**System Health Reporting Instructions**

**System Health Session Retention**

By default the System Health Session captures 4 files of 5MB each - <http://msdn.microsoft.com/en-us/library/ff877955.aspx>

Those files could rollover if there are significant issues. I would recommend changing that to a higher number of files or larger size per file. In order to do this on the instances:

In the script below, we are increasing it to 5 files of 25MB each.

ALTER EVENT SESSION [system\_health] ON SERVER STATE = STOP

go

ALTER EVENT SESSION [system\_health] ON SERVER

DROP TARGET package0.event\_file

ALTER EVENT SESSION [system\_health] ON SERVER

ADD TARGET package0.event\_file

(SET filename=N'system\_health.xel'

,max\_file\_size=(25),

max\_rollover\_files=(5))

GO

ALTER EVENT SESSION [system\_health] ON SERVER STATE = START

go

**System Health Session Import Procedure**

1. **Create Import DB and Schema:**

* Run the file CreateSystemhealthDBAndSchema.sql which creates a database XEvents\_ImportSystemHealth and the procedures required.
* On subsequent imports, tables are dropped and recreated
* These stored procedures can be optimized further to improve import performance

1. **Import the System Health Sessions into a Database**

Note: The Import procedure is not totally optimized, so it can be further optimized.

**Via TSQL**

* + **Process a SQL instance:** Running the procedure below without any input parameters will query the current instance for the system health XEvent session file location and import the files into the current database. Run this if you have the database deployed on the instance from which you want to analyze the Xevent files.

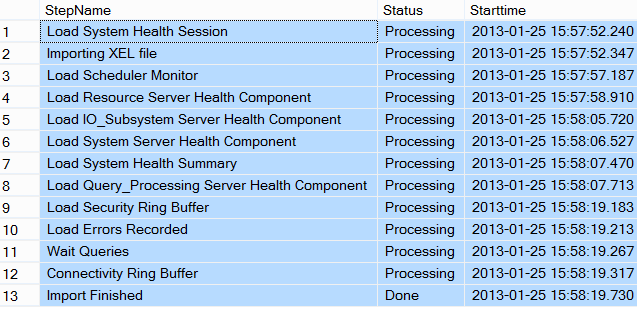
Exec spLoadSystemHealthSession

* + **Import System Health files from UNC:** If you have received the files from another instance and have the files stored on any UNC path, use the @path\_to\_health\_session input parameters to pull the files from a UNC location. **Note** that in this case you will have to specify the UNC name and the UTC offset in order to have the datetime data imported using the appropriate time zone. If you do not do this, the time is stored in GMT within the XEL files, and will be imported as such.

exec spLoadSystemHealthSession @path\_to\_health\_session='D:\XELFiles\system\_health\*.xel',@UTDDateDiff=-6

* + **Import Status:** If you want to check the status of the import while it is going

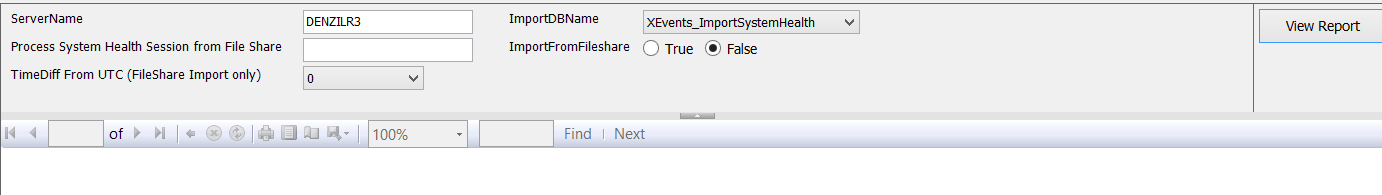
Select \* from tbl\_ImportStatus



**Via Report**

* + You can also import it via a report – ImportSessionStatus.rdl after deploying the reports. Again in the report below, you have to choose the Server and DatabaseName that has the schema that was created above, and you will also need to specify if the import is from the server itself ( Default) or the import is from a fileshare ( via the Import from FileShare Radio button) and indicate the file share location.

