Mass Data Runtime - Administrators Guide

Manual

7.2 — Last update: 2015/03/04

Basis Technologies

Table of Contents

Audience

This guide has been developed for the following audience:

- SAP® Basis Administrators
- SAP® performance specialists or advisers

Introduction

Basis Technologies’ Mass Data Runtime (MDR) provides a framework and run-time environment for custom developed and standard SAP programs that must work with large volumes of data. It provides for parallel execution such that these programs can run within acceptable time constraints to deliver timelier business information. This is achieved by:

- Providing a frame-work and methodology for development of reports such that they can be scaled according to the resources available in order to maximize use of the available hard-ware
- Providing a frame-work for execution of ABAP code using parallel processing such that report run faster and more efficiently
- Enabling huge volumes of data to be processed within dramatically reduced timescales
- Providing a frame-work toolset for the monitoring and administration of the parallel execution of reports
- Facilitating the presentation and delivery of reports, enabling the business to interactively access current and historical report information

When to use MDR

MDR is essential when the data volumes are so large that the processing time to run your reports is unacceptable. If you are able to write a report that runs within acceptable time-constraints then the use of MDR may not be required. However, even if your report runs within 2 hours (and this is considered acceptable) you are still able to use MDR to bring this run-time down even further.

It is our consideration that almost any report that executes in the background can gain benefits by being implemented in MDR. This is because the MDR format not only promotes performance improvements, but also cost reductions in maintenance by having a generic format. It also provides the benefit of separating the processing and presentation logic. Separate presentation logic allows the data to be viewed interactively and in a user-friendly manner – far easier than lengthy static list output.

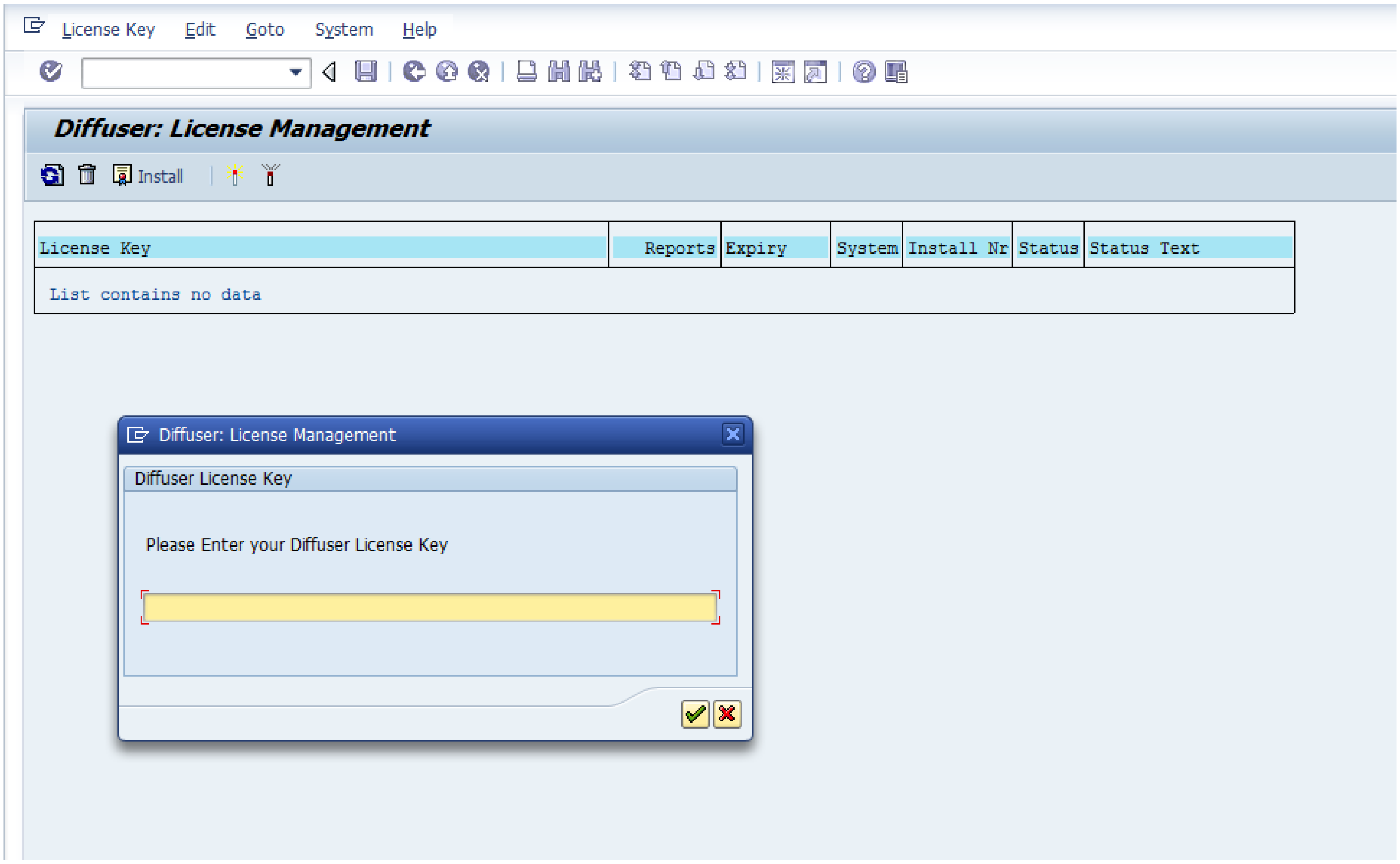
Setting Up MDR

Once the MDR program is constructed you define the MDR program, setup the technical settings and generate intervals where required. The Definition of an MDR program is set up via the transaction /BTR/MDR.

- Program Definition
- Default Technical Settings
- Interval Generation

License Keys

Use the transaction /N/BTR/MDRLICENSE then select the option “Install” and a dialog appears into which you can enter the MDR license key.



Diffuser: License Management

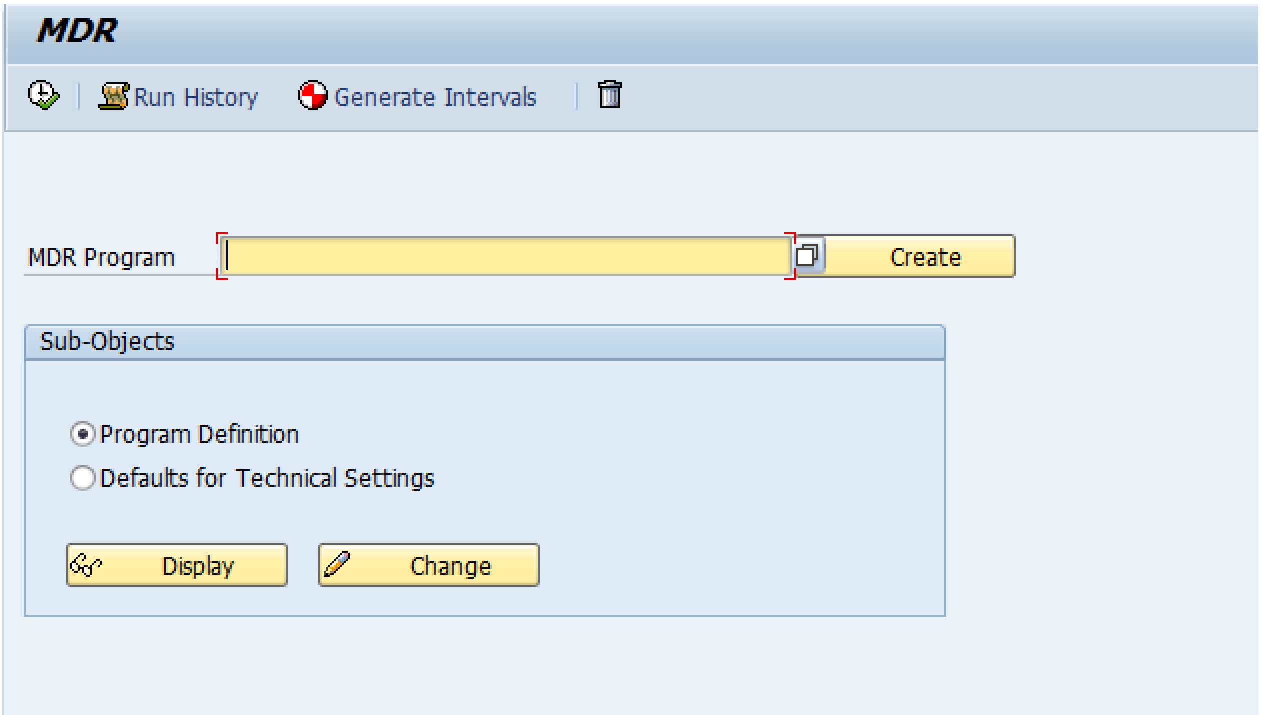
License Key	Reports	Expiry	System	Install Nr	Status	Status Text

List contains no data

Diffuser: License Management

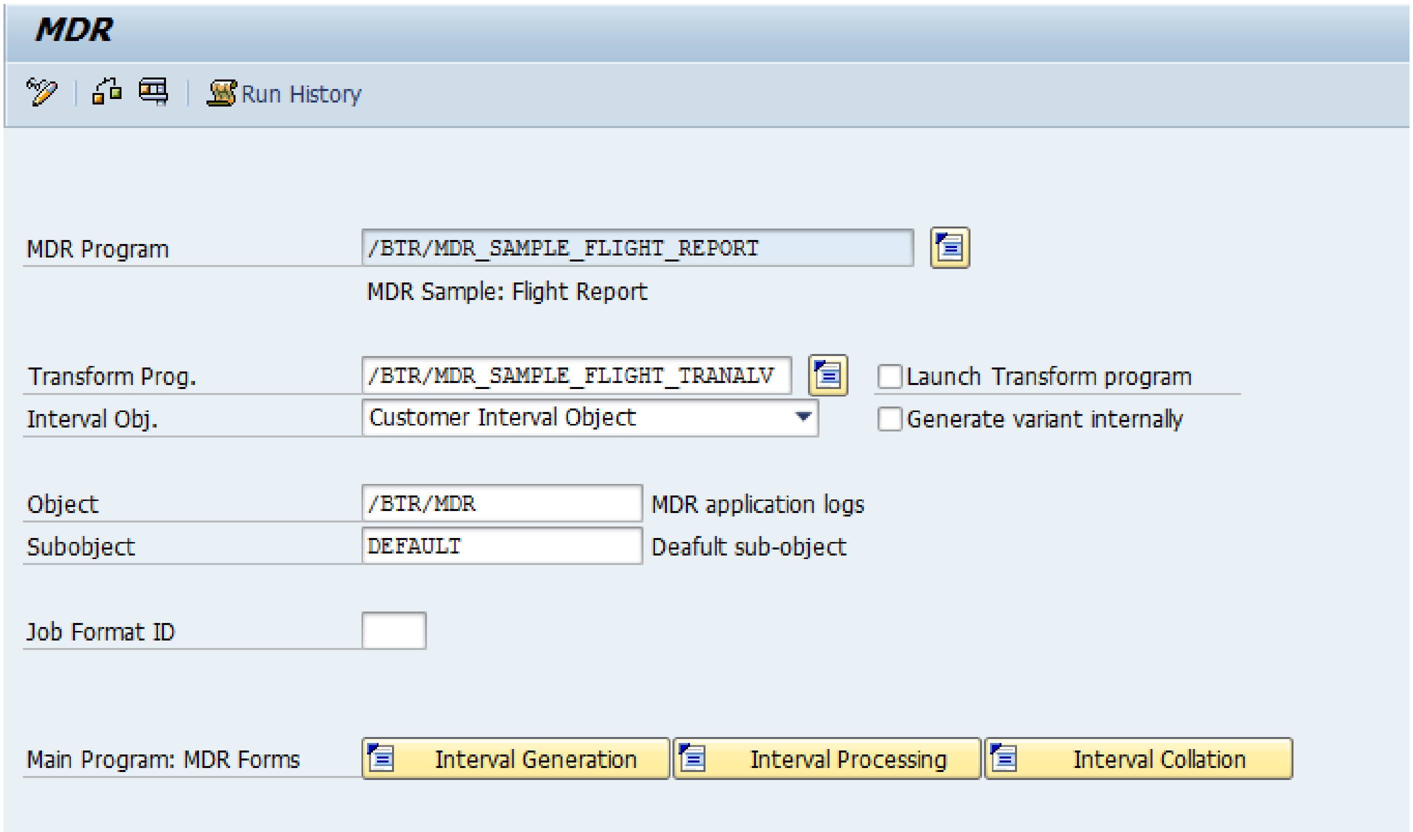
Program Definition

The definition of an MDR program is set up via the transaction /BTR/MDR.



Simply enter the program name and press the create button for new programs or change button for an existing program with the sub-object as program definition.

The definition is now displayed as below.



MDR Program /BTR/MDR_SAMPLE_FLIGHT_REPORT

MDR Sample: Flight Report

Transform Prog. /BTR/MDR_SAMPLE_FLIGHT_TRANALV

☐ Launch Transform program

Interval Obj. Customer Interval Object

☐ Generate variant internally

Object /BTR/MDR MDR application logs

Subobject DEFAULT Deafult sub-object

Job Format ID

The main program is already populated from the first screen, a transformation program can now be configured it required.

