How DLM Dashboard works

For DLM Dashboard to work, you need to run a setup script on servers you want to monitor. It installs several objects:

- a DDL trigger called *RG_SQLLighthouse_DDLTrigger*
  DDL triggers run a stored procedure in response to a specified Data Definition Language (DDL) event. For more information, see Understanding DDL Triggers (MSDN).
- a database called *RedGate*
- three stored procedures called *RG_SQLLighthouse_WriteEvent*, *RG_SQLLighthouse_ReadEvents*, and *RG_SQLLighthouse_WriteError*

DLM Dashboard detects schema changes with the DDL trigger. Whenever a database's schema changes, the DDL trigger fires, triggering the stored procedure *RG_SQLLighthouse_WriteEvent*.

The stored procedure adds details of the changes to the database called *RedGate*, which DLM Dashboard's monitoring service polls regularly. If a schema has changed, the monitoring service updates the DLM Dashboard interface, and alerts you.

DLM Dashboard also takes a snapshot of the database schema when a database has changed. It compares these schema snapshots to a snapshot of the previous schema version, and records the differences in the DLM Dashboard's storage service so you can see the changes in the dashboard.

Running the setup script

To install the objects that DLM Dashboard needs, run this SQL script on each server you want to monitor:

Installation script

```
/*
 *    Red Gate
 *
 *
 *  _____   _       __  __       _____                _      _                              _
 * |  __ \ | |     |  \/  |     |  __ \              | |    | |                            | |
 * | |  | || |     | \  / |     | |  | |  __ _  ___ | |__  | |__    ___    __ _  _ __  __| |
 * | |  | || |     | |\/| |     | |  | | / _` |/ __|| '_ \ | '_ \  / _ \  / _` || '__|/ _` |
 * | |__| || |____ | |  | |     | |__| || (_| |\__ \| | | || |_) || (_) || (_| || |  | (_| |
 * |_____/ |______||_|  |_|     |_____/  \__,_||___/|_| |_||_.__/  \___/  \__,_||_|   \__,_|
 *
 *
 *
 *
 *    For information about these objects, see: http://www.red-gate.com/dlm-dashboard/ddl-trigger
 *
 *    If you have any problems, or need to change these objects, we really want to know,
 *    please email us at: dlmdashboardsupport@red-gate.com
 *
 *    Copyright © Red Gate Software Ltd. 2015 | Version 1.11 | 2015-06-15
 */
USE master;
GO
/* Make sure RedGate database exists, and writes to it don't block transactions in Snapshot isolation level */
IF DB_ID(N'RedGate') IS NULL
        CREATE DATABASE RedGate
ALTER DATABASE RedGate SET RECOVERY SIMPLE
ALTER DATABASE RedGate SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE RedGate SET READ_COMMITTED_SNAPSHOT ON
GO
/* Create/update objects necessary for the DLM Dashboard change detection trigger in a transaction */
BEGIN TRANSACTION RG_SQLLighthouse_CreateTrigger
IF EXISTS (SELECT 1 FROM sys.server_triggers WHERE name = 'RG_SQLLighthouse_DDLTrigger')
        DROP TRIGGER RG_SQLLighthouse_DDLTrigger ON ALL SERVER
GO
IF OBJECT_ID(N'dbo.RG_SQLLighthouse_WriteEvent', N'P') IS NOT NULL
        DROP PROCEDURE dbo.RG_SQLLighthouse_WriteEvent;
GO
IF OBJECT_ID(N'dbo.RG_SQLLighthouse_WriteError', N'P') IS NOT NULL
        DROP PROCEDURE dbo.RG_SQLLighthouse_WriteError;
GO
IF OBJECT_ID(N'dbo.RG_SQLLighthouse_ReadEvents', N'P') IS NOT NULL
        DROP PROCEDURE dbo.RG_SQLLighthouse_ReadEvents;
GO
/* Ensure SQLLighthouse schema exists */
```

```
USE RedGate;
GO
IF OBJECT_ID(N'SQLLighthouse.DDL_Events', N'U') IS NOT NULL
        DROP TABLE SQLLighthouse.DDL_Events
GO
IF OBJECT_ID(N'SQLLighthouse.Trigger_Errors', N'U') IS NOT NULL
        DROP TABLE SQLLighthouse.Trigger_Errors
GO
IF SCHEMA_ID(N'SQLLighthouse') IS NOT NULL
        DROP SCHEMA SQLLighthouse
GO
CREATE SCHEMA SQLLighthouse AUTHORIZATION dbo
GO
CREATE TABLE SQLLighthouse.DDL_Events (
