



WiX managed SDK

WixToolset.Dtf.WindowsInstaller namespace

Database Class

Database Class

Accesses a Windows Installer database.

Methods

Method	Description
<code>ApplyTransform(transformFile)</code>	Apply a transform to the database, suppressing any error conditions specified by the transform's summary information.
<code>ApplyTransform(transformFile, errorConditionsToSuppress)</code>	Apply a transform to the database, specifying error conditions to suppress.
<code>Commit()</code>	Finalizes the persistent form of the database. All persistent data is written to the writeable database, and no temporary columns or rows are written.
<code>CountRows(table)</code>	Gets the count of all rows in the table.
<code>CountRows(table, where)</code>	Gets the count of all rows in the table that satisfy a given condition.

Method	Description
CreateRecord(fieldCount)	Creates a new record object with the requested number of fields.
CreateTransformSummaryInfo(referenceDatabase, transformFile, errors, validations)	Creates and populates the summary information stream of an existing transform file, and fills in the properties with the base and reference ProductCode and ProductVersion.
DeleteOnClose(path)	Schedules a file or directory for deletion after the database handle is closed.
Dispose(disposing)	Closes the database handle. After closing a handle, further method calls may throw «see T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException» .
Execute(sqlFormat, args)	Executes the query specified by a SQL string. The query may not be a SELECT statement.
Execute(sql, record)	Executes the query specified by a SQL string. The query may not be a SELECT statement.
ExecuteIntegerQuery(sqlFormat, args)	Executes the specified SQL SELECT query and returns all results as integers.
ExecuteIntegerQuery(sql, record)	Executes the specified SQL SELECT query and returns all results as integers.

Method	Description
<code>ExecutePropertyQuery(property)</code>	Returns the value of the specified property.
<code>ExecuteQuery(sqlFormat, args)</code>	Executes the specified SQL SELECT query and returns all results.
<code>ExecuteQuery(sql, record)</code>	Executes the specified SQL SELECT query and returns all results.
<code>ExecuteScalar(sqlFormat, args)</code>	Executes the specified SQL SELECT query and returns a single result.
<code>ExecuteScalar(sql, record)</code>	Executes the specified SQL SELECT query and returns a single result.
<code>ExecuteStringQuery(sqlFormat, args)</code>	Executes the specified SQL SELECT query and returns all results as strings.
<code>ExecuteStringQuery(sql, record)</code>	Executes the specified SQL SELECT query and returns all results as strings.
<code>Export(table, exportFilePath)</code>	Copies the structure and data from a specified table to a text archive file.
<code>ExportAll(directoryPath)</code>	Exports all database tables, streams, and summary information to archive files.
<code>FromHandle(handle, ownsHandle)</code>	Creates a new Database object from an integer database handle.
<code>GenerateTransform(referenceDatabase,</code>	Creates a transform that, when applied to the object database, results

Method	Description
<code>transformFile)</code>	in the reference database.
<code>Import(importFilePath)</code>	Imports a database table from a text archive file, dropping any existing table.
<code>ImportAll(directoryPath)</code>	Imports all database tables, streams, and summary information from archive files.
<code>IsColumnPersistent(table, column)</code>	Checks whether a table contains a persistent column with a given name.
<code>IsTablePersistent(table)</code>	Checks whether a table exists and is persistent in the database.
<code>IsTransformValid(transformFile)</code>	Checks whether a transform is valid for this Database, according to its validation data and flags.
<code>IsTransformValid(transformSummaryInfo)</code>	Checks whether a transform is valid for this Database, according to its SummaryInfo data.
<code>Merge(otherDatabase, errorTable)</code>	Merges another database with this database.
<code>Merge(otherDatabase)</code>	Merges another database with this database.
<code>OpenView(sqlFormat, args)</code>	Gets a View object representing the query specified by a SQL string.

Method	Description
<code>ToString()</code>	Returns the file path of this database, or the handle value if a file path was not specified.
<code>ViewTransform(transformFile)</code>	Apply a transform to the database, recording the changes in the "_TransformView" table.