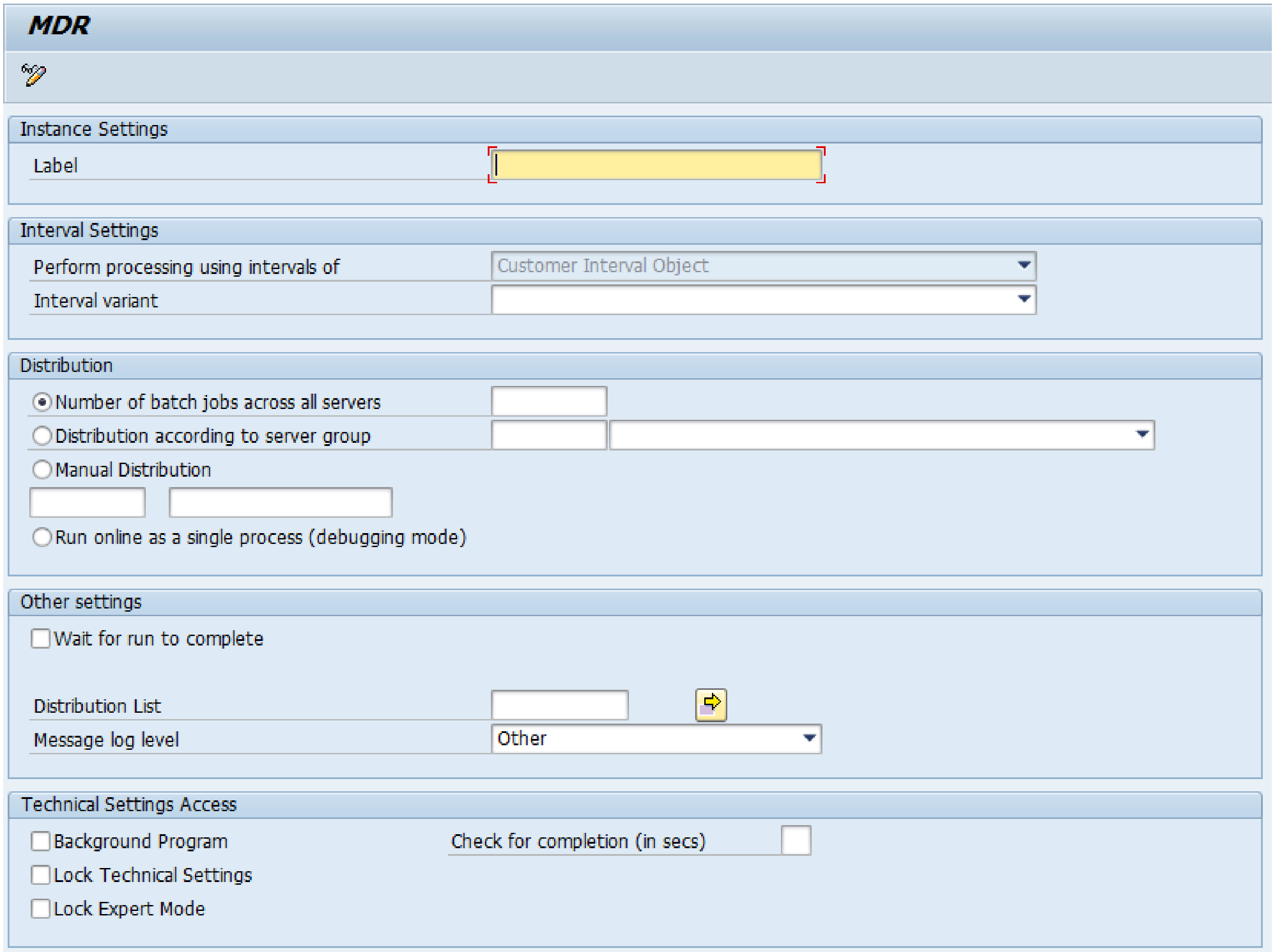
The “Interval Object” object is also populated here, refer here for more information on intervals.

The program definition also allows the user to configure which application log object and sub-object any messages are written to that are called during the execution of the program. The default object and sub-object are /BTR/MDR/DEFAULT.

Furthermore, the transaction /BTR/MDR allows the developer to maintain the Main Program and Transformation Program directly instead of using the standard SAP transaction SE38, with the code buttons at the bottom linking directly to the right subroutines.

Default Technical Settings

The second sub-object managed through the transaction /BTR/MDR is “Defaults for Technical Settings” this screen contains two main sections. The first section “Defaults for Technical Settings” allows to set default values for a specific MDR program. Once set, these values will always appear on the Technical Settings’ sub screen for that program see Technical Settings under Running MDR Programs for more details.



MDR

Instance Settings

Label	

Interval Settings

Perform processing using intervals of	Customer Interval Object
Interval variant	

Distribution

● Number of batch jobs across all servers
○ Distribution according to server group
○ Manual Distribution
○ Run online as a single process (debugging mode)

Other settings

☐ Wait for run to complete

Distribution List	
Message log level	Other

Technical Settings Access

☐ Background Program
☐ Lock Technical Settings
☐ Lock Expert Mode

Check for completion (in secs)	

The second section “Technical Settings Access” we will explore in more detail.

Background Program

NOTE: This functionality requires “Wait for run to complete” to be set.

This option supresses the default display of the Run History screen after completion of an instance run. This is a useful feature that allows an MDR program to be called from another program without interrupting the latter with the Run History display.

The input field “Check for completion (in sec)” allows to set in seconds a time interval in which the parent job of a running MDR instance will wake up and check if all child parallel processes have completed. The default wake up and check time is 30 seconds which is suitable for very long running programs but not for speeding up, say, web services where every second counts.

Lock Technical Settings

This options allows to lock all input fields for Technical Settings. This is useful if when a program can repeatedly run with the same default values and users should not change those values. When this option is set, the Expert Mode in the Run History will be locked as well.

This restriction applies at program level and not at user level. That is, once set the Technical Settings will be locked for all users. Restrictions at user level can be implemented with the MDR enhancement spots (see MDR Enhancement Spots document).

Lock Expert Mode

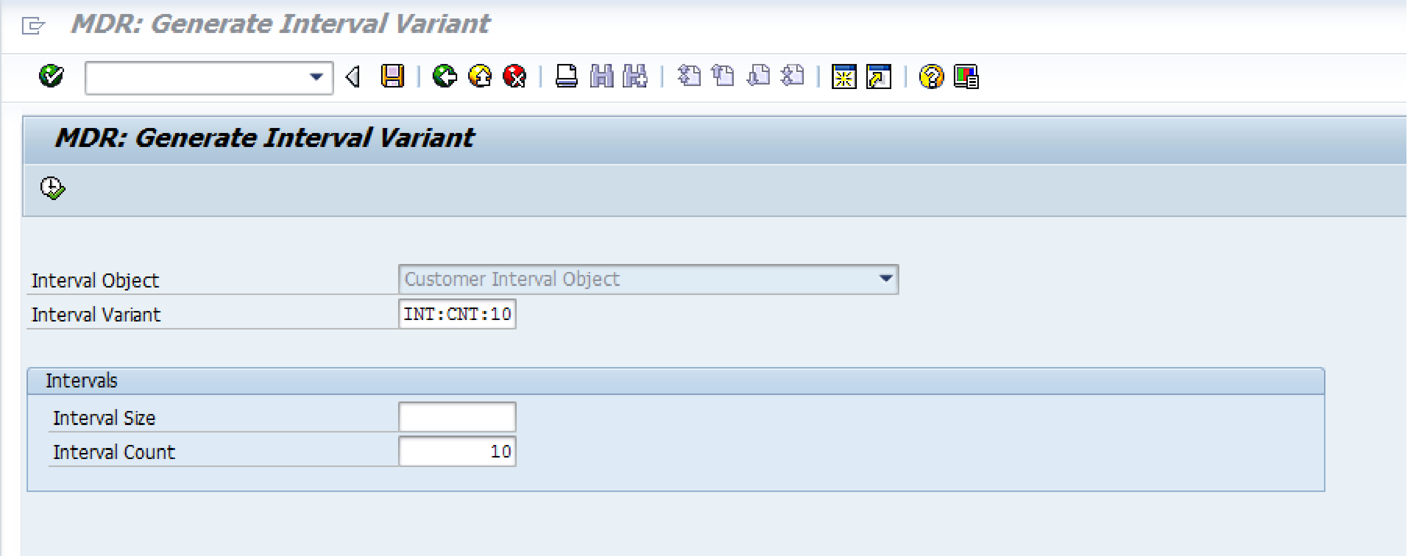
This option is similar to “Lock Technical Settings”. The only difference is that on the Technical settings screen only the input fields under “Distribution” are locked. This allows the user to change settings like label name while protecting the more critical job distribution section from potential misuse. This option applies at program level as well. Restrictions at user level can be implemented with see the MDR Developers Guide Authority Checks

Interval Generation

When an MDR program uses Interval Objects, an Interval Variant needs to be created from the Interval Object, before the MDR program is run. An Interval Variant can be thought of as the set of Intervals that the MDR program is going to use. It is necessary that new Interval Variants are generated regularly (potentially before each batch run) to ensure that the intervals are split evenly as the data in the system grows.

To use an Interval Objects, they must first be configured into the framework via table /BTR/INTVALOBJ.

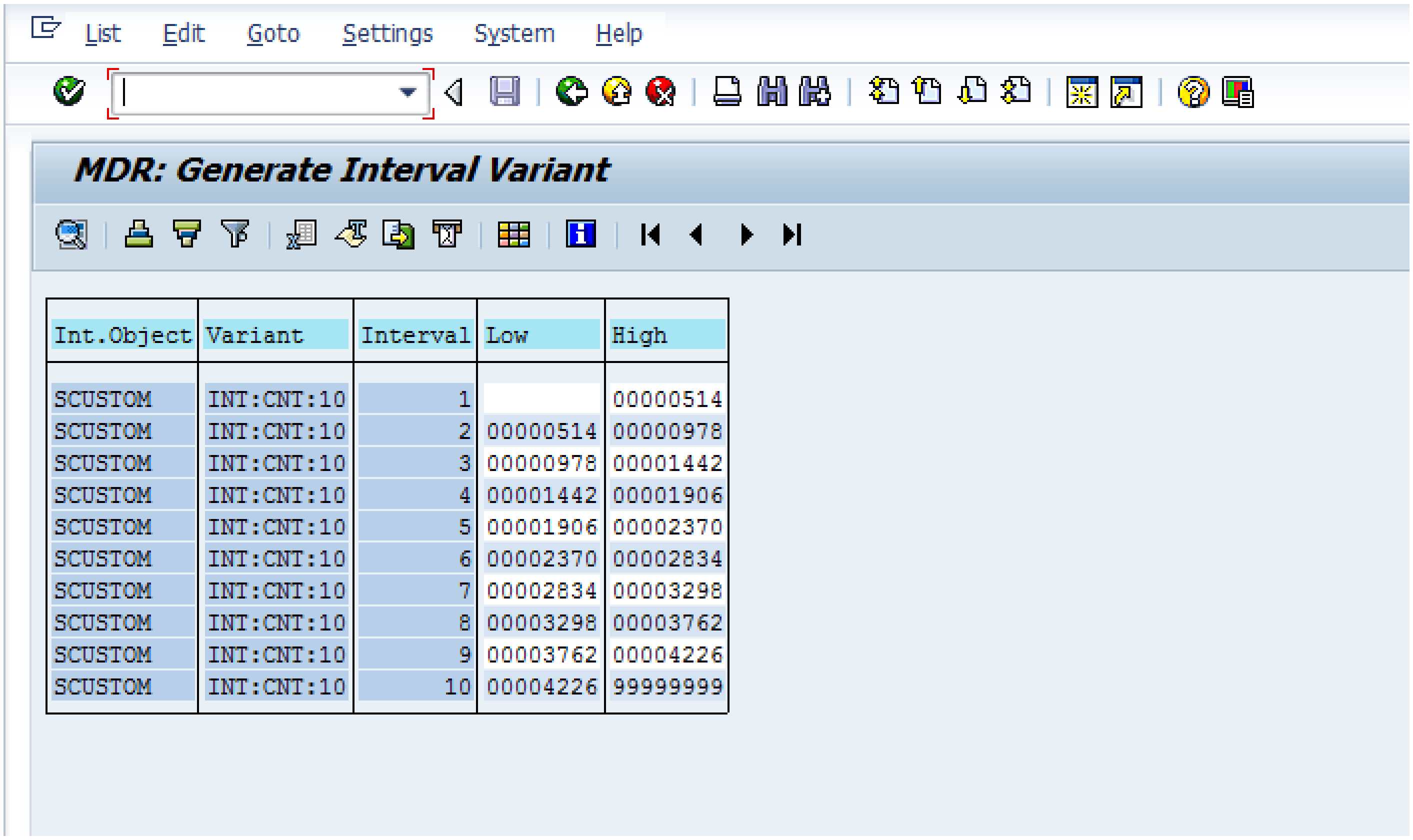
There are two different types of Interval Objects; standard SAP Mass Run Interval Objects and MDR Interval Objects. Both functions in the same manner, except for the generation of the Interval Variants is performed differently. The Intervals (or Interval Variant) are created before the MDR program is executed; this is done via the program /BTR/MDR_INTERVAL_GENERATION or alternately for standard SAP Interval variants via transaction FQD2. Either an Interval Count or Interval Size can be used as parameters to how the Intervals get generated.


