

SQL Server Agent Job and Execution Logs Extraction

Contents

- 1. Introduction3
- 2. SQL Server Agent Job Extraction3
 - 2.1 Bulk Export3
 - 2.2 Single Job Export4
- 3. SQL Server Agent Job Execution History5
- 4. Getting Help.....7

1. Introduction

LeapLogic's Assessment profiles existing inventory, identifies complexity, performs dependency analysis, and provides recommendations for migration to the modern data platform.

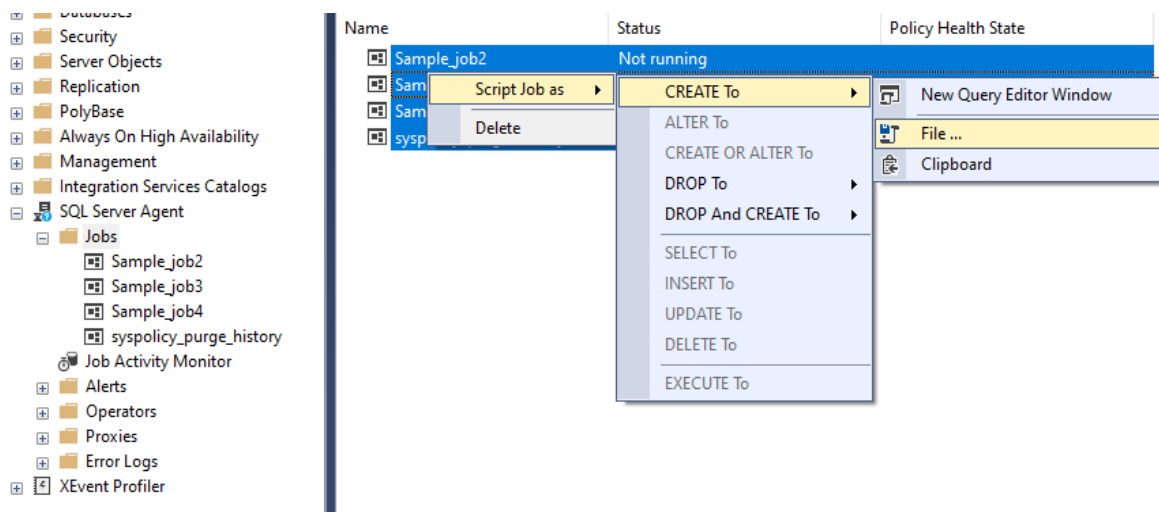
2. SQL Server Agent Job Extraction

LeapLogic requires SQL Server agent jobs exported in .sql or .txt format which typically contains information about all the steps, jobs etc. Follow the below given steps to start exporting the jobs using the SQL Server Management Studio 9SSMS.

2.1 Bulk Export

For exporting multiple SQL Server agent jobs at a time, please follow the below steps.

