CTX-Logging
Deployment Plan

Contents

Versions

Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
04/12/2018	1.0	First release
11/03/2019	2.0	Updated to deploy the Partitioned Solution.
01/07/2021	2.3	Updated to reflect Replication parameters and stored procedures

Module Versions

This version of the CTX-Logging deployment plan is relevant up to version 2.3 of the CTX-Logging module.

Preface

About this Manual

This document provides a guide on how to deploy the CTX-Logging module in your Cortex system.

Audience

This document is intended for those who require the use of CTX-Logging module.

Related Material

Document
CTX-Logging – User Guide
CTX-Logging.studiopkg
Cortex-Logging-Install.sql

Abbreviations used in this Document

SQL Structured Query Language

DB Database

1 Requirements

This document details all the steps required to deploy the CTX-Logging module.

Requirements:

- SQL Server Management Studio Access to the Cortex Database Server
- Minimum Cortex v6.4 installed on the Cortex Application Server
- Minimum SQL Server 2012 (version 11.0.7001.0) installed on the Cortex Database Server.
    - SQL Server Express is not supported for this module.

2 Import CTX-Logging

To deploy the CTX-Logging module on your Cortex system, CTX-Logging Studio Package needs to be imported on your Cortex system. To do this:

- Download the CTX-Logging Studio Package
- Import the Studio Package in Cortex Gateway
- Ensure the relevant users have the required permissions in 'Studio Authorisation'

After this, all users in the authorised groups will be able to view and execute the subtasks.

3 Deploy Cortex-Logging Database

3.1 Overview

For the CTX-Logging module to work, the Cortex-Logging database along with the schema must exist on the server where the Cortex databases exist. The following steps instruct you how to deploy the database and schema.

3.2 Create Database

1 Open Remote Desktop Connection to the Cortex database server or SQL Server Management Studio Remote Access

2 Copy the 'Cortex-Logging-Install.sql' script to the Cortex database server

3 Open 'Cortex-Logging-Install.sql' in SQL Server Management Server (SSMS) and connect to the DB engine where the query should be executed (this is where Cortex DBs are hosted).

4 Click on ***Query*** -> ***SQLCMD Mode*** and execute the query

5 Replace the SQL Command variables as required. For example, see **Appendix A**.

```
:setvar CortexDBUser "DOMAIN\ctx_DB"

:setvar DatabaseFilePath "C:\ProgramData\Cortex\Databases\"

:setvar DatabaseLogPath "C:\ProgramData\Cortex\Databases\"

:setvar InstanceName "server/instance hostname"

:setvar numPartitions "7"

:setvar retentionLengthHours "48"

:setvar SQLAdministrator "DOMAIN\ctx_SQL"

:setvar Distribution_DataPath "N/A if replication not required"

:setvar Distribution_LogPath "N/A if replication not required"

:setvar isReplicated "false if replication not required, otherwise true"

:setvar MachineName "SQL server hostname"

:setvar REPL_Admin_User "DOMAIN\ctx_sql"

:setvar ResilientInstanceName "SQL server hostname if not replicated. otherwise resiliant name"

:setvar DatabaseName "Cortex-Logging"
```

Variable	Description
CortexDBUser	The Cortex Database Interface username on your Cortex system. Example: AD\CTX_CerberusDB

