

System Overview

March 2017



Contents

1	Overview.....	4
2	The Development Cycle.....	5
3	Check Out	8
3.1	Description.....	8
3.2	Process Flow.....	8
3.3	User Interface.....	9
3.4	User Interface Fields.....	10
3.5	User Interface Actions	10
4	Check In	13
4.1	Description.....	13
4.2	Process Flow.....	13
4.3	User Interfaces	14
4.4	User Interface Fields.....	14
4.5	User Interface Actions	15
5	Undo Check Out	16
5.1	Description.....	16
5.2	Process Flow.....	16
5.3	User Interfaces	17
5.4	User Interface Fields.....	18
5.5	User Interface Actions	18
6	Take Object Back to Development.....	19
6.1	Description.....	19
6.2	Process Flow.....	19
6.3	User Interfaces	20
6.4	User Interface Fields.....	21
6.5	User Interface Actions	21
7	Quick Backup	22
7.1	Description.....	22
7.2	Process Flow.....	22
7.3	User Interfaces	23
8	Compare	24
8.1	Description.....	24
8.2	Process Flow.....	24

8.3	User Interfaces	25
8.4	User Interface Actions	28
9	View History.....	30
9.1	Description.....	30
9.2	Process Flow.....	30
9.3	User Interfaces	31
10	Project Maintenance	33
10.1	Description.....	33
10.2	Process Flow.....	34
10.3	User Interfaces	35
10.4	User Interface Fields.....	37
10.5	User Interface Actions	37
11	Release.....	38
11.1	Description.....	38
11.2	Process Flow.....	38
11.3	User Interfaces	39
11.4	User Interface Actions	40
12	Search	41
12.1	Description.....	41
12.2	User Interfaces	41
13	Backup and Restore.....	44
13.1	Description.....	44
13.2	User Interfaces	44
13.3	User Interface Actions	46
13.4	User Interface Actions	47
14	Administration – User Maintenance	48
14.1	Description.....	48
14.2	User Interfaces	48
14.3	User Interface Fields.....	49
14.4	User Interface Actions	50
15	Server Setup.....	51
15.1	Description.....	51
15.2	User Interfaces	51
15.3	User Interface Fields.....	52

15.4	User Interface Actions	52
16	Administration – Role Settings.....	53
16.1	Description	53
16.2	User Interfaces	53
17	Data Store Settings.....	55
17.1	Description	55
17.2	User Interfaces	55
17.3	User Interface Fields.....	56
17.4	User Interface Actions	56

1 Overview

SQL Sure protects your database objects' source code. Fully-integrated in Microsoft™ SQL Server™ Management Studio. Check the object out for development and lock it to protect the artefact from being overwritten. A backup copy is always available with a full history of every change.

- Compatible with SSMS 2012, 2014 and 2016.
- Repository is kept in the database, no external file systems to maintain.
- Intuitive, simple, effective source control.

Link objects to a release project for simple deployment across environments. Development to UAT to Production – simple. There is always a backup copy made before deploying the object to the next environment.

- Create a deployment project during check-out or simply add more objects to the project.
- Include file based scripts such as data inserts, configuration scripts and the likes for deployment - all script data is stored in the database.
- Select your project, select the “from” and “to” environments and click “Go”, no more release disasters.
- All release objects are automatically backed-up before changes are released.

A developer's dream...

- Powerful search in all database objects.
- Compare your object's code across multiple environments.
- View object differences between environments, line by line.
- View an object's full history trail.
- User administration and role-based security.
- Simple, effective source control and deployment for your database.

2 The Development Cycle

The development cycle for a typical programmable database object is done across multiple environments.

In the Microsoft™ SQL Server™ environment, some of these programmable objects are:

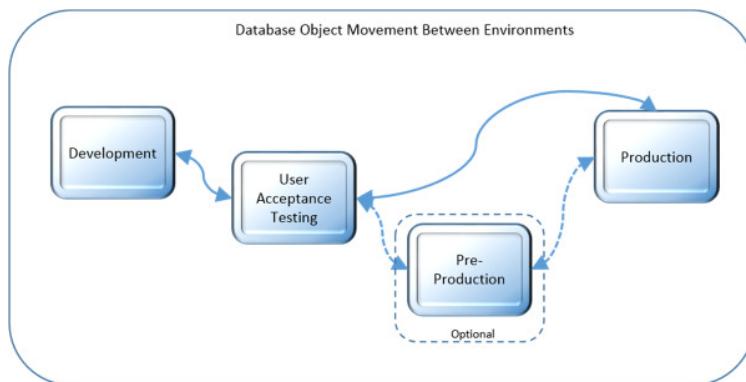
- Stored Procedures
- Table-valued Functions
- Scalar-valued Functions
- Aggregate Functions

The developer typically starts the development cycle by creating a new object or by modifying an existing object on the development server. In the case of an existing object, the developer will normally take a working copy of the artefact from the production server in order to have the latest available source code for the object.

Once all the development has been completed and the object tested in the development environment, it will be moved to the user acceptance testing (UAT) environment where it will be exposed to rigorous testing by a specialized team. Should the object fail in this environment due to a development defect or a possible design flaw, it will be moved back to the development environment for rework.

After a successful UAT outcome, the object or group of objects forming the development project, can be moved to a pre-production environment for a round of sanity testing. This phase of the development cycle is normally done for complex projects with a big impact on the production system. The system is tested at a high level to ensure that all the components act together in the intended manner. The release process to be followed in implementing the changes to the production environment is also tested during this phase. This process is not normally followed for minor releases and is indicated as optional in the diagram below.

On completion of all the development and testing rounds, the objects are released to the production server where basic sanity testing is done by the business users. In the worst of scenarios, a defect in the production environment will require a roll-back to the previous working set of objects, thereby undoing the new implementation. All the objects involved in this scenario will start the cycle from the beginning and go through each phase until a successful production version is created.



The software system described in this document will assist each of the role players in this process to protect the integrity of the programmable database objects and to reduce complexity in releasing objects from one environment to the next.

Assistance is given in the following aspects:

Development phase:

- Checking out of the object on the development server, thereby ensuring a single code base that is protected from changes by other developers.
- Creating a source base from a working copy of the object and creating a backup of the original development version.
- Adding visual comments in the object for others to note the status of the object.
- During the development cycle, the developer can create “Quick Backups” of the object with a single click on a menu. This prevents code losses over extended object development times.

Release to UAT, Pre-Prod or Production phases:

- A release project is created for each release. This is simply a grouping of all objects participating in the new version as intended by the business requirements.
- During the check-out function, the developer marks the object as being part of the release project. This will alleviate the possibility of objects being left out when moving them between the various environments.
- Moving the object grouping (release project) from one environment to the next, will change the status of each of the objects to clearly indicate where the object is within the total cycle. Typically no further development is allowed on an object which is undergoing testing in UAT.
- After a successful release to production, the object’s status will allow the system developers to take ownership of the object in the development environment again, thus starting the cycle all over again.

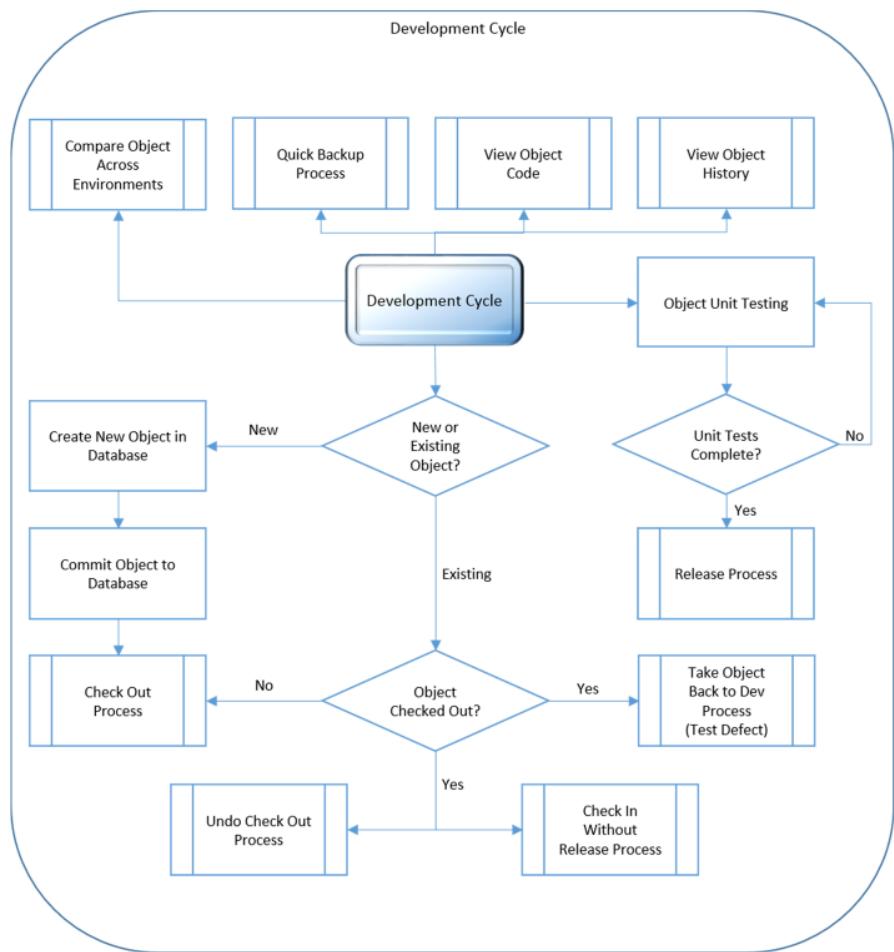
Exceptions to the cycle:

- An object checked out in development might no longer require the suggested changes. The owner (responsible person) of this object, can “undo” the checkout and restore the previous version of the object. The object will become available to other developers wanting to make changes to the object.
- In the case of multiple developers contributing to the changes of the same object, the first developer can check-in the object after making the required changes. The artefact will remain intact and the second developer can take ownership of the object. The remaining process is followed as normal.

Maintenance utilities:

- Compare object across multiple environments by clicking on the compare menu option. A quick visual indication shows whether the object’s code is different in other environments. Clicking the compare toolbar will show the detail of the differences between two environments.
- A comprehensive set of backup and restore functions are available for full or incremental backups across multiple environments.
- A full search in the object definition and object name is available in the system.
- General administrative functions for the software configuration, including user maintenance and access rights, data store setup and server environment setup is also done through this system.

The diagram below shows the system functions available at a high level.



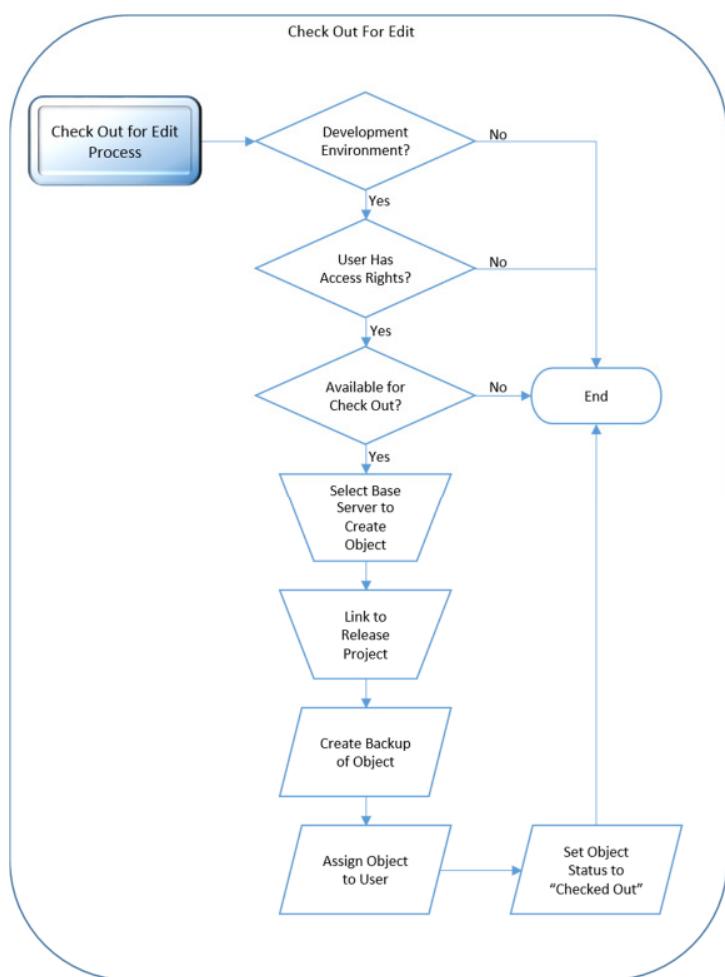
3 Check Out

3.1 Description

Checking the database object out for edit allows the system to protect the artefact from being overwritten in the development environment. The developer assigned to the object during the check-out phase will be responsible for this object until such time as the object is released to the next environment, typically the user acceptance testing area (UAT).