        id bigint NOT NULL IDENTITY (1,1) PRIMARY KEY,
        PostTime datetimeoffset,
        transaction_id bigint, spid smallint, options integer, nestlevel int, langid smallint,
client_net_address varchar(48), appname nvarchar(128),
        eventdata XML
  )
GO
CREATE TABLE SQLLighthouse.Trigger_Errors (
        id BIGINT NOT NULL IDENTITY(1, 1) PRIMARY KEY,
        error_time DATETIMEOFFSET,
        error_number INT,
        error_severity INT,
        error_state INT,
        error_procedure sysname,
        error_line INT,
        error_message NVARCHAR(4000),
)
GO
USE master;
GO
/*
 *    Red Gate
 *
 *
 *  _____   _       __  __       _____                _      _                            _
 * |  __ \ | |     |  \/  |     |  __ \              | |    | |                          | |
 * | |  | || |     | \  / |     | |  | |  __ _  ___ | |__  | |__    ___    __ _  _ __  __| |
 * | |  | || |     | |\/| |     | |  | | / _` |/ __|| '_ \ | '_ \  / _ \  / _` || '__|/ _` |
 * | |__| || |____ | |  | |     | |__| || (_| |\__ \| | | || |_) || (_) || (_| || |  | (_| |
 * |_____/ |______||_|  |_|     |_____/  \__,_||___/|_| |_||_.__/  \___/  \__,_||_|   \__,_|
 *
 *
 *
 *
 *    For information about these objects, see: http://www.red-gate.com/dlm-dashboard/ddl-trigger
 *
 *    If you have any problems, or need to change these objects, we really want to know,
 *    please email us at: dlmdashboardsupport@red-gate.com
 *
 *    Copyright © Red Gate Software Ltd. 2015 | Version 1.11 | 2015-06-15
 */
/*****************************************************************************************************
 * Stored procedure: master.dbo.RG_SQLLighthouse_WriteEvent
 * Used by trigger RG_SQLLighthouse_DDLTrigger to write change events to RedGate.SQLLighthouse.DDL_Events
 *****************************************************************************************************/
CREATE PROCEDURE dbo.RG_SQLLighthouse_WriteEvent
        @eventdata XML
AS
        INSERT INTO RedGate.SQLLighthouse.DDL_Events
        VALUES  (
                TODATETIMEOFFSET(CONVERT(varchar(23), @eventdata.query('data(/EVENT_INSTANCE/PostTime)')),
DATEPART(tz, SYSDATETIMEOFFSET())),
                (SELECT transaction_id from sys.dm_tran_current_transaction), @@SPID, @@OPTIONS, @@NESTLEVEL,
@@LANGID, CONVERT(varchar(48), CONNECTIONPROPERTY('client_net_address')), APP_NAME(),
                @eventdata
           )
        IF (SELECT COUNT(*) FROM RedGate.SQLLighthouse.DDL_Events) >= 512
```

```
        BEGIN
                DELETE FROM RedGate.SQLLighthouse.DDL_Events WHERE id <= SCOPE_IDENTITY() - 420
        END
GO
/*
 *    Red Gate
 *
 *
 *   _____   _       __  __       _____               _      _                            _
 * |  __ \ | |     |  \/  |     |  __ \             | |    | |                          | |
 * | |  | || |     | \  / |     | |  | | __ _  ___ | |__  | |__    ___    __ _  _ __  __| |
 * | |  | || |     | |\/| |     | |  | | / _` |/ __|| '_ \ | '_ \  / _ \  / _` || '__|/ _` |
 * | |__| || |____ | |  | |     | |__| || (_| |\__ \| | | || |_) || (_) || (_| || |  | (_| |
 * |_____/ |______||_|  |_|     |_____/  \__,_||___/|_| |_||_.__/  \___/  \__,_||_|   \__,_|
 *
 *
 *
 *
 *    For information about these objects, see: http://www.red-gate.com/dlm-dashboard/ddl-trigger
 *
 *    If you have any problems, or need to change these objects, we really want to know,
 *    please email us at: dlmdashboardsupport@red-gate.com
 *
 *    Copyright © Red Gate Software Ltd. 2015 | Version 1.11 | 2015-06-15
 */
/*****************************************************************************************************
**
 * Stored procedure: master.dbo.RG_SQLLighthouse_ReadEvents
 * Used by the DLM Dashboard change detection service to read change events from RedGate.SQLLighthouse.