Properties

Property	Description
<code>CodePage</code>	Gets or sets the code page of the Database.
<code>FilePath</code>	Gets the file path the Database was originally opened from, or null if not known.
<code>IsReadOnly</code>	Gets a boolean value indicating whether this database was opened in read-only mode.
<code>OpenMode</code>	Gets the open mode for the database.
<code>SummaryInfo</code>	Gets the SummaryInfo object for this database that can be used to examine and modify properties to the summary information stream.
<code>Tables</code>	Gets the collection of tables in the Database.

Remarks

The «see `M:WixToolset.Dtf.WindowsInstaller.Database.Commit`» method must be called before the Database is closed to write out all persistent changes. If the Commit method is not called, the installer performs an implicit rollback upon object destruction. The client can use the following procedure for data access:

- Obtain a Database object using one of the Database constructors.
- Initiate a query using a SQL string by calling the «see `M:WixToolset.Dtf.WindowsInstaller.Database.OpenView(System.String,System.Object[])`» method of the Database.
- Set query parameters in a «see `T:WixToolset.Dtf.WindowsInstaller.Record`» and execute the database query by calling the «see `M:WixToolset.Dtf.WindowsInstaller.View.Execute(WixToolset.Dtf.WindowsInstaller.Record)`» method of the «see `T:WixToolset.Dtf.WindowsInstaller.View`» . This produces a result that can be fetched or updated.
- Call the «see `M:WixToolset.Dtf.WindowsInstaller.View.Fetch`» method of the View repeatedly to return Records.
- Update database rows of a Record object obtained by the Fetch method using one of the «see `M:WixToolset.Dtf.WindowsInstaller.View.Modify(WixToolset.Dtf.WindowsInstaller.ViewModifyMode,WixToolset.Dtf.WindowsInstaller.Record)`» methods of the View.
- Release the query and any unfetched records by calling the «see `M:WixToolset.Dtf.WindowsInstaller.InstallerHandle.Close`» method of the View.
- Persist any database updates by calling the Commit method of the Database.

`WixToolset.Dtf.WindowsInstaller.dll` version `4.0.2+644ed0118bae1aa885bb6cc906dc6c0b904de4d9`

ApplyTransform(transformFile) Method

Apply a transform to the database, suppressing any error conditions specified by the transform's summary information.

Declaration

```
public void ApplyTransform(  
    string transformFile  
)
```

Parameters

Parameter	Type	Description
transformFile	string	Path to the transform file

Remarks

Win32 MSI API: [MsiDatabaseApplyTransform](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the transform could not be applied
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ApplyTransform(transformFile, errorConditionsToSuppress) Method

Apply a transform to the database, specifying error conditions to suppress.

Declaration

```
public void ApplyTransform(  
    string transformFile,  
    TransformErrors errorConditionsToSuppress  
)
```

Parameters

Parameter	Type	Description
transformFile	string	Path to the transform file
errorConditionsToSuppress	TransformErrors	Error conditions that are to be suppressed

Remarks

Win32 MSI API: [MsiDatabaseApplyTransform](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the transform could not be applied
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

Commit() Method

Finalizes the persistent form of the database. All persistent data is written to the writeable database, and no temporary columns or rows are written.

Declaration

```
public void Commit()
```

Remarks

For a database open in «see F:WixToolset.Dtf.WindowsInstaller.DatabaseOpenMode.ReadOnly» mode, this method has no effect. For a database open in «see F:WixToolset.Dtf.WindowsInstaller.DatabaseOpenMode.CreateDirect» or «see F:WixToolset.Dtf.WindowsInstaller.DatabaseOpenMode.Direct» mode, it is not necessary to call this method because the database will be automatically committed when it is closed. However this method may be called at any time to persist the current state of tables loaded into memory. For a database open in «see F:WixToolset.Dtf.WindowsInstaller.DatabaseOpenMode.Create» or «see

F:WixToolset.Dtf.WindowsInstaller.DatabaseOpenMode.Transact» mode, no changes will be persisted until this method is called. If the database object is closed without calling this method, the database file remains unmodified. Win32 MSI API:

MsiDatabaseCommit

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

CountRows(table) Method

Gets the count of all rows in the table.

Declaration

```
public int CountRows(  
    string table  
)
```

Parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
table	string	Name of the table whose rows are to be counted

Return value

`int` The count of all rows in the table

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

CountRows(table, where) Method

Gets the count of all rows in the table that satisfy a given condition.