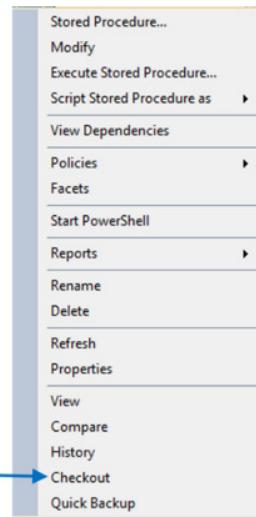
Should the object fail during the UAT phase due to a defect or other reason, the original person responsible can take the object back into the development environment. The object can be modified as if under development, and once the defect has been corrected, the object can be released to the next environment through the release process described later in this document.

3.2 Process Flow

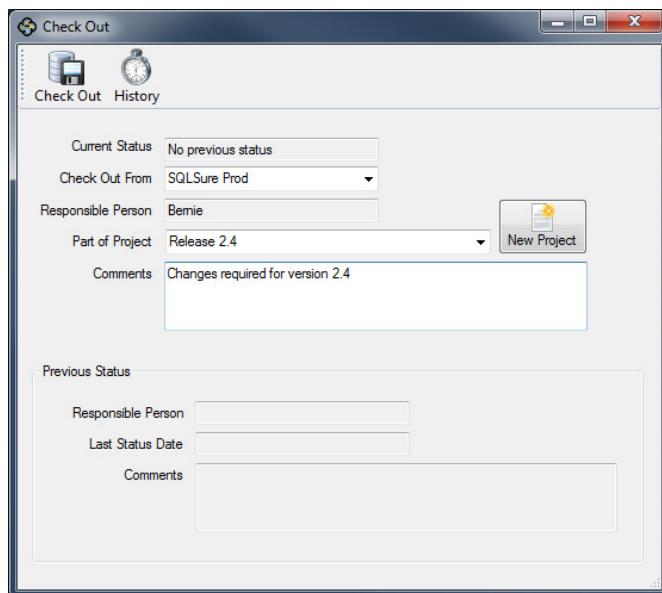


3.3 User Interface

Right-clicking on any programmable database object in SQL Server Management Studio will show the popup menu below:



Selecting the “Checkout” menu option will show the screen below. This functionality is only available on the development server.

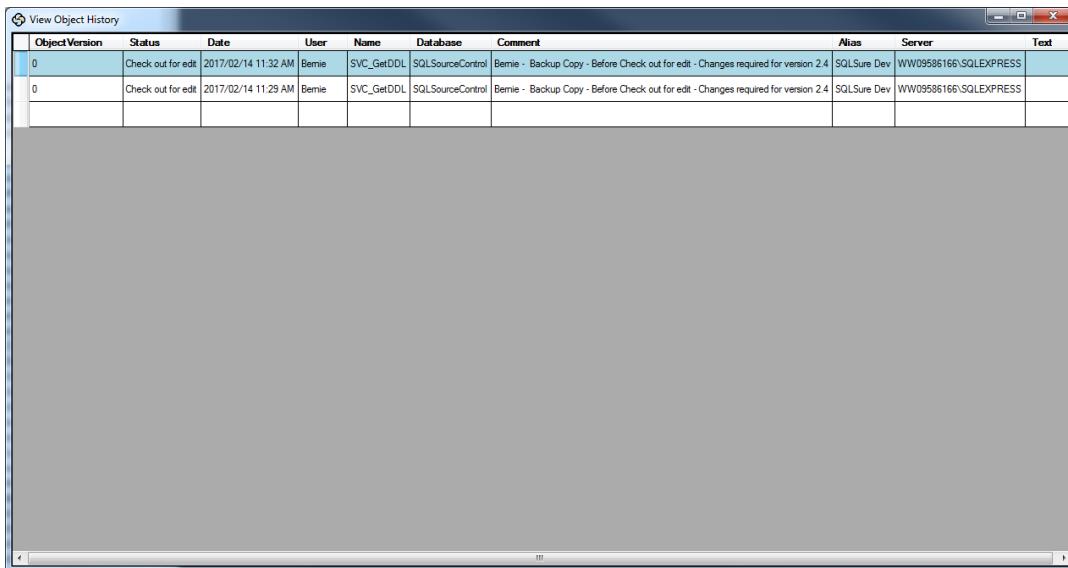


3.4 User Interface Fields

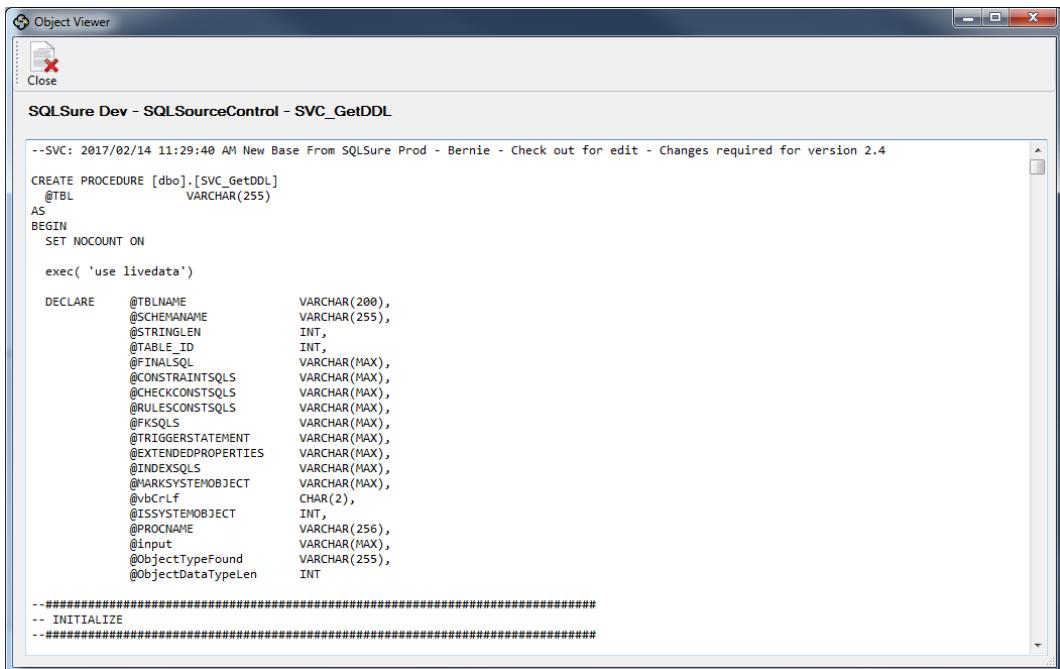
Field	Description	Explanation
Current Status	Displays the last known status of the object	Read only for information.
Check Out From	Select the environment to use as the new source code base for the object.	<ul style="list-style-type: none"> • Select “Use Current Version” to use the version currently on the development server. This is normally for new objects created during development. • For existing objects, it is suggested that you use the production version for the new object base. • A backup copy of the development version will be made before creating the new source base.
Responsible Person	Displays the name person that is about to check the object out.	Read only for information.
Part of Project	Select the project that will be used to release this object. (Described elsewhere in this document)	Select the name of an existing project or click the “New Project” button to create a new release project.
Comments	Enter a descriptive comment for the check-out reason.	This comment will be prepended to the object’s source code.

3.5 User Interface Actions

Toolbar Button	Description	Function
	View history.	Clicking on the “History” tool button will display a list of previous actions performed on this object.



You can click on any of the history items to view the code associated with the specific version of the object.



```
--SVC: 2017/02/14 11:29:40 AM New Base From SQLSure Prod - Bernie - Check out for edit - Changes required for version 2.4
CREATE PROCEDURE [dbo].[SVC_GetDDL]
    @TBL          VARCHAR(255)
AS
BEGIN
    SET NOCOUNT ON

    exec( 'use livedata')

    DECLARE      @TBLNAME        VARCHAR(200),
                @SCHEMANAME     VARCHAR(255),
                @STRINGLEN       INT,
                @TABLE_ID        INT,
                @FINALSQL        VARCHAR(MAX),
                @CONSTRAINTSQLS VARCHAR(MAX),
                @CHECKCONSTSQLS VARCHAR(MAX),
                @RULESCONSTSQLS VARCHAR(MAX),
                @FKSQLS          VARCHAR(MAX),
                @TRIGGERSTATEMENT VARCHAR(MAX),
                @EXTENDEDPROPERTIES VARCHAR(MAX),
                @INDEXSQLS       VARCHAR(MAX),
                @MARKSYSTEMOBJECT VARCHAR(MAX),
                @vbCrLf          CHAR(2),
                @ISSYSTEMOBJECT  INT,
                @PROCNAME        VARCHAR(256),
                @Input           VARCHAR(MAX),
                @ObjectTypeFound VARCHAR(255),
                @ObjectTypeLen   INT

    --#####
    -- INITIALIZE
    --#####

```

Toolbar Button	Description	Function
	Check Out.	<p>Clicking on the “Check Out” tool button will perform the following actions:</p> <ul style="list-style-type: none"> • Create a backup of the current version of the object. This will form part of the item’s history and will be available for viewing through the item’s history. • Create a new base version from the selected environment (UAT, Production, etc.) or use the current development version in the case of selecting “Use Current Version” as the “Check Out From” option. • Prepend the comments of the comments-field to the object’s code. • Assign the object to the person checking the object out. • Update the status of the object to “Check out for edit”.

The table below shows icon displayed in SSMS per status for each environment:

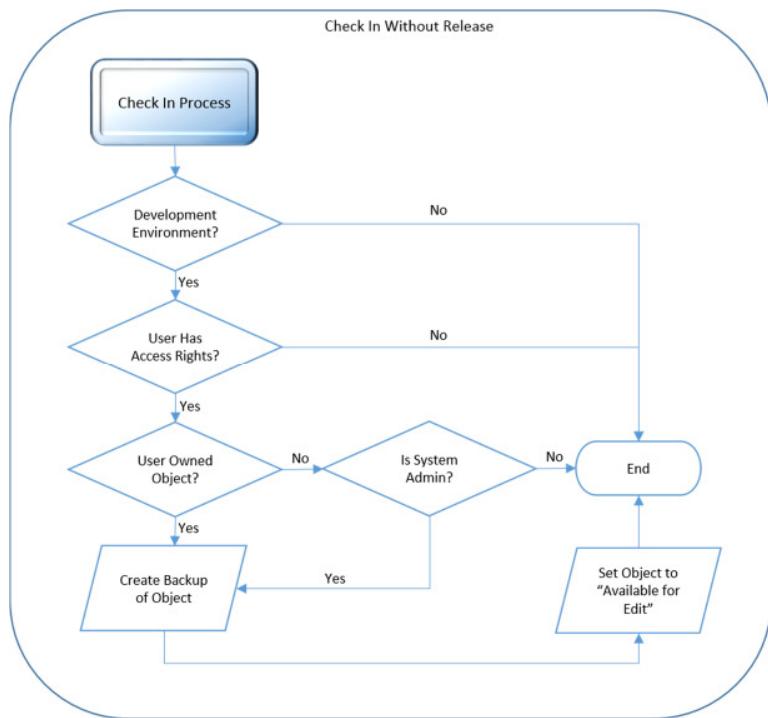
Status	Development	UAT, Pre-Production, Production
Check out for edit (Object Owner)	Available for edit	Caution, object in development cycle
Check out for edit (NOT Object Owner)	Not available for edit	Caution, object in development cycle
Released to UAT (Check out user)	Not available for edit (Can take object back)	Caution, object in development cycle
Released to Pre-Prod (Check out user)	Not available for edit (Can take object back)	Caution, object in development cycle
Released to UAT (NOT Check out user)	Not available for edit	Caution, object in development cycle
Released to Pre-Prod (NOT Check out user)	Not available for edit	Caution, object in development cycle
Released to Production	Available for edit	Available for edit
Check In	Available for edit	Available for edit
Check-out Undo	Available for edit	Available for edit
Backup	Available for edit	Available for edit

4 Check In

4.1 Description

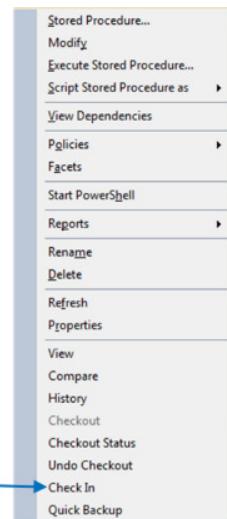
In the case of multiple developers contributing to the changes of the same object, the first developer can check-in the object after making the required changes. The artefact will remain intact and the second developer can take ownership of the object. The remaining process is followed as normal.

