SQL Object Level Recovery Native 1.1

September 2009

Note: these pages apply to a version of
this product that is not the current released version.

For the latest support documentation, please see
http://documentation.red-gate.com

Contents

Getting started 3
Using SQL Object Level Recovery Native 4
Limitations 8
Acknowledgements 13

Getting started

SQL Object Level Recovery Native enables you to recover individual database objects from Microsoft® Tape Format (MTF) SQL Server backup files (.bak), and restore them to a database of your choice.

This is useful, for example, if a user accidentally drops a table, or deletes important data from a production database. By recovering only the objects you specify, SQL Object Level Recovery Native can save you a considerable amount of time when compared to restoring from a full backup. You can also save space, as you do not need to restore a complete database.

SQL Object Level Recovery Native recovers objects from SQL Server backup files created by SQL Server 2000, SQL Server 2005, and SQL Server 2008. You cannot restore objects to a version of SQL Server that is earlier than the version of SQL Server used to create the backup.

Using SQL Object Level Recovery Native

Using SQL Object Level Recovery Native (page 4) provides an overview of how to select a backup source, choose the objects that you want to recover, and restore them to a database.

Technical notes

SQL Object Level Recovery Native has some limitations on the types of object it can recover, and the types of backup file it can recover objects from. For example, SQL Object Level Recovery Native cannot recover objects from backup files created by SQL Backup (.sqb files).

See Limitations (page 8) for more information.

Related products

SQL Object Level Recovery Pro is included with SQL Backup Pro (http://www.red-gate.com/products/SQL_Backup/index.htm), and enables you to recover objects from SQL Backup (.sqb) files. SQL Object Level Recovery Pro requires a SQL Backup Pro license on the server you are restoring objects to.

For more complex recovery scenarios, you should consider using Red Gate SQL Compare (http://www.red-gate.com/products/SQL_Compare/index.htm) and SQL Data Compare (http://www.red-gate.com/products/SQL_Data_Compare/index.htm). These enable you to compare the contents (object schema, and data) of SQL Server backup files (.bak) or SQL Backup (.sqb) files, with a live database, and then synchronize the database with the backup file contents while maintaining object dependencies.

Using SQL Object Level Recovery Native

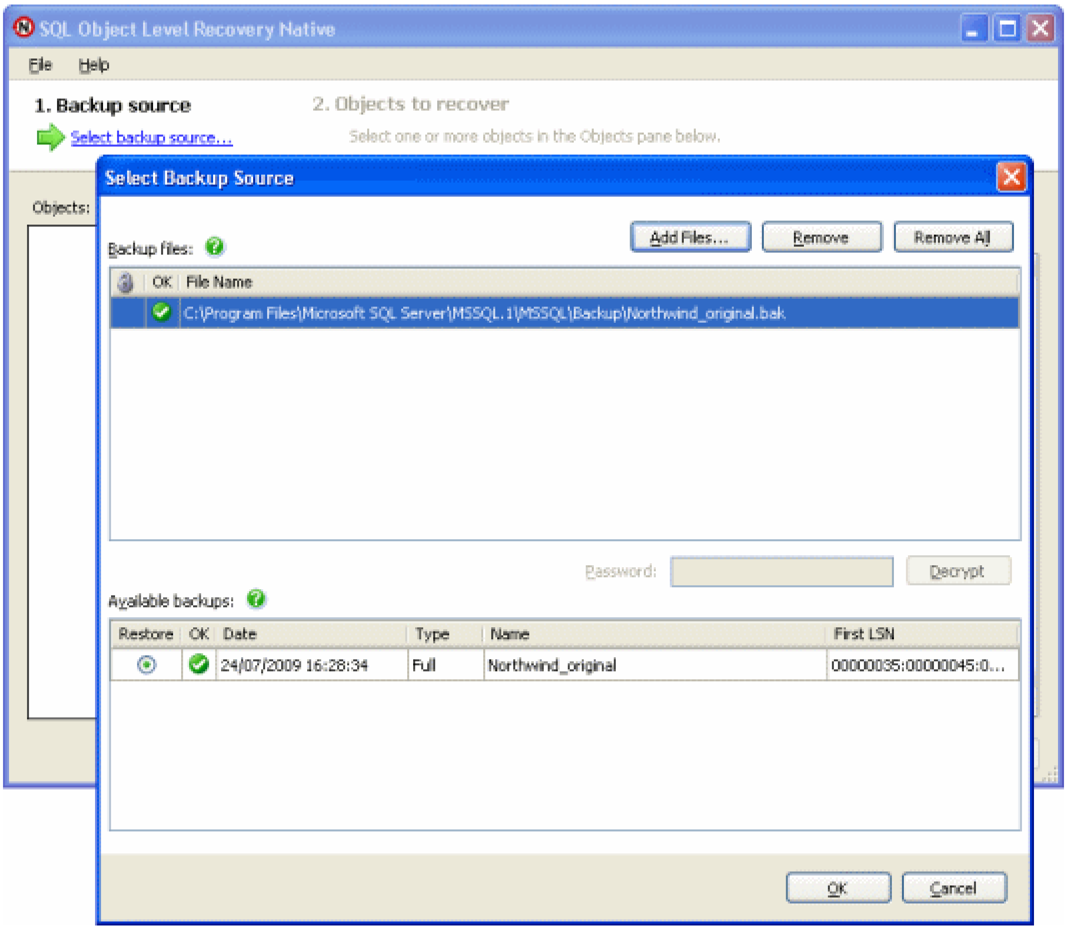
This topic describes the basic steps to recover objects from a backup file using SQL Object Level Recovery Native, including tips to help you get the most from the application.