Declaration

```
public int CountRows(  
    string table,
```

```
    string where  
)
```

Parameters

Parameter	Type	Description
table	string	Name of the table whose rows are to be counted
where	string	Conditional expression, such as could be placed on the end of a SQL WHERE clause

Return value

`int` The count of all rows in the table satisfying the condition

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL WHERE syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

CreateRecord(fieldCount) Method

Creates a new record object with the requested number of fields.

Declaration

```
public Record CreateRecord(  
    int fieldCount  
)
```

Parameters

Parameter	Type	Description
fieldCount	int	Required number of fields, which may be 0. The maximum number of fields in a record is limited to 65535.

Return value

Record A new record object that can be used with the database.

Remarks

This method is equivalent to directly calling the «see T:WixToolset.Dtf.WindowsInstaller.Record» constructor in all cases outside of a custom action context. When in a custom action session, this method allows creation of a record that can work with a database

other than the session database. The Record object should be «see `M:WixToolset.Dtf.WindowsInstaller.InstallerHandle.Close`» d after use. It is best that the handle be closed manually as soon as it is no longer needed, as leaving lots of unused handles open can degrade performance. Win32 MSI API: `MsiCreateRecord`

CreateTransformSummaryInfo(referenceDatabase, transformFile, errors, validations) Method

Creates and populates the summary information stream of an existing transform file, and fills in the properties with the base and reference ProductCode and ProductVersion.

Declaration

```
public void CreateTransformSummaryInfo(  
    Database referenceDatabase,  
    string transformFile,  
    TransformErrors errors,  
    TransformValidations validations  
)
```

Parameters

Parameter	Type	Description
referenceDatabase	Database	Database that does not include the changes

Parameter	Type	Description
transformFile	string	Name of the generated transform file
errors	TransformErrors	Error conditions that should be suppressed when the transform is applied
validations	TransformValidations	Defines which properties should be validated to verify that this transform can be applied to a database.

Remarks

Win32 MSI API: [MsiCreateTransformSummaryInfo](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the transform summary info could not be
generated	
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	a Database handle is invalid

DeleteOnClose(path) Method

Schedules a file or directory for deletion after the database handle is closed.

Declaration

```
public void DeleteOnClose(  
    string path  
)
```

Parameters

Parameter	Type	Description
path	string	File or directory path to be deleted. All files and subdirectories under a directory are deleted.

Remarks

Once an item is scheduled, it cannot be unscheduled. The items cannot be deleted if the Database object is auto-disposed by the garbage collector; the handle must be explicitly closed. Files which are read-only or otherwise locked cannot be deleted, but they will not cause an exception to be thrown.

Dispose(disposing) Method

Closes the database handle. After closing a handle, further method calls may throw «see `T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException`» .

Declaration

```
protected void Dispose(  
    bool disposing  
)
```

Parameters

Parameter	Type	Description
disposing	bool	If true, the method has been called directly or indirectly by a user's code, so managed and unmanaged resources will be disposed. If false, only unmanaged resources will be disposed.

Execute(sqlFormat, args) Method

Executes the query specified by a SQL string. The query may not be a SELECT statement.

Declaration

```
public void Execute(  
    string sqlFormat,  
    System.Object[] args  
)
```

Parameters

Parameter	Type	Description
sqlFormat	string	SQL query string, which may contain format items
args	System.Object[]	Zero or more objects to format

Remarks

The *sqlFormat* parameter is formatted using «see `M:System.String.Format(System.String,System.Object[])`» . Win32 MSI APIs: [MsiDatabaseOpenView](#) , [MsiViewExecute](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

Execute(sql, record) Method

Executes the query specified by a SQL string. The query may not be a SELECT statement.

Declaration

```
public void Execute(  
    string sql,  
    Record record  
)
```

Parameters

Parameter	Type	Description
sql	string	SQL query string
record	Record	Optional Record object containing the values that replace the parameter tokens (?) in the SQL query.

Remarks

Win32 MSI APIs: [MsiDatabaseOpenView](#) , [MsiViewExecute](#)

Exceptions

Exception	Description
-----------	-------------

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ExecuteIntegerQuery(sqlFormat, args) Method

Executes the specified SQL SELECT query and returns all results as integers.