**Note:** Import from a file share via SSRS reports may require Kerberos if it is a double hop.

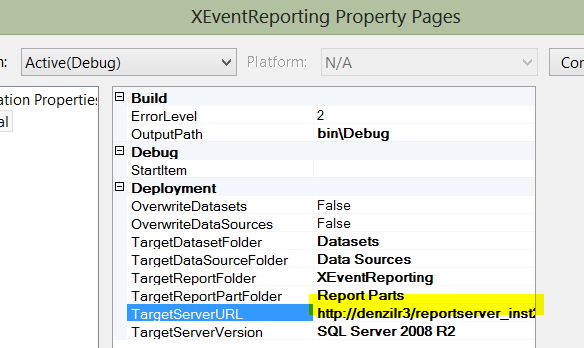


1. **Imported relational tables:**

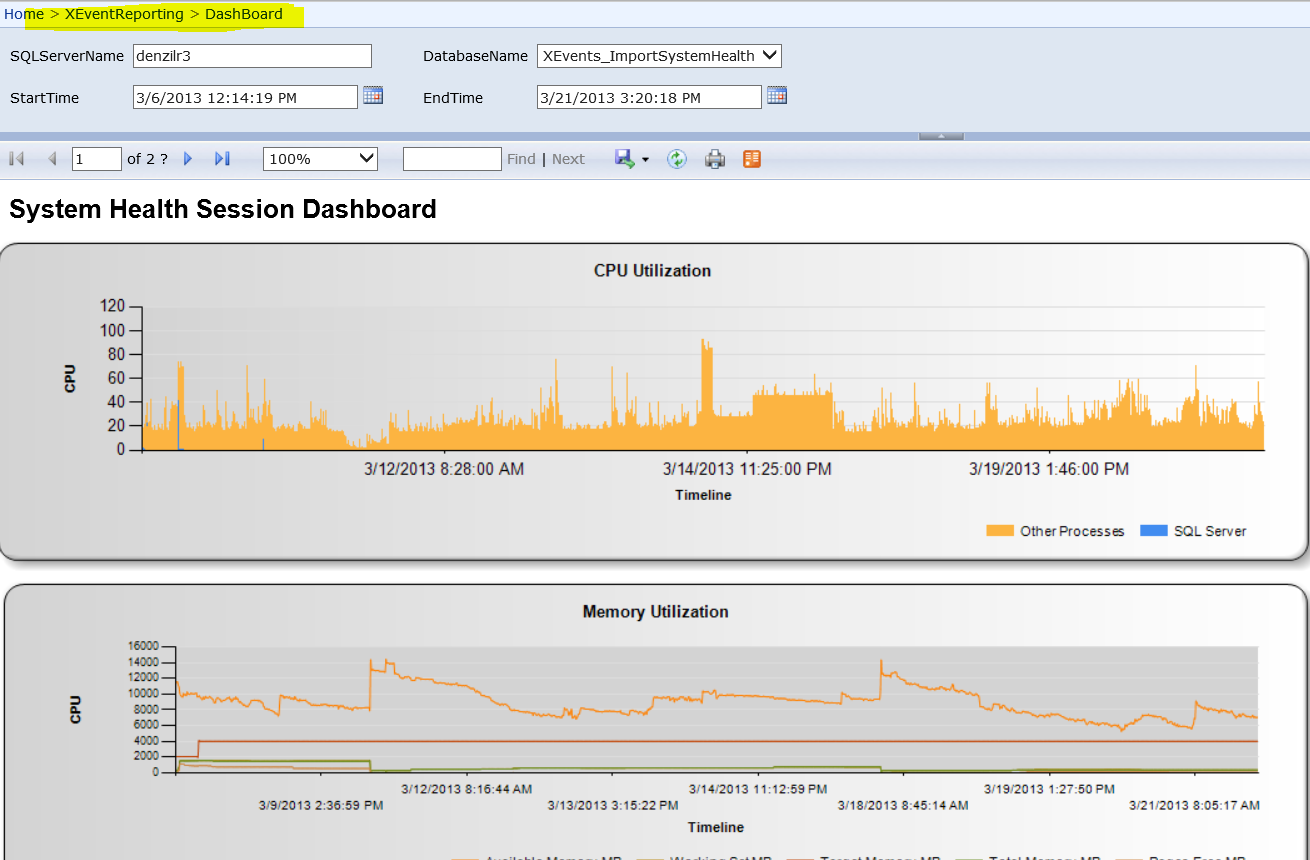
|  |  |
| --- | --- |
| **Name** | **Description** |
| tbl\_ImportStatus | Import Stats , each step Start time |
| tbl\_scheduler\_monitor | Pulls out details from RING\_BUFFER\_SCHEDULER\_MONITOR |
| tbl\_Resource | Details from sp\_server\_diagnostics RESOURCE component, mainly Memory info |
| tbl\_IO\_SUBSYSTEM | Details from sp\_server\_diagnostics IO component |
| tbl\_SYSTEM | Details from sp\_server\_diagnostics System component |
| tbl\_Summary | Summary States for sp\_server\_diagnostic\_component\_result individual components |
| tbl\_OS\_WAIT\_STATS\_byDuration | Top 10 Waits since Server Restart ( not deltas) |
| tbl\_QUERY\_PROCESSING | Details from sp\_server\_diagnostics System component |
| tbl\_BlockingXeOutput | Details of any significant blocking > 30 seconds |
| tbl\_security\_ring\_buffer | Security Ring buffer Details |
| tbl\_errors | Details from RING\_BUFFER\_EXCEPTION |
| tbl\_waitqueries | Any queries with significant waits > 15 seconds or 30 seconds depending on wait\_type |
| tbl\_connectivity\_ring\_buffer | Details from RING\_BUFFER\_CONNECTIVITY |
| tbl\_DeadlockReport | Times of Deadlock occurrences |