DDL_Events

*****************************************************************************************************
*/
CREATE PROCEDURE dbo.RG_SQLLighthouse_ReadEvents
AS
        SELECT * FROM RedGate.SQLLighthouse.DDL_Events
GO
/*
 *    Red Gate
 *
 *
 *   _____   _       __  __       _____               _      _                            _
 * |  __ \ | |     |  \/  |     |  __ \             | |    | |                          | |
 * | |  | || |     | \  / |     | |  | | __ _  ___ | |__  | |__    ___    __ _  _ __  __| |
 * | |  | || |     | |\/| |     | |  | | / _` |/ __|| '_ \ | '_ \  / _ \  / _` || '__|/ _` |
 * | |__| || |____ | |  | |     | |__| || (_| |\__ \| | | || |_) || (_) || (_| || |  | (_| |
 * |_____/ |______||_|  |_|     |_____/  \__,_||___/|_| |_||_.__/  \___/  \__,_||_|   \__,_|
 *
 *
 *
 *
 *    For information about these objects, see: http://www.red-gate.com/dlm-dashboard/ddl-trigger
 *
 *    If you have any problems, or need to change these objects, we really want to know,
 *    please email us at: dlmdashboardsupport@red-gate.com
 *
 *    Copyright © Red Gate Software Ltd. 2015 | Version 1.11 | 2015-06-15
 */
/*****************************************************************************************************
 * Stored procedure: master.dbo.RG_SQLLighthouse_WriteError
 * Used by trigger RG_SQLLighthouse_DDLTrigger to write error logs to RedGate.SQLLighthouse.Trigger_Errors
 *****************************************************************************************************/
CREATE PROCEDURE dbo.RG_SQLLighthouse_WriteError
        @error_number INT,
        @error_severity INT,
        @error_state INT,
        @error_procedure sysname,
        @error_line INT,
        @error_message NVARCHAR(4000)
AS
        INSERT  INTO RedGate.SQLLighthouse.Trigger_Errors
```

```
        VALUES  (
                SYSDATETIMEOFFSET(), @error_number, @error_severity, @error_state, @error_procedure,
@error_line, @error_message
                )
        IF (SELECT COUNT(*) FROM RedGate.SQLLighthouse.Trigger_Errors) >= 512
        BEGIN
                DELETE FROM RedGate.SQLLighthouse.Trigger_Errors WHERE id <= SCOPE_IDENTITY() - 420
        END
GO
/*
 *    Red Gate
 *
 *
 *   _____   _       __  __       _____                _      _                            _
 * |  __ \ | |     |  \/  |     |  __ \              | |    | |                          | |
 * | |  | || |     | \  / |     | |  | |  __ _  ___ | |__  | |__    ___    __ _  _ __  __| |
 * | |  | || |     | |\/| |     | |  | | / _` |/ __|| '_ \ | '_ \  / _ \  / _` || '__|/ _` |
 * | |__| || |____ | |  | |     | |__| || (_| |\__ \| | | || |_) || (_) || (_| || |  | (_| |
 * |_____/ |______||_|  |_|     |_____/  \__,_||___/|_| |_||_.__/  \___/  \__,_||_|   \__,_|
 *
 *
 *
 *
 *    For information about these objects, see: http://www.red-gate.com/dlm-dashboard/ddl-trigger
 *
 *    If you have any problems, or need to change these objects, we really want to know,
 *    please email us at: dlmdashboardsupport@red-gate.com
 *
 *    Copyright © Red Gate Software Ltd. 2015 | Version 1.11 | 2015-06-15
 */
/*****************************************************************************************************
**
 * Server level trigger: RG_SQLLighthouse_DDLTrigger
 * Detects DDL events and writes them to the table (via master.dbo.RG_SQLLighthouse_WriteEvent)

*****************************************************************************************************
*/
CREATE TRIGGER RG_SQLLighthouse_DDLTrigger
    ON ALL SERVER
    WITH EXECUTE AS 'sa'
    AFTER DDL_EVENTS
AS
        IF OBJECT_ID(N'master.dbo.RG_SQLLighthouse_WriteEvent', N'P') IS NOT NULL
        BEGIN
                -- see http://msdn.microsoft.com/en-us/library/ms190763.aspx