MDR: Generate Interval Variant
MDR: Generate Interval Variant
Intervals

Interval Object: Customer Interval Object
Interval Variant: INT:CNT:10
Interval Size:
Interval Count: 10

The above example shows the generation of a new Interval Variant called INT:CNT:10 from the Interval Object “Customer Interval Object”, with the requirement that 10 Intervals are to be created.

As can be seen below, the result is 10 generated Intervals.



MDR: Generate Interval Variant

Int.Object	Variant	Interval	Low	High
SCUSTOM	INT:CNT:10	1		00000514
SCUSTOM	INT:CNT:10	2	00000514	00000978
SCUSTOM	INT:CNT:10	3	00000978	00001442
SCUSTOM	INT:CNT:10	4	00001442	00001906
SCUSTOM	INT:CNT:10	5	00001906	00002370
SCUSTOM	INT:CNT:10	6	00002370	00002834
SCUSTOM	INT:CNT:10	7	00002834	00003298
SCUSTOM	INT:CNT:10	8	00003298	00003762
SCUSTOM	INT:CNT:10	9	00003762	00004226
SCUSTOM	INT:CNT:10	10	00004226	99999999

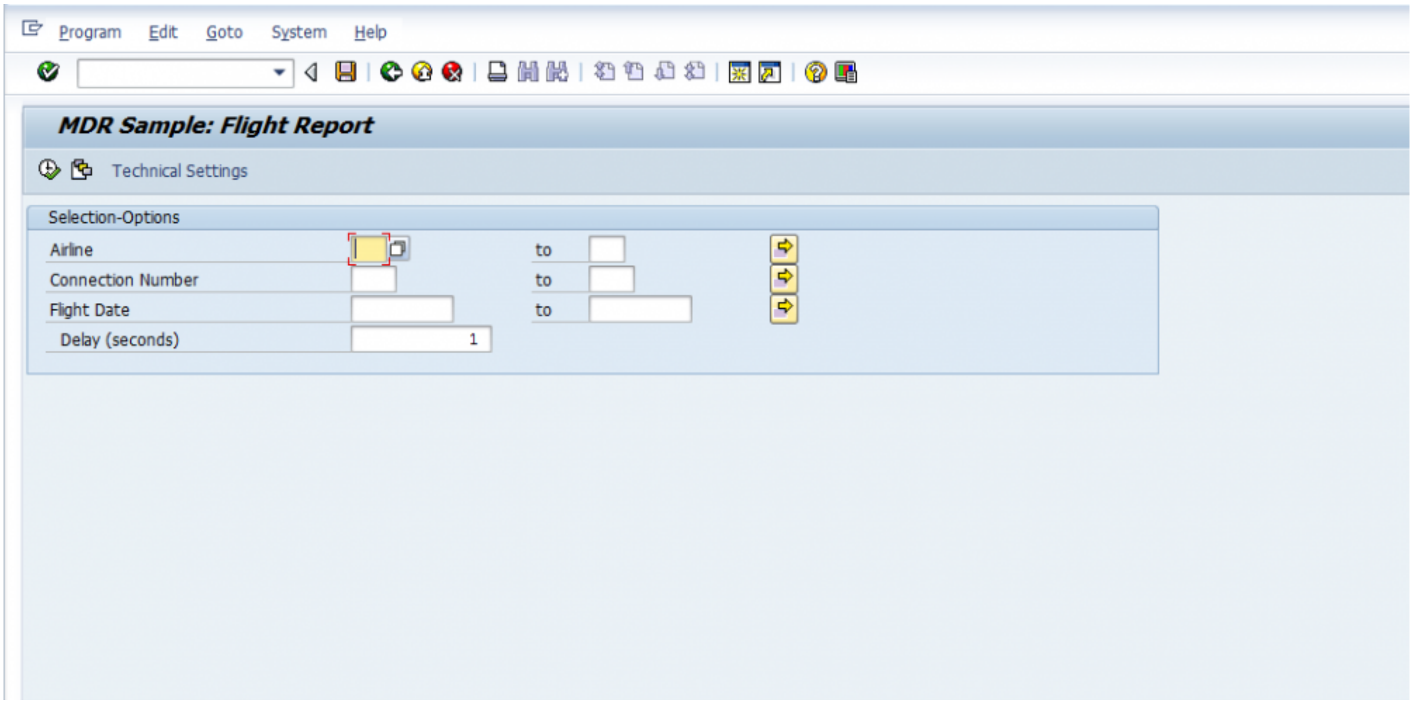
Running MDR Programs

Before running an MDR program the technical settings need to be completed, once running to administer the program see Administering MDR Programs section.

- Technical Settings

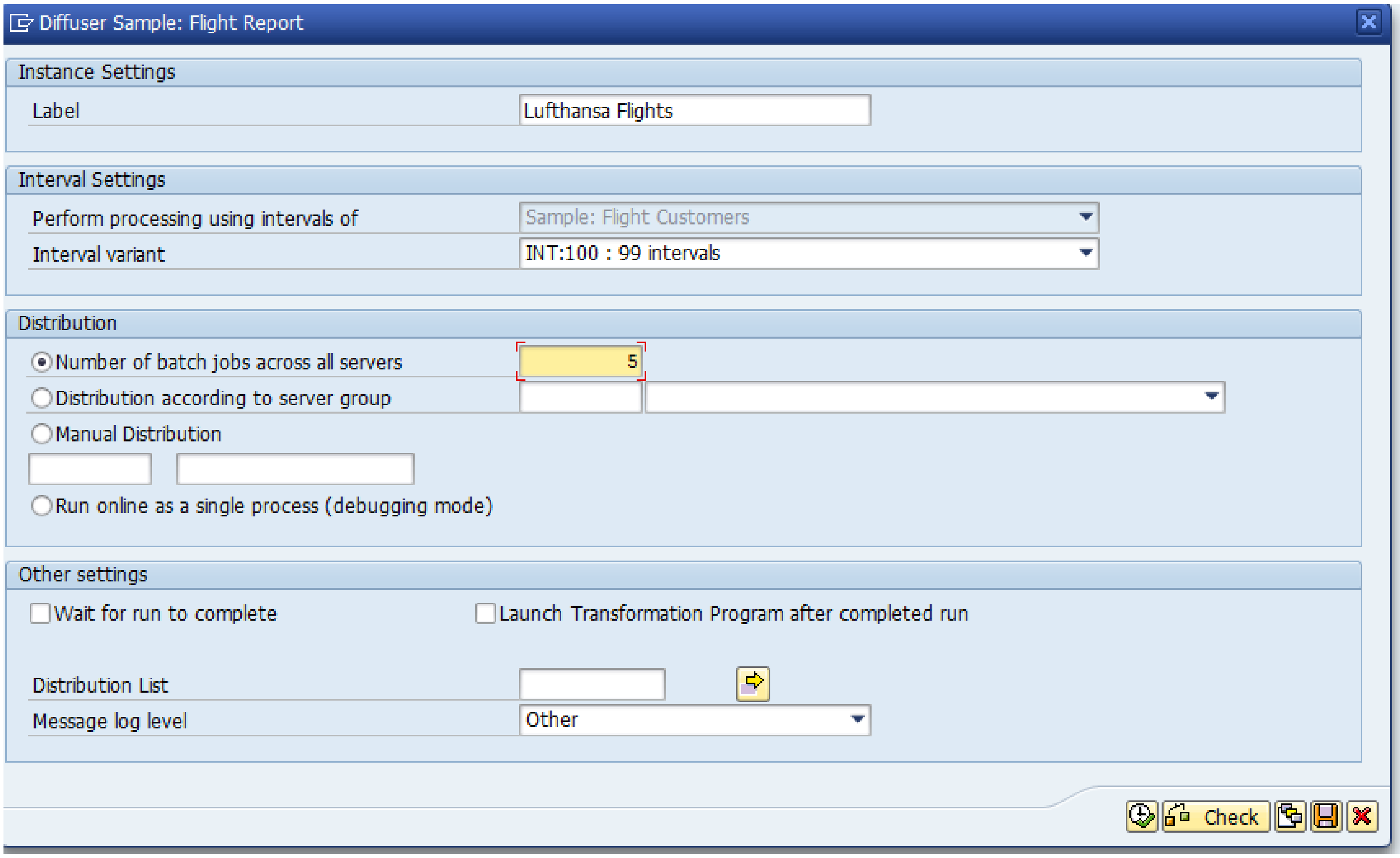
Technical Settings

The key part that the user sees is the “Technical Settings” button as below.



Delay (seconds) 1

If you select the “Technical Settings” button, you will be prompted for the MDR specific technical settings.



Diffuser Sample: Flight Report

Instance Settings

Label: Lufthansa Flights

Interval Settings

Perform processing using intervals of: Sample: Flight Customers

Interval variant: INT:100 : 99 intervals

Distribution

● Number of batch jobs across all servers: 5

○ Distribution according to server group

○ Manual Distribution

○ Run online as a single process (debugging mode)

Other settings

☐ Wait for run to complete

☐ Launch Transformation Program after completed run

Distribution List

Message log level: Other

These “Technical Settings” are important when executing an MDR program be it in a production environment, or when performing unit testing of your MDR program.

Note that you can setup Default Technical Settings and setup user authorizations to control the users ability to change these settings, for more information refer to the Mass Data Runtime – Developers Guide Authority Checks.

An explanation for the function of each field on the “Technical Settings” screen is as below:

- Label – The first is a label that can be specified to identify this particular execution
- Perform processing using intervals of – This is the interval object confirmed in the Program Definition
- Interval Variant – The Interval variant provides you with a list of different pre-generated Intervals. As detailed in the section Interval Generation, the interval variants are pre-generated using program /BTR/MDR_INTERVAL_GENERATION
- Number of batch jobs across all servers –This specifies the number of processes with which to run the MDR program

- Distribution according to server grouping –This allows the distribution of jobs over one server group to control the number of processors available to this MDR instance

- Manual Distribution –The server grouping above can also be distributed manually

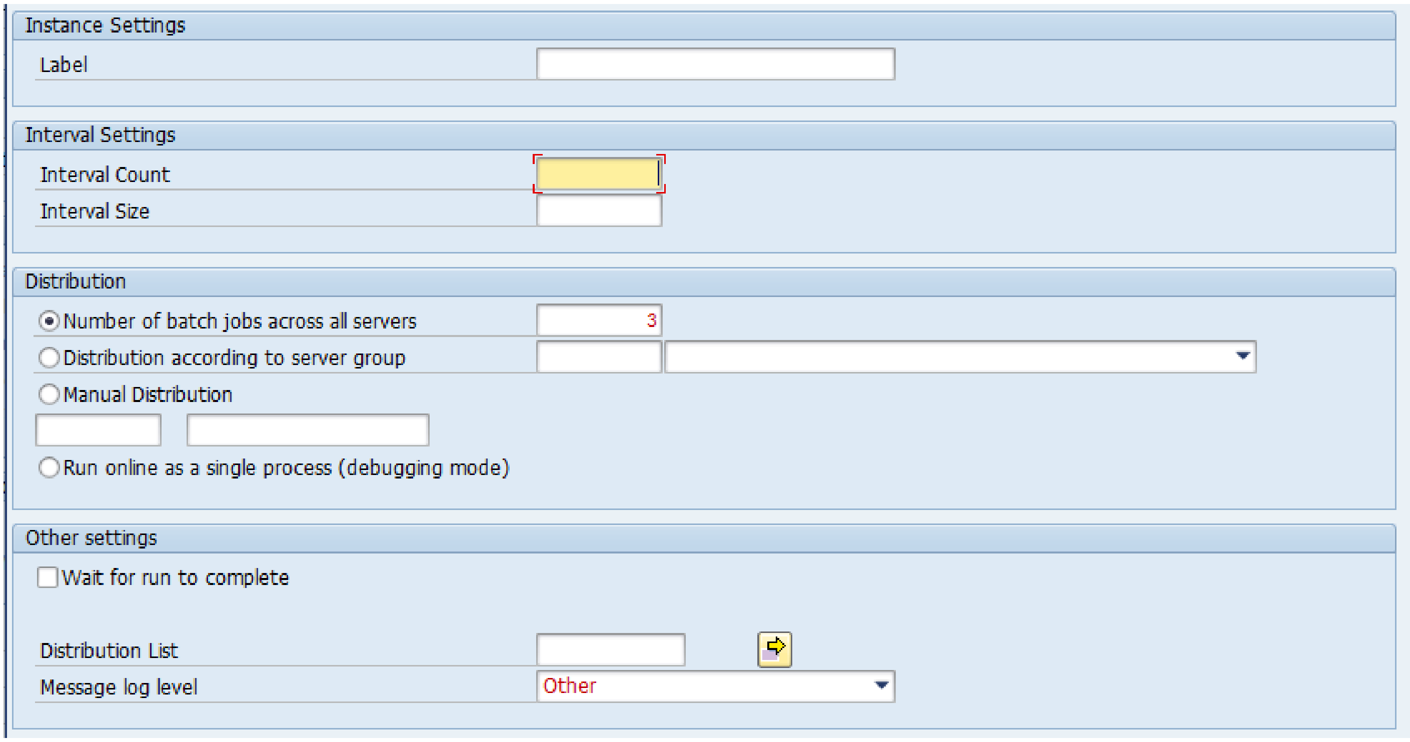
- Run online as a single process (debugging mode) –This is only used when debugging MDR programs and ensures the whole program runs sequentially

- Wait for run to complete before finishing –This is often used when running MDR programs on-line or when executing them via a job scheduler. It will ensure the parent process waits until all child processes have completed. Once all child processes have finished, control is returned to the parent for completion

- Distribution List –After a program completes within the MDR framework it is able to send a SAP office document or external email to a set of recipients that can be specified here

- Message log level – Lower limit for the priority of messages output to the application log by MDR. For example, you can restrict output of informational application log messages by increasing the log level via this parameter

When using Dynamic Intervals as set out in the Mass Data Runtime – Developers Guide Dynamic Interval Generation” the “Interval Settings” section will change and be replaced with the two options Interval Count and Interval Size introduced, this looks as per the screenshot below.