DatabaseFilePath	The directory to install the database Datefile to. Inside this directory there must be the folders <Database-Name>\Datafile, e.g. `C:\Cortex Databases\Cortex-Logging\Datafile` - note that the highlighted section should not be included in the variable as this will be appended automatically.
DatabaseLogPath	The directory to install the database Logfile to. Inside this directory there must be the folders <Database-Name>\Logfile, e.g. `C:\Cortex Databases\Cortex-Logging\Logfile` - note that the highlighted section should not be included in the variable as this will be appended automatically.
InstanceName	The name of the SQL Server or SQL Server Instance. If there are no named instances then this should just be the server name
numPartitions	The number of partitioned tables to create. This combined with Retention Length sets the data retention – is it suggested to have at least 3 more partitions than required.
retentionLengthHours	For partitions which change daily, this will be 24. Weekly = 168 Monthly (4 weeks) = 672
SQLAdministrator	The SQL Administrator, e.g. AD\CTX_SQL_Admin
Distribution_DataPath	For replication, the root folder where the Distribution Database is stored e.g. `C:\Cortex Databases` . This folder must contain the `Distribution\Datafile` folders within it. For non-replicated scenario's, use N/A
Distribution_LogPath	For replication, the root folder where the Distribution Database is stored e.g. `C:\Cortex Databases` . This folder must contain the `Distribution\Logfile` folders within it. For non-replicated scenario's, use N/A
isReplicated	True / False depending on whether Replication is required
MachineName	Server name – for replication, this should be the server it is currently being run on
REPL_Admin_User	For replication, the SQL Admin User. Usually the same as SQLAdministrator

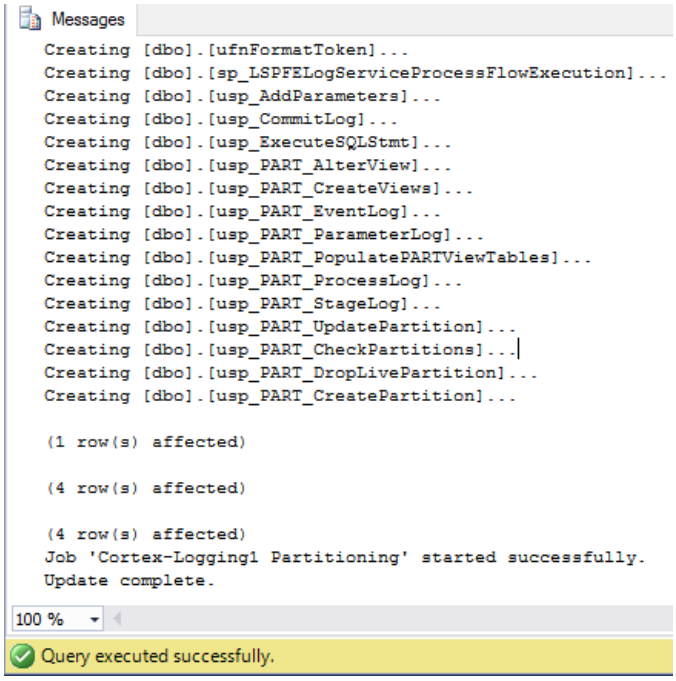
DatabaseName	The Logging database name. It is advised to leave the default value 'Cortex-Logging'. Changing this value require updating the module flows/subtasks default values.

Retention / Partition Examples:

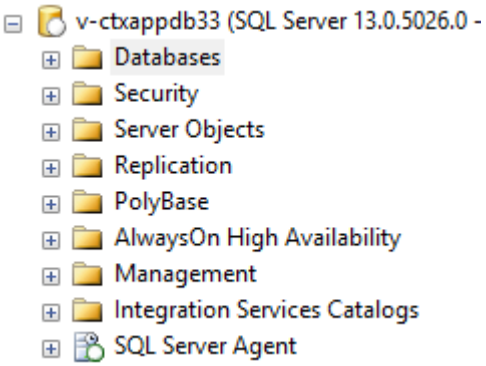
Partitions	Description
Weekly Rolling Partitions	To get partitions which last a week each, the retentionLengthHours should be set to 168. Based on this, the Number of Partitions should also be set – so for data retention of a year this would need to be 52. Ideally there should be 3 extra partitions than the required number.
Monthly Rolling Partitions	To get partitions which last a month (4 weeks) each, the retentionLengthHours should be set to 672. Based on this, the Number of Partitions should also be set – so for data retention of a year this would need to be (52 / 4) = 13. Ideally there should be 3 extra partitions than the required number.