Declaration

```
public IList<System.Int32> ExecuteIntegerQuery(  
    string sqlFormat,  
    System.Object[] args  
)
```

Parameters

Parameter	Type	Description
sqlFormat	string	SQL query string, which may contain format items

Parameter	Type	Description
args	System.Object[]	Zero or more objects to format

Return value

`IList<System.Int32>` All results combined into an array

Remarks

The *sqlFormat* parameter is formatted using «see `M:System.String.Format(System.String,System.Object[])`». Multiple rows columns will be collapsed into a single one-dimensional list. Win32 MSI APIs: [MsiDatabaseOpenView](#) , [MsiViewExecute](#) , [MsiViewFetch](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ExecuteIntegerQuery(sql, record) Method

Executes the specified SQL SELECT query and returns all results as integers.

Declaration

```
public IList<System.Int32> ExecuteIntegerQuery(  
    string sql,  
    Record record  
)
```

Parameters

Parameter	Type	Description
sql	string	SQL SELECT query string
record	Record	Optional Record object containing the values that replace the parameter tokens (?) in the SQL query.

Return value

`IList<System.Int32>` All results combined into an array

Remarks

Multiple rows columns will be collapsed into a single one-dimensional list. Win32 MSI APIs: [MsiDatabaseOpenView](#) , [MsiViewExecute](#) , [MsiViewFetch](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ExecutePropertyQuery(property) Method

Returns the value of the specified property.

Declaration

```
public string ExecutePropertyQuery(  
    string property  
)
```

Parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
property	string	Name of the property to retrieve.

ExecuteQuery(sqlFormat, args) Method

Executes the specified SQL SELECT query and returns all results.

Declaration

```
public System.Collections.IList ExecuteQuery(  
    string sqlFormat,  
    System.Object[] args  
)
```

Parameters

Parameter	Type	Description
sqlFormat	string	SQL query string, which may contain format items
args	System.Object[]	Zero or more objects to format

Return value

`System.Collections.IList` All results combined into an array

Remarks

The *sqlFormat* parameter is formatted using «see `M:System.String.Format(System.String,System.Object[])`» . Multiple rows columns will be collapsed into a single one-dimensional list. Win32 MSI APIs: [MsiDatabaseOpenView](#) , [MsiViewExecute](#) , [MsiViewFetch](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ExecuteQuery(sql, record) Method

Executes the specified SQL SELECT query and returns all results.

Declaration

```
public System.Collections.IList ExecuteQuery(  
    string sql,  
    Record record  
)
```

sql	string	SQL SELECT query string
record	Record	Optional Record object containing the values that replace the parameter tokens (?) in the SQL query.

Return value

`System.Collections.IList` All results combined into an array

Remarks

Multiple rows columns will be collapsed into a single one-dimensional list. Win32 MSI APIs: [MsiDatabaseOpenView](#) , [MsiViewExecute](#) , [MsiViewFetch](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ExecuteScalar(sqlFormat, args) Method

Executes the specified SQL SELECT query and returns a single result.

Declaration

```
public System.Object ExecuteScalar(  
    string sqlFormat,  
    System.Object[] args  
)
```

Parameters

Parameter	Type	Description
sqlFormat	string	SQL query string, which may contain format items
args	System.Object[]	Zero or more objects to format

Return value

`System.Object` First field of the first result

Remarks

The *sqlFormat* parameter is formatted using «see `M:System.String.Format(System.String,System.Object[])`» . Win32 MSI APIs:

[MsiDatabaseOpenView](#) , [MsiViewExecute](#) , [MsiViewFetch](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
or the query returned 0 results	
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ExecuteScalar(sql, record) Method

Executes the specified SQL SELECT query and returns a single result.

Declaration

```
public System.Object ExecuteScalar(  
    string sql,  
    Record record  
)
```

Parameters

Parameter	Type	Description
sql	string	SQL SELECT query string
record	Record	Optional Record object containing the values that replace the parameter tokens (?) in the SQL query.

Return value

`System.Object` First field of the first result

Remarks

Win32 MSI APIs: [MsiDatabaseOpenView](#) , [MsiViewExecute](#) , [MsiViewFetch](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
or the query returned 0 results	
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ExecuteStringQuery(sqlFormat, args) Method

Executes the specified SQL SELECT query and returns all results as strings.

