Script Builder

Prepared By:
Sumit Uppal

Script Builder - Features

Script Builder is a comprehensive and easy to use tool to script sql server database tables, views, stored procedures, user-defined functions, and even data, making it easy to synchronize the databases. And to creates script for database deployment. It can also create C# code and basic sql stored procedures.

Features:

- Generates **SQL scripts for tables, user defined functions, views and stored procedures**. Along with constraints on these database objects. It can also create **C# code** and **basic sql stored procedures.**

- It can create a scripts that reproduce the tables and the data contained in it . It can create insert statements to populate data in tables.

- It gives the option to generate script of selected objects. User can select the objects from checkboxes or to load the list from a feed file.

- It gives the option to generate C# code of selected objects. User can select the objects from checkboxes or to load the list from a feed file.

- Generates log file to validate the objects in the scripts.

- Generates print statements within the scripts to help in error debugging.

Script Builder - Advantages

UnitedHealth Group®

Advantages of Script Builder

Script builder can be useful in many ways:

- In creating consolidated script of complete database for database migration
- In creating consolidated script for build deployment.
- **Reduce build errors** as it creates log file to validate the number of object scripted.
- **Saves developer time** as he has to provide only the object name and not the complete script that needs to be deployed. And improves the overall productive and process.
- It can be used as **database viewer**. On machines which doesn't have sql server client installed. This tool can be used to view database objects and their attributes.

Script Builder – Advantages

UnitedHealth Group®

Advantages of Script Builder

- **Saves the build manager/dba time** as he has to provide only the feed file containing the names of the objects changed. Then the script builder would script all the objects in the feed file.

- Can help in debugging of any error occurred during deployment. As it creates log file and contain print messages in the script.

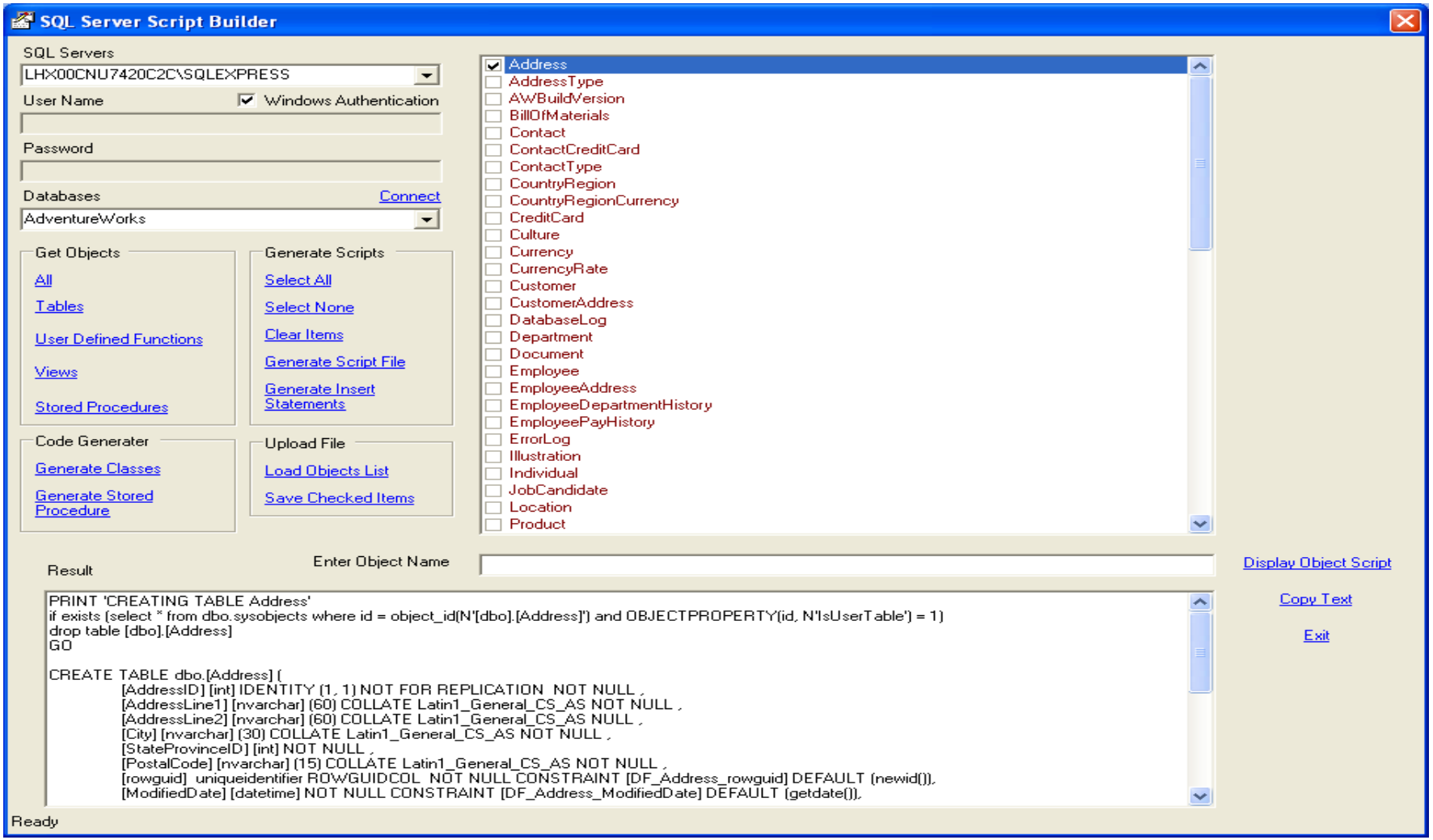
- User can create feed file from the objects selected in the check boxes and later use that feed file to script objects.