Instance Settings

Label

Interval Settings

Interval Count

Interval Size

Distribution

● Number of batch jobs across all servers: 3

○ Distribution according to server group

○ Manual Distribution

○ Run online as a single process (debugging mode)

Other settings

☐ Wait for run to complete

Distribution List

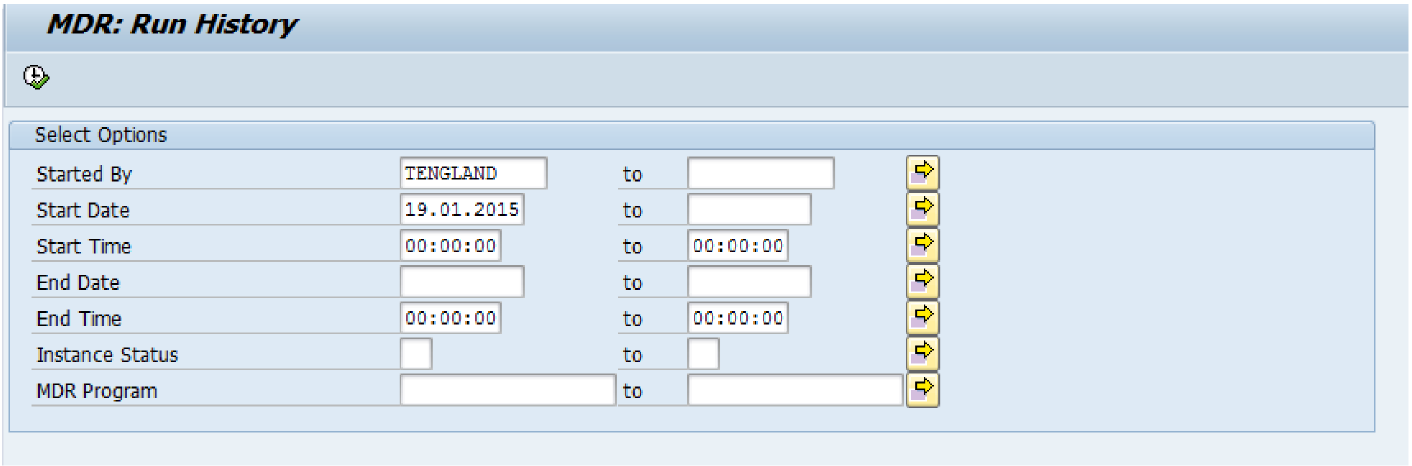
Message log level: Other

The impact of the two fields is as below:

- Interval Count –This specifies the number of intervals (chunks) that the total amount of work to be done is to be broken up into
- Interval Size –This specifies the “number of objects” to be put into each interval to be then worked upon independently

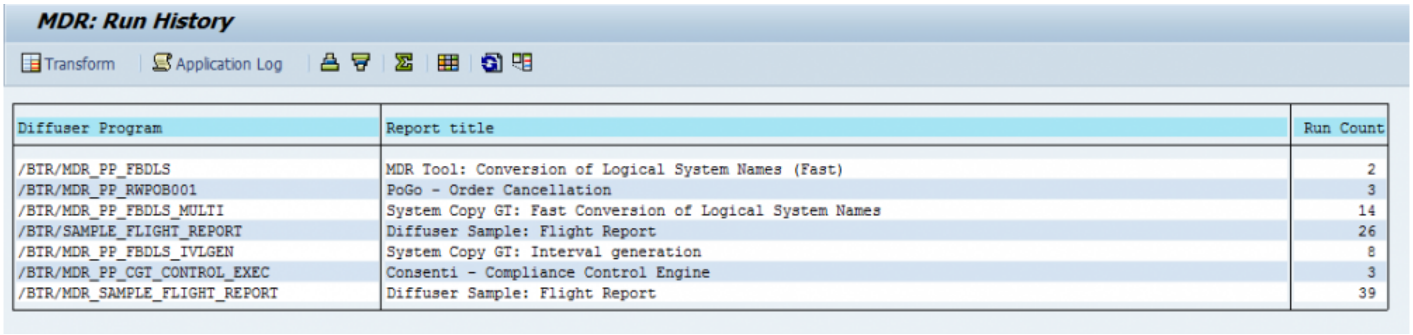
Accessing Run History

Historical instance runs and results can be easily accessed via transaction /BTR/MDRH. It allows to search by user, time period, status and program. This is especially useful for making result data available to users without having to rerun the programs. In the selection screen insert your search criteria and execute.



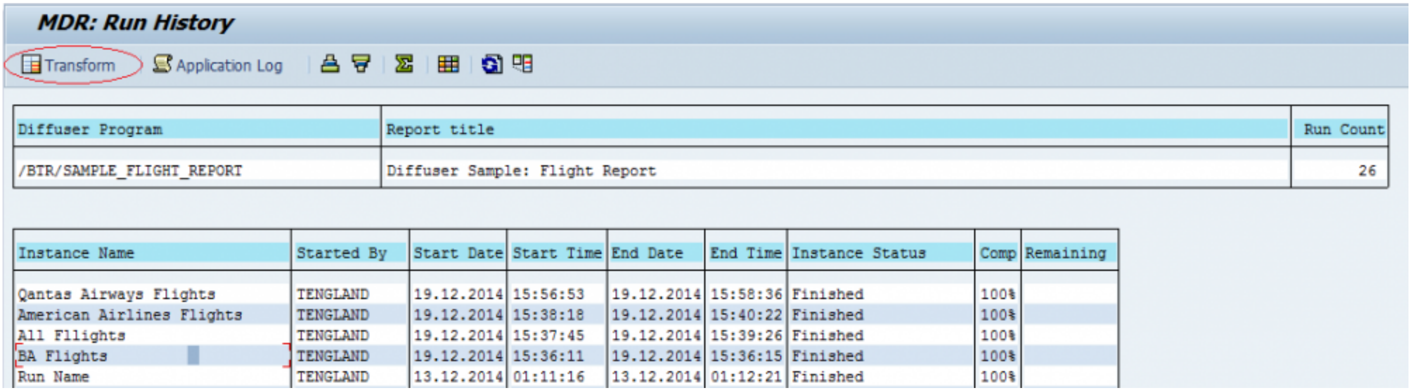
Started By	TENGLAND	to	
Start Date	19.01.2015	to	
Start Time	00:00:00	to	00:00:00
End Date		to	
End Time	00:00:00	to	00:00:00
Instance Status		to	
MDR Program		to	

Run History will show a list of the MDR defined program(s) with instances relevant to the search criteria.



Diffuser Program	Report title	Run Count
/BTR/MDR_PP_FBDLS	MDR Tool: Conversion of Logical System Names (Fast)	2
/BTR/MDR_PP_RWPOB001	PoGo - Order Cancellation	3
/BTR/MDR_PP_FBDLS_MULTI	System Copy GT: Fast Conversion of Logical System Names	14
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26
/BTR/MDR_PP_FBDLS_IVLGEN	System Copy GT: Interval generation	8
/BTR/MDR_PP_CGT_CONTROL_EXEC	Consenti - Compliance Control Engine	3
/BTR/MDR_SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	39

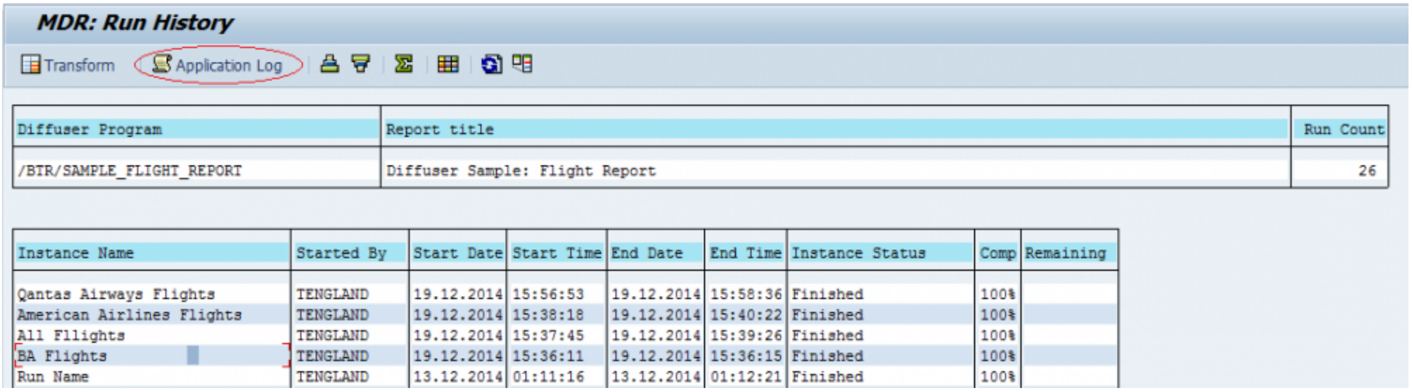
By drilling down on the program name the user will access the programs instance runs. Select an instance and click “Transform” to display the results of the run.



Diffuser Program	Report title	Run Count
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26

Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining
Qantas Airways Flights	TENGLAND	19.12.2014	15:56:53	19.12.2014	15:58:36	Finished	100%	
American Airlines Flights	TENGLAND	19.12.2014	15:38:18	19.12.2014	15:40:22	Finished	100%	
All Fllights	TENGLAND	19.12.2014	15:37:45	19.12.2014	15:39:26	Finished	100%	
BA Flights	TENGLAND	19.12.2014	15:36:11	19.12.2014	15:36:15	Finished	100%	
Run Name	TENGLAND	13.12.2014	01:11:16	13.12.2014	01:12:21	Finished	100%	

In the same manner you can check the application log for error messages.



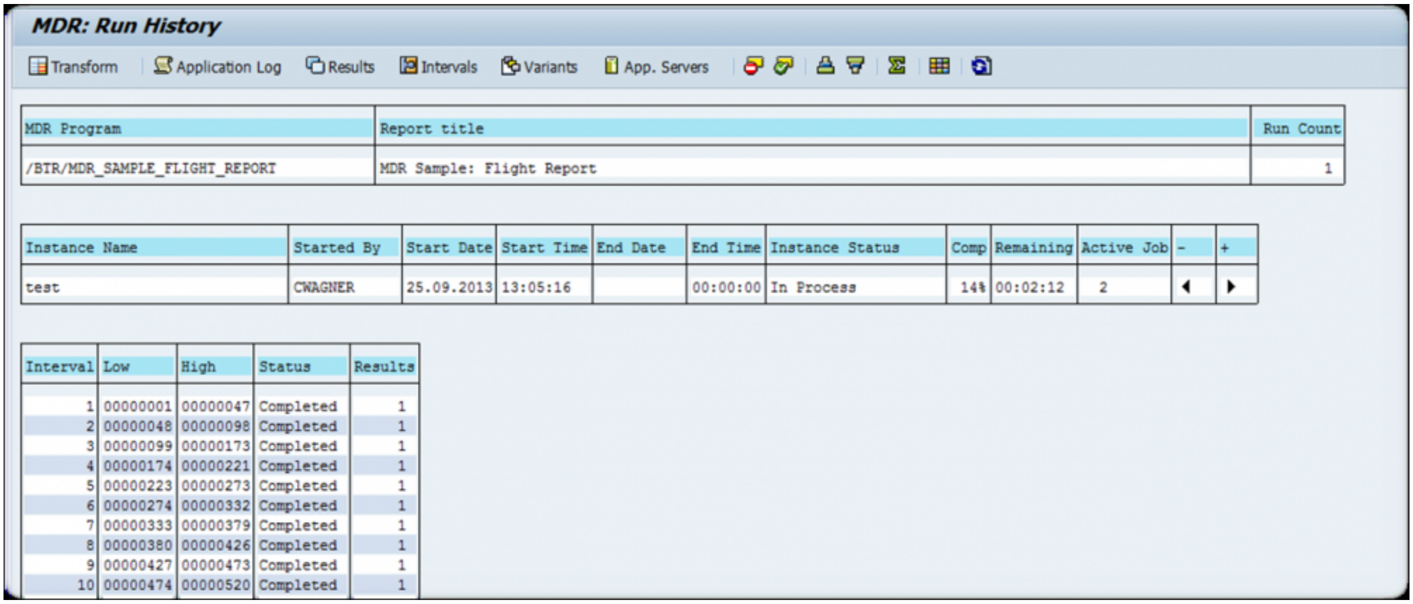
Diffuser Program	Report title	Run Count
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26

Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining
Qantas Airways Flights	TENGLAND	19.12.2014	15:56:53	19.12.2014	15:58:36	Finished	100%	
American Airlines Flights	TENGLAND	19.12.2014	15:38:18	19.12.2014	15:40:22	Finished	100%	
All Fllights	TENGLAND	19.12.2014	15:37:45	19.12.2014	15:39:26	Finished	100%	
BA Flights	TENGLAND	19.12.2014	15:36:11	19.12.2014	15:36:15	Finished	100%	
Run Name	TENGLAND	13.12.2014	01:11:16	13.12.2014	01:12:21	Finished	100%	

Once on the screen above the user will be able to see and administer historical data as well as instances in progress using the functionality mentioned in Administering MDR Programs

Administering MDR Programs

MDR provides the advanced user a number of powerful administrative capabilities via the “Run History” report accessible via transaction /BTR/MDRH (see screen below). These capabilities provide a powerful way of managing your MDR programs.