Declaration

```
public IList<System.String> ExecuteStringQuery(  
    string sqlFormat,  
    System.Object[] args  
)
```

Parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
sqlFormat	string	SQL query string, which may contain format items
args	System.Object[]	Zero or more objects to format

Return value

`IList<System.String>` All results combined into an array

Remarks

The *sqlFormat* parameter is formatted using «see `M:System.String.Format(System.String,System.Object[])`». Multiple rows columns will be collapsed into a single on-dimensional list. Win32 MSI APIs: [MsiDatabaseOpenView](#) , [MsiViewExecute](#) , [MsiViewFetch](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ExecuteStringQuery(sql, record) Method

Executes the specified SQL SELECT query and returns all results as strings.

Declaration

```
public IList<System.String> ExecuteStringQuery(  
    string sql,  
    Record record  
)
```

Parameters

Parameter	Type	Description
sql	string	SQL SELECT query string
record	Record	Optional Record object containing the values that replace the parameter tokens (?) in the SQL query.

Return value

`IList<System.String>` All results combined into an array

Remarks

Multiple rows columns will be collapsed into a single on-dimensional list. Win32 MSI APIs: [MsiDatabaseOpenView](#) , [MsiViewExecute](#) , [MsiViewFetch](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

Export(table, exportFilePath) Method

Copies the structure and data from a specified table to a text archive file.

Declaration

```
public void Export(  
    string table,  
    string exportFilePath  
)
```

Parameters

Parameter	Type	Description
table	string	Name of the table to be exported
exportFilePath	string	Path to the file to be created

Remarks

Win32 MSI API: [MsiDatabaseExport](#)

Exceptions

Exception	Description
T:System.IO.FileNotFoundException	the file path is invalid
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ExportAll(directoryPath) Method

Exports all database tables, streams, and summary information to archive files.

Declaration

```
public void ExportAll(  
    string directoryPath  
)
```

Parameters

Parameter	Type	Description
directoryPath	string	Path to the directory where archive files will be created

Remarks

The directory will be created if it does not already exist. Win32 MSI API: [MsiDatabaseExport](#)

Exceptions

Exception	Description
T:System.IO.FileNotFoundException	the directory path is invalid
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

FromHandle(handle, ownsHandle) Method

Creates a new Database object from an integer database handle.

Declaration

```
public static Database FromHandle(  
    IntPtr handle,  
    bool ownsHandle  
)
```

Parameters

Parameter	Type	Description
handle	IntPtr	Integer database handle
ownsHandle	bool	true to close the handle when this object is disposed

Remarks

This method is only provided for interop purposes. A Database object should normally be obtained from «see `P:WixToolset.Dtf.WindowsInstaller.Session.Database`» or a public Database constructor.

GenerateTransform(referenceDatabase, transformFile) Method

Creates a transform that, when applied to the object database, results in the reference database.

Declaration

```
public bool GenerateTransform(  
    Database referenceDatabase,  
    string transformFile  
)
```

Parameters

Parameter	Type	Description
referenceDatabase	Database	Database that does not include the changes
transformFile	string	Name of the generated transform file, or null to only check whether or not the two database are identical

Return value

`bool` true if a transform is generated, or false if a transform is not generated because there are no differences between the two

databases.

Remarks

A transform can add non-primary key columns to the end of a table. A transform cannot be created that adds primary key columns to a table. A transform cannot be created that changes the order, names, or definitions of columns. If the transform is to be applied during an installation you must use the «see

M:WixToolset.Dtf.WindowsInstaller.Database.CreateTransformSummaryInfo(WixToolset.Dtf.WindowsInstaller.Database,System.String,WixToolset.Dtf.WindowsInstaller.TransformErrors,WixToolset.Dtf.WindowsInstaller.TransformValidations)» method to populate the summary information stream. Win32 MSI API: [MsiDatabaseGenerateTransform](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the transform could not be generated
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	a Database handle is invalid

Import(importFilePath) Method

Imports a database table from a text archive file, dropping any existing table.

Declaration

```
public void Import(  
    string importFilePath  
)
```

importFilePath	string	Path to the file to be imported. The table name is specified within the file.
----------------	--------	---

Remarks

Win32 MSI API: [MsiDatabaseImport](#)

Exceptions

Exception	Description
T:System.IO.FileNotFoundException	the file path is invalid
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ImportAll(directoryPath) Method

Imports all database tables, streams, and summary information from archive files.