4.2 Process Flow

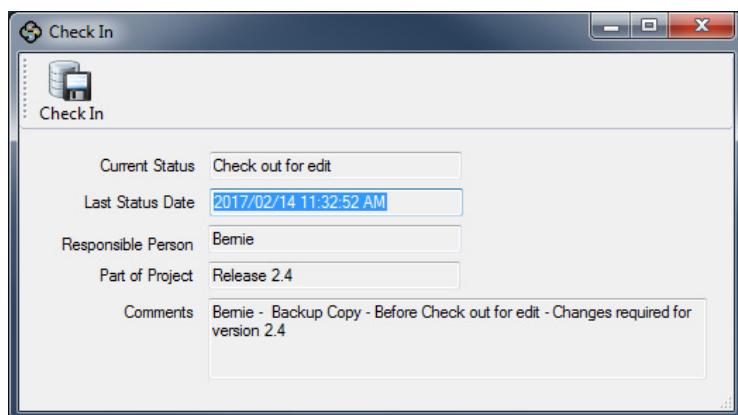


4.3 User Interfaces

Right-clicking on any programmable database object in SQL Server Management Studio will show the popup menu below:



Selecting the “Check In” menu option will show the screen below. This functionality is only available on the development server.



4.4 User Interface Fields

Field	Description	Explanation
Current Status	Displays the last known status of the object	Read only for information.
Last Status Date	The date of the last status change of the object.	Read only for information.
Responsible Person	Displays the name person that is about to check the object out.	Read only for information.
Part of Project	Displays the project that will be used to release this object.	Read only for information.
Comments	Shows the descriptive comment for the previous status	Read only for information.

4.5 User Interface Actions

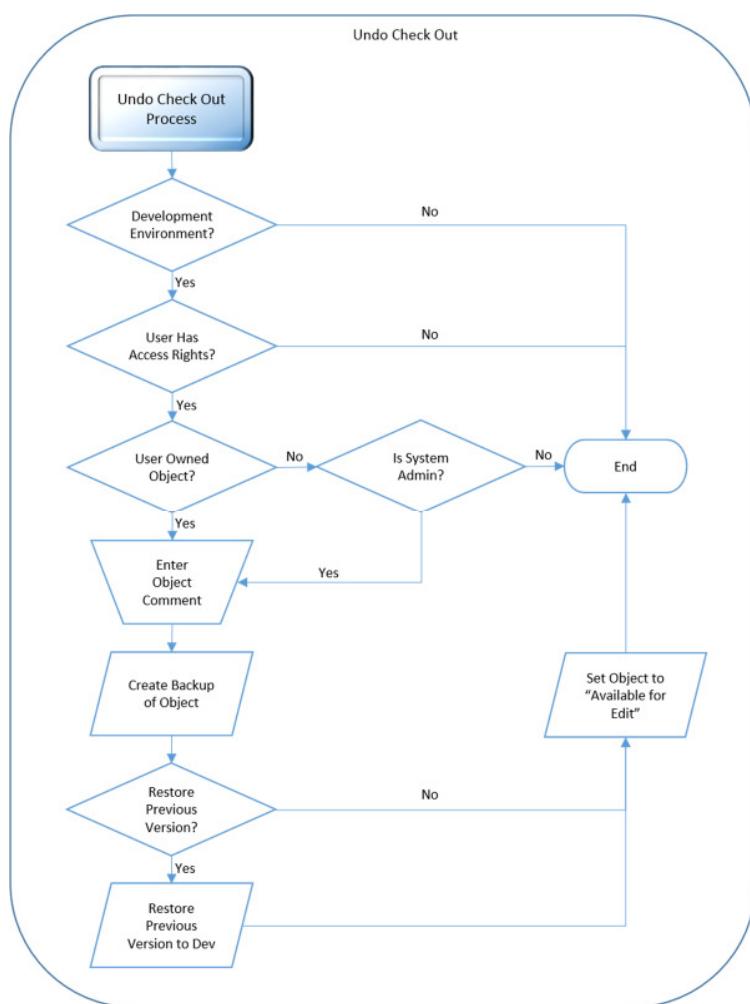
Toolbar Button	Description	Function
	Check In.	<p>Clicking on the “Check In” tool button will perform the following actions:</p> <ul style="list-style-type: none">• Prepend the comments of the comments-field to the object’s code.• Remove the assigned person from the object.• Update the status of the object to “Check in” – the object will now be available for editing.

5 Undo Check Out

5.1 Description

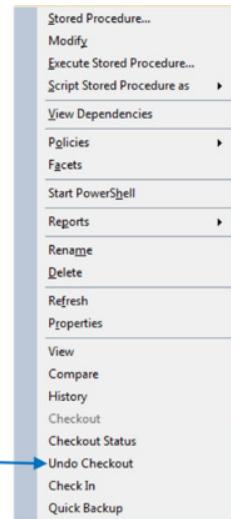
An object checked out in development might no longer require the suggested changes. The owner (responsible person) of this object, can “undo” the checkout and optionally restore the previous version of the object. The object will become available to other developers wanting to make changes to the object.

5.2 Process Flow

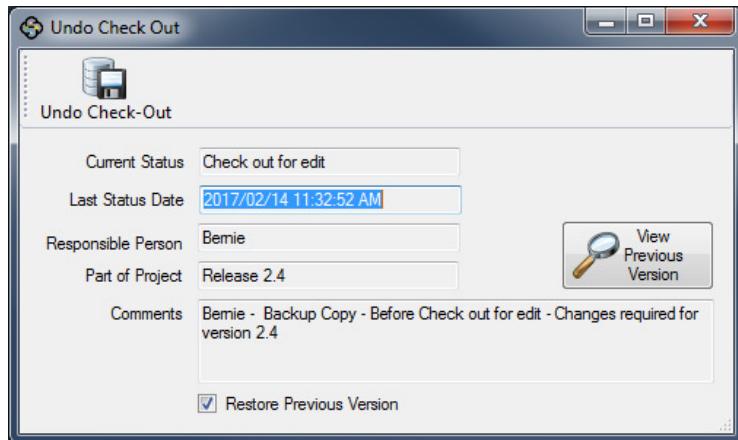


5.3 User Interfaces

Right-clicking on any programmable database object in SQL Server Management Studio will show the popup menu below:



Selecting the “Undo Checkout” menu option will show the screen below. This functionality is only available on the development server.



5.4 User Interface Fields

Field	Description	Explanation
Current Status	Displays the last known status of the object	Read only for information.
Last Status Date	The date of the last status change of the object.	Read only for information.
Responsible Person	Displays the name person that is about to check the object out.	Read only for information.
Part of Project	Displays the project that will be used to release this object.	Read only for information.
Comments	Shows the descriptive comment for the previous status	Read only for information.
Restore Previous Version	Selected = Yes, not selected = No	Selecting this option will restore the version of the object as it was during the check-out phase. Not selecting this option will leave the current version intact. Both options will set the object's status to allow for editing.

5.5 User Interface Actions

Toolbar Button	Description	Function
	Undo Check-Out.	<p>Clicking on the “Undo Check-Out” tool button will perform the following actions:</p> <ul style="list-style-type: none"> Prepend the comments of the comments-field to the object’s code. Remove the assigned person from the object. Restore the version of the object as it was during the check-out phase, if this option was selected. Update the status of the object to “Undo check out” – the object will now be available for editing.

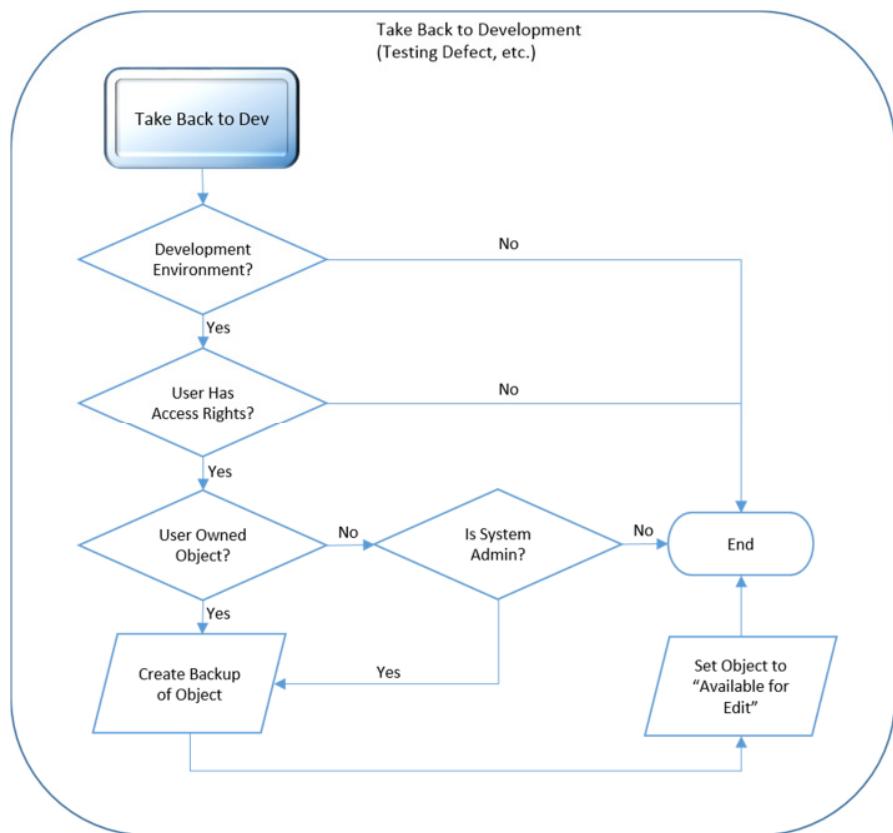
6 Take Object Back to Development

6.1 Description

Once all the development has been completed and the object tested in the development environment, it will be moved to the user acceptance testing (UAT) environment where it will be exposed to rigorous testing by a specialized team.

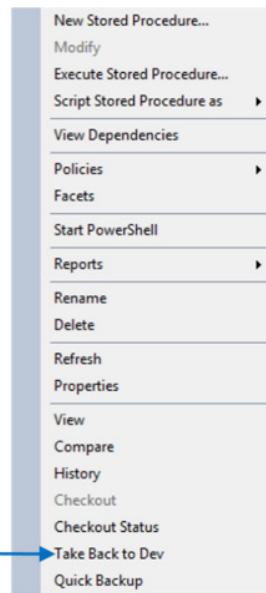
Should the object fail in this environment due to a development defect or a possible design flaw, it will be moved back to the development environment for rework.