MDR Program	Report title	Run Count
/BTR/MDR_SAMPLE_FLIGHT_REPORT	MDR Sample: Flight Report	1

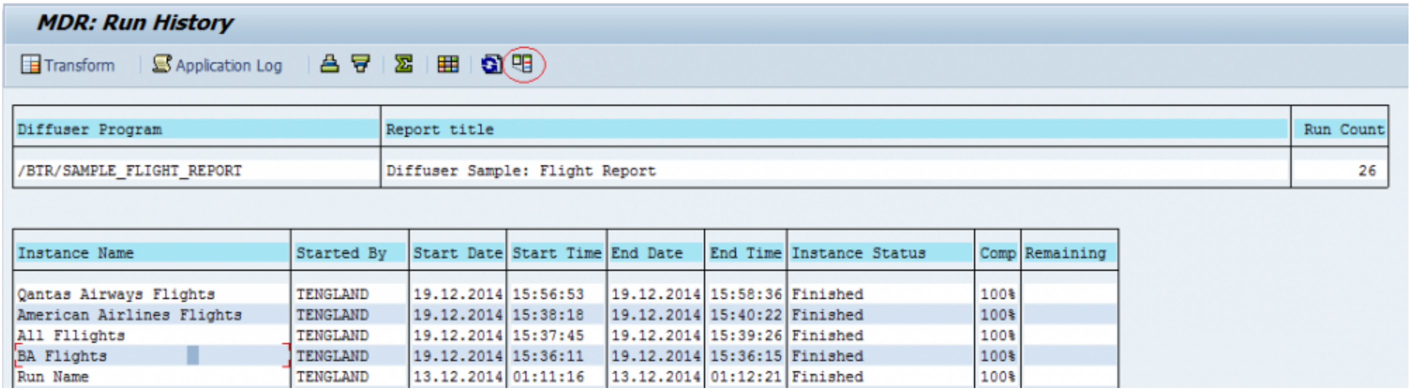
Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining	Active Job	-	+
test	CWAGNER	25.09.2013	13:05:16		00:00:00	In Process	14%	00:02:12	2	◀	▶

Interval	Low	High	Status	Results
1	00000001	00000047	Completed	1
2	00000048	00000098	Completed	1
3	00000099	00000173	Completed	1
4	00000174	00000221	Completed	1
5	00000223	00000273	Completed	1
6	00000274	00000332	Completed	1
7	00000333	00000379	Completed	1
8	00000380	00000426	Completed	1
9	00000427	00000473	Completed	1
10	00000474	00000520	Completed	1

- Expert Mode
- Results
- Intervals
- Variants
- App Servers
- Increase Jobs
- Decrease Jobs
- Stop
- Resume
- Delete
- Force Error

Expert Mode

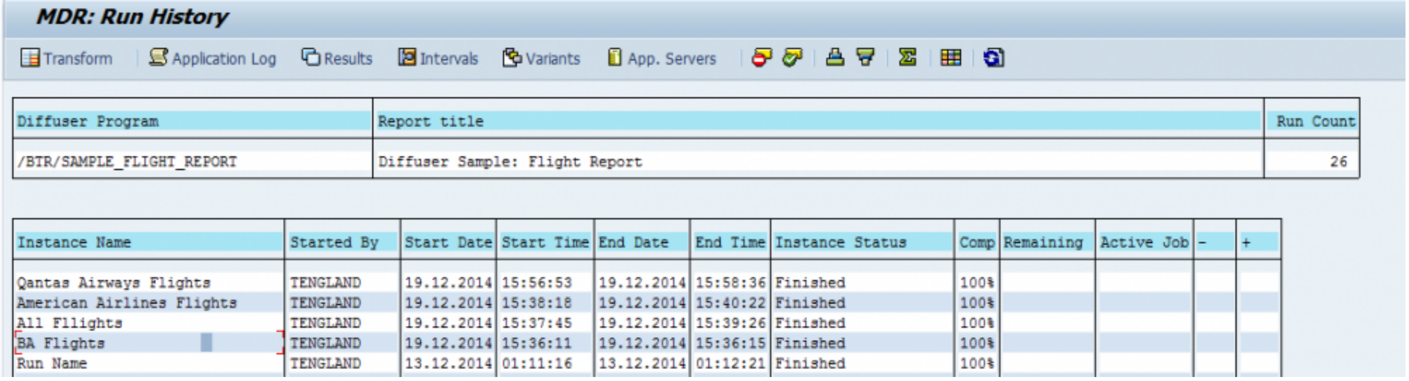
To access the expert mode click the expert mode button as below.



Diffuser Program	Report title	Run Count
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26

Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining
Qantas Airways Flights	TENGLAND	19.12.2014	15:56:53	19.12.2014	15:58:36	Finished	100%	
American Airlines Flights	TENGLAND	19.12.2014	15:38:18	19.12.2014	15:40:22	Finished	100%	
All Fllights	TENGLAND	19.12.2014	15:37:45	19.12.2014	15:39:26	Finished	100%	
BA Flights	TENGLAND	19.12.2014	15:36:11	19.12.2014	15:36:15	Finished	100%	
Run Name	TENGLAND	13.12.2014	01:11:16	13.12.2014	01:12:21	Finished	100%	

If authorized a number of other functions will be revealed.

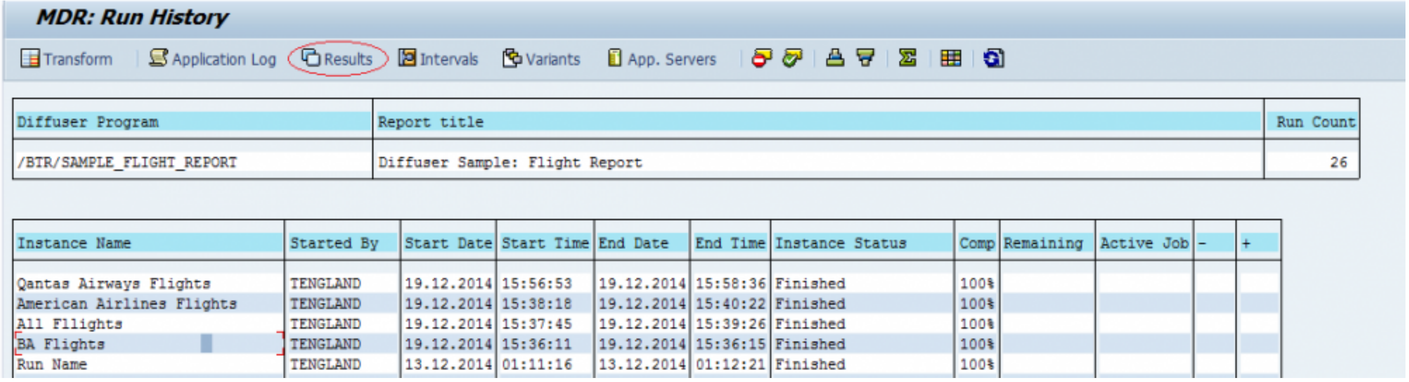


Diffuser Program	Report title	Run Count
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26

Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining	Active Job	-	+
Qantas Airways Flights	TENGLAND	19.12.2014	15:56:53	19.12.2014	15:58:36	Finished	100%				
American Airlines Flights	TENGLAND	19.12.2014	15:38:18	19.12.2014	15:40:22	Finished	100%				
All Flights	TENGLAND	19.12.2014	15:37:45	19.12.2014	15:39:26	Finished	100%				
BA Flights	TENGLAND	19.12.2014	15:36:11	19.12.2014	15:36:15	Finished	100%				
Run Name	TENGLAND	13.12.2014	01:11:16	13.12.2014	01:12:21	Finished	100%				

Results

To access the raw results stored against the instance click the results button as below.



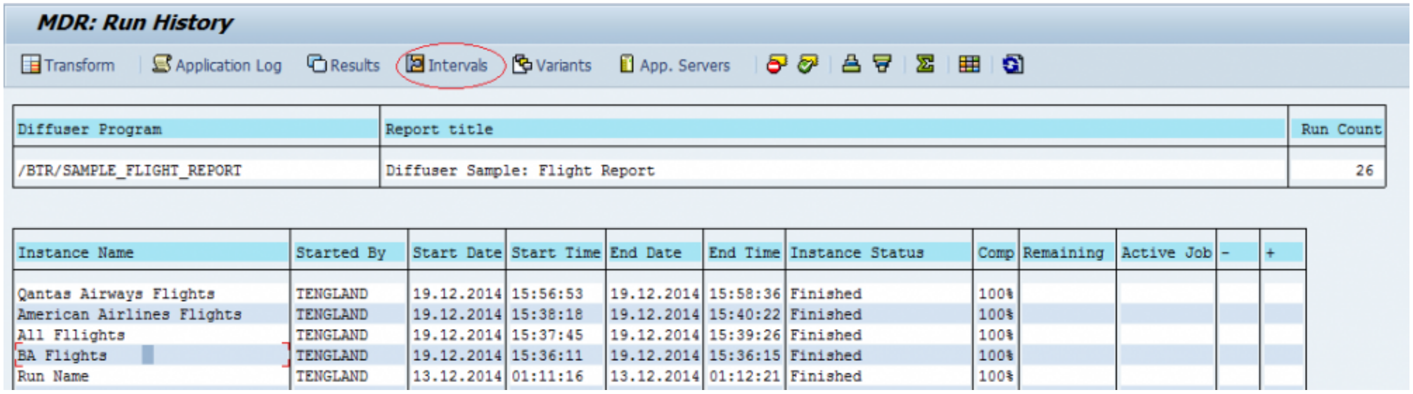
Diffuser Program	Report title	Run Count
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26

Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining	Active Job	-	+
Qantas Airways Flights	TENGLAND	19.12.2014	15:56:53	19.12.2014	15:58:36	Finished	100%				
American Airlines Flights	TENGLAND	19.12.2014	15:38:18	19.12.2014	15:40:22	Finished	100%				
All Fllights	TENGLAND	19.12.2014	15:37:45	19.12.2014	15:39:26	Finished	100%				
BA Flights	TENGLAND	19.12.2014	15:36:11	19.12.2014	15:36:15	Finished	100%				
Run Name	TENGLAND	13.12.2014	01:11:16	13.12.2014	01:12:21	Finished	100%				

You can also select an interval and view the raw results stored against each interval.

Intervals

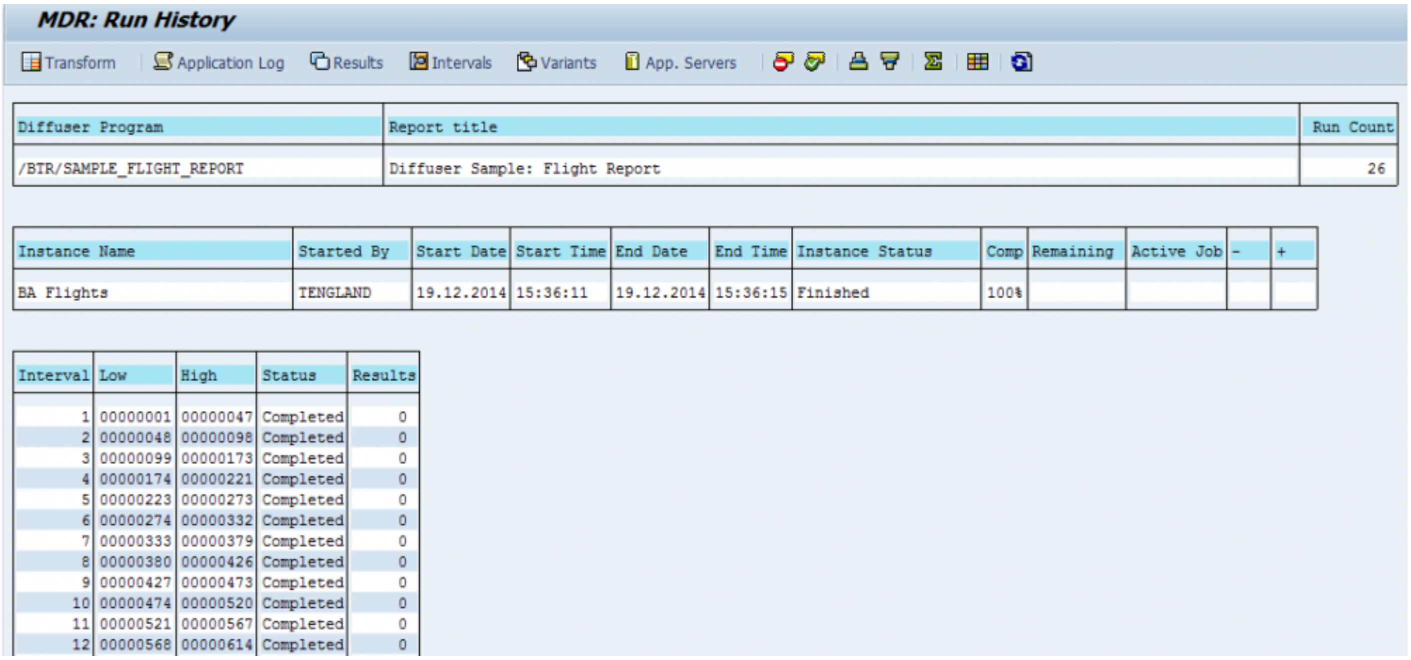
By drilling down on the program name the user will access the programs instance runs. Select an instance and click “Intervals” to display the intervals to that specific instance run.