6 Before proceeding, ensure that the SQL Server Agent is running. This can be checked under Services > SQL Server Agent (MSSQLSERVER) for a Default SQL Server Instance or SQL Server Agent (<Instance Name>) for a Named SQL Server Instance

7 On the messages panel, you should see no errors on the messages and the below text

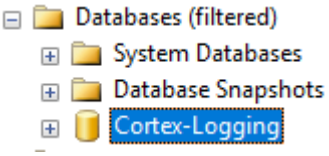


8 In the left-hand panel, click the plus to the left of 'Databases' to expand 'Databases'



9 Right click ‘Databases’ and click ‘Refresh’.

10 Validate the ‘Cortex-Logging’ database has been created.



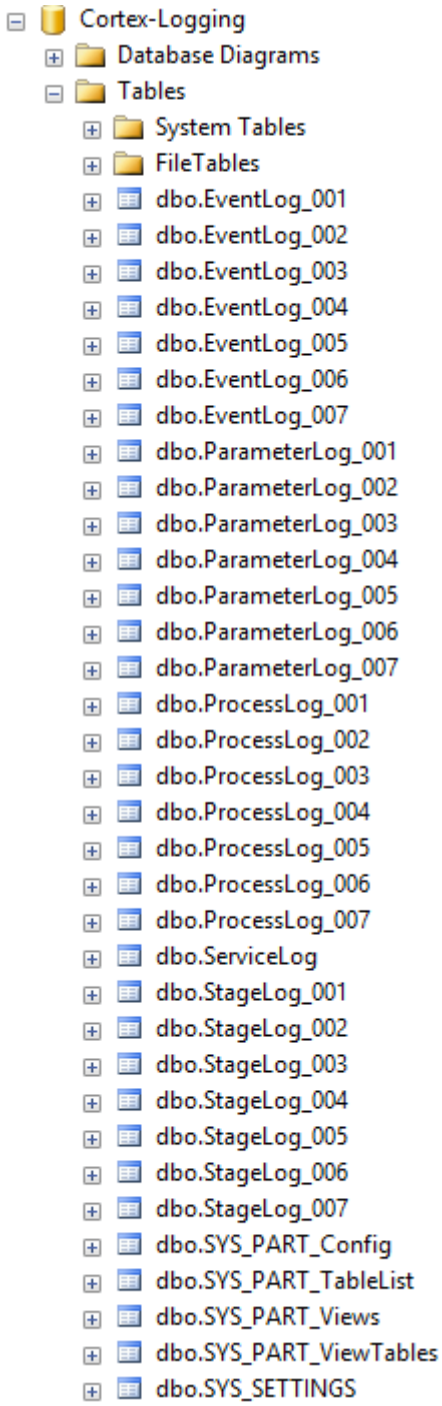
11 Expand ‘Cortex-Logging’ (presuming the default Database Name was selected)