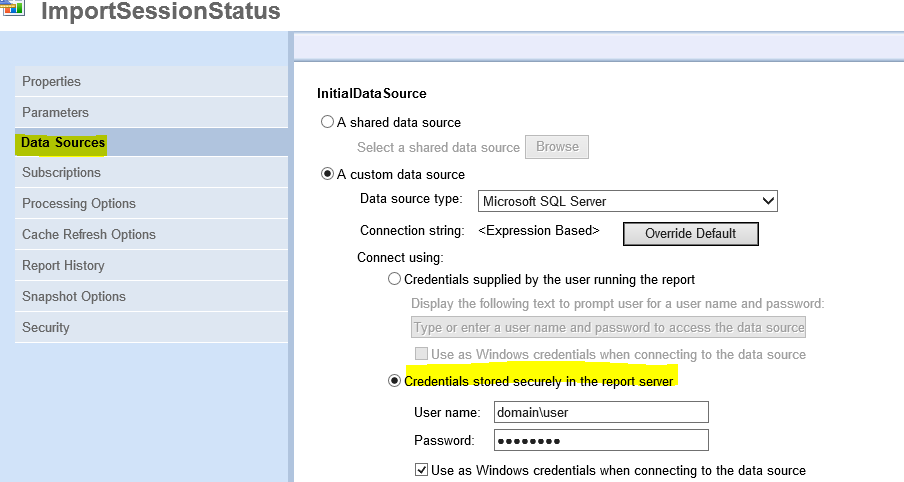
**System Health Dashboard Reports**

1. **Report Deployment:** Extract XEventReporting.zip
   * Open the XEventreporting.sln ( may have to open Visual Studio in Admin mode to deploy)
   * Configure the Report Server you want to Deploy to



1. **Reports:** Start with the Dashboard Report ( Dashboard.rdl), You can choose the Server and Database you imported the data into, and then drill through from the Dashboard to other reports.



1. **Data Source Configuration:** If Kerberos is not configured, you may have to store credentials at the data sources for each report.

**Individual Reports:**

Each report (other than the main Dashboard) has a description of the data source.

**Dashboard Report ( Dashboard.rdl)**

* Parameters allow you to drill into a specific time frame, once chosen, any drill through activity should carry those filtered date times.
* CPU is obtained from the scheduler monitor ring buffer
* Memory is from the Resource component of sp\_server\_diagnostics
* Wait\_queries drill through is from the QUERY\_PROCESSING component of sp\_server\_diagnostics
* The drill through blocking report is not an exact time frame, rather 5 minutes before and 5 minutes after the sp\_server\_diagnostics time frame that is clicked. Blocked processes are captured as part of the QUERY\_PROCESSING sp\_server\_diagnostic component if blocking exists > 30 seconds

Feedback always welcome!

-**Denzil Ribeiro** – Sr. SQL Premier Field Engineer, MSFT