SQL Object Level Recovery Native is licensed per user. You do not need to install any software on the SQL Server that you recover objects to.

You recover objects in four steps:

1. Select a backup source (page 4)
2. Select objects to recover (page 5)
3. Select a destination database (page 6)
4. Recover the objects (page 7)

1. Select a backup source



Select one or more objects in the Objects pane below.

OK	File Name
✓	C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup\Northwind_original.bak

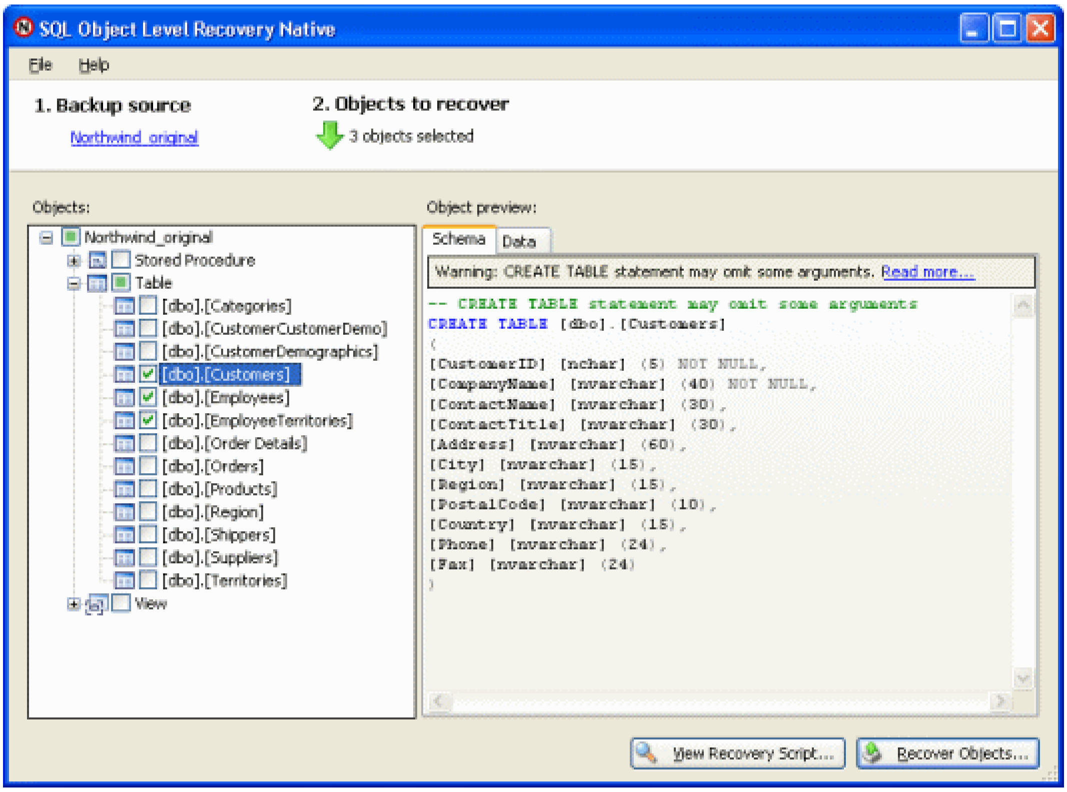
Restore	OK	Date	Type	Name	First LSN
●	✓	24/07/2009 16:28:34	Full	Northwind_original	00000035:00000045:0...

- you can add full backup files or differential backup files (SQL Server MTF files only; these usually have a .bak extension)

  If you add a differential backup file, you will also need to add the related full backup file.