Declaration

```
public void ImportAll(  
    string directoryPath  
)
```

Parameters

Parameter	Type	Description
directoryPath	string	Path to the directory from which archive files will be imported

Remarks

Win32 MSI API: [MsiDatabaseImport](#)

Exceptions

Exception	Description
T:System.IO.FileNotFoundException	the directory path is invalid
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

IsColumnPersistent(table, column) Method

Checks whether a table contains a persistent column with a given name.

Declaration

```
public bool IsColumnPersistent(  
    string table,  
    string column  
)
```

Parameters

Parameter	Type	Description
table	string	The table to the checked
column	string	The name of the column to be checked

Return value

`bool` true if the column exists in the table; false if the column is temporary or does not exist.

Remarks

To check whether a column exists regardless of persistence, use «see `M:WixToolset.Dtf.WindowsInstaller.ColumnCollection.Contains(System.String)`» .

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the View could not be executed
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

IsTablePersistent(table) Method

Checks whether a table exists and is persistent in the database.

Declaration

```
public bool IsTablePersistent(  
    string table  
)
```

Parameters

Parameter	Type	Description
table	string	The table to the checked

Return value

`bool` true if the table exists and is persistent in the database; false otherwise

Remarks

To check whether a table exists regardless of persistence, use «see

M:WixToolset.Dtf.WindowsInstaller.TableCollection.Contains(System.String)» . Win32 MSI API: [MsiDatabasesIsTablePersistent](#)

Exceptions

Exception	Description
T:System.ArgumentException	the table is unknown
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

IsTransformValid(transformFile) Method

Checks whether a transform is valid for this Database, according to its validation data and flags.

Declaration

```
public bool IsTransformValid(  
    string transformFile
```

```
)
```

Parameters

Parameter	Type	Description
transformFile	string	Path to the transform file

Return value

`bool` true if the transform can be validly applied to this Database; false otherwise

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the transform could not be applied
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

IsTransformValid(transformSummaryInfo) Method

Checks whether a transform is valid for this Database, according to its SummaryInfo data.

Declaration

```
public bool IsTransformValid(  
    SummaryInfo transformSummaryInfo  
)
```

Parameters

Parameter	Type	Description
transformSummaryInfo	SummaryInfo	SummaryInfo data of a transform file

Return value

`bool` true if the transform can be validly applied to this Database; false otherwise

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	error processing summary info
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database or SummaryInfo handle is invalid

Merge(otherDatabase, errorTable) Method

Merges another database with this database.

Declaration

```
public void Merge(  
    Database otherDatabase,  
    string errorTable  
)
```

Parameters

Parameter	Type	Description
otherDatabase	Database	The database to be merged into this database
errorTable	string	Optional name of table to contain the names of the tables containing merge conflicts, the number of conflicting rows within the table, and a reference to the table with the merge conflict.

Remarks

Merge does not copy over embedded cabinet files or embedded transforms from the reference database into the target database. Embedded data streams that are listed in the Binary table or Icon table are copied from the reference database to the target database. Storage embedded in the reference database are not copied to the target database. The Merge method merges the data

of two databases. These databases must have the same codepage. The merge fails if any tables or rows in the databases conflict. A conflict exists if the data in any row in the first database differs from the data in the corresponding row of the second database. Corresponding rows are in the same table of both databases and have the same primary key in both databases. The tables of non-conflicting databases must have the same number of primary keys, same number of columns, same column types, same column names, and the same data in rows with identical primary keys. Temporary columns however don't matter in the column count and corresponding tables can have a different number of temporary columns without creating conflict as long as the persistent columns match. If the number, type, or name of columns in corresponding tables are different, the schema of the two databases are incompatible and the installer will stop processing tables and the merge fails. The installer checks that the two databases have the same schema before checking for row merge conflicts. If the schemas are incompatible, the databases have be modified. If the data in particular rows differ, this is a row merge conflict, the merge fails and creates a new table with the specified name. The first column of this table is the name of the table having the conflict. The second column gives the number of rows in the table having the conflict. Win32 MSI API: [MsiDatabaseMerge](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.MergeException	merge failed due to a schema difference or data conflict
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

Merge(otherDatabase) Method

Merges another database with this database.