- Can be helpful in populating the tables containing static tables. Like of roles and functionalities tables. As it can create insert statements to populate those tables.

- As it's a stand alone executable. Does not require admin rights on machine to install this application.

Script Builder – ScreenShot

UnitedHealth Group®



Script Builder – Sample Object Script

UnitedHealth Group®

```
PRINT 'CREATING TABLE EDICMProviderStatus'
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[EDICMProviderStatus]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[EDICMProviderStatus]
GO

CREATE TABLE dbo.[EDICMProviderStatus] (
    [EDICMProviderStatusID] [bigint] NOT NULL ,
    [EDICMProviderID] [bigint] NOT NULL ,
    [EDIFileHeaderID] [bigint] NOT NULL ,
    [InactiveReasonCode] [nvarchar] (100) COLLATE Latin1_General_CI_AS NULL ,
    [BeginDate] [datetime] NOT NULL ,
    [EndDate] [datetime] NULL ,
    [CreateBy] [bigint] NOT NULL ,
    [CreateDate] [datetime] NOT NULL ,
    [UpdateBy] [bigint] NOT NULL ,
    [UpdateDate] [datetime] NOT NULL ,
    [RecordStatus] [tinyint] NOT NULL ,
    [RecordVersion] [int] NOT NULL ,
    [RecordSource] [tinyint] NOT NULL ,
    [SyncDate] [datetime] NULL ,
    [RecordUniqueID] [int] NOT NULL ,
    [ParentRecordUniqueID] [int] NULL ,
    [ProviderStatus] [nvarchar] (100) COLLATE Latin1_General_CI_AS NULL ,
    CONSTRAINT [PK_EDICMProviderStatus] PRIMARY KEY  CLUSTERED
    (
        [EDICMProviderStatusID]
    )  ON [PRIMARY] ,
    CONSTRAINT [FK_EDICMProviderStatus_EDIFileHeader] FOREIGN KEY
    (
        [EDIFileHeaderID]
    ) REFERENCES [EDIFileHeader] (
        [EDIFileHeaderID]
    )
) ON [PRIMARY]
GO
```

Script Builder – Sample Insert Script

UnitedHealth Group®

```
PRINT 'INSERTING DATA INTO TABLE ActivityInstance'

--TRUNCATE TABLE ActivityInstance
ALTER TABLE [ActivityInstance] NOCHECK CONSTRAINT ALL
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
INSERT INTO [ActivityInstance] ([ActivityInstanceId],[ProcessflowInstanceId],[QualifiedName],[ContextGuid],
```

Script Builder – Sample C# Code

UnitedHealth Group®

```
public class tbl_SPRF_Tracking
{
    /// <summary>
    /// Default Contructor
    /// <summary>
    public tbl_SPRF_Tracking()
    {}


    private string _strReleaseId;
    public string ReleaseId
    {
        get { return _strReleaseId; }
        set { _strReleaseId = value; }
    }


    private int _nSPRFID;
    public int SPRFID
    {
        get { return _nSPRFID; }
        set { _nSPRFID = value; }
    }


    private string _strProgramID;
    public string ProgramID
    {
```

Script Builder – Sample Stored Procedures

UnitedHealth Group®

```
--STORED PROCEDURE TO INSERT DATA IN tbl_BS_Manager
CREATE PROCEDURE dbo.tbl_BS_Manager_INSERT
    @BSMRID int,
    @BSID int,
    @ManagerID int
AS
INSERT INTO [tbl_BS_Manager] (
    BSMRID,
    BSID,
    ManagerID)
VALUES (
    @BSMRID,
    @BSID,
    @ManagerID)
GO
--STORED PROCEDURE TO UPDATE DATA IN tbl_BS_Manager
CREATE PROCEDURE dbo.tbl_BS_Manager_UPDATE
    @BSMRID int,
    @BSID int,
    @ManagerID int
AS
UPDATE [tbl_BS_Manager]
SET
    BSMRID = @BSMRID,
    BSID = @BSID,
    ManagerID = @ManagerID
GO
```

Script Builder – Technical Specifications

Technical Specifications

- Works with Sql Server 2000, Sql Server 2005
- Requires .Net framework 2.0 or above

Script Builder – Release Notes

Release Notes

Script Builder

UnitedHealth Group®

Demo

Questions?

UnitedHealth Group®

Thanks