



Extracting Data from I3 PW
Functional Spec

Document Date: April 17th

Created by: Eyal Markovich.

Table of Contents

1. Introduction	3
Purpose	3
The solution	3
Solution Constraints.....	3
2. Solution Specification	4
Statistics to extract	4
Two modes of operation	5
Procedure <i>I3ExtractData</i>	5
Procedure <i>I3ExtractDataExt</i>	6
Creating the External tables	6
I3InstanceStats	7
I3DatabaseStats	7
I3ProcessorStats	8
I3PhysicalDisksStats	8
3. Example of Usage.....	9
4. Adding new counters.....	10

1. Introduction

Purpose

...

The solution

The solution will be based on SQL stored procedures. Total APM will implement the solution as a set of stored procedures located in the I3 PW database.

There will be two modes of activation. The first mode will allow to specify a time range for which the required counters will be extracted from the PW. The second mode of operation is an incremental mode that allows new data to be added to the external tables.

A sample SQL Agent job that will use the incremental load will be supplied with a daily scheduler.

Solution Constraints

The following list describes the agreed constraints and requirements:

- The external tables will not be part of I3 and such integration will not be supplied. Inform/Foresight or any other I3 modules will not access these tables.
- As the archive database (e.g. the database where the external tables reside) can be remote to the PW instance, a linked server is required between I3 PW and the external archive database.

2. Solution Specification

The following section describes the list of statistics that will be extracted, the modes of operation and the structure of the archive tables.

Statistics to extract

The list below summarizes the statistics that will be extracted from I3:

- AccessMethods
 - Page Splits/sec [avg]
 - Table Lock Escalations/Sec [avg]
- Buffer Manager
 - Buffer CacheHit Ratio [avg]
- General Statistics
 - Logins/sec [avg]
 - User Connections [avg]
- PhysicalDisk
 - Average Disk Queue Length
 - Average Disk sec/Read
 - Average Disk sec/Write
 - Average Writes Sec
 - Average Reads Sec
- Process
 - % Processor Time [avg,max]
 - Non SQL Server % Processor Time [avg,max]
- Processor
 - % Processor Time [avg,max]
- SQL Statistics
 - Batch Requests/sec [avg,max]
 - SQL Compilations/sec [avg]
 - SQL Re-Compilations/sec [avg]
- Load
 - Transactions/sec [avg,max]
- Activity
 - Totals Sessions [sum]
 - Executions [sum]
 - cpu time [sum]
 - physical I/O [sum]

Two modes of operation

The solution will consist of two modes of operation. The first mode will allow to specify a time range for which the required counters will be extracted from the PW. The second mode of operation is an incremental mode that allows new data to be added to the external tables.

The table below lists the two stored procedure one for each mode:

Stored procedure /Mode	Description
I3ExtractData	Incremental extraction to add additional new data to the extracted table
I3ExtractDataExt	specify a time range for which counters will be extracted

Procedure *I3ExtractData*

The *I3ExtractData* stored procedure will load new data (data that has been added to I3 since the last extraction) to the external tables. If the external tables do not exist an error will be returned.

Parameters:

Name	Description	Type	Default value
@p_path	Full path to the archive tables (including the linked server if needed)	String	NULL
@p_instance_name	Instance Name or 'All' for all instances	String	All
@p_db_level	Load database level for the statistics and Activity?	1 – Yes; 0 – No	0

Procedure *I3ExtractDataExt*

The *I3ExtractDataExt* stored procedure will load statistics to the external tables based on the time range provided.

Parameters:

Name	Description	Type	Default value
@p_path	Full path to the archive tables (including the linked server if needed)	String	NULL
@p_instance_name	Instance Name or 'All' for all instances	String	All
@p_db_level	Load database level for the statistics and Activity?	1 – Yes; 0 – No	0
@p_from_date	Starting date to copy the data from	Date	Yesterday
@p_to_date	Starting date to copy the data from	Date	Now

Ignoring Instances

A table called *I3IgnoreInstances* will be created in the PW database. This table will allow to exclude several instances while using the 'ALL' mode.

To ignore an instance run the command:

Insert into I3IgnoreInstances (InstanceName) Values ('XXX').

Creating the External tables

In order to start using the solution, you will need to create the archive tables. The statistics extracted from the PW will be loaded to the archive tables.

A script named *I3SetupExternalSchema.sql* will be supplied in order to create the required schema. The script will need to run in the designated database for the archive tables.