- for backups that are split across several files, you must add every file from the backup set.
- if there are problems validating the file or backup, ✖ is displayed in the **OK** column; select the file or backup to see a detailed error message in the lower pane

2. Select objects to recover



Warning: CREATE TABLE statement may omit some arguments. Read more...

```
-- CREATE TABLE statement may omit some arguments
CREATE TABLE [dbo].[Customers]
(
[CustomerID] [nchar] (5) NOT NULL,
[CompanyName] [nvarchar] (40) NOT NULL,
[ContactName] [nvarchar] (30),
[ContactTitle] [nvarchar] (30),
[Address] [nvarchar] (60),
[City] [nvarchar] (15),
[Region] [nvarchar] (15),
[PostalCode] [nvarchar] (10),
[Country] [nvarchar] (15),
[Phone] [nvarchar] (24),
[Fax] [nvarchar] (24)
)
```

- SQL Object Level Recovery Native does not automatically check for dependencies between objects, but objects are recovered in an order that reduces the possibility of errors caused by missing dependents.

  Depending on the contents of the destination database, you may have to select several objects to ensure that the necessary object dependencies are maintained.

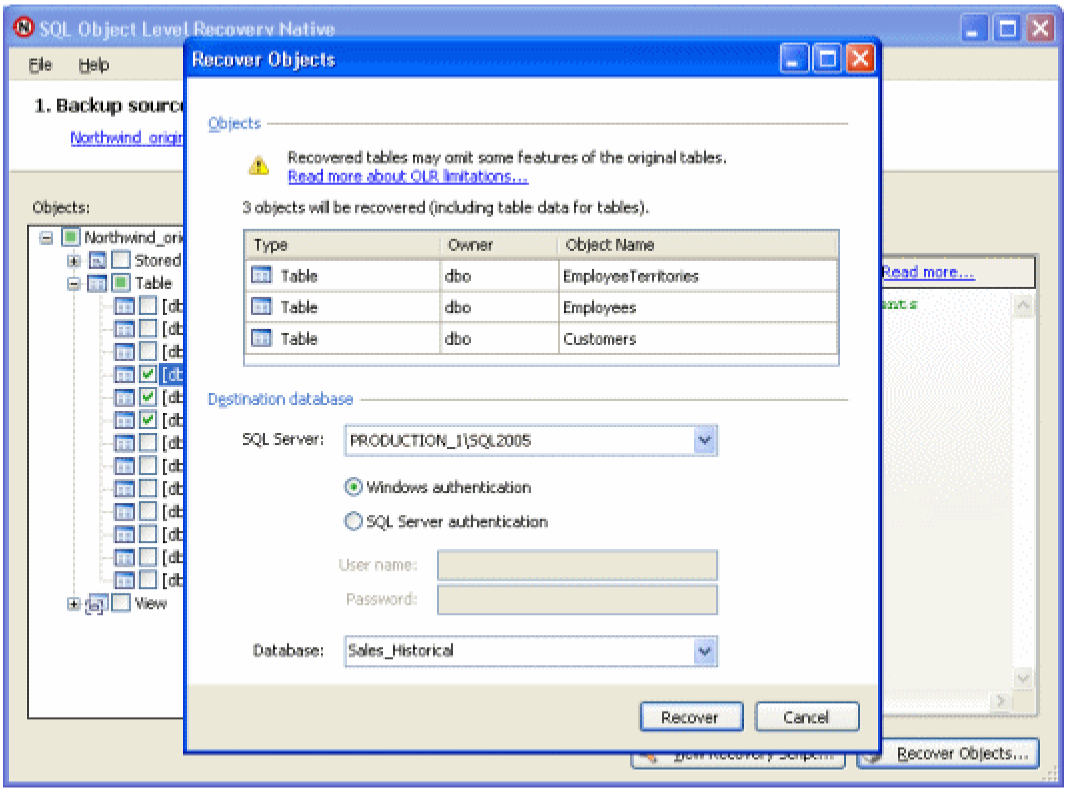
- the **Data** tab (available for table objects only) shows the first 100 rows of data in a table

  Some types of data (for example, varchar and nvarchar) may not be shown if it is too large to display.

- **View Recovery Script** displays a complete object and data recovery script for the objects you have selected; you can edit this script if required, then save it and run it on the target database to restore the objects
- SQL Object Level Recovery Native does not support all possible arguments of the CREATE TABLE statement

  For more information, see Limitations (page 8).

3. Select a destination database



Objects

Recovered tables may omit some features of the original tables.
Read more about OLR limitations...

3 objects will be recovered (including table data for tables).