Diffuser Program	Report title	Run Count
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26

Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining	Active Job	-	+
Qantas Airways Flights	TENGLAND	19.12.2014	15:56:53	19.12.2014	15:58:36	Finished	100%				
American Airlines Flights	TENGLAND	19.12.2014	15:38:18	19.12.2014	15:40:22	Finished	100%				
All Fllights	TENGLAND	19.12.2014	15:37:45	19.12.2014	15:39:26	Finished	100%				
BA Flights	TENGLAND	19.12.2014	15:36:11	19.12.2014	15:36:15	Finished	100%				
Run Name	TENGLAND	13.12.2014	01:11:16	13.12.2014	01:12:21	Finished	100%				

The details of each intervals are then displayed as below.



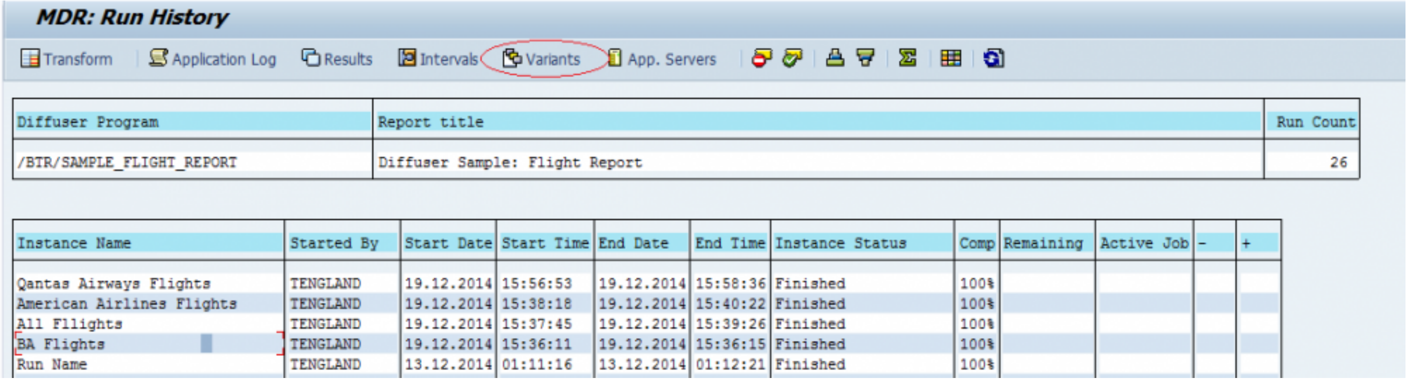
Diffuser Program	Report title	Run Count
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26

Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining	Active Job	-	+
BA Flights	TENGLAND	19.12.2014	15:36:11	19.12.2014	15:36:15	Finished	100%				

Interval	Low	High	Status	Results
1	00000001	00000047	Completed	0
2	00000048	00000098	Completed	0
3	00000099	00000173	Completed	0
4	00000174	00000221	Completed	0
5	00000223	00000273	Completed	0
6	00000274	00000332	Completed	0
7	00000333	00000379	Completed	0
8	00000380	00000426	Completed	0
9	00000427	00000473	Completed	0
10	00000474	00000520	Completed	0
11	00000521	00000567	Completed	0
12	00000568	00000614	Completed	0

Variants

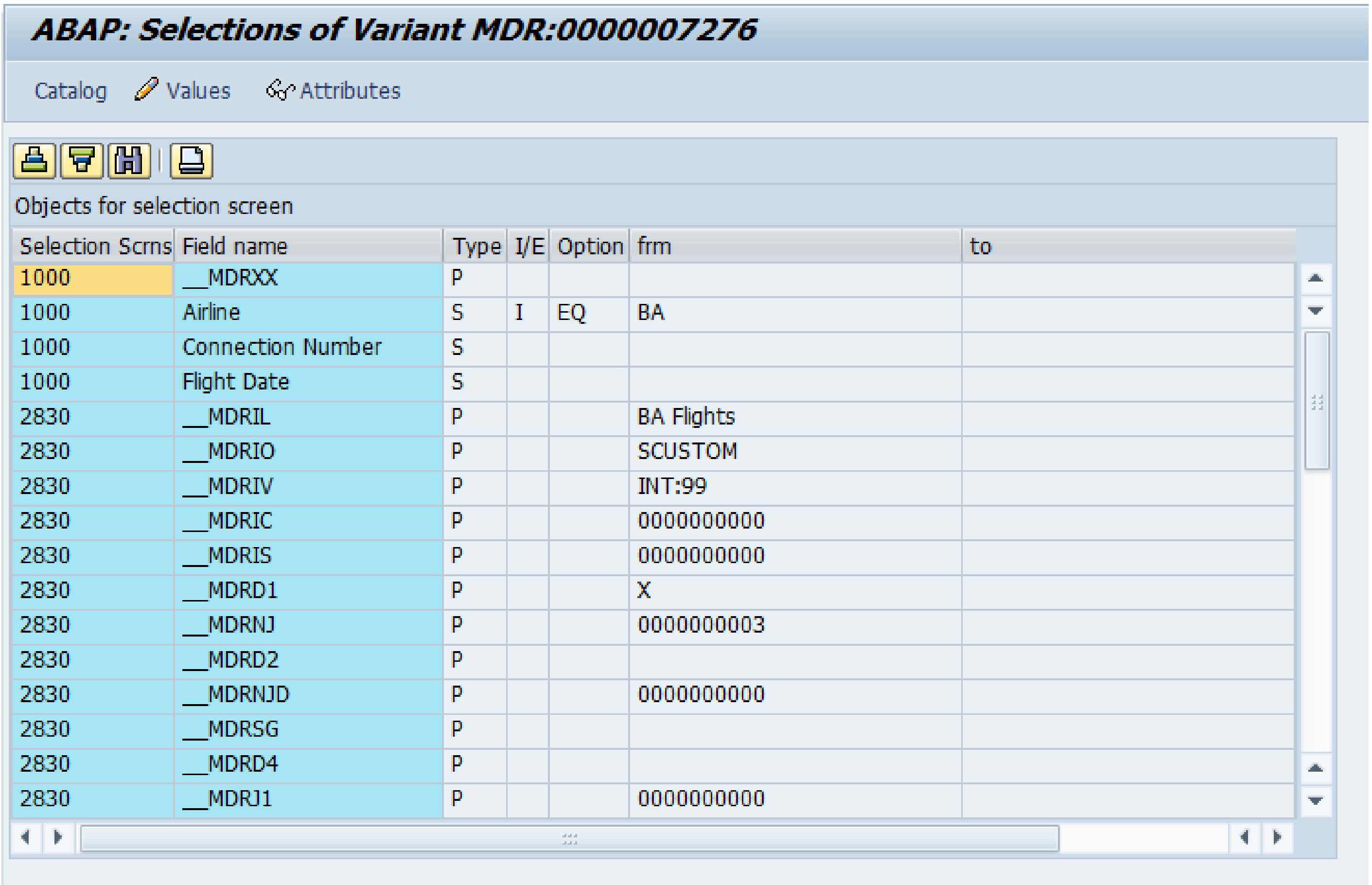
To access the raw results stored against the instance click the results button as below.



Diffuser Program	Report title	Run Count
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26

Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining	Active Job	-	+
Qantas Airways Flights	TENGLAND	19.12.2014	15:56:53	19.12.2014	15:58:36	Finished	100%				
American Airlines Flights	TENGLAND	19.12.2014	15:38:18	19.12.2014	15:40:22	Finished	100%				
All Fllights	TENGLAND	19.12.2014	15:37:45	19.12.2014	15:39:26	Finished	100%				
BA Flights	TENGLAND	19.12.2014	15:36:11	19.12.2014	15:36:15	Finished	100%				
Run Name	TENGLAND	13.12.2014	01:11:16	13.12.2014	01:12:21	Finished	100%				

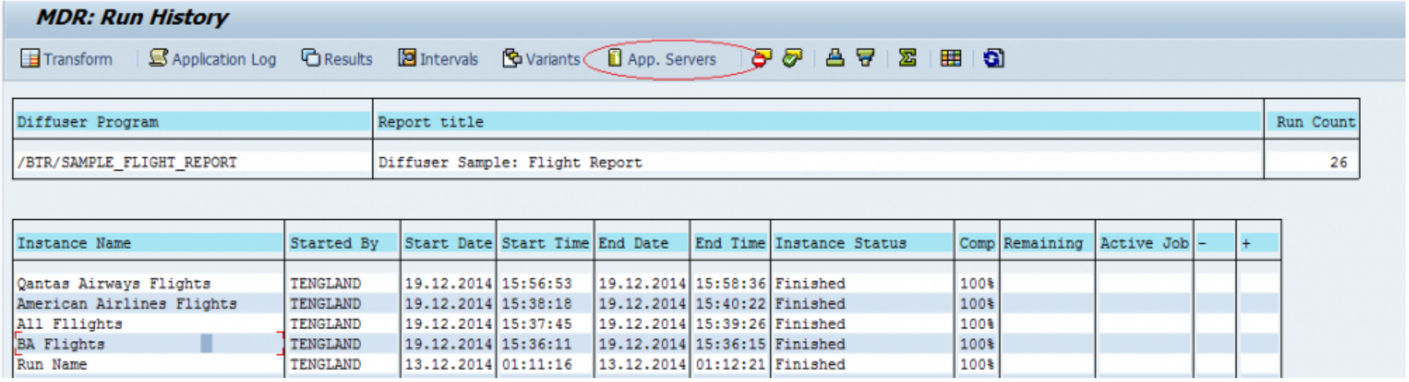
This enables the variant details entered on the selection screen to be viewed.


ABAP: Selections of Variant MDR:0000007276
Catalog
Values
Attributes
Objects for selection screen

Selection Scrns	Field name	Type	I/E	Option	frm	to
1000	__MDRXX	P				
1000	Airline	S	I	EQ	BA	
1000	Connection Number	S				
1000	Flight Date	S				
2830	__MDRIL	P			BA Flights	
2830	__MDRIO	P			SCUSTOM	
2830	__MDRIV	P			INT:99	
2830	__MDRIC	P			0000000000	
2830	__MDRIS	P			0000000000	
2830	__MDRD1	P			X	
2830	__MDRNJ	P			0000000003	
2830	__MDRD2	P				
2830	__MDRNJD	P			0000000000	
2830	__MDRSG	P				
2830	__MDRD4	P				
2830	__MDRJ1	P			0000000000	

App Servers

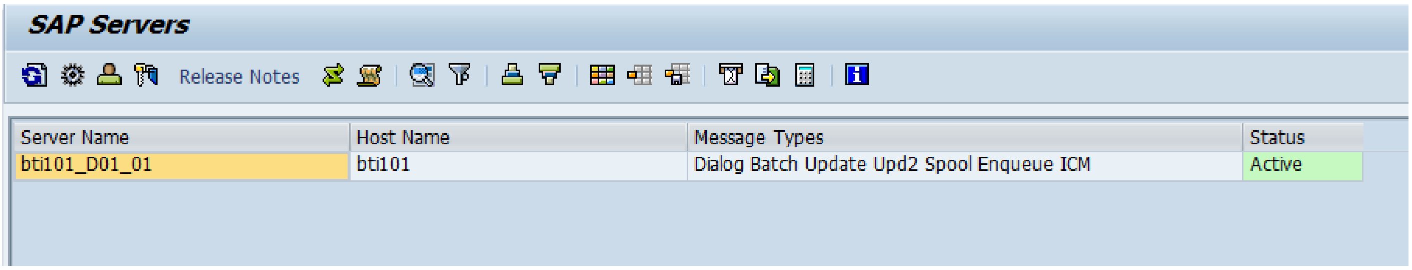
To view the application servers click the App Server button as below.