Declaration

```
public void Merge(  
    Database otherDatabase  
)
```

Parameters

Parameter	Type	Description
otherDatabase	Database	The database to be merged into this database

Remarks

MsiDatabaseMerge does not copy over embedded cabinet files or embedded transforms from the reference database into the target database. Embedded data streams that are listed in the Binary table or Icon table are copied from the reference database to the target database. Storage embedded in the reference database are not copied to the target database. The Merge method merges the data of two databases. These databases must have the same codepage. The merge fails if any tables or rows in the databases conflict. A conflict exists if the data in any row in the first database differs from the data in the corresponding row of the second database. Corresponding rows are in the same table of both databases and have the same primary key in both databases. The tables of non-conflicting databases must have the same number of primary keys, same number of columns, same column types, same column names, and the same data in rows with identical primary keys. Temporary columns however don't matter in the column count and corresponding tables can have a different number of temporary columns without creating conflict as long as the persistent columns match. If the number, type, or name of columns in corresponding tables are different, the schema of the two databases are incompatible and the installer will stop processing tables and the merge fails. The installer checks that the two

databases have the same schema before checking for row merge conflicts. If the schemas are incompatible, the databases have be modified. Win32 MSI API: [MsiDatabaseMerge](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.MergeException	merge failed due to a schema difference or data conflict
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

OpenView(sqlFormat, args) Method

Gets a View object representing the query specified by a SQL string.

Declaration

```
public View OpenView(  
    string sqlFormat,  
    System.Object[] args  
)
```

Parameters

Parameter	Type	Description
sqlFormat	string	SQL query string, which may contain format items
args	System.Object[]	Zero or more objects to format

Return value

View A View object representing the query specified by a SQL string

Remarks

The *sqlFormat* parameter is formatted using «see `M:System.String.Format(System.String,System.Object[])`». The View object should be «see `M:WixToolset.Dtf.WindowsInstaller.InstallerHandle.Close`» d after use. It is best that the handle be closed manually as soon as it is no longer needed, as leaving lots of unused handles open can degrade performance. Win32 MSI API: [MsiDatabaseOpenView](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.BadQuerySyntaxException	the SQL syntax is invalid
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

ToString() Method

Returns the file path of this database, or the handle value if a file path was not specified.

Declaration

```
public string ToString()
```

ViewTransform(transformFile) Method

Apply a transform to the database, recording the changes in the "_TransformView" table.

Declaration

```
public void ViewTransform(  
    string transformFile  
)
```

Parameters

Parameter	Type	Description
transformFile	string	Path to the transform file

Remarks

Win32 MSI API: [MsiDatabaseApplyTransform](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InstallerException	the transform could not be applied
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

CodePage Property

Gets or sets the code page of the Database.

Declaration

```
public int CodePage { get; set; }
```

Remarks

Getting or setting the code page is a slow operation because it involves an export or import of the codepage data to/from a temporary file.

Exceptions

Exception	Description
T:System.IO.IOException	error exporting/importing the codepage data
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

FilePath Property

Gets the file path the Database was originally opened from, or null if not known.

Declaration

```
public string FilePath { get; set; }
```

IsReadOnly Property

Gets a boolean value indicating whether this database was opened in read-only mode.

Declaration

```
public bool IsReadOnly { get; set; }
```

Win32 MSI API: [MsiGetDatabaseState](#)

OpenMode Property

Gets the open mode for the database.

Declaration

```
public DatabaseOpenMode OpenMode { get; set; }
```

SummaryInfo Property

Gets the SummaryInfo object for this database that can be used to examine and modify properties to the summary information stream.

Declaration

```
public SummaryInfo SummaryInfo { get; set; }
```

Remarks

The object returned from this property does not need to be explicitly persisted or closed. Any modifications will be automatically saved when the database is committed. Win32 MSI API: [MsiGetSummaryInformation](#)

Exceptions

Exception	Description
T:WixToolset.Dtf.WindowsInstaller.InvalidHandleException	the Database handle is invalid

Tables Property

Gets the collection of tables in the Database.

Declaration

```
public TableCollection Tables { get; set; }
```