



UnitedHealth Group®

## 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.

## 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.

## 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



**SQL Server Script Builder**

SQL Servers  
LHX00CNU7420C2C\SQLEXPRESS

User Name ☒ Windows Authentication

Password

Databases [Connect](#)  
AdventureWorks

Get Objects  
[All](#)  
[Tables](#)  
[User Defined Functions](#)  
[Views](#)  
[Stored Procedures](#)

Generate Scripts  
[Select All](#)  
[Select None](#)  
[Clear Items](#)  
[Generate Script File](#)  
[Generate Insert Statements](#)

Code Generator  
[Generate Classes](#)  
[Generate Stored Procedure](#)

Upload File  
[Load Objects List](#)  
[Save Checked Items](#)

☒ Address  
☐ AddressType  
☐ AWBuildVersion  
☐ BillOfMaterials  
☐ Contact  
☐ ContactCreditCard  
☐ ContactType  
☐ CountryRegion  
☐ CountryRegionCurrency  
☐ CreditCard  
☐ Culture  
☐ Currency  
☐ CurrencyRate  
☐ Customer  
☐ CustomerAddress  
☐ DatabaseLog  
☐ Department  
☐ Document  
☐ Employee  
☐ EmployeeAddress  
☐ EmployeeDepartmentHistory  
☐ EmployeePayHistory  
☐ ErrorLog  
☐ Illustration  
☐ Individual  
☐ JobCandidate  
☐ Location  
☐ Product

Result  
Enter Object Name

PRINT 'CREATING TABLE Address'  
if exists (select \* from dbo.sysobjects where id = object\_id(N'[dbo].[Address]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)  
drop table [dbo].[Address]  
GO  
  
CREATE TABLE dbo.[Address] (  
[AddressID] [int] IDENTITY (1, 1) NOT FOR REPLICATION NOT NULL ,  
[AddressLine1] [nvarchar] (60) COLLATE Latin1\_General\_CS\_AS NOT NULL ,  
[AddressLine2] [nvarchar] (60) COLLATE Latin1\_General\_CS\_AS NULL ,  
[City] [nvarchar] (30) COLLATE Latin1\_General\_CS\_AS NOT NULL ,  
[StateProvinceID] [int] NOT NULL ,  
[PostalCode] [nvarchar] (15) COLLATE Latin1\_General\_CS\_AS NOT NULL ,  
[rowguid] uniqueidentifier ROWGUIDCOL NOT NULL CONSTRAINT [DF\_Address\_rowguid] DEFAULT (newid()),  
[ModifiedDate] [datetime] NOT NULL CONSTRAINT [DF\_Address\_ModifiedDate] DEFAULT (getdate()),  
)

[Display Object Script](#)  
[Copy Text](#)  
[Exit](#)

Ready

# 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
```

## UnitedHealth Group®

Any use, copying or distribution without written permission from UnitedHealth Group is prohibited.

# 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



UnitedHealth Group®

## Technical Specifications

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

# Script Builder – Release Notes



UnitedHealth Group®

## Release Notes



Microsoft Word  
Document



**Demo**



**Questions?**



**Thanks**