

Example - selecting tables with unrelated names

This example illustrates how to select a number of individual tables for comparison when their names are not related in any way.

In this example, the databases contain the following tables:

- Product
- Supplier
- ProductCategory
- SpecialOffer
- Customer
- Order
- Invoice

You are interested only in the schema differences between the *Product*, *Customer*, *Order*, and *Invoice* tables in two different versions of your database, *Customers1* and *Customers2*; you are not interested in any of the other tables, or any other objects in the databases.

Using the command line

To specify the list of tables to include, you use the */include* switch. You could use an *include* switch for each table that you want to compare. However, this could get unwieldy if you have a long list of tables. Instead, you can use the pipe character (|) to separate the table names:

```
sqlcompare /db1:Customers1 /db2:Customers2 /Include:table /Include:table:\[Product\]^|Customer^|Order^|Invoice
```

where:

/db1:Customers1 specifies that you want to compare the database *Customers1*

/db2:Customers2 specifies that you want to compare the database *Customers2*

/Include:table specifies that you want to compare only tables; you do not want to compare other objects such as views, stored procedures, and so on. If you omit this argument, SQL Compare compares all tables that match the second */Includeswitch* and all other objects in the databases.

To specify more than one object type for inclusion, use multiple */Include* switches. For example, to include only tables and views, enter:

```
/Include:table /Include:view /Include:table:\[Product\]^|Customer^|Order^|Invoice
```

specifies that you want to compare only the tables that have a name that includes the strings *[Product]*, or *Customer*, or *Order*, or *Invoice*



Because you use .NET standard regular expressions to define the */Include* and */Exclude* arguments you must escape the square brackets ([]) with the backslash character (\). Regular expression syntax is beyond the scope of this online help; refer to your Microsoft .NET framework documentation for more information.

You must include the brackets ([]) in the string; if you specify the argument without the brackets, */Include:table:Product*, the *ProductCategory* table is included because it contains the string *Product*. The full SQL Server table names are qualified by the owner name in SQL Server 2000, and the schema name in SQL Server 2005/2008, and include brackets. For example (in SQL Server 2000):

```
[dbo].[Product]  
[dbo].[ProductCategory]
```

and so on. Therefore, the brackets indicate that you are specifying the full table name. To include the owner (or schema) name in the regular expression, you also need to escape the dot (.):

```
/Include:table:\[dbo\]\. \[Product\]
```

The pipe character (|) in a regular expression is interpreted as a logical OR. The character must be escaped by the caret character (^), to prevent the operating system shell from interpreting it as the pipe operator.



If you want to use the caret character itself as part of your regular expression, it must be escaped by a second caret.

Using XML

You can use XML like this:

```
<?xml version="1.0"?>
<commandline>
  <database1>Customers1</database1>
  <database2>Customers2</database2>
  <sync/>
  <include>Table</include>
  <include>Table:\[Product\]|Customer|Order|Invoice</include>
</commandline>
```



The pipe character (|) (and other operating system operators) do not have to be escaped by the caret character (^) when they are specified in the XML file.

To execute the comparison using the XML file, enter the following command, where *XMLFileName* is the name of the XML file:

```
sqlcompare /Argfile:XMLFileName.xml
```