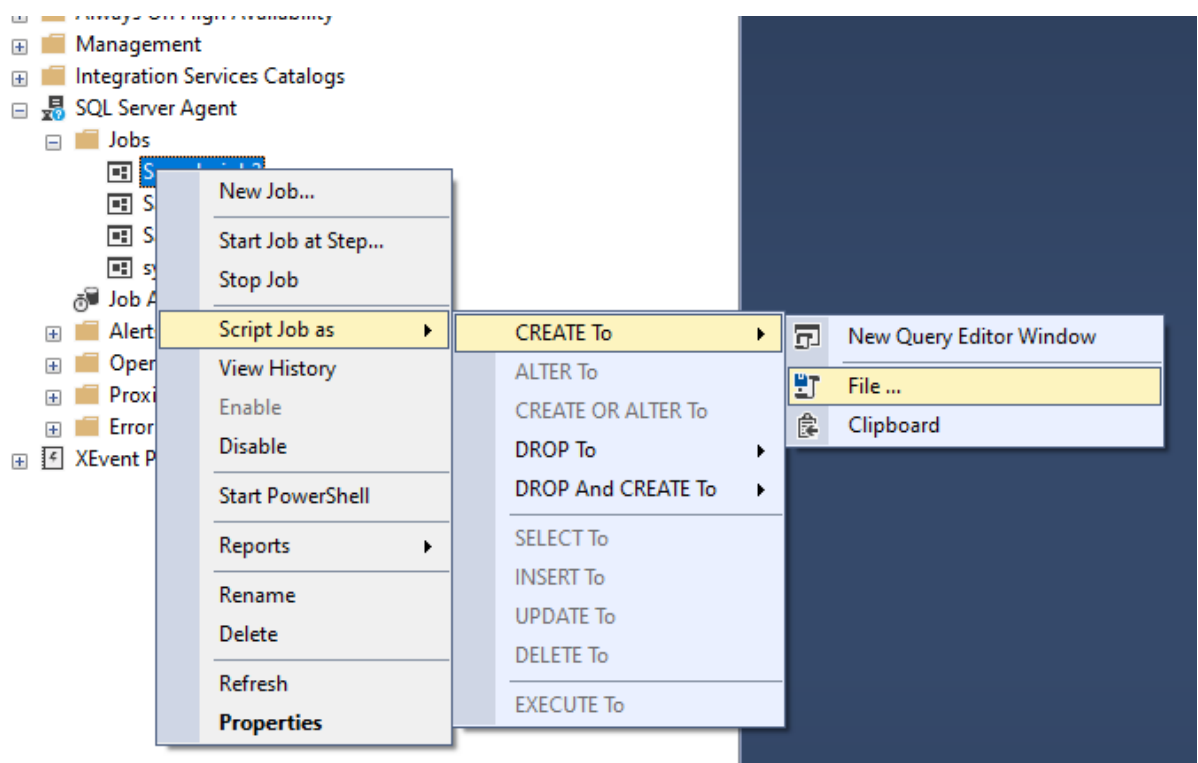
1. Open **SQL Server Management Studio**.
2. Expand **SQL Server Agent** and expand **Jobs**.
3. Hit the **F7** key on your keyboard to open the Object Explorer Details.
4. Select all the **jobs** you want to export.
5. Right-click the highlighted jobs then select **Script Job as**.
6. Select **CREATE To**, then select **File** to select a destination for the scripts. Typically, the destination is a file with a .sql extension.



2.2 Single Job Export

For exporting a single SQL Server agent job at a time, please follow the below steps.

1. Open **SQL Server Management Studio**.
2. Expand **SQL Server Agent** and then **Jobs**.
3. Right-click the **job** you want to create a backup script for, and then select **Script Job as**.
4. Select **CREATE To**, then select **File** to select a destination for the script. Typically, the destination is a file with a .sql extension.
5. Repeat this procedure from **Step 3** for each job you want to export.



3. SQL Server Agent Job Execution History

To export the execution history of SQL Server Agent jobs, please run the following query and save the results to a flat file.