6.2 Process Flow

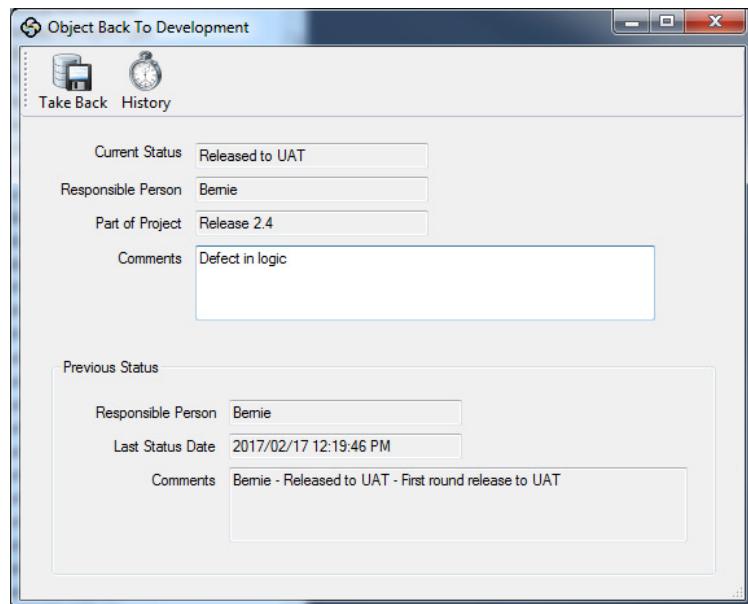


6.3 User Interfaces

Right-clicking on any programmable database object in SQL Server Management Studio will show the popup menu below:



Selecting the “Take Back to Dev” menu option will show the screen below. This functionality is only available on the development server.



6.4 User Interface Fields

Field	Description	Explanation
Current Status	Displays the last known status of the object	Read only for information.
Responsible Person	Displays the name person that is about to check the object out.	Read only for information.
Part of Project	Displays the project that will be used to release this object.	Read only for information.
Comments	Enter a descriptive comment for the reason for taking the object back to development.	

6.5 User Interface Actions

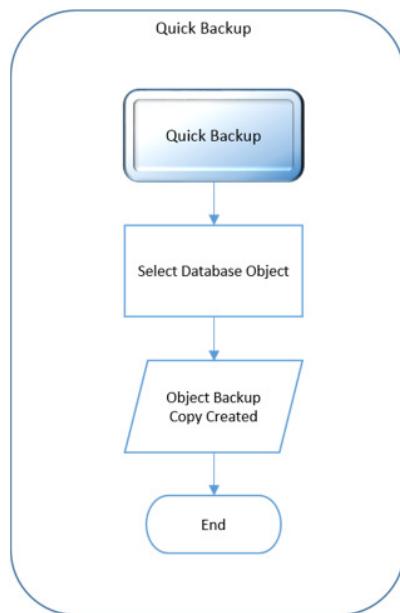
Toolbar Button	Description	Function
	Take Back.	<p>Clicking on the “Take Back” tool button will perform the following actions:</p> <ul style="list-style-type: none">• Prepend the comments of the comments-field to the object’s code.• Make a backup copy of the current development version.• Update the status of the object to “Check out for edit” – the object will now be available for editing.

7 Quick Backup

7.1 Description

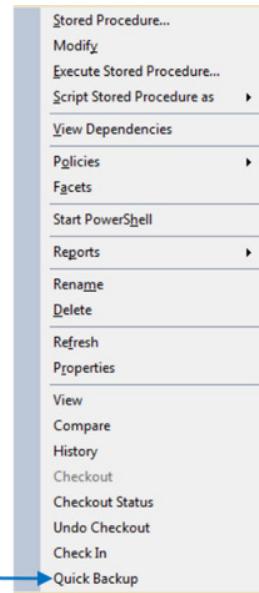
During the development cycle, the developer can create “Quick Backups” of the object with a single click on a menu. This prevents code losses over extended object development times. This functionality is also available in all other environments.

7.2 Process Flow

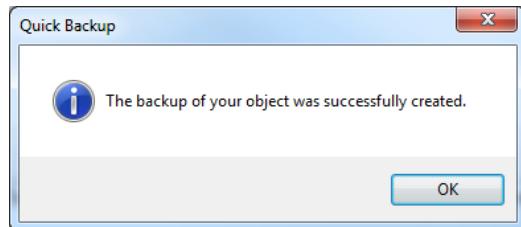


7.3 User Interfaces

Right-clicking on any programmable database object in SQL Server Management Studio will show the popup menu below:



No additional user interface is shown apart from a confirmation message that the backup of the selected object was successful.

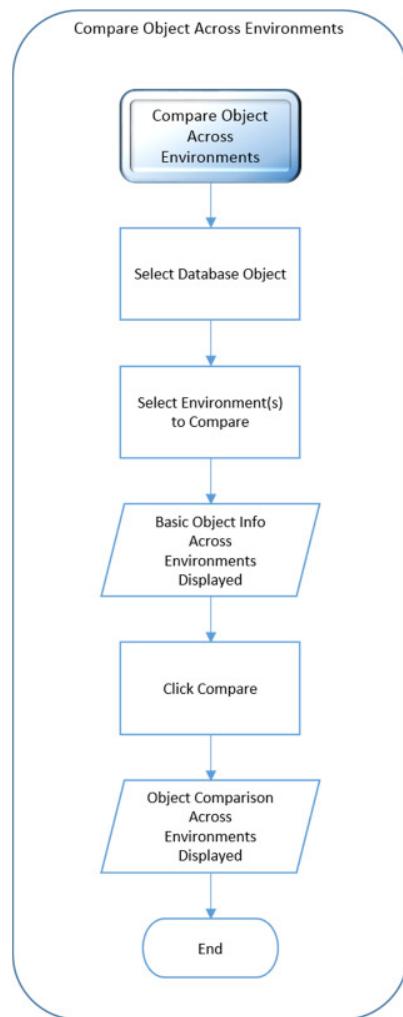


8 Compare

8.1 Description

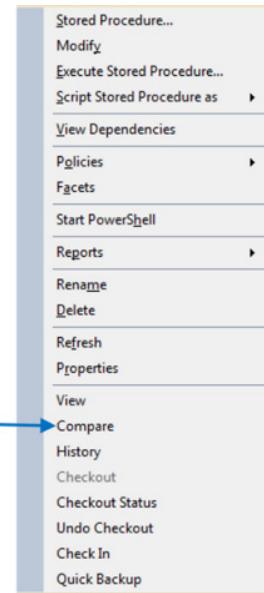
Compare object across multiple environments by clicking on the compare menu option. A quick visual indication shows whether the object's code is different in other environments. Clicking the compare toolbar will show the detail of the differences between two environments.

8.2 Process Flow

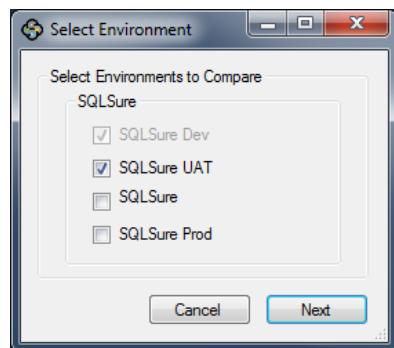


8.3 User Interfaces

Right-clicking on any programmable database object in SQL Server Management Studio will show the popup menu below:



Selecting the “Compare” menu option will show the screen below. This functionality is available in all environments



After selecting one or more servers, a basic visual compare screen is shown.

A screenshot of the "Compare Objects" interface. The window has two panes. The left pane is for "SQLSure Dev" and the right pane is for "SQLSure UAT". Both panes show details for a stored procedure named "SVC_GetDDL".

SQLSure Dev		SQLSure UAT	
Database	SQLSourceControl	Database	SQLSourceControl
Object	SVC_GetDDL	Object	SVC_GetDDL
Type	Stored Procedure	Type	Stored Procedure
Date Created	2017/02/14 11:29:40 AM	Date Created	2017/02/14 11:24:55 AM
Date Modified	2017/02/14 11:29:40 AM	Date Modified	2017/02/14 11:24:55 AM
Status	Check out for edit	Status	
User	Bernie	User	
Status Date	2017/02/14 11:29:40 AM	Status Date	
Comment	Bernie - Backup Copy - Before Check out for edit - Changes required for version 2.4	Comment	

Selecting the “Compare” toolbar button, the following detail compare is shown:

WinMerge - [SQLSure Dev - SQLSure UAT]

File Edit View Merge Tools Plugins Window Help

SQLSure Dev SQLSure UAT

SQlSure Dev - SVC: 2017/02/14 11:29:40 AM New Base From SQLSure Prod - Bernie - Check out for edit - Chang

```
CREATE PROCEDURE [dbo].[SVC_GetDDL]
    @TBLNAME          VARCHAR(255)
AS
BEGIN
    SET NOCOUNT ON

    exec('use liveidata')

    DECLARE    @TBLNAME           VARCHAR(200),
                @SCHEMANAME        VARCHAR(255),
                @STRINGLEN          INT,
                @TABLE_ID           INT,
                @FINALSQL           VARCHAR(MAX),
                @CONSTRAINTSQLS    VARCHAR(MAX),
                @CHECKCONSTSQLS    VARCHAR(MAX),
                @RULESCONSTSQLS    VARCHAR(MAX),
                @TRIGGERSQL         VARCHAR(MAX),
                @TRIGGERSTATEMENT  VARCHAR(MAX),
                @EXTENDEDPROPERTIES VARCHAR(MAX),
                @INDEXSQLS          VARCHAR(MAX),
                @MARKSYSTEMOBJECT   VARCHAR(MAX),
                @vbCrLf              CHAR(2),
                @ISSYSTEMOBJECT     INT,
                @PROCNAME            VARCHAR(256),
                @input               VARCHAR(MAX),
                @ObjectTypeFound    VARCHAR(255),
                @ObjectDataTypelen  INT

--#####
-- INITIALIZE
--#####
SET @input = ''
-- new code: determine whether this proc is marked as a system proc with sp_ms_marksystemobject
-- which flips the is_ms_shipped bit in sys.objects
-- SELECT @ISSYSTEMOBJECT = ISNULL(is_ms_shipped,0),@PROCNAME = ISNULL(name,'SVC_GetDDL') FROM sys.objects
IF @ISSYSTEMOBJECT IS NULL
```