Diffuser Program	Report title	Run Count
/BTR/SAMPLE_FLIGHT_REPORT	Diffuser Sample: Flight Report	26

Instance Name	Started By	Start Date	Start Time	End Date	End Time	Instance Status	Comp	Remaining	Active Job	-	+
Qantas Airways Flights	TENGLAND	19.12.2014	15:56:53	19.12.2014	15:58:36	Finished	100%				
American Airlines Flights	TENGLAND	19.12.2014	15:38:18	19.12.2014	15:40:22	Finished	100%				
All Fllights	TENGLAND	19.12.2014	15:37:45	19.12.2014	15:39:26	Finished	100%				
BA Flights	TENGLAND	19.12.2014	15:36:11	19.12.2014	15:36:15	Finished	100%				
Run Name	TENGLAND	13.12.2014	01:11:16	13.12.2014	01:12:21	Finished	100%				

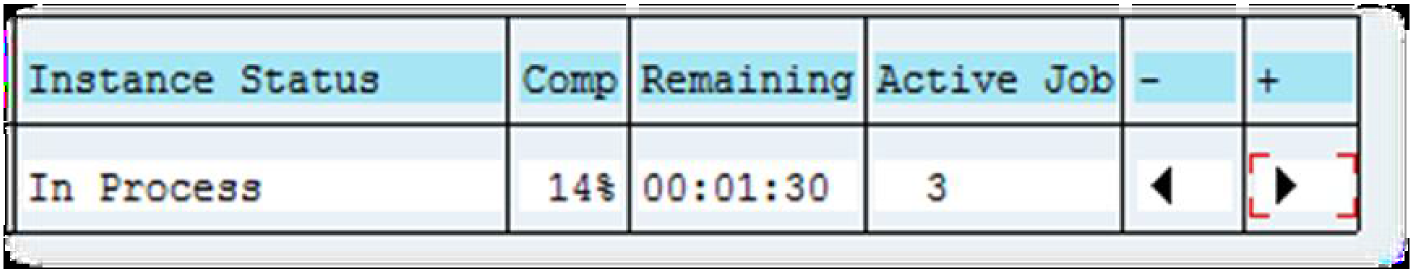
This then displays the available App Servers



Server Name	Host Name	Message Types	Status
bti101_D01_01	bti101	Dialog Batch Update Upd2 Spool Enqueue ICM	Active

Increase Jobs

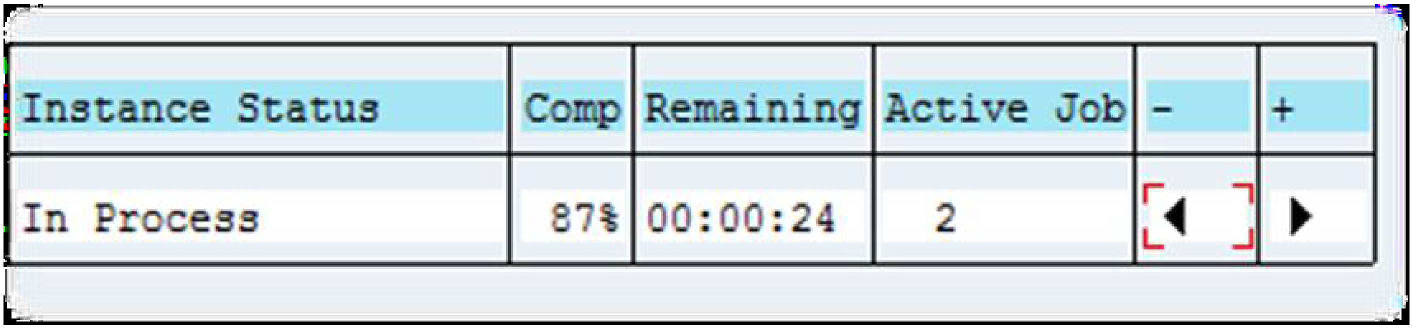
On the “Run History” report, the user can see historical instances of a program, as well as any currently executing program instances. The user can also see the number of active jobs for each program instance currently running. It is now possible to select an instance and then the “Increase Jobs” option, or alternately the right arrow icon against the instance. By selecting this option another job is scheduled immediately to assist in processing any unprocessed intervals. After a short period, and after selecting refresh the user will notice that the number of active jobs increase by 1, or by the number of times the increase jobs option is selected.



Instance Status	Comp	Remaining	Active Job	-	+
In Process	14%	00:01:30	3	◀	▶

Decrease Jobs

It is also possible to select an instance and decrease the number of jobs by selecting the “Decrease Jobs” option, or selecting the left arrow icon beside the instance. It is only possible to select this option for instances that are currently “In Progress”, and have more than one active job. MDR will prevent you decreasing the number of jobs to zero, if it is the users intention to stop the processing of the MDR report, then the user should select the “Stop” option. After a short period, and after selecting refresh the user will notice that the number of active jobs decreases by 1.



Instance Status	Comp	Remaining	Active Job	-	+
In Process	87%	00:00:24	2	◀	▶

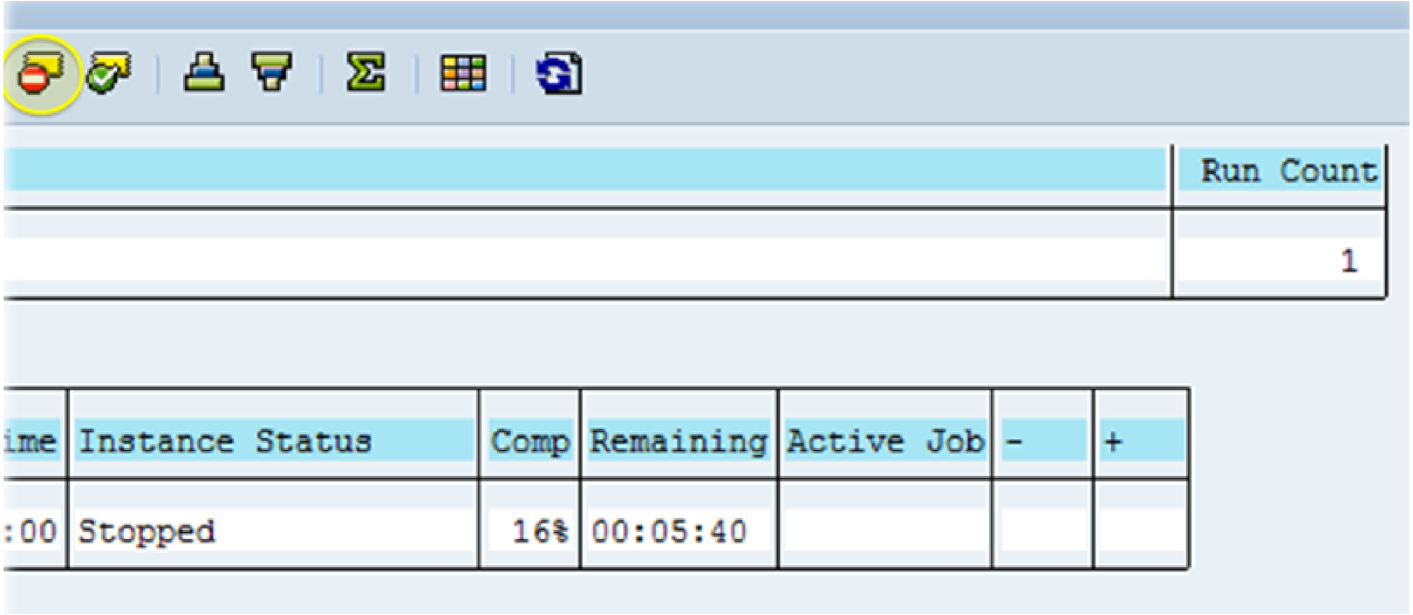
Stop

It is possible to Stop (or Pause) a program instance using the “Stop” option. By selecting this option after selecting a program instance, the MDR framework tells the currently executing jobs to no longer process any more intervals after it completes the processing of the current interval. The status of the Instance, and unprocessed intervals changes to “Stopped”. This is a powerful option that is used typically when an MDR program needs to be stopped temporarily due to the need to free up batch resources, or stopped permanently if the report run is no longer required.

The benefit MDR has over the traditional approach to executing reports is that the MDR program does not need to start over again, execution can continue from where it left off. The intervals that have already been processed do not need to be reprocessed unless of course it is deemed necessary by the user due to perhaps a substantial amount of time passing before the program is allowed to continue. It is only possible to stop an Instance that is currently in the “In Progress” status.

When the instance is paused, the MDR framework will not immediately stop all jobs that are currently running. It will instead prevent any new intervals from being started.

The more intervals there are the more control over the execution of the instance an administrator will have.

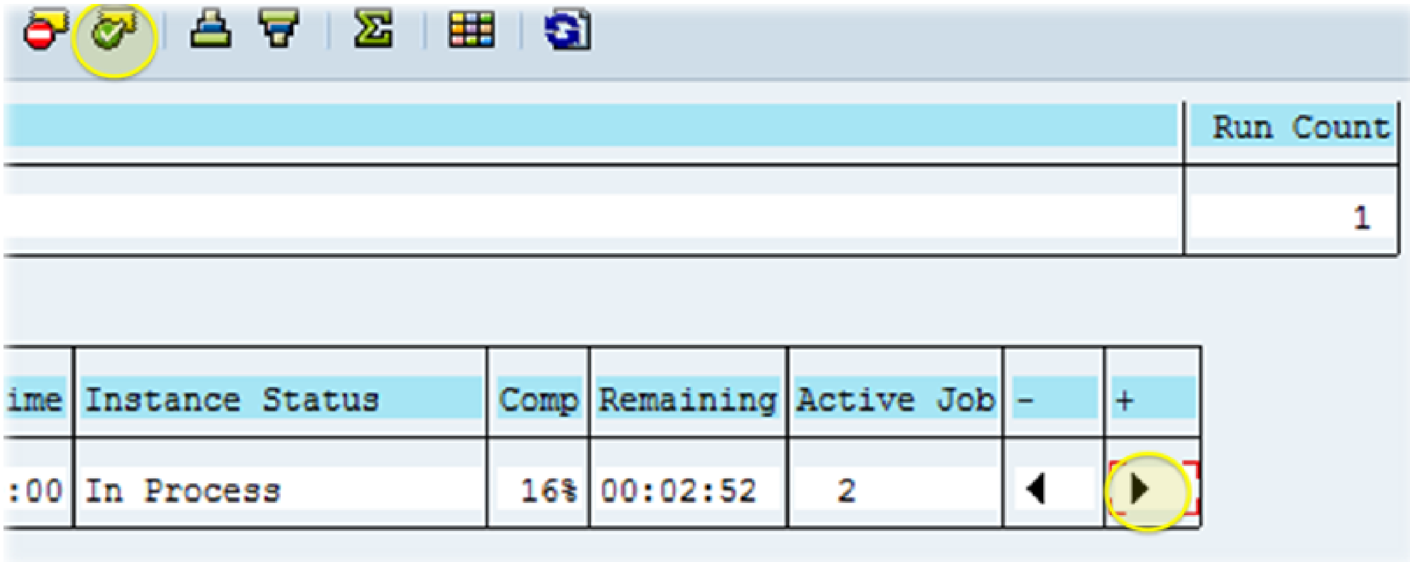


	Run Count
	1

ime	Instance Status	Comp	Remaining	Active Job	-	+
:00	Stopped	16%	00:05:40			

Resume

The “Resume” option allows the selected program instance to continue from the point it was stopped or paused. This option uses the Technical Settings of the original program instance to reschedule the report. By resuming an instance it does not reprocess any intervals that have a status of “Completed”, it changes the status of a “Paused” interval to “Ready”. The restart option can only be selected for instances with the status “Paused” or “Error”.



Run Count
1

ime	Instance Status	Comp	Remaining	Active Job	-	+
:00	In Process	16%	00:02:52	2	◀	▶

Delete

It is possible to delete a program instance by selecting the instance and then the “Delete” option, this in turn deletes all the intervals and results belonging to the program instance. After the delete option is selected the user is faced with a confirmation window to ensure the deletion was intentional. This option is particularly useful in a testing environment and with instances that have errored. It is only possible to delete instances with the status “Error” or “Completed”. The system will not allow an instance “In Progress” to be deleted due to possible data inconsistencies.

Force Error

By selecting an Instance, right-clicking and then the “Force error” option, the status of the program instance is changed to “Error”. This allows instances that have technically completed successfully to be changed to Error. This is basically an override function. It is only possible to set an instance to “Error” if there are no active jobs executing the instance.

Scheduling MDR Programs

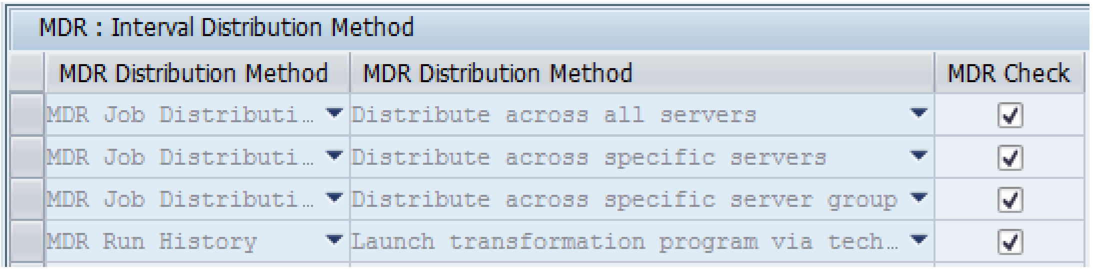
An MDR program can be scheduled just like any other background program. Typically this is done using the standard transaction SM36. The program variants can also be saved as per normal.

MDR in most cases, however, does require another program to be scheduled for it to operate efficiently in a production environment. The program function is to regenerate the Interval Variant. The purpose of regenerating an Interval Variant is such that as the master or transactional data grows, the intervals can be recalculated to ensure that each interval is evenly spread. This then ensures the MDR program is processed as efficiently as possible.