Type	Owner	Object Name
Table	dbo	EmployeeTerritories
Table	dbo	Employees
Table	dbo	Customers

Destination database

SQL Server: PRODUCTION_1\SQL2005

● Windows authentication

○ SQL Server authentication

User name:

Password:

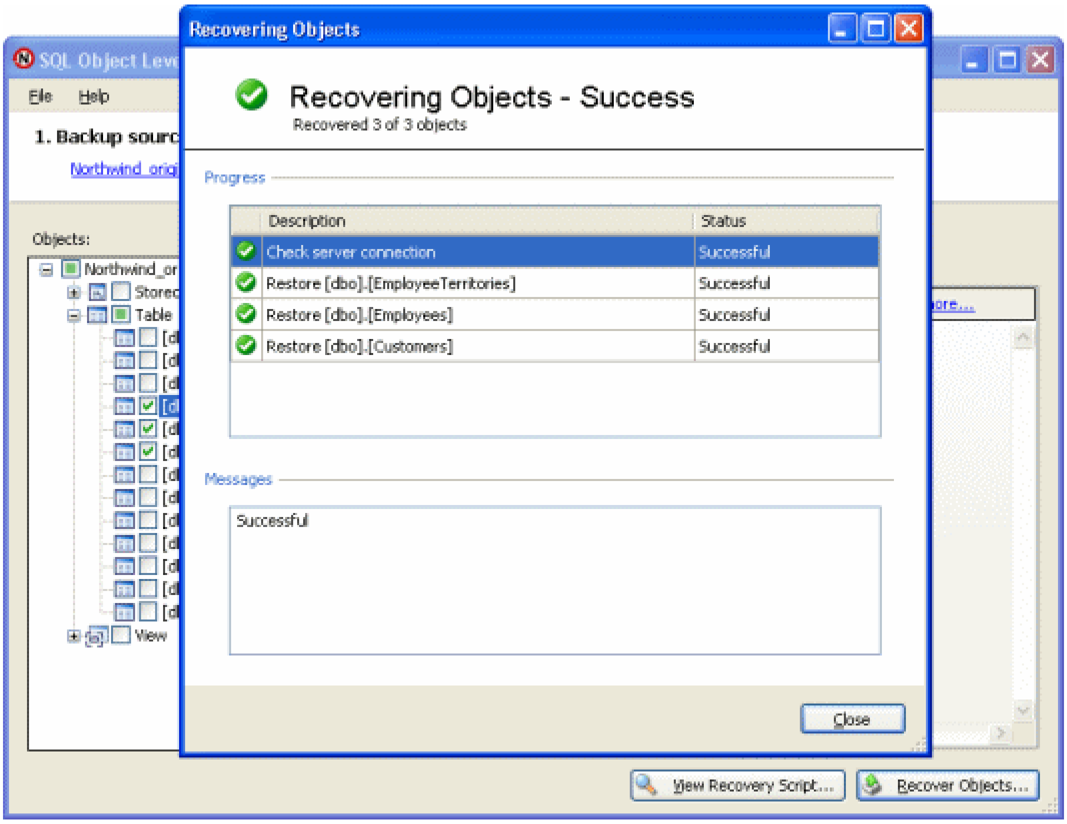
Database: Sales_Historical

- you can restore objects to SQL Server 2000, SQL Server 2005, or SQL Server 2008 databases

  You cannot restore objects to a version of SQL Server that is earlier than the version of SQL Server used to create the backup.
- the destination database must be open, and in a suitable state to create recovered objects

  For example, a database that is currently being restored cannot be used as a destination database.
- you must have appropriate security permissions to create objects in the destination database

4. Recover the objects



Recovering Objects - Success

Recovered 3 of 3 objects

Progress

	Description	Status
✓	Check server connection	Successful
✓	Restore [dbo].[EmployeeTerritories]	Successful
✓	Restore [dbo].[Employees]	Successful
✓	Restore [dbo].[Customers]	Successful

Messages

Successful

- when you click **Recover**, the **Recovering Objects** dialog shows the status of each object you selected for recovery
- objects marked with ☒ could not be recovered. Select an object in the Progress list to display a detailed error message for that object in the Messages pane.
- SQL Object Level Recovery Native will not alter or overwrite existing objects in the destination database

  See Limitations (page 8) for more information.

Limitations

You can use SQL Object Level Recovery Native to recover various object types from SQL Server Backup files, with some limitations. For example, dependencies between objects are not handled automatically, and some features of SQL Server tables are not supported. The following sections provide more detailed information about these limitations:

- Existing objects (page 8)
- Object dependencies (page 8)
- Supported backup types (page 9)
- Supported object types (page 9)
- Supported CREATE TABLE arguments (page 11)

Refer to your SQL Server documentation (http://msdn2.microsoft.com/en-us/library/bb545450.aspx) for detailed information about specific object types, and table arguments.