SQLSure UAT

```
create PROCEDURE [dbo].[SVC_GetDDL]
    @TBLNAME          VARCHAR(255)
AS
BEGIN
    SET NOCOUNT ON

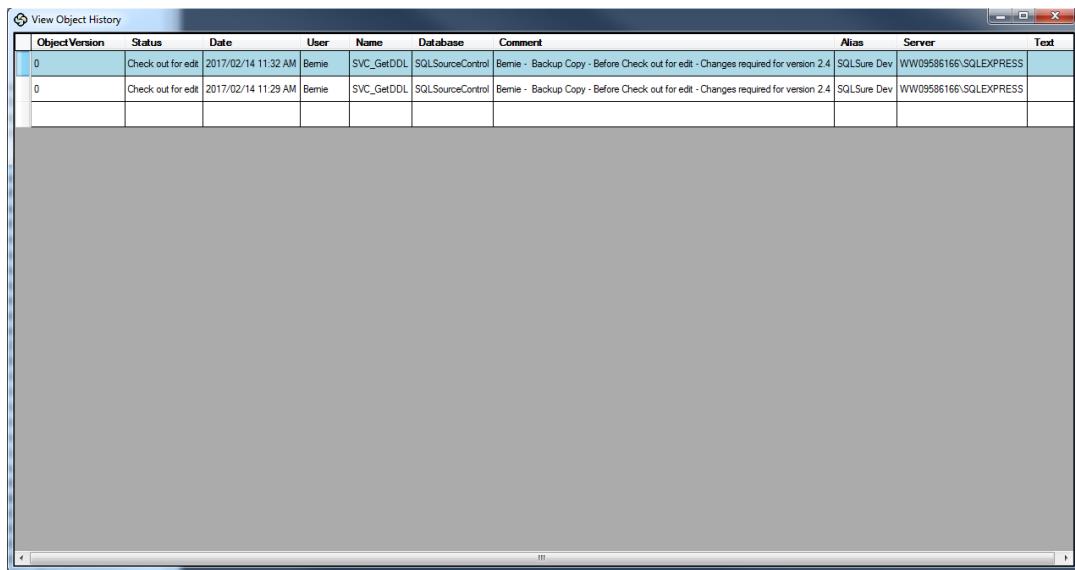
    exec(' use liveidata')

    DECLARE    @TBLNAME           VARCHAR(200),
                @SCHEMANAME        VARCHAR(255),
                @STRINGLEN          INT,
                @TABLE_ID           INT,
                @FINALSQL           VARCHAR(MAX),
                @CONSTRAINTSQLS    VARCHAR(MAX),
                @CHECKCONSTSQLS    VARCHAR(MAX),
                @RULESCONSTSQLS    VARCHAR(MAX),
                @TRIGGERSQL         VARCHAR(MAX),
                @TRIGGERSTATEMENT  VARCHAR(MAX),
                @EXTENDEDPROPERTIES VARCHAR(MAX),
                @INDEXSQLS          VARCHAR(MAX),
                @MARKSYSTEMOBJECT   VARCHAR(MAX),
                @vbCrLf              CHAR(2),
                @ISSYSTEMOBJECT     INT,
                @PROCNAME            VARCHAR(256),
                @input               VARCHAR(MAX),
                @ObjectTypeFound    VARCHAR(255),
                @ObjectDataTypelen  INT

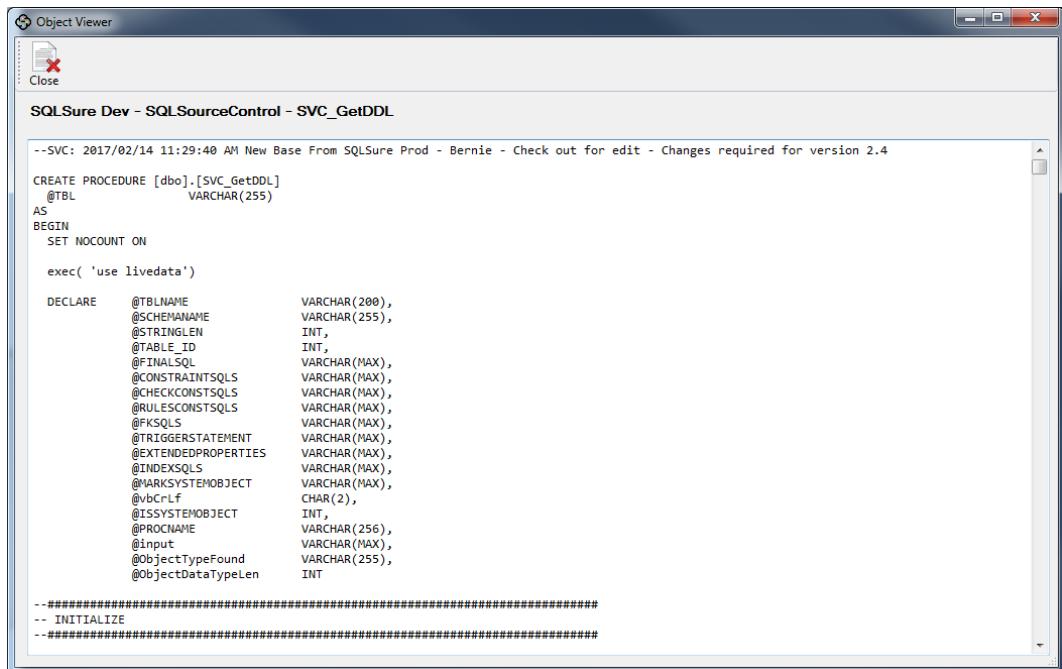
--#####
-- INITIALIZE
--#####
SET @input = ''
-- new code: determine whether this proc is marked as a system proc with sp_ms_marksystemobject
-- which flips the is_ms_shipped bit in sys.objects
-- SELECT @ISSYSTEMOBJECT = ISNULL(is_ms_shipped,0),@PROCNAME = ISNULL(name,'SVC_GetDDL') FROM sys.objects
IF @ISSYSTEMOBJECT IS NULL
```

8.4 User Interface Actions

Toolbar Button	Description	Function
	View history.	Clicking on the “History” tool button will display a list of previous actions performed on this object.



You can click on any of the history items to view the code associated with the specific version of the object.



The screenshot shows the SQLSure Object Viewer window with the title "SQLSure Dev - SQLSourceControl - SVC_GetDDL". The window displays the T-SQL code for the stored procedure. The code includes declarations for parameters (@TBLNAME, @SCHEMANAME, etc.), variables (@STRINGLEN, @TABLE_ID, etc.), and local variables (@FINALSQL, @CONSTRAINTSQLS, etc.). It also contains logic for setting NOCOUNT ON, executing a USE statement, and handling triggers and extended properties. A note at the bottom indicates the code is for version 2.4.

```
--SVC: 2017/02/14 11:29:40 AM New Base From SQLSure Prod - Bernie - Check out for edit - Changes required for version 2.4
CREATE PROCEDURE [dbo].[SVC_GetDDL]
    @TBL          VARCHAR(255)
AS
BEGIN
    SET NOCOUNT ON

    exec( 'use livedata')

    DECLARE      @TBLNAME        VARCHAR(200),
                @SCHEMANAME     VARCHAR(255),
                @STRINGLEN       INT,
                @TABLE_ID        INT,
                @FINALSQL        VARCHAR(MAX),
                @CONSTRAINTSQLS VARCHAR(MAX),
                @CHECKCONSTSQLS VARCHAR(MAX),
                @RULESCONSTSQLS VARCHAR(MAX),
                @FKSQLS          VARCHAR(MAX),
                @TRIGGERSTATEMENT VARCHAR(MAX),
                @EXTENDEDPROPERTIES VARCHAR(MAX),
                @INDEXSQLS       VARCHAR(MAX),
                @MARKSYSTEMOBJECT VARCHAR(MAX),
                @vbCrLf          CHAR(2),
                @ISSYSTEMOBJECT  INT,
                @PROCNAME         VARCHAR(256),
                @Input            VARCHAR(MAX),
                @objectTypeFound  VARCHAR(255),
                @ObjectTypeDataLen INT

    --#####
    -- INITIALIZE
    --#####

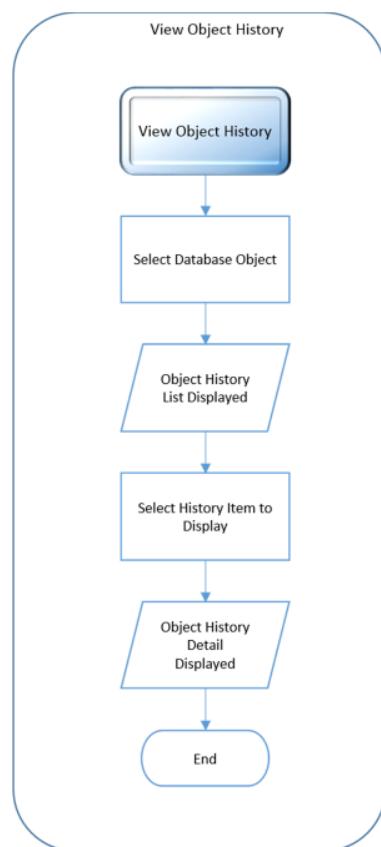
```

9 View History

9.1 Description

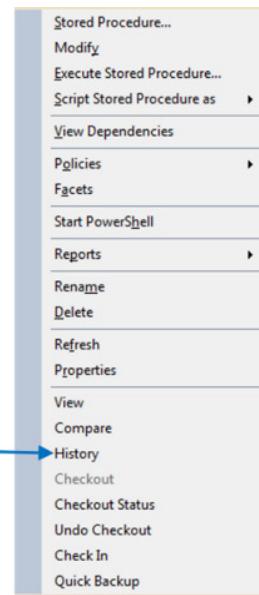
The history of an object will display a list of previous actions performed on this object.

9.2 Process Flow



9.3 User Interfaces

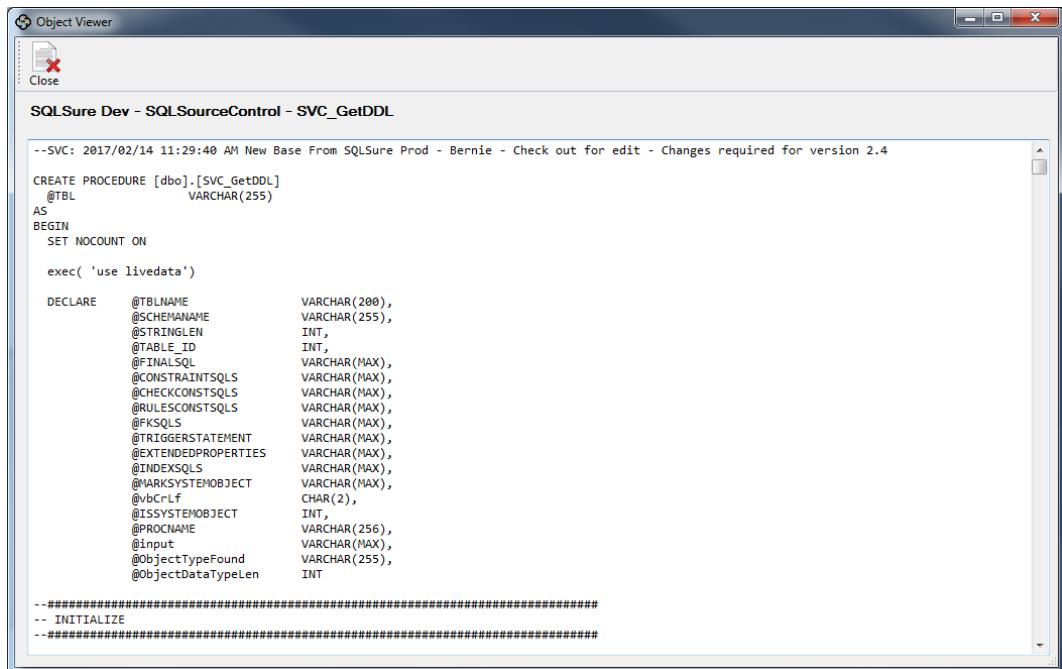
Right-clicking on any programmable database object in SQL Server Management Studio will show the popup menu below:



Selecting the “History” menu option will show the screen below. This functionality is available in all environments.

ObjectVersion	Status	Date	User	Name	Database	Comment	Alias	Server	Text
0	Check out for edit	2017/02/14 11:32 AM	Bernie	SVC_GetDDL	SQLSourceControl	Bernie - Backup Copy - Before Check out for edit - Changes required for version 2.4	SQLSure Dev	WW09586166\SQLEXPRESS	
0	Check out for edit	2017/02/14 11:29 AM	Bernie	SVC_GetDDL	SQLSourceControl	Bernie - Backup Copy - Before Check out for edit - Changes required for version 2.4	SQLSure Dev	WW09586166\SQLEXPRESS	

You can click on any of the history items to view the code associated with the specific version of the object.



