Limitations

SQL Object Level Recovery Pro has 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
- Object dependencies
- Supported backup types
- Supported object types
- Supported data types
- Supported CREATE TABLE arguments
- .NET 3.5 or greater

Refer to your SQL Server documentation for detailed information about specific object types, data types, and table arguments.

For more complex recovery scenarios, you should consider using Redgate SQL Compare and SQL Data Compare. These enable you to compare the contents (object schema, and data) of SQL Backup .sqb files 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 Pro. 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 Pro 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

You can use SQL Object Level Recovery Pro 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 Pro does not support:

- filegroup backups
- transaction log backups
- backups from databases that use Transparent Data Encryption (TDE)
- native SQL Server backups (.bak files)

Supported object types

Object types marked  can be recovered from SQL Backup .sqb 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 data types

Table data with types marked ✅ can be recovered from SQL Backup .sqb files.

Group	Data type	Supported
Exact numerics	bit	✅
	tinyint	✅
	smallint	✅
	int	✅
	bigint	✅
	numeric	✅
	decimal	✅

	smallmoney	✅
	money	✅
Approximate numerics	float	✅
	real	✅
Date and time	datetime	✅
	smalldatetime	✅
	date	✅
	time	✅
	datetimeoffset	✅
	datetime2	✅
Character strings	char	✅
	varchar / varchar(max)	✅
	text	✅
Unicode character strings	nchar	✅
	nvarchar / nvarchar(max)	✅
	ntext	✅
Binary strings	binary	✅
	varbinary / nvarbinary(max)	✅
	image	✅
Other data types	sql_variant	✅
	timestamp	✅
	uniqueidentifier	✅
	xml	✅
CLR data types	hierarchyid	✅
Spatial data types	geometry	✅
	geography	✅

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	

.Net 3.5 or greater

In order to run SQL Object Level Recovery Pro you will need to install NET 3.5 or greater. Currently SQL Backup Pro only insists on .NET 2 or greater. However to run this functionality you will need to install a later version of .NET.

Alternatively you can copy the contents of the *SQBObjectLevelRecovery* (by default it is installed to *C:\Program Files\Red Gate\SQL Backup 8\SQBObjectLevelRecovery\SQLObjectLevelRecoveryPro.exe*) folder to any server (with .NET 3.5 or greater installed) , and run SQL Object Level Recovery Pro independently of SQL Backup Pro. In addition this is useful, for example, if you are recovering objects from a large backup, and want to avoid streaming large amounts of data across the network. The server you are recovering objects to must have a valid SQL Backup Pro license.