The following table lists the external tables:

Table name	Level	Description
I3InstanceStats	Instance	Hold statistics for the Instance level. For all database counters it will also load a total value
I3DatabaseStats	Database	Hold statistics for database level
I3ProcessorStats	Processor	Hold statistics for processor level
I3PhysicalDisksStats	Disk	Hold statistics for disk level

I3InstanceStats

The I3InstanceStats table holds data for instance level. The table below documents the structure of the table:

Column Name	Type
InstanceName	varchar(200) not null,
SampleDate	datetime not null,
PageSplitsSec_AVG	float (53)
TableLockEscalationsSec_AVG	float (53)
BufferCacheHitRatio_AVG	float (53)
LoginsSec_AVG	float (53)
UserConnections_AVG	float (53)
SQLServerProcessTimePercentage_AVG	float (53)
SQLServerProcessTimePercentage_MAX	float (53)
NonSQLServerProcessTimePercentage_AVG	float (53)
NonSQLServerProcessTimePercentage_MAX	float (53)
BatchRequestsSec_Avg	float (53)
BatchRequestsSec_MAX	float (53)
CompliationsSec_AVG	float (53)
ReCompliationsSec_AVG	float (53)
TransactionsSec_AVG	float (53)
TransactionsSec_MAX	float (53)
TotalSessions_SUM	bigint,
TotalExecutions_SUM	bigint,
CPUTime_SUM	float (53)
PhysicalIO_SUM	Bigint
DiskQueueLength_AVG	float (53)
DiskSecReads_AVG	float (53)
DiskSecWrites_AVG	float (53)
DiskReadsSec_AVG	float (53)
DiskWritesSec_AVG	float (53)
DiskTransfersSec_AVG	float (53)

I3DatabaseStats

The I3DatabaseStats table holds data for the database level. The table below documents the structure of this table:

Column Name	Type
InstanceName	varchar(200)
SampleDate	datetime
DatabaseName	varchar(200)
TransactionsSec_AVG	float (53)
TransactionsSec_MAX	float (53)
TotalSessions_SUM	bigint
TotalExecutions_SUM	bigint
CPUTime_SUM	float (53)
PhysicalIO_SUM	bigint

I3ProcessorStats

The I3processorStats table holds statistics for each processor level. The table below documents the structure of the table:

Column Name	Type
InstanceName	varchar(200)
SampleDate	datetime
CPUNumber	int
ProcessorTimePercentage_AVG	float (53)
ProcessorTimePercentage_MAX	float (53)

I3PhysicalDisksStats

The table holds statistics for each disk. The table below documents the structure of the table:

Column Name	Type
InstanceName	varchar(200)
SampleDate	datetime
DiskName	varchar(200)
DiskQueueLength_AVG	float (53)
DiskSecReads_AVG	float (53)
DiskSecWrites_AVG	float (53)

3. Example of Usage

Examples

The following are some examples of using the stored procedures:

Extracting 3 months of data:

```
declare @d as datetime
set @d=dateadd(month,-3,getdate())
exec I3ExtractDataExt 'dblink.db', 'ALL', 0, @d,NULL
```

Adding last day activity (assuming the previous load ran yesterday):

```
exec I3ExtractData 'dblink.db', 'ALL',0
```

Extracting one instance data:

```
exec I3ExtractData 'dblink.db', 'instance1',0
```

Adding database level :

```
exec I3ExtractData 'dblink.db', 'ALL',1
```

Extracting Large Set of Data

Running the extract stored procedure on a long timeframe can result with a large set of rows that need to be inserted to the archive tables. In some cases this can take a long time or fail on log operation. To overcome this problem, consider running the stored procedure with a smaller date range and/or one instance at a time.

For example,

To extract two years of data for all instances run the commands:

```
declare @d as datetime
set @d=dateadd(month,-24,getdate())
exec I3ExtractDataExt 'dblink.db', 'ALL', 0, @d,NULL
```

However if this fails, run the following commands that will perform the same results

```
declare @d as datetime
declare @d1 as datetime
set @d=dateadd(month,-24,getdate())
set @d1=dateadd(month,1,@d)

while (@d<getdate())
begin
    exec I3ExtractDataExt '[e-markovich-lap\sqlxpress].archive', 'ALL', 0, @d,@d1
    set @d=@d1
    set @d1=dateadd(month,1,@d)
end
```

4. Adding new counters

The following section documents the steps required in order to add additional counters to the extracted table. This section can serve as a guide if in the future additional counters would be required to be added:

Adding a new counter requires the following changes (we will assume an instance counter):

1. Change the archive *I3InstancesStats* table DDL. This is done in *I3SetupExternalSchema.sql*.
2. Make the same change to the *#I3InstancesStats* temporary table.
3. Find the appropriate statement that update the *#I3InstancesStat* that is based on the I3 table that holds the counter. For example, if the counter is from “Access method” group (the I3 table is *PW_SQAM_ACCESS_METHOD_STATS_T*) the first update method should be modified.
4. Modify the *Update* statement with the info of the new counter.
5. Modify the dynamic Insert statement from the temporary *I3InstancesStats* table to the *I3InstancesStats* archive table to include the new counter.