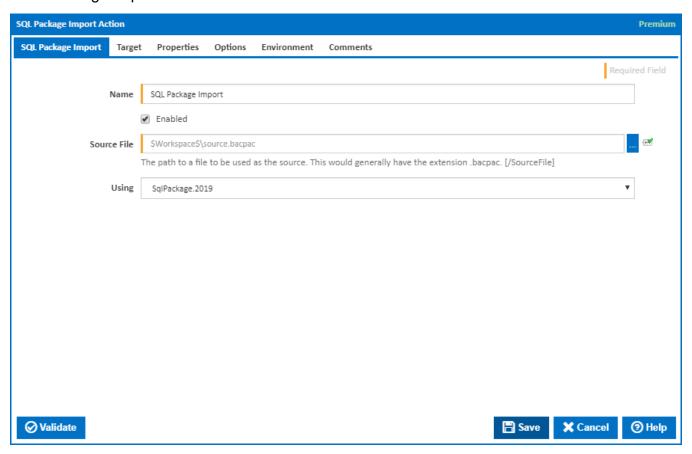# SQL Package Import Action

> ℹ️ The SQL Package actions in Continua are a wrapper around the SqlPackage command line. If you're having trouble using any of the SQL Package actions, please refer to the Command Line Reference.

The SQL Package Import action is a premium action that imports the schema and table data from a BACPAC package - .bacpac file - into a new or empty database in SQL Server or Azure SQL Database.

Note: At the time of the import operation to an existing database, the target database cannot contain any user-defined schema objects.

SQL Package Import

Name

A friendly name for this action (will be displayed in the actions workflow area).

Enabled

Determines if this action will be run within the relevant stage.

Source File

The path to a file to be used as the source. This would generally have the extension .bacpac. [/SourceFile]

Using

The Using drop down is populated with any property collector whose namespace matches the pattern defined by the SQL Package Import action. The pattern for this action is `^SqlPackage\..*`

If you create a property collector for this action, make sure you select the **Path Finder PlugIn** type and give it a name that will match the pattern above in blue. Example names listed here, search the table's Plugin column for "**SQL Package Import**".

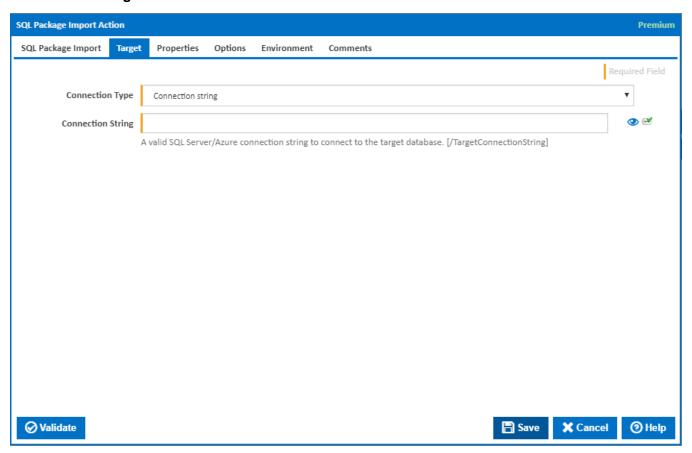For more in-depth explanations on property collectors see Property Collectors.

Alternatively, you can select the **Custom** option from the Using drop down list and specify a path in the resulting input field that will be displayed. Please read Why it's a good idea to use a property collector before using this option.

Target

Connection Type

Whether to use a connection string or specify each part of the connection separately.
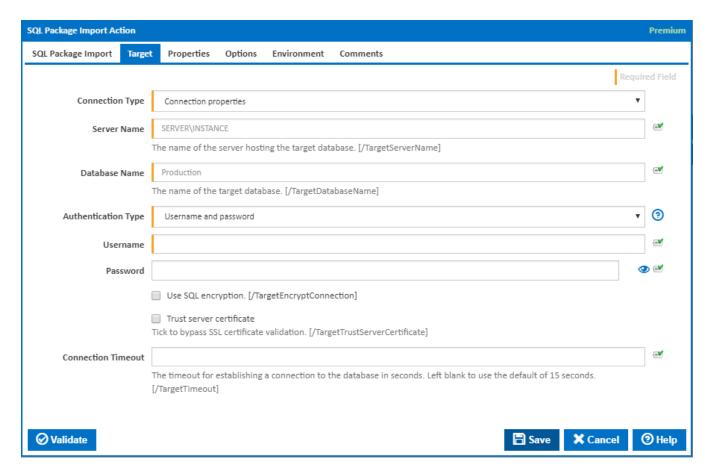
Connection String

Connection String

The SQL Server/Azure connection string to connect to the database. [/TargetConnectionString]

An example of a valid connection string is:

```
Provider=SQLNCLI11;Password=myPassword;User ID=[username]@[servername];Initial Catalog=databasename;Data
Source=tcp:[servername].database.windows.net;
```

Connection Properties

Server Name

The name of the server that hosts the database. The server should be accessible by the agent running the action. [/TargetServerName]

Database Name

The name of the source database. [/TargetDatabaseName]

Authentication Type

The authentication method. Either by using a Username and Password or Windows Authentication.