For more complex recovery scenarios, you should consider using Red Gate SQL Compare (http://www.red-gate.com/products/SQL_Compare/index.htm) and SQL Data Compare (http://www.red-gate.com/products/SQL_Data_Compare/index.htm). These enable you to compare the contents (object schema, and data) of SQL Server backup files (.bak) with a live database, and then synchronize the database with the backup file contents while maintaining object dependencies.

Existing objects

Objects that already exist in the destination database will not be modified or overwritten by SQL Object Level Recovery Native. Attempting to recover such objects results in an error.

It is usually safer to recover an object to a test or staging database first, and then transfer the object to its final destination database manually. If you want to recover an object directly from a backup file to its final destination database, you will have to drop the object first. Make sure you have a recent valid backup of the object before you drop it.

Object dependencies

Objects you attempt to recover may have dependencies on other objects in the destination database. For example, a view may refer to several tables; successful recovery of the view depends on these tables being present in the destination database.

SQL Object Level Recovery Native does *not* attempt to resolve dependencies automatically. To avoid dependency errors, you may need to recover multiple dependent objects together.

If you have selected objects of more than one type, they are recovered in the following order:

1. SCHEMA
2. TYPE (user defined type)
3. XML SCHEMA COLLECTION
4. FUNCTION
5. TABLE
6. VIEW
7. PROCEDURE (stored procedure)

Recovering objects in this order reduces the possibility of failures caused by dependencies on missing objects.

Supported backup types

SQL Object Level Recovery Native supports:

- full backups
- differential backups

  If you need to recover objects from a differential backup, you will also need to provide the associated full backup.

You can use SQL Object Level Recovery Native with full backups and differential backups. To recover objects from a differential backup, you will also need to provide the associated full backup.

SQL Object Level Recovery Native does not support:

- filegroup backups
- transaction log backups
- backups from databases that use Transparent Data Encryption (TDE)
- SQL Backup (.sqb) backups

  If you need to restore an object from a SQL Backup backup file, you can do so using SQL Backup Pro (from version 6.2). For more information, see the SQL Backup Product page (http://www.red-gate.com/products/SQL_Backup/index.htm).
- restoring to a version of SQL Server that is earlier than the version of SQL Server used to create the backup

Supported object types

Object types marked ✓ can be recovered from SQL Server .bak files. Other object types are not supported.

Object type	Supported
ASSEMBLY	

ASYMMETRIC KEY	
CERTIFICATE	
CONTRACT	
DEFAULT	
EVENT NOTIFICATION	
FULLTEXT CATALOG	
FULLTEXT STOPLIST	
FUNCTION	✓
INDEX	
MESSAGE TYPE	
PARTITION FUNCTION	
PARTITION SCHEME	
PROCEDURE (stored procedure)	✓
QUEUE	
REMOTE SERVICE BINDING	
ROLE	
ROUTE	
RULE	
SCHEMA	✓
SERVICE	
SYMMETRIC KEY	
SYNONYM	
TABLE	✓
TRIGGER	
TYPE (user defined type)	✓
USER	
VIEW	✓
XML SCHEMA COLLECTION	✓

Supported CREATE TABLE arguments

CREATE TABLE arguments marked ✓ are supported. All other CREATE TABLE arguments are ignored. For example, if the table to be recovered includes a FOREIGN KEY ... REFERENCES argument, this will not be created in the recovered table.

Argument	Supported
ALLOW_PAGE_LOCKS	
ALLOW_ROW_LOCKS	
CLUSTERED	
COLLATE	
computed_column_expression	✓
CONSTRAINT	
CONTENT	
DATA_COMPRESSION	
DEFAULT	✓
DOCUMENT	
FILESTREAM_ON	
FOREIGN KEY ... REFERENCES	
IDENTITY	✓
IGNORE_DUP_KEY	
NONCLUSTERED	
NOT FOR REPLICATION	
NULL	✓
ON filegroup	
ON partition scheme	
ON DELETE	
ON UPDATE	
PAD_INDEX	
PERSISTED	✓
PRIMARY KEY	

RANGE	
ROWGUIDCOL	
SPARSE	✓
STATISTICS_NORECOMPUTE	
TEXTIMAGE_ON	
UNIQUE	
WITH FILLFACTOR	
XML COLUMN_SET FOR ALL_SPARSE_COLUMNS	

Acknowledgements

Trademarks and registered trademarks

Copyright information