The program /BTR/MDR_INTERVAL_REGENERATION is used for this purpose. This job should typically be scheduled nightly at the beginning of the batch window, and can be executed for individual Interval Objects, individual Interval Variants, or for all Interval Variants by adjusting the parameters on the selection screen. For the Interval Regeneration to operate, you will need to configure the table /BTR/INTVALVARC. Here you define an Interval Object, Interval Variant and the refresh age. The refresh age defines how frequently the Interval Variant is refreshed. For example if for Object SCUSTOMID, Variant SAMPLE, if the refresh age is 7, the interval variant will only be regenerated every 7 days, even if the regeneration job is scheduled nightly. This functionality allows you to avoid scheduling individual regeneration jobs in different reoccurring cycles. You can override the refresh age functionality by selecting the “Force regeneration” check-box.

Job Distribution

MDR threads can now be run on a specific server/servers or a server group. This functionality can be activated on table /BTR/MDR_C via transaction code SM30 as shown below.



MDR : Interval Distribution Method

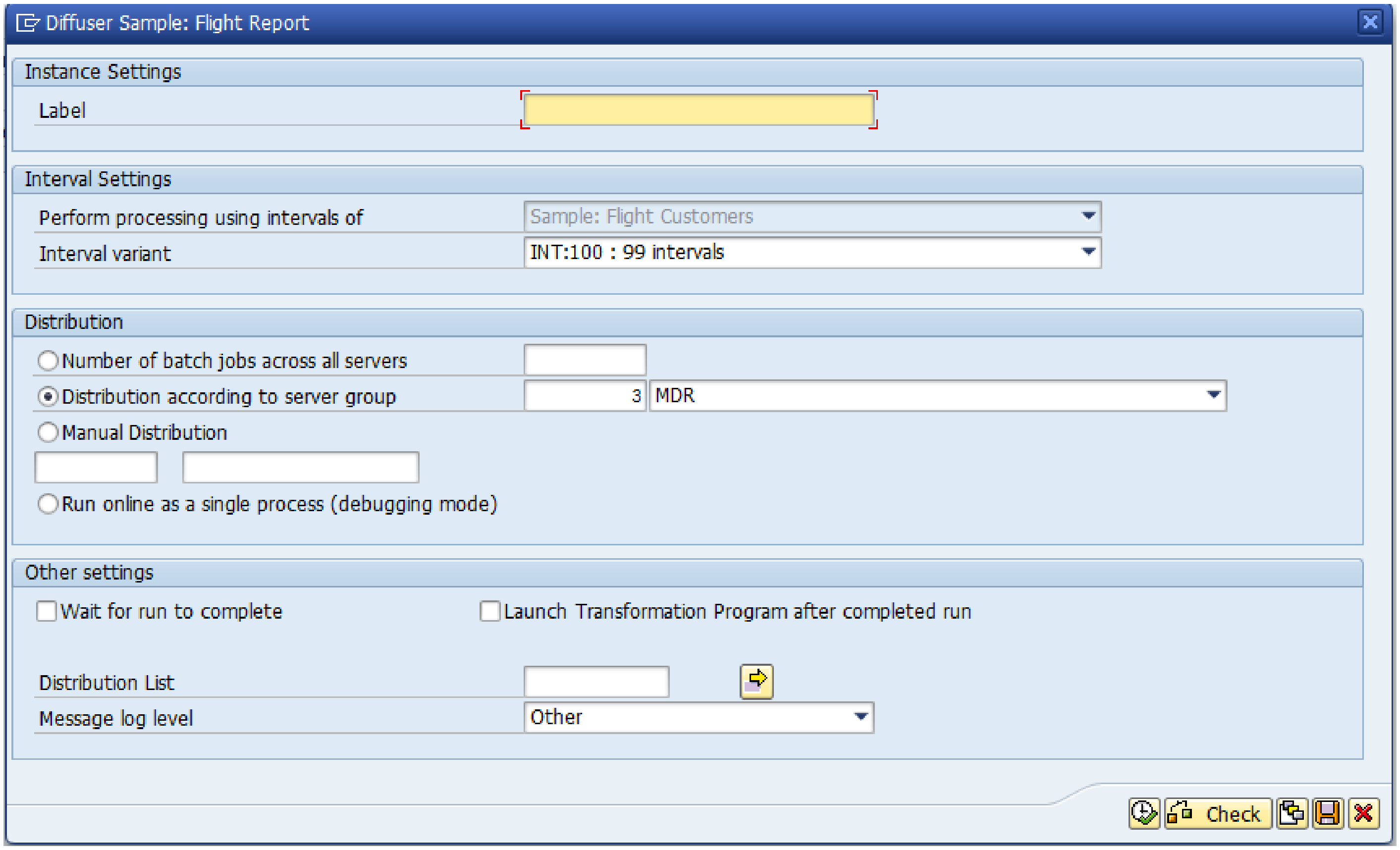
MDR Distribution Method	MDR Distribution Method	MDR Check
MDR Job Distributi…	Distribute across all servers	☑
MDR Job Distributi…	Distribute across specific servers	☑
MDR Job Distributi…	Distribute across specific server group	☑
MDR Run History	Launch transformation program via tech…	☑

- Server Group Distribution
- Manual Distribution

Server Group Distribution

If option “Distribute across specific server group” in config table /BTR/MDR_C is checked an additional option will appear on the technical settings screen.

Specify the number of jobs and predefined server group in the corresponding input fields (see below) and run your MDR program as usual.



Label

Perform processing using intervals of: Sample: Flight Customers
Interval variant: INT:100 : 99 intervals

○ Number of batch jobs across all servers
● Distribution according to server group: 3 MDR
○ Manual Distribution
○ Run online as a single process (debugging mode)

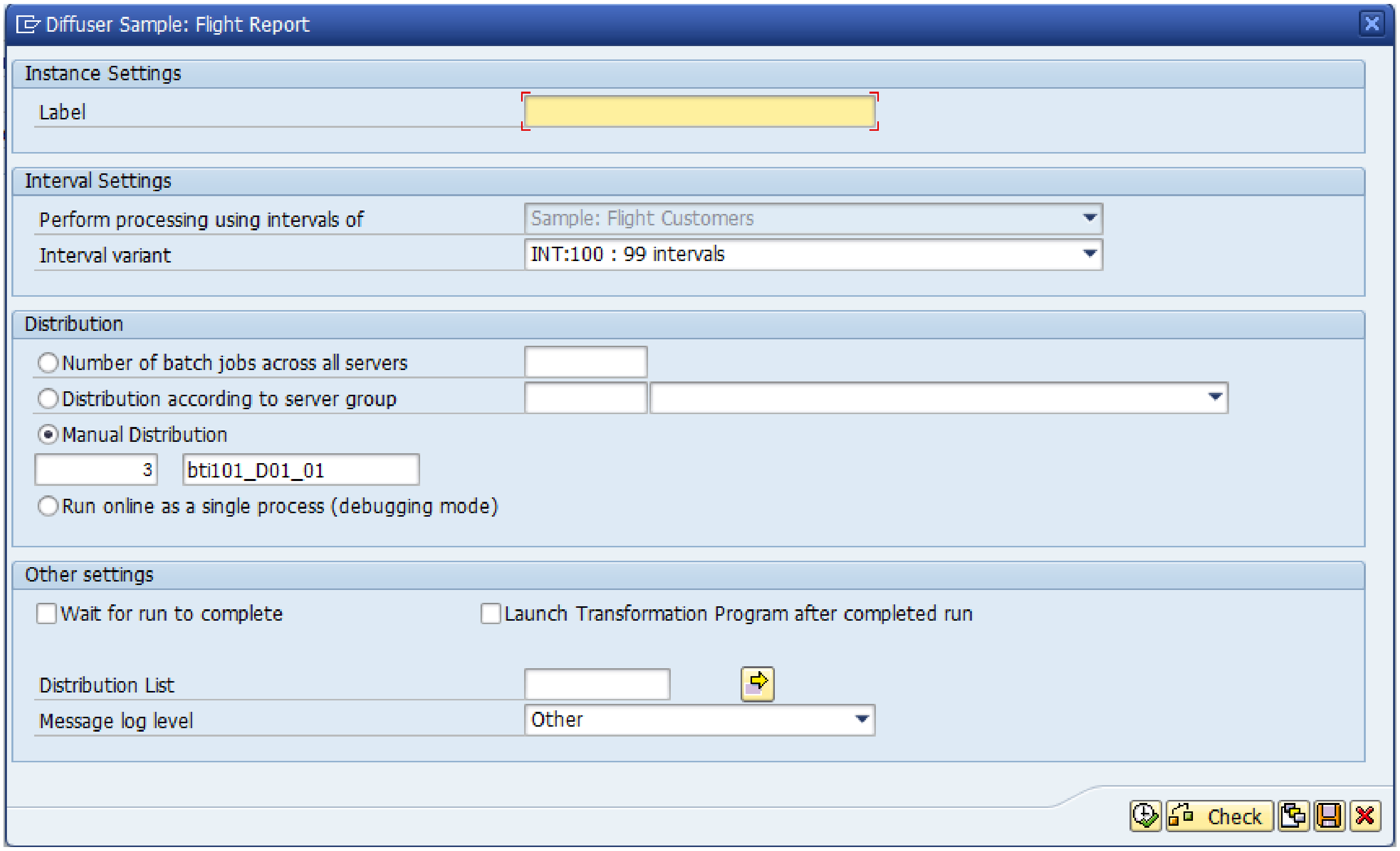
☐ Wait for run to complete
☐ Launch Transformation Program after completed run

Distribution List
Message log level: Other

Manual Distribution

If option “Distribute across specific servers” in config table /BTR/MDR_C is checked an additional option will appear on the technical settings screen. The MDR Framework determines how many application servers are available (up to 10 servers) and displays the corresponding number of input field rows (in the example below only one).

Specify the number of jobs as well as application server in the corresponding input fields by means of the search help and run your MDR program as usual (see below).



Diffuser Sample: Flight Report

Instance Settings

Label:

Interval Settings

Perform processing using intervals of: Sample: Flight Customers

Interval variant: INT:100 : 99 intervals

Distribution

○ Number of batch jobs across all servers

○ Distribution according to server group

● Manual Distribution

3 bti101_D01_01

○ Run online as a single process (debugging mode)

Other settings

☐ Wait for run to complete ☐ Launch Transformation Program after completed run

Distribution List:

Message log level: Other

Maintenance of MDR results

The MDR framework provides program /BTR/MDR_INSTANCE_DELETE, this program allows you to completely delete the stored result sets for one or more MDR instance runs and the result sets of all their corresponding intervals. Alternatively, you can choose to delete only the result sets of the intervals of an instance run. If you don't need results at interval level you can choose the second options since the result sets at instance level are an aggregation of the intervals.

To run the program /BTR/MDR_INSTANCE_DELETE you need to have added the programs you want to delete the data for in the table /BTR/PROGARCH using SM30. This allows you to restrict by program and numbers of days of instance runs that are stored, the configuration table makes it difficult for anyone to accidentally delete results. For example a sales team might want to see their historical data for the last 30 days and you can add 30 as the limit, then when running the /BTR/MDR_INSTANCE_DELETE it will only delete data for that program that is older than the 30 day limit.

It is recommended to schedule this program to do regular cleaning in your system to avoid accumulation of unnecessary historical data.

If you only need to delete one run the “Delete” functionality for individual instances mentioned in Administering MDR Programs can be useful.

Performance considerations

Limits to the number of parallel processes that can be used

The limitation of MDR is purely based upon the available hardware within the SAP landscape. We have seen, on larger SAP customers, 20+ application servers with over 700 background processes. In this configuration, it has been possible to run an MDR program with as much as 300 parallel processes without causing contention upon the database and seeing an effect on scalability.

Contention with other batch jobs and dialog processes

It is important to know what other jobs are running in the batch schedule so that the SAP system is not overloaded. This goes for both MDR programs and their child processes as well as other batch jobs or user / dialog activity. This is the key benefit of a batch scheduler to ensure that jobs are orchestrated together. MDR arms you with the necessary control to run an ABAP program in a short burst of activity, so you can get it out of the way by maximizing the available hardware and system configuration.

MDR and your SAP batch scheduler

MDR integrates seamlessly with any job scheduler e.g. Automic / UC4, Tivoli / Maestro, Redwood. The scheduler itself triggers the same MDR ABAP program with the same variant. Within this variant, we set a parameter flag (see Wait for run to complete in section 1.2) that ensures that the “parent” job (triggered by the job scheduler) waits while the child parallel jobs finish. This means there are no changes from a batch scheduler perspective, as it completes the job as per normal but just N times faster. One of the key strengths of an MDR report / program is that it looks like any other ABAP report / program. The number of “intervals” and the number of “parallel jobs” to be started must be specified, however, these can be defaulted into the program itself or into the variant. The batch scheduler will still continue scheduling the program to run with the ABAP program name and the variant name, but the parameters within that variant will be enhanced only.

Security Enhancements

As mentioned above, the Defaults for Technical Settings section offers two options for functionality restrictions, “Lock Technical Settings” and “Lock Expert Mode”. These work at a program level and once set they will apply for every user.

However, the MDR Framework also provides an enhancement spot to allow developers to apply customer specific authority checks. This can be used to restrict technical as well as administrative settings at user and at program level. For more information refer to the Mass Data Runtime – Developers Guide Authority Checks