The screenshot shows a Windows application window titled "Object Viewer". Inside, there's a tab labeled "SQLSure Dev - SQLSourceControl - SVC_GetDDL". The main pane displays the following T-SQL code:

```
--SVC: 2017/02/14 11:29:40 AM New Base From SQLSure Prod - Bernie - Check out for edit - Changes required for version 2.4
CREATE PROCEDURE [dbo].[SVC_GetDDL]
    @TBL          VARCHAR(255)
AS
BEGIN
    SET NOCOUNT ON

    exec( 'use liveData')

    DECLARE      @TBLNAME        VARCHAR(200),
                @SCHEMANAME     VARCHAR(255),
                @STRINGLEN       INT,
                @TABLE_ID        INT,
                @FINALSQL        VARCHAR(MAX),
                @CONSTRAINTSQLS VARCHAR(MAX),
                @CHECKCONSTSQLS VARCHAR(MAX),
                @RULESCONSTSQLS VARCHAR(MAX),
                @FKSQLS          VARCHAR(MAX),
                @TRIGGERSTATEMENT VARCHAR(MAX),
                @EXTENDEDPROPERTIES VARCHAR(MAX),
                @INDEXSQLS       VARCHAR(MAX),
                @MARKSYSTEMOBJECT VARCHAR(MAX),
                @vbCrLf          CHAR(2),
                @ISSYSTEMOBJECT  INT,
                @PROCNAME        VARCHAR(256),
                @Input           VARCHAR(MAX),
                @objectTypeFound VARCHAR(255),
                @objectdatatypeLen INT

    --#####
    -- INITIALIZE
    --#####

```

10 Project Maintenance

10.1 Description

A release project is created for each release. This is simply a grouping of all objects participating in the new version as intended by the business requirements. During the check-out function, the developer marks the object as being part of the release project. This will alleviate the possibility of objects being left out when moving them between the various environments.

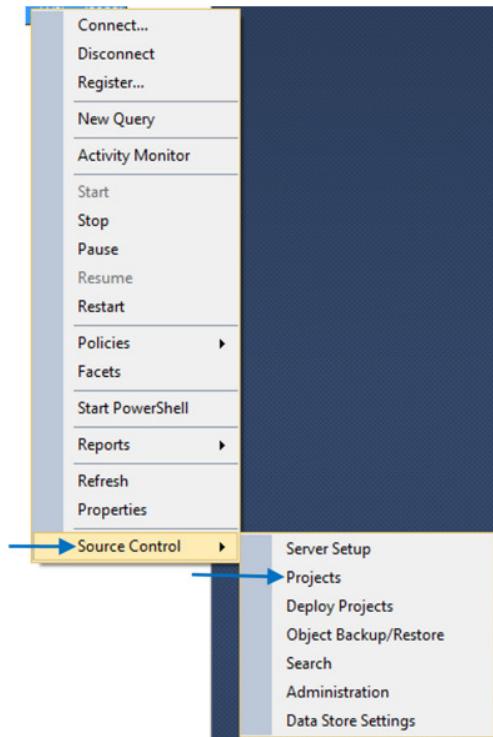
Moving the object grouping (release project) from one environment to the next, will change the status of each of the objects to clearly indicate where the object is within the total cycle. Typically no further development is allowed on an object which is undergoing testing in UAT. After a successful release to production, the object's status will allow the system developers to take ownership of the object in the development environment again, thus starting the cycle all over again.

10.2 Process Flow

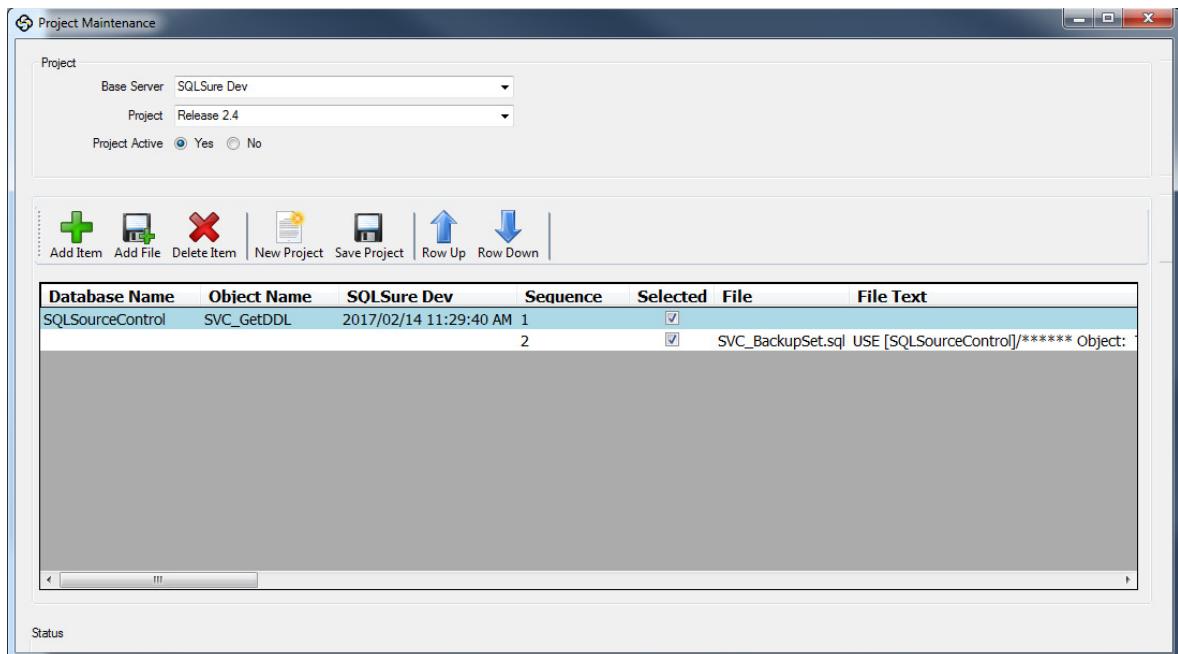


10.3 User Interfaces

Right-clicking on any database or on the top level root item in the object explorer in SQL Server Management Studio, will show the popup menu below:



Selecting the “Projects” menu option will show the screen below.



10.4 User Interface Fields

Field	Description	Explanation
Base Server	Select the base server from the list.	Selecting the base server sets the project's server group.
Project	Select an existing project.	The unique name defined for this release project.
Project Active	Setting the project inactive will remove it from the visible list of projects.	All relevant project details will remain, even if the project is set to Active = No.

10.5 User Interface Actions

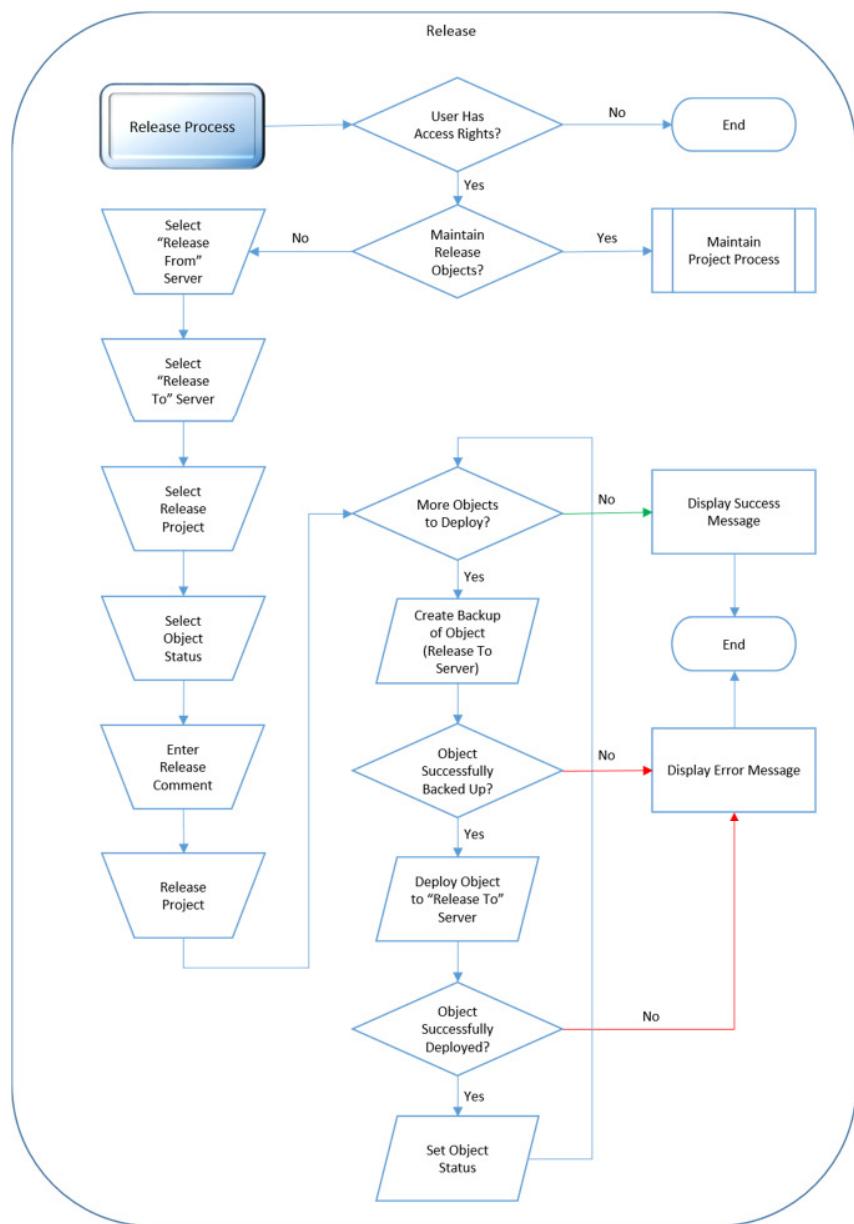
Toolbar Button	Description	Function
	Add new database object.	The “Add database object” dialog screen will be shown. Select the database name followed by the object name.
	Add new file script object.	The “Add file object” dialog screen will be shown. Select the file location in this dialog.
	Remove database or file object from the project.	Remove the item from the project.
	Create new project.	Create new project. See the process for creating a new release project elsewhere in this document.
	Save release project.	Commits the release project to the data store. All script file objects will also be stored in a database table.
	Move row up.	If the release order of the object is important, use this function to move the object up in the release order. Typically “table create” or data scripts are run before object modification scripts.
	Move row down.	See item above.

11 Release

11.1 Description

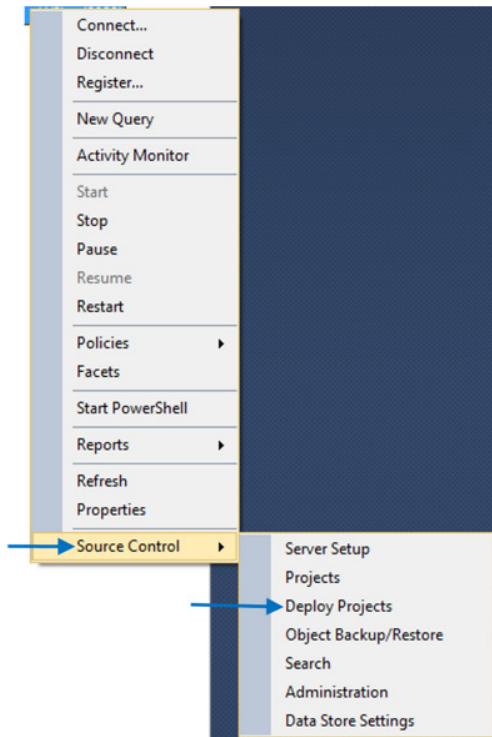
The release process is the process of moving one or more objects grouped as a “release project” from one environment to the next.