                DECLARE @restoreXACT_ABORT bit = 16384 & (SELECT @@OPTIONS);
                SET XACT_ABORT OFF
                DECLARE @restoreANSI_PADDING bit = 16 & (SELECT @@OPTIONS);
                SET ANSI_PADDING ON
                DECLARE @restoreANSI_WARNINGS bit = 8 & (SELECT @@OPTIONS);
                SET ANSI_WARNINGS ON
                DECLARE @restoreCONCAT_NULL_YIELDS_NULL bit = 4096 & (SELECT @@OPTIONS);
                SET CONCAT_NULL_YIELDS_NULL ON
                DECLARE @restoreNUMERIC_ROUNDABORT bit = 8192 & (SELECT @@OPTIONS);
                SET NUMERIC_ROUNDABORT OFF
                DECLARE @restoreNOCOUNT bit = 512 & (SELECT @@OPTIONS);
                SET NOCOUNT ON
                BEGIN TRY
                        DECLARE @eventdata XML = EVENTDATA();
                        EXECUTE master.dbo.RG_SQLLighthouse_WriteEvent @eventdata
                END TRY
                BEGIN CATCH
                        IF OBJECT_ID(N'master.dbo.RG_SQLLighthouse_WriteError', N'P') IS NOT NULL
                        BEGIN
                                BEGIN TRY
                                        DECLARE @error_number INT = ERROR_NUMBER(),
                                                @error_severity INT = ERROR_SEVERITY(),
                                                @error_state INT = ERROR_STATE(),
                                                @error_procedure SYSNAME = ERROR_PROCEDURE(),
                                                @error_line INT = ERROR_LINE(),
                                                @error_message NVARCHAR(4000) = ERROR_MESSAGE();
```

```
                                        EXECUTE master.dbo.RG_SQLLighthouse_WriteError @error_number,
                                                @error_severity, @error_state, @error_procedure,
                                                @error_line, @error_message;
                                END TRY
                                BEGIN CATCH
                                        -- Left intentionally blank :(
                                END CATCH
                        END
                END CATCH
                IF @restoreXACT_ABORT = 1
                BEGIN
                        SET XACT_ABORT ON
                END
                IF @restoreANSI_PADDING = 0
                BEGIN
                        SET ANSI_PADDING OFF
                END
                IF @restoreANSI_WARNINGS = 0
                BEGIN
                        SET ANSI_WARNINGS OFF
                END
                IF @restoreCONCAT_NULL_YIELDS_NULL = 0
                BEGIN
                        SET CONCAT_NULL_YIELDS_NULL OFF
                END
                IF @restoreNUMERIC_ROUNDABORT = 1
                BEGIN
                        SET NUMERIC_ROUNDABORT ON
                END
                IF @restoreNOCOUNT = 0
                BEGIN
                        SET NOCOUNT OFF
                END
        END
GO
/***************************************************************************
 * Cleanup obsolete objects (from previous versions of DLM Dashboard)
 ***************************************************************************/
IF OBJECT_ID(N'tempdb.dbo.RG_SQLLighthouse_v1', N'U') IS NOT NULL
DROP TABLE tempdb.dbo.RG_SQLLighthouse_v1
IF OBJECT_ID(N'tempdb.dbo.RG_SQLLighthouse_Errors_v1', N'U') IS NOT NULL
DROP TABLE tempdb.dbo.RG_SQLLighthouse_Errors_v1
IF OBJECT_ID(N'master.dbo.RG_SQLLighthouse_Sproc', N'P') IS NOT NULL
DROP PROCEDURE dbo.RG_SQLLighthouse_Sproc
COMMIT TRANSACTION RG_SQLLighthouse_CreateTrigger
```

Removing the DLM Dashboard objects from SQL Server

To uninstall the DLM Dashboard objects from your SQL Server, run this uninstall script. The uninstall script *does not* delete the RedGate database - you'll need to delete it manually.