB2 Administration Guide.

The fields Begin Timestamp and End Timestamp enable you to define the exact time you want to execute this JCL. However, since we want this Job Profile to run at regular intervals, the Interval Options have been defined instead. This batch JCL will execute every Saturday of the month throughout the year at 12 noon to evaluate the objects in the corresponding Object Profile that need a RUNSTATS and a REORG run.

```

AUTOTOOL V4R1 ----- Schedule DB2 Admin Task ----- 2011/10/04 12:45:28
Option ==> _____ Scroll ==> CSR

Task Name . . . . batch_evaluation >
Task Description . . . evaluate objs for runstats and reorg >

Begin Timestamp . . _____ (DB2 Timestamp)
End Timestamp . . . _____ (DB2 Timestamp)
Max Invocations . . 1 (Integer, Blank)
SSID . . . . . DATA (Blank for any datasharing member)
Job Wait . . . . . Y (Y - Yes, N - No, P - Purge)
Execution Threads . . 001 (Integer)

Invocation Options:
Interval (minutes) _____ (Integer, Blank)
-Or-
Trigger:
Task Name . . . _____ >
Cond . . . . . _____ (GT,GE,EQ,LT,LE,NE)
Code . . . . . _____ (Integer, Blank)
-Or-
Point in Time . . 0 12 * * sat >

```

Figure 14: Execute JCL at a specific time or during an interval.

The values in the Point in Time fields are:

Position 1: Minute. Valid values are 0 through 59.

Position 2: Hour. Valid values are 1 through 23.

Position 3: Day of the Month. Valid values are 1-31.

Position 4: Month. Valid values are 1 through 12, where 1 is January. Three-character strings are also allowed, i.e., JAN, FEB, etc.

Position 5: Day of the week. Valid values are 0 through 7, where 0 or 7 is Sunday. Three-character strings are also allowed, i.e., SUN, SAT, etc.

Ranges and lists are allowed. Ranges are simply two values separated by a hyphen, and mean all values are inclusive. Lists are simply two or more values separated by a comma, and mean only those values are included.

When you have completed adding the Job Profile to the DB2 administrative task scheduler, you can then add the utility JCL jobs to the DB2 administrative task scheduler, too. You will see the same 2 options as before, 'Schedule Job' and 'Update Options'.

```
Build Job for CSJENN.BATCH EVALUATION

You have selected your job to be built in a batch mode. The
batch generation JCL will be stored in dataset
CSJENN.HAA410.JCL(REORGB).
Please specify in the dataset and member below where you want the
JCL built by the batch module to be placed.

Schedule Job . . . . y (Yes/No)  Update options . . y (Yes/No)

Build job in Dataset. . CSJENN.HAA410.JCL_____
Member . . reorgj__

Jobcard data to be used on the generated job:
=> //REORGJ_____JOB CSJENN.CLASS=A.NOTIFY=&SYSUID_____
=> //*_____
=> /* THIS IS THE ACTUAL UTILITY JCL_____
=> /*_____
```

Figure 15: Utility JCL can also be added to the Scheduler.

When the panel Schedule DB2 Admin Task is displayed, you will specify when the resultant maintenance utility JCL should be run. Remember that the previous DB2 admin task panel enabled you to define when to run the Job Profile that evaluates your objects against the exception criteria and generates utility JCL. This time, you are defining when to run the actual utilities themselves.

In this example, the generated maintenance utility jobs will run between the hours of midnight and 3 AM each Sunday. The field 'Execution Threads' specifies that up to 5 jobs can run at once.

```

AUTOTOOL V4R1 ----- Schedule DB2 Admin Task ----- 2011/10/04 13:14:46
Option ==> _____ Scroll ==> CSR_

Task Name . . . . batch_evaluation_utility_jobs >
Task Description these_jobs_are_the_actual_utility_jobs >

Begin Timestamp . . _____ (DB2 Timestamp)
End Timestamp . . . . _____ (DB2 Timestamp)
Max Invocations . . 1 _____ (Integer, Blank)
SSID . . . . . DA1A (Blank for any datasharing member)
Job Wait . . . . . Y (Y - Yes, N - No, P - Purge)
Execution Threads . . 005 (Integer)

Invocation Options:
Interval (minutes) _____ (Integer, Blank)
-Or-
Trigger:
Task Name . . . . _____ >
Cond . . . . . _____ (GT,GE,EQ,LT,LE,NE)
Code . . . . . _____ (Integer, Blank)
-Or-
Point in Time . . 0 00-03 * * sun >

```

Figure 16: Schedule the resultant utility JCL with different parameters.

3 Evolve your strategy with an intelligent solution

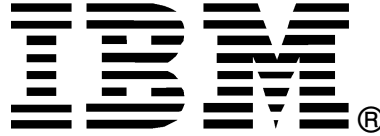
The maintenance utility jobs that were generated by DB2 Automation Tool can now be scheduled and automatically run with DB2 Automation Tool's interface to the DB2 administrative task scheduler. Now, you can automate your database maintenance strategy with DB2 Automation Tool from start to finish; from finding which objects to include to generating and submitting utility JCL. And your DBA intervention can be reduced to a minimum, allowing you to spend your time on other business critical tasks, while still running important database maintenance in the background. The combination of DB2 Automation Tool v4.1 for z/OS and the DB2 Administrative Scheduler allow you to fully automate your database maintenance to achieve maximum control and availability of your critical DB2 objects, while minimizing the system and staff resources that cost you time and money.

REFERENCES

IBM DB2 Automation Tool for z/OS v4.1 User's Guide, SC19-3494-00

IBM DB2 Version 9.1 for z/OS Administration Guide, SC18-9840-00

IBM DB2 10 for z/OS Administration Guide, SC19-2968-00



© Copyright IBM Corporation 2011

IBM United States of America

Produced in the United States of America

All Rights Reserved

The e-business logo, the eServer logo, IBM, the IBM logo, OS/390, zSeries, SecureWay, S/390, Tivoli, DB2, Lotus and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both.

Lotus, Lotus Discovery Server, Lotus QuickPlace, Lotus Notes, Domino, and Sametime are trademarks of Lotus Development Corporation and/or IBM Corporation.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PAPER "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Information in this paper as to the availability of products (including portlets) was believed accurate as of the time of publication. IBM cannot guarantee that identified products (including portlets) will continue to be made available by their suppliers.

This information could include technical inaccuracies or typographical errors. Changes may be made periodically to the information herein; these changes may be incorporated in subsequent versions of the paper. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this paper at any time without notice.

Any references in this document to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
4205 South Miami Boulevard
Research Triangle Park, NC 27709 U.S.A.