

SQL Schema Compare Excluding Objects by Type

Type-based method allows you to include or exclude an entire class of objects, such as tables or views. To exclude objects by type, use the <ObjectTypeSelection> element and specify two attributes:

- **Type**. Indicates the object type
- **Include="true|false"**. A flag that indicates whether the specified type should be included or excluded

The following xml fragment excludes tables and views from the comparison.

```
<ObjectTypeSelection>
  <ObjectType Type="Table" Include="false" />
  <ObjectType Type="View" Include="false" />
</ObjectTypeSelection>
```

The command line provides a list of predefined types that you can choose to include or exclude:

- Table
- MemoryTable
- View
- StoredProcedure
- UserDefinedFunction
- UserDefinedDataType
- ClrDataType
- UserDefinedTableType
- XmlSchemaCollection
- Default
- Rule
- Schema
- User
- DatabaseRole
- ApplicationRole
- Certificate
- AsymmetricKey
- SymmetricKey
- PartitionFunction
- PartitionScheme
- FullTextCatalog
- Synonym
- Assembly
- Aggregate
- Sequence
- DatabaseTrigger
- PrimaryKey
- UniqueConstraint
- Index
- XmlIndex
- SpatialIndex

- FullTextIndex
- ForeignKey
- DefaultConstraint
- CheckConstraint
- Trigger
- Statistic
- BrokerMessageType
- BrokerContract
- BrokerQueue
- BrokerRoute
- BrokerService
- BrokerRemoteServiceBinding
- BrokerPriority
- Permission
- ExtendedProperty
- FullTextStopList
- FullTextStopListWord
- SearchPropertyList
- SecurityPolicy
- ColumnMasterKey
- ColumnEncryptionKey

In addition to these types, the command line supports also three macro-types in order to facilitate the selection:

- **All**. When specified it includes all object types
- **None**. When specified it excludes all object types
- **DefaultObjects**. When specified it includes the most common objects.

The Include attribute is ignored for macro-types. Object types included by DefaultObjects macro-option are those that you see in the Schema Compare User-Interface option screen when you click on the **Restore Defaults** button. More details are provided [here](#).

Depending on your scenario, you could combine one of the macro-types with the actual types to achieve the desired result. For example, to exclude only views and procedures, you could use the following xml fragment:

```
<ObjectTypeSelection>
  <ObjectType Type="DefaultObjects" />
  <ObjectType Type="View" Include="false" />
  <ObjectType Type="StoredProcedure" Include="false" />
</ObjectTypeSelection>
```

or just:

```
<ObjectTypeSelection>
  <ObjectType Type="View" Include="false" />
  <ObjectType Type="StoredProcedure" Include="false" />
</ObjectTypeSelection>
```

To include only tables and exclude all other objects, you could combine the None macro-type with the Table type as in the following xml fragment:

```
<ObjectTypeSelection>  
  <ObjectType Type="None" />  
  <ObjectType Type="Table" Include="true" />  
</ObjectTypeSelection>
```

Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more](#) > >

IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------