Important Note: Make sure to update the **highlighted** date range to export the required job execution logs.

```
DECLARE @start_date date = '2024-10-01'
DECLARE @end_date date = '2024-10-23'

SELECT DISTINCT
S.job_id
,s.subsystem
,s.database_name AS [Database]
,J.name
,CAST(S.command AS NVARCHAR(MAX)) as Command
,J.enabled
,CASE WHEN SD.step_id IS NULL AND (S.on_success_action=1 or
S.on_success_action=2) then 'INDEPENDENT'
      WHEN SD.step_id IS NOT NULL and (SD.on_success_action=3
or SD.on_success_action=4) THEN 'DEPENDS ON :'+CAST(SD.step_id as
VARCHAR)
      ELSE 'INDEPENDENT'
      END as DEPENDeNCY
,S.step_id
,S.step_name
,H.run_date
,H.run_time
,H.run_duration
,((H.run_duration/10000*3600 + (H.run_duration/100)%100*60 +
H.run_duration%100 + 31 ) / 60)
      as 'RunDurationMinutes'
,CONVERT(VARCHAR(16), J.date_created, 120) date_created
,CONVERT(VARCHAR(16), J.date_modified,120) date_modified
,LEFT(CAST(s.last_run_date AS VARCHAR),4)+ '-'
+SUBSTRING(CAST(s.last_run_date AS VARCHAR),5,2)+'-'
+SUBSTRING(CAST(s.last_run_date AS VARCHAR),7,2) last_run_date
,CASE
  WHEN LEN(CAST(S.last_run_time AS VARCHAR)) = 6
    THEN SUBSTRING(CAST(S.last_run_time AS VARCHAR),1,2)
      +':' + SUBSTRING(CAST(S.last_run_time AS VARCHAR),3,2)
      +':' + SUBSTRING(CAST(S.last_run_time AS VARCHAR),5,2)
  WHEN LEN(CAST(S.last_run_time AS VARCHAR)) = 5
    THEN '0' + SUBSTRING(CAST(S.last_run_time AS VARCHAR),1,1)
      +':' + SUBSTRING(CAST(S.last_run_time AS VARCHAR),2,2)
      +':' + SUBSTRING(CAST(S.last_run_time AS VARCHAR),4,2)
  WHEN LEN(CAST(S.last_run_time AS VARCHAR)) = 4
    THEN '00:'
      + SUBSTRING(CAST(S.last_run_time AS VARCHAR),1,2)
      +':' + SUBSTRING(CAST(S.last_run_time AS VARCHAR),3,2)
  WHEN LEN(CAST(S.last_run_time AS VARCHAR)) = 3
    THEN '00:'
      + '0' + SUBSTRING(CAST(S.last_run_time AS VARCHAR),1,1)
      +':' + SUBSTRING(CAST(S.last_run_time AS VARCHAR),2,2)
```

```

WHEN LEN(CAST(S.last_run_time AS VARCHAR)) = 2 THEN '00:00:' +
CAST(S.last_run_time AS VARCHAR)
WHEN LEN(CAST(S.last_run_time AS VARCHAR)) = 1 THEN '00:00:' + '0'+
CAST(S.last_run_time AS VARCHAR)
END last_run_time
,CAST(SC.next_scheduled_run_date AS DATE) next_run_date
,convert(char(5), SC.next_scheduled_run_date, 108) as next_run_time
,CASE WHEN freq_type=1 THEN 'One time only'
      WHEN freq_type=2 THEN 'Daily'
      WHEN freq_type=8 THEN 'Weekly'
      WHEN freq_type=16 THEN 'Monthly'
      WHEN freq_type=32 THEN 'Monthly, relative to
freq_interval'
      WHEN freq_type=64 THEN 'Runs when the SQL Server Agent
service starts'
      WHEN freq_type=128 THEN 'Runs when the computer is idle'
      ELSE ''
      END AS Frequency
,CASE WHEN freq_type =1 THEN 'Sunday'
      WHEN freq_type =2 THEN 'Monday'
      WHEN freq_type =3 THEN 'Tuesday'
      WHEN freq_type =4 THEN 'Wednesday'
      WHEN freq_type =5 THEN 'Thursday'
      WHEN freq_type =6 THEN 'Friday'
      WHEN freq_type =7 THEN 'Saturday'
      WHEN freq_type =8 THEN 'Day'
      WHEN freq_type =9 THEN 'Weekday'
      WHEN freq_type =10 THEN 'Weekend day'
      ELSE ''
      END AS InterValType
FROM msdb.dbo.sysjobsteps AS S
INNER JOIN msdb.dbo.sysjobs as J
ON S.job_id = J.job_id
INNER JOIN msdb.dbo.sysjobhistory H
ON S.job_id = H.job_id
INNER JOIN
(SELECT job_id,MIN(next_scheduled_run_date) as next_scheduled_run_date
FROM
msdb.dbo.sysjobactivity
GROUP BY job_id
) AS SC
ON J.job_id = SC.job_id
LEFT JOIN msdb.dbo.sysjobsteps AS SD
ON S.job_id = SD.job_id
and S.step_id-1 = SD.step_id
LEFT JOIN msdb.dbo.sysjobschedules AS JS on J.job_id = JS.job_id
LEFT join msdb.dbo.sysschedules AS SS on JS.schedule_id = SS.schedule_id
Where convert(date,convert(varchar(8),H.run_date),101)
BETWEEN @start_date and @end_date -- Change for date range queries
-- and LEFT(j.name,1) NOT IN ('1','2','3','4','5','6','7','8','9','0')
and CHARINDEX('-',j.name) = 0
and j.name <> 'syspolicy_purge_history'
ORDER BY S.job_id, S.step_id;

```

4. Getting Help

Contact LeapLogic technical support at info@leaplogic.io