12 Expand ‘Tables’

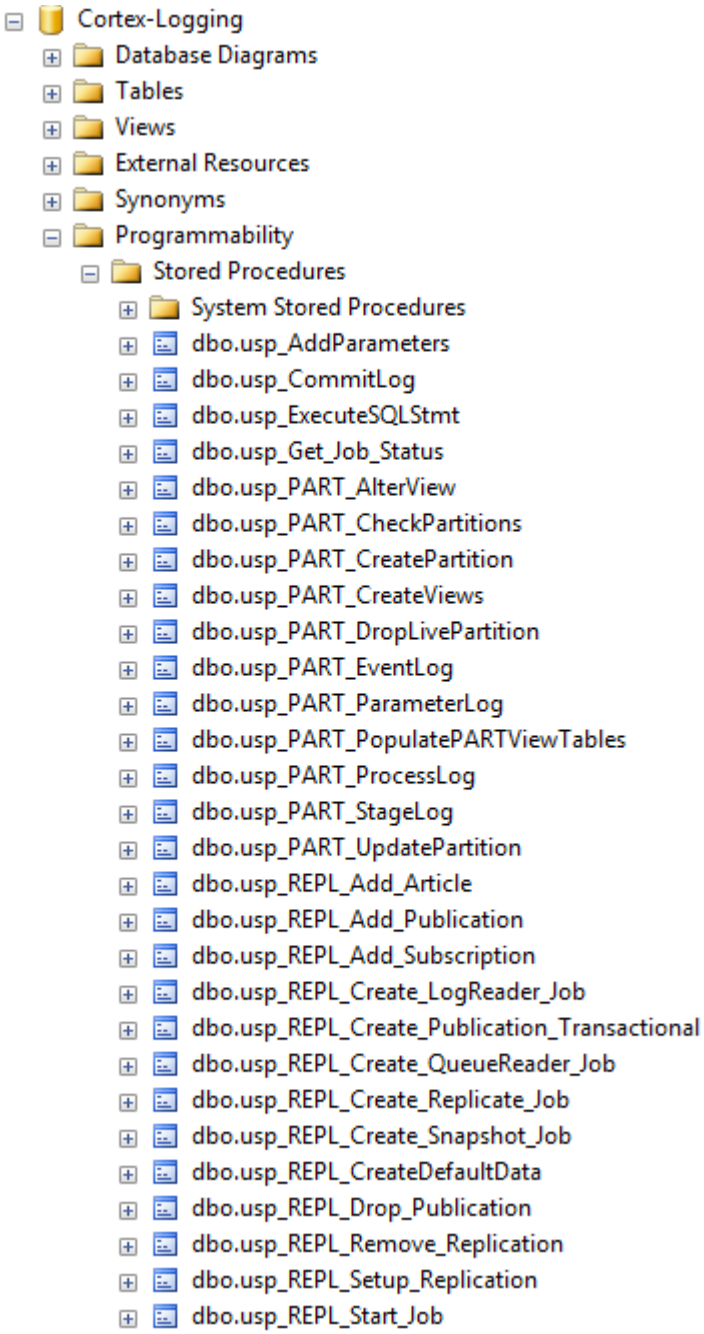
13 Expand ‘Programmability’ > ‘Stored Procedures’

14 Validate the tables and stored procedures shown below are present (the below example is for 7 partitions, meaning 7 instances of every partitioned table)

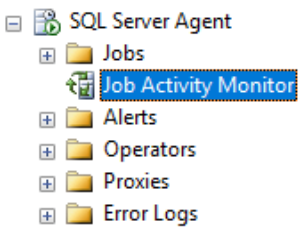
Tables



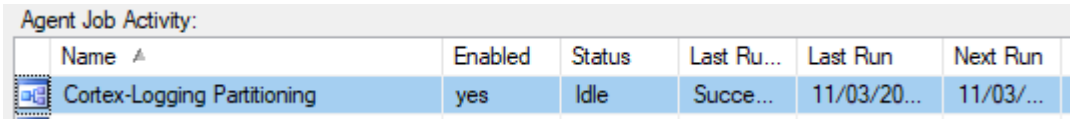
Stored Procedures



15 Expand SQL Server Agent and double click the Job Activity Monitor



16 Check that the Cortex Logging Partitioning Job has run successfully. This should be running as the SQL Administrator.



You can validate this by checking the ‘Last Run’ date. For more details it is possible to right-click this and select ‘View History’.

3.3 Replicated Setups

If replication is required, the deployment script will need to be run on each instance of SQL Server. The following parameters will need to be adjusted depending on which server it is running on:

```
:setvar InstanceName "currentServer"
:setvar isReplicated "true"
:setvar MachineName "currentServer"
:setvar ResilientInstanceName "otherServer"
```

Examples of these parameters are shown in **Appendix A**.