Username and Password

Username

The name of the user who has permissions to perform the operation on the database. [/TargetUser]

Password

The password for the user connecting to the database. [/TargetPassword]

Windows Authentication

If this is selected, windows authentication is used to gain permission to perform the operation on the database.

Use SQL encryption

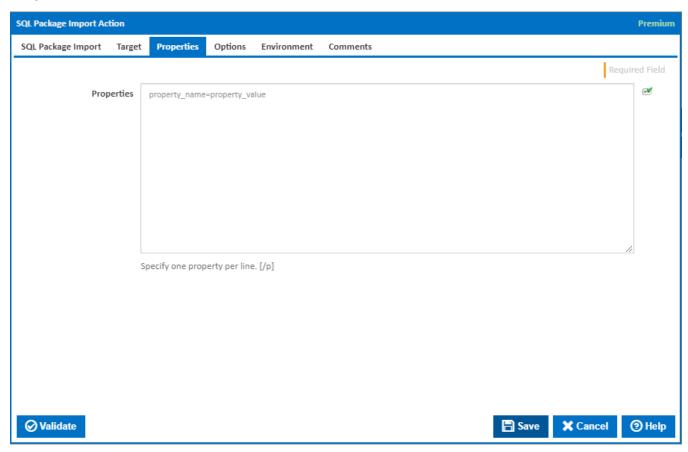If this is ticked, SQL encryption is used. [/TargetEncryptConnection]

Trust server certificate

If this is ticked, SSL certificate validation is bypassed. [/TargetTrustServerCertificate]

Connection Timeout

The timeout for establishing a connection to the database in seconds. Leave blank to use the default of 15 seconds. [/TargetTimeout]
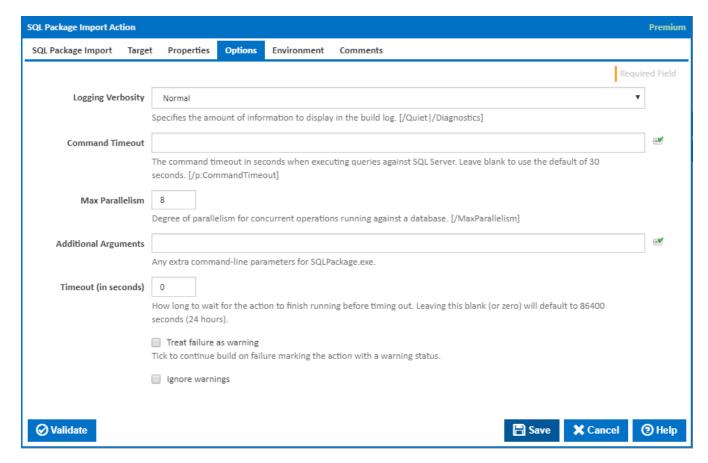
Properties

Properties

Specify one property per line in the format of **property_name=property_value**. The full list of properties can be found at SqlPackage.exe. [/p:]

Options

Logging Verbosity

Specifies the amount of information to display in the build log. [/Quiet | /Diagnostics]

Command Timeout

The command timeout in seconds when executing queries against SQL Server. Leave blank to use the default of 30 seconds. [/p:CommandTimeout=]

Max Parallelism

Degree of parallelism for concurrent operations running against a database. [/MaxParallelism]

Additional Arguments

Any extra command-line parameters for SQLPackage.exe. The full list of extra arguments can be found at SqlPackage.exe.

Timeout (in seconds)

How many seconds to wait for the action to finish before timing out. The default is 86400 seconds (24 hours).
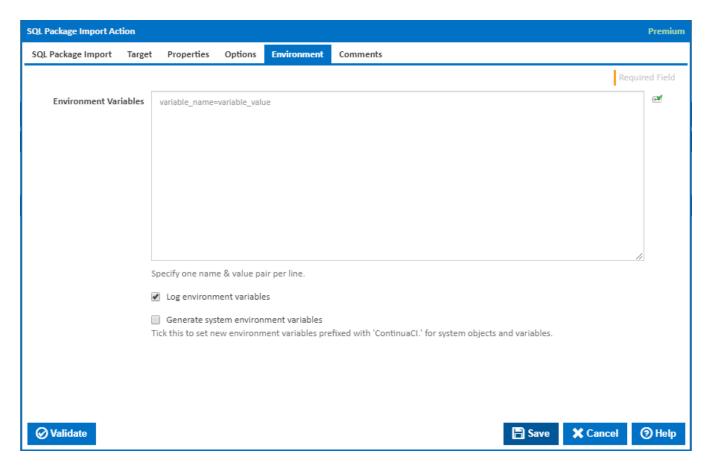
Treat failure as warning

Tick to continue build on failure marking the action with a warning status.

Ignore warnings

If this is ticked, any warnings logged will not mark the action with a warning status.

Environment

Environment Variables

Multiple environment variables can be defined - one per line. These are set before the command line is run.

Log environment variables

If this is ticked, environment variable values are written to the build log.

Generate system environment variables

Tick this checkbox to set up a list of new environment variables prefixed with 'ContinuaCI.' for all current system expression objects and variables.

Mask sensitive variable values in system environment variables

This checkbox is visible only if the '**Generate system environment variables**' checkbox is ticked.

If this is ticked, the values of any variables marked as sensitive will be masked with **** when setting system environment variables. Clear this to expose the values.