11.2 Process Flow

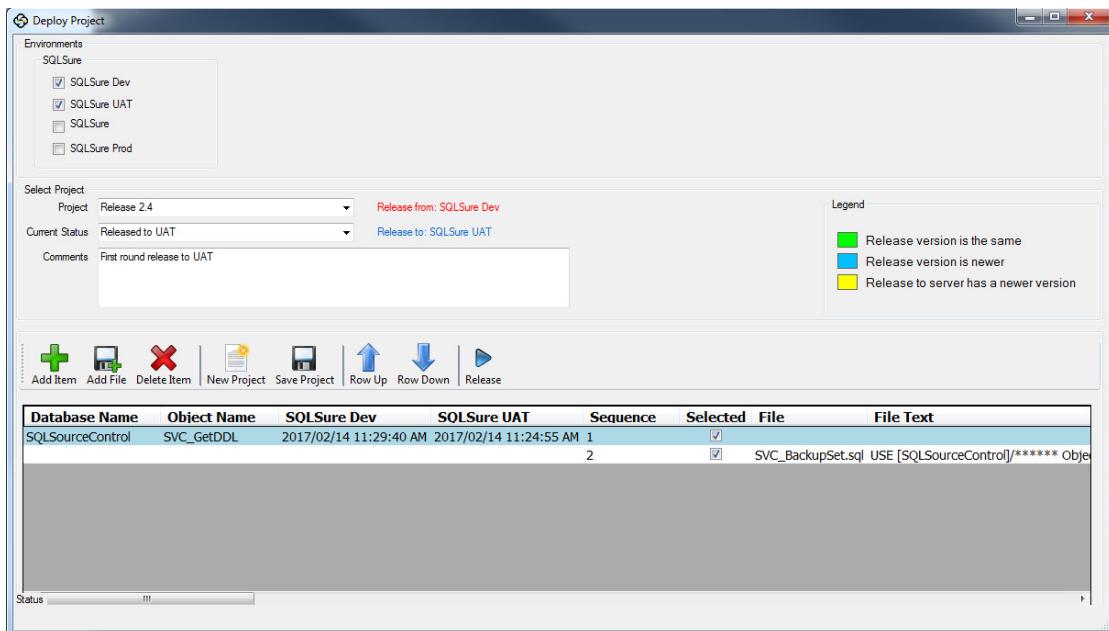


11.3 User Interfaces

Right-clicking on any database or on the top level root item in the object explorer in SQL Server Management Studio, will show the popup menu below:



Selecting the “Deploy Projects” menu option will show the screen below.



11.4 User Interface Actions

Toolbar Button	Description	Function
	Release the objects.	<p>The following system functions are performed when releasing an object:</p> <ul style="list-style-type: none">• Create a backup of the object on the target server.• Add release comments to the object's code.• Deploy the new or modified object to the target server.• Set the object's status.

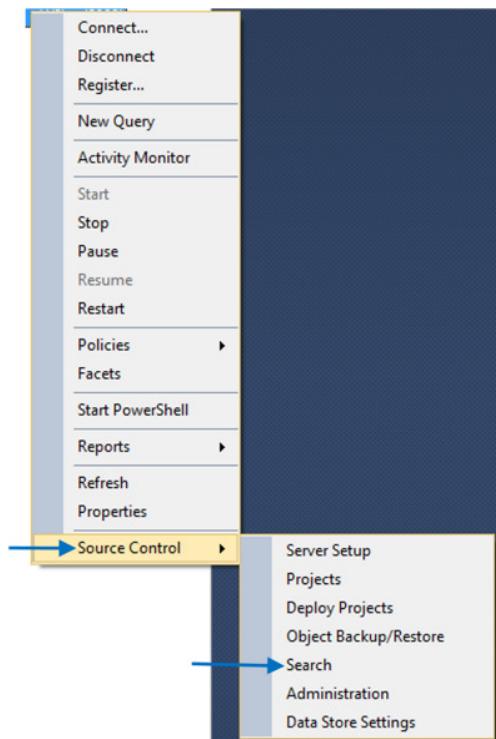
12 Search

12.1 Description

The search function does a full search on the object definition and object name as specified in the search criteria.

12.2 User Interfaces

Right-clicking on any database or on the top level root item in the object explorer in SQL Server Management Studio, will show the popup menu below:



Selecting the “Search” menu option will show the screen below.

The screenshot shows the SQL Sure application interface. At the top, there is a search bar with the text "FunctionName" and a "Go" button. Below the search bar is a table with the following columns: Object Name, Schema, Database, Type, Matches On, and Object Detail. The table contains three rows:

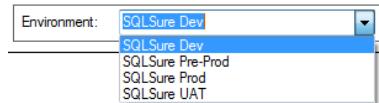
Object Name	Schema	Database	Type	Matches On	Object Detail
SVC_Security_InsertUpdateRoleFunction	dbo	SQLSourceControl	Stored Procedure	Content	CREATE PROCEDURE [dbo].[SVC_Security_InsertUpdateRoleFunction]@RoleName varchar(50),@FunctionName varchar(50),@SystemUser varchar(50)ASBEGINSET NOCOUNT ON
SVC_Security_SystemRoleHasFunction	dbo	SQLSourceControl	Stored Procedure	Content	CREATE PROCEDURE [dbo].[SVC_Security_SystemRoleHasFunction]@RoleName varchar(50),@FunctionName varchar(50)ASBEGINSET NOCOUNT ONselect SR.RoleName, SF.Fu
SVC_SystemFunction	dbo	SQLSourceControl	User Table	Column	SystemFunctionId int, FunctionName varchar(50)

Below the table, a specific stored procedure is selected: "Stored Procedure SVC_Security_InsertUpdateRoleFunction". The code for this procedure is displayed in a large text area:

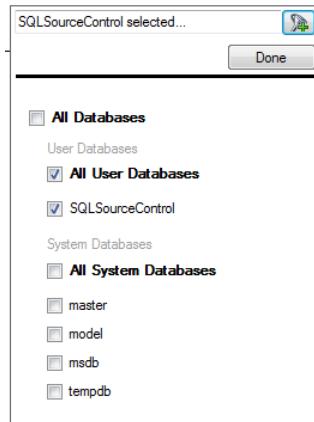
```
CREATE PROCEDURE [dbo].[SVC_Security_InsertUpdateRoleFunction]
    @RoleName varchar(50),
    @FunctionName varchar(50),
    @SystemUser varchar(50)
AS
BEGIN
    SET NOCOUNT ON
    declare @TmpStr varchar(100)
    declare @RoleId int
    declare @FunctionId int
    if (select count(*) from SVC_RoleFunction as RF INNER JOIN
        SVC_SystemRole as SR ON RF.RoleId = SR.RoleId INNER JOIN
        SVC_SystemFunction as SF ON RF.SystemFunctionId = SF.SystemFunctionId
        where SR.RoleName = @RoleName
        and SF.FunctionName = @FunctionName) = 0
    begin
        --Insert
    end
    else
        --Update
    end
END
```

Setting the search criteria

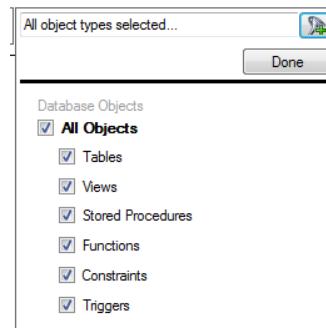
Environment:



Databases to include:



Object types to include:



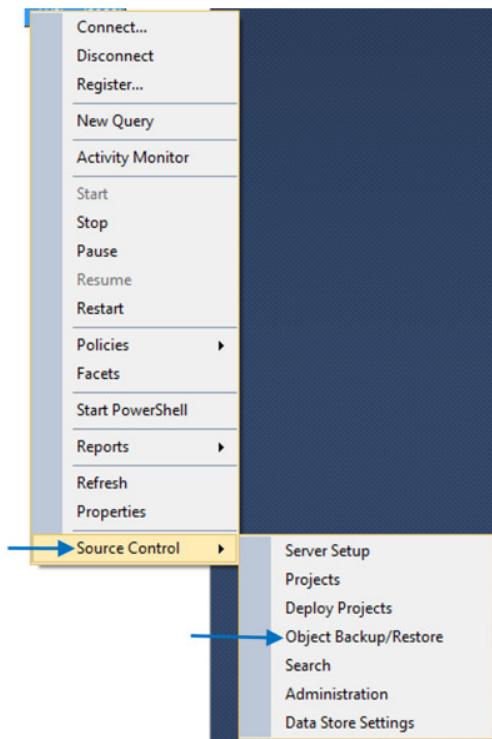
13 Backup and Restore

13.1 Description

The system includes a comprehensive set of backup and restore functions for full or incremental backups across multiple environments.

13.2 User Interfaces

Right-clicking on any database or on the top level root item in the object explorer in SQL Server Management Studio, will show the popup menu below:



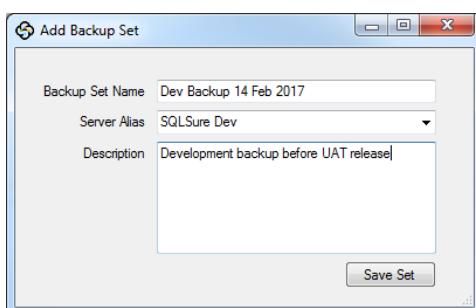
Selecting the “Object Backup/Restore” menu option will show the screen below.

Select the “Backup” tab for the backup function.

The screenshot shows the SQLSure Backup and Restore application window. The title bar reads "Backup and Restore". The main menu bar has "Backup" and "Restore" tabs, with "Backup" currently selected. The toolbar includes icons for New Set, Save Set, Select All, De-Select All, Clear Grid, Get Objects, and Backup. On the left, there's a "Set Details" panel with fields for Backup Set Name (Dev Backup 14 Feb 2017), Server (SQLSure Dev), Backup Set Description (Development backup before UAT release), Create Date (2017/02/14 11:42:13 AM), and Set Active (Yes). A "Databases" list on the right includes master, model, msdb, SQLSourceControl (selected with a checked checkbox), and tempdb. An "Object Types" section on the far right lists Stored Procedures, Functions, Tables, Views, and Triggers, with all checkboxes checked. Below these are sections for "Set Criteria (Objects modified since)" (All selected) and "and with status" (Include All). The main grid area is titled "Objects" and contains a table with columns: Database, Name, Type, User, Status, and Text. The table lists various objects from the SQLSourceControl schema, such as SVC_BackupSet, SVC_BackupSetDatabase, SVC_BackupSetObject, SVC_BackupSetObjectType, SVC_ExternalChangeCheck, SVC_FileObject, SVC_GetBackupSetDatabase, SVC_GetBackupSetObjectType, SVC_GetDDL, SVC_GetProjectObjects, SVC_GetRestoreSetObject, SVC_GetServerRole, and SVC_LU_ObjectStatus. The "Text" column shows the corresponding CREATE TABLE or stored procedure definitions. At the bottom of the grid, there's a message: "Done - Success count: 72, Fail count: 1".

13.3 User Interface Actions

Toolbar Button	Description	Function
	Add new backup set.	Adds a new backup set definition, see the user interface below.
	Save backup set definition.	Saves the backup set definition.
	Select all databases.	Select all available databases for the backup operation.
	De-select all databases.	De-select all selected databases for the backup operation.
	Clear the grid.	Clear the object grid.
	Get all affected objects.	Fetch all the objects specified by the set criteria.
	Backup objects.	Save all objects in the grid in the database.



Select the “Restore” tab for the restore functions.

Database	Name	Type	User	Status	Backup Date	Text
SQLSourceControl	SVC_ExternalChangeCheck	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_GetBackupSetDatabase	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_GetBackupSetObjectType	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_GetDDL	Stored Procedure	Bernie	Check out for edit	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_GetProjectObjects	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_GetRestoreSetObject	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_GetServerRole	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_ObjectCreate	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_ObjectGetVersionAtCheckOut	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_ObjectSaveHistory	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_SaveBackupSet	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie
SQLSourceControl	SVC_SaveBackupSetDatabase	Stored Procedure	Bernie	Backup	2017/02/14 11:43:04 AM	-- ====== Author:Bernie

13.4 User Interface Actions

Toolbar Button	Description	Function
	Restore the backup set.	Restore all objects specified by the filter criteria, listed in the detail grid. Each item affected will have a backup created of the existing object before being restored.