Once the database is deployed on both servers, the following Stored Procedures should be run:

SQL Statement	Server
`exec [dbo].[usp_REPL_Setup_Replication]`	Primary
`exec [dbo].[usp_REPL_Setup_Replication]`	Secondary
`exec [dbo].[usp_REPL_Create_Publication_Transactional]`	Primary
`exec [dbo].[usp_REPL_Create_Publication_Transactional]`	Secondary

4 Appendix A – Example Parameters

4.1 Single-Server Deployment, Default SQL Instance

```
:setvar CortexDBUser "myDomain\ctx_databaseUser"
:setvar DatabaseFilePath "C:\Cortex Databases\"
:setvar DatabaseLogPath "C:\Cortex Databases\"
:setvar InstanceName "myDbServer"
:setvar numPartitions "7"
:setvar retentionLengthHours "48"
:setvar SQLAdministrator "myDomain\ctx_sqlAdminUser"
:setvar Distribution_DataPath "N/A"
:setvar Distribution_LogPath "N/A"
:setvar isReplicated "false"
:setvar MachineName "myDbServer"
:setvar REPL_Admin_User "N/A "
:setvar ResilientInstanceName "myDbServer"
:setvar DatabaseName "Cortex-Logging"
```

Note that the ResilientInstanceName is set to the same server we are deploying on in this scenario. SQL will validate the server name, so this cannot be set to N/A

4.2 Single-Server Deployment, Named SQL Instance

```
:setvar CortexDBUser "myDomain\ctx_databaseUser"
:setvar DatabaseFilePath "C:\Cortex Databases\"
:setvar DatabaseLogPath "C:\Cortex Databases\"
:setvar InstanceName "myDbServer\namedInstance"
:setvar numPartitions "7"
:setvar retentionLengthHours "48"
:setvar SQLAdministrator "myDomain\ctx_sqlAdminUser"
:setvar Distribution_DataPath "N/A"
:setvar Distribution_LogPath "N/A"
:setvar isReplicated "false"
:setvar MachineName "myDbServer"
:setvar REPL_Admin_User "N/A "
:setvar ResilientInstanceName "myDbServer\namedInstance"
:setvar DatabaseName "Cortex-Logging"
```

In this scenario, we are setting InstanceName and resilientInstanceName to include the Instance Name, as it is not the default SQL Instance.

4.3 Resilient / Replicated SQL Setup

4.3.1 Primary Server Deployment

The example Primary Server Hostname is ‘primaryDbServer’ and the secondary is ‘secondaryDbServer’. The parameters which need to be changed between each server are highlighted in yellow.

```
:setvar CortexDBUser "myDomain\ctx_databaseUer"
:setvar DatabaseFilePath "C:\Cortex Databases\"
:setvar DatabaseLogPath "C:\Cortex Databases\"
:setvar InstanceName "primaryDbServer"
```



```
:setvar numPartitions "7"
:setvar retentionLengthHours "48"
:setvar SQLAdministrator "myDomain\ctx_sqlAdminUser"
:setvar Distribution_DataPath "C:\Cortex Databases\"
:setvar Distribution_LogPath "C:\Cortex Databases\"
:setvar isReplicated "true"
:setvar MachineName "primaryDbServer"
:setvar REPL_Admin_User "myDomain\ctx_dbAdminUser"
:setvar ResilientInstanceName "secondaryDbServer"
:setvar DatabaseName "Cortex-Logging"
```

4.3.2 Secondary Server Deployment

The example Primary Server Hostname is ‘primaryDbServer’ and the secondary is ‘secondaryDbServer’. The parameters which need to be changed between each server are highlighted in yellow.

```
:setvar CortexDBUser "myDomain\ctx_databaseUer"
:setvar DatabaseFilePath "C:\Cortex Databases\"
:setvar DatabaseLogPath "C:\Cortex Databases\"
:setvar InstanceName "secondaryDbServer"
:setvar numPartitions "7"
:setvar retentionLengthHours "48"
:setvar SQLAdministrator "myDomain\ctx_sqlAdminUser"
:setvar Distribution_DataPath "C:\Cortex Databases\"
:setvar Distribution_LogPath "C:\Cortex Databases\"
:setvar isReplicated "true"
:setvar MachineName "secondaryDbServer"
:setvar REPL_Admin_User "myDomain\ctx_dbAdminUser"
:setvar ResilientInstanceName "primaryDbServer"
:setvar DatabaseName "Cortex-Logging"
```