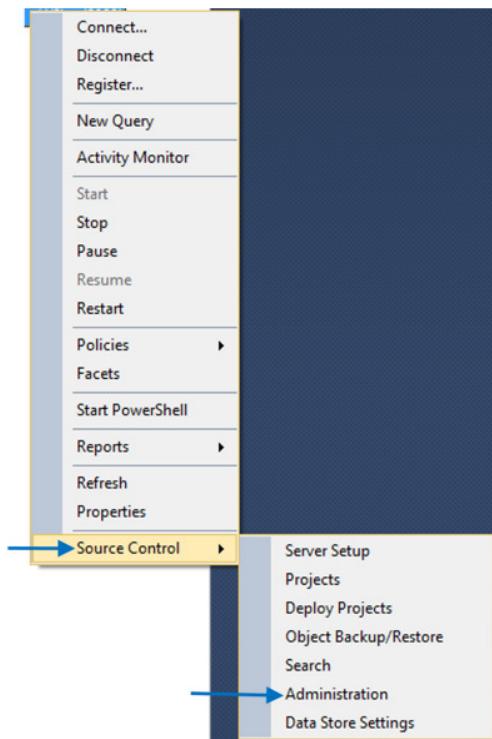
14 Administration – User Maintenance

14.1 Description

This section is used for the general maintenance of the system users, associating the user with a pre-defined role and setting the access rights per role.

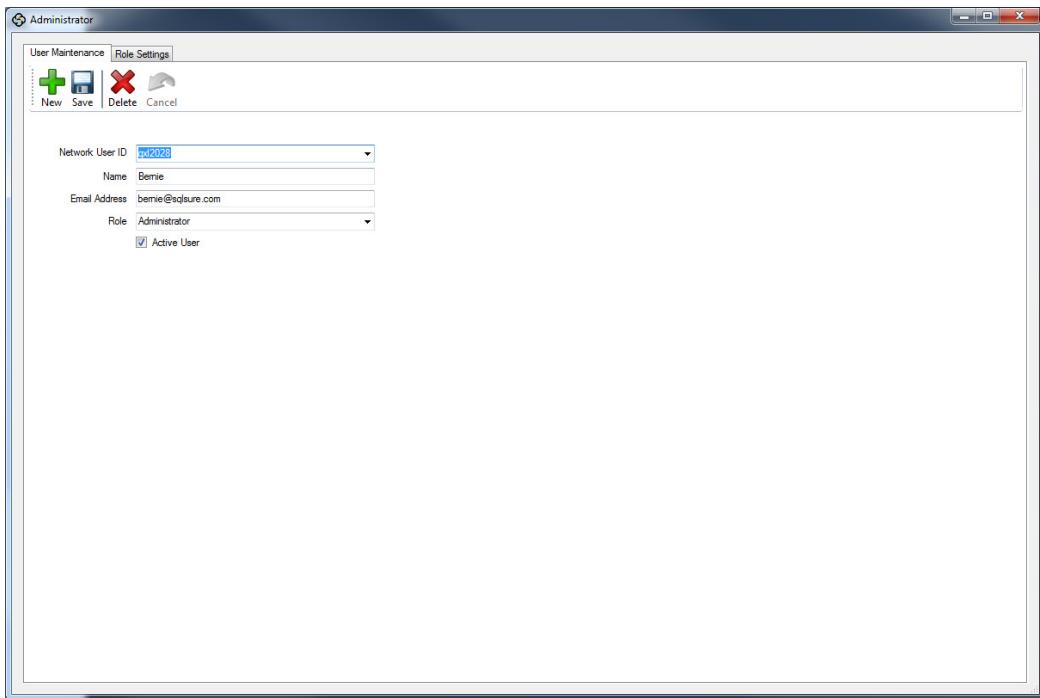
14.2 User Interfaces

Right-clicking on any database or on the top level root item in the object explorer in SQL Server Management Studio, will show the popup menu below:



Selecting the “Administration” menu option will show the screen below.

Select the “User Maintenance” tab to maintain the user details.



14.3 User Interface Fields

Field	Description
Network User ID	The network id associated with the user.
Name	The name of the user.
Email Address	The email address of the user for system notifications.
Role	Select one of the pre-defined roles for this user.
Active User	Setting the user to Active allows access to the system.

14.4 User Interface Actions

Toolbar Button	Description
	Add a new user.
	Save the user details.
	Removes the user definition.
	Cancel the current operation.

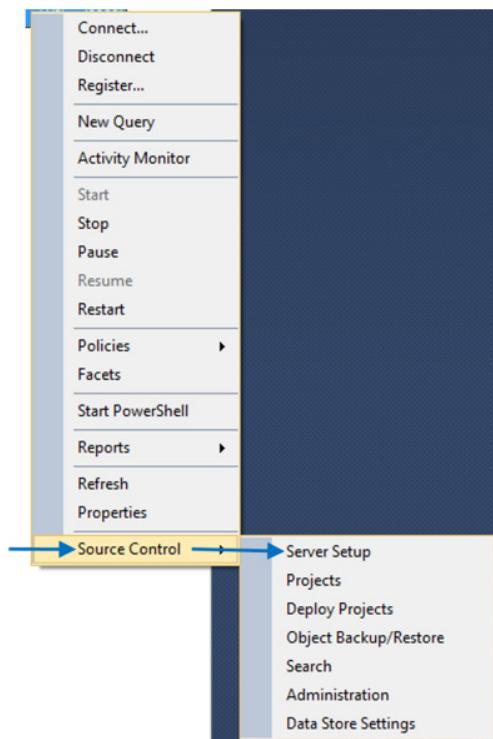
15 Server Setup

15.1 Description

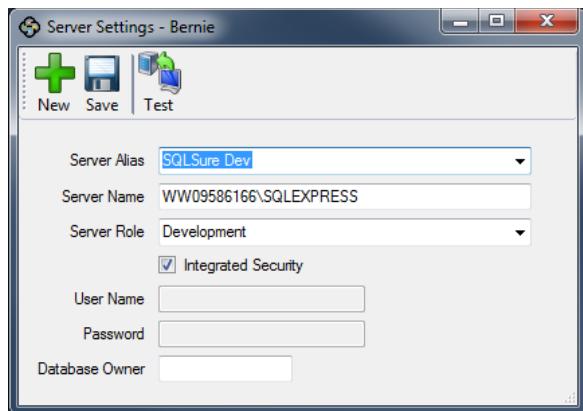
Every user is uniquely identified in the system. Every action of the user affecting the database object is recorded with the user's credentials. Access to database objects is determined by the set of user servers and the authentication used to connect to the server. These permissions are set in the database by existing functionality embedded in the database management system.

15.2 User Interfaces

Right-clicking on any database or on the top level root item in the object explorer in SQL Server Management Studio, will show the popup menu below:



Selecting the “Server Setup” menu option will show the screen below.



15.3 User Interface Fields

Field	Description	Explanation
Server Alias	Select one of the defined server aliases in the list.	A list of server aliases set up during the software installation.
Server Name	The name of the SQL Server instance on the network.	The fully qualified SQL Server network instance name.
Server Role	Select the server role.	
Integrated Security	Use the Windows integrated security for the connection to this server.	
User Name	If not integrated security, specify the user name for this server connection.	
Password	If not integrated security, specify the user password for this server connection.	
Database Owner	The default database owner to use.	Optional field.

15.4 User Interface Actions

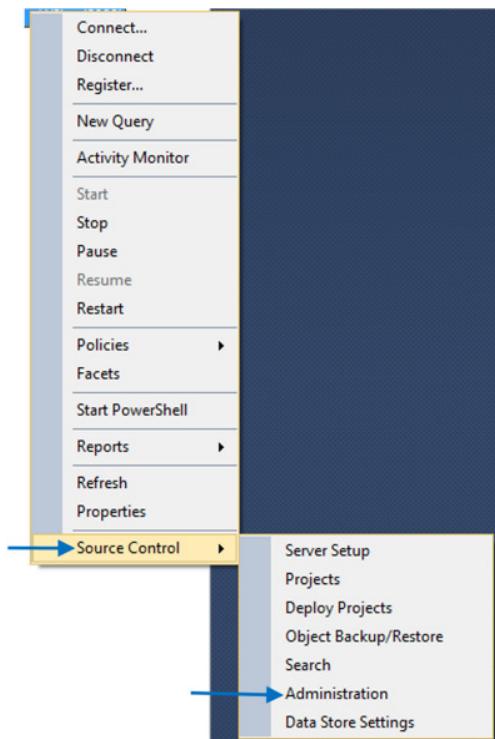
Toolbar Button	Description
	Add new user server.
	Save the server settings.
	Test the connection settings.

16 Administration – Role Settings

16.1 Description

16.2 User Interfaces

Right-clicking on any database or on the top level root item in the object explorer in SQL Server Management Studio, will show the popup menu below:



Selecting the “Administration” menu option will show the screen below. Select the “Role Settings” tab to show the setting per role.

	Developer	Releaser UAT	Releaser Pre-Prod	Releaser Prod	Administrator
Build (Check Out)					
Development	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Allow "Modify" if checked out					
UAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pre-Production	<input type="checkbox"/>				
Production	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Projects					
Project Maintenance	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Release					
Release to UAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Release to Pre-Production	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Release to Production	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Create Backup Set					
Development	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
UAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pre-Production	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Production	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Restore Objects					
Development	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
UAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pre-Production	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Production	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Administration					
Administer System	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

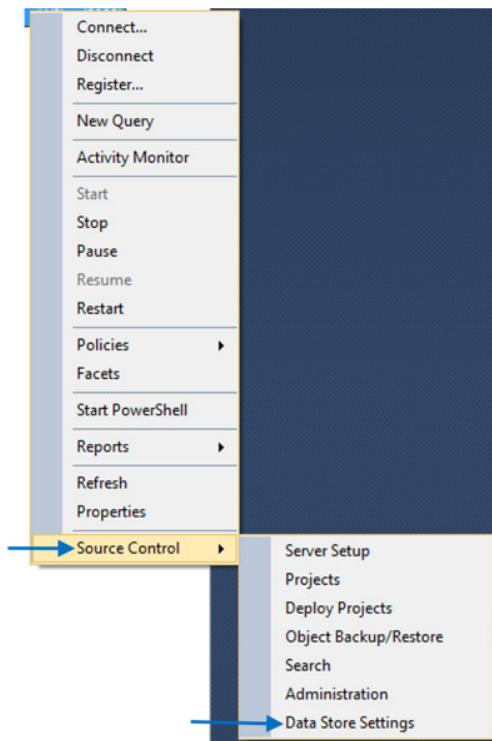
17 Data Store Settings

17.1 Description

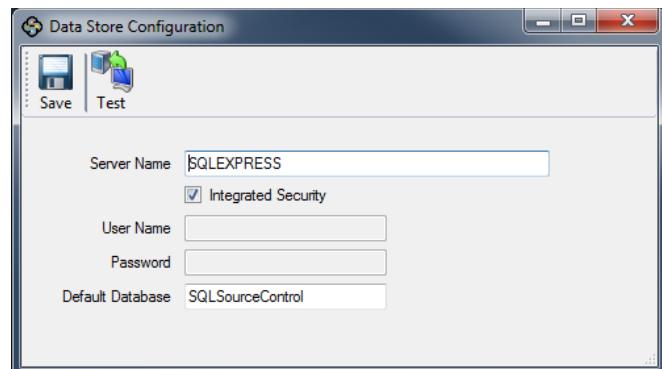
The data store is typically the location of the database where all object changes are stored. This setting is the main connection used to access the source control database.

17.2 User Interfaces

Right-clicking on any database or on the top level root item in the object explorer in SQL Server Management Studio, will show the popup menu below:



Selecting the “Data Store Settings” menu option will show the screen below.



17.3 User Interface Fields

Field	Description	Explanation
Server Name	The name of the SQL Server instance on the network.	The fully qualified SQL Server network instance name.
Integrated Security	Use the Windows integrated security for the connection to this server.	
User Name	If not integrated security, specify the user name for this server connection.	
Password	If not integrated security, specify the user password for this server connection.	
Database Owner	The default database owner to use.	Optional field.

17.4 User Interface Actions

Toolbar Button	Description
Save	Save the data store settings.
Test	Test the data store connection.