 RED   
Templates and Extended Properties  
Standards & Guidelines

WhereScape Ltd.

March 2018 Version 8.0.1.0

|  |
| --- |
|  |
| |  |  |  |  | | --- | --- | --- | --- | | WhereScape Europe Ltd |  | WhereScape USA |  | | 450 Brook Drive  Green Park  Reading  RG2 6UU  United Kingdom |  | 1915 NW AmberGlen Parkway  Beaverton  Oregon  97006  United States of America |  | | T: +44 (0)118-914-4509  F: +44 (0)118-914-4508 |  | T: +1 (503) 466-3979 |  | |
|  |
|  |

Overview

This document provides standards and guidelines regarding the development of templates. A standard is noted by S: to preface the instruction. A guideline will be denoted via G:

The intent of this document is to be WhereScape target platform independent, although there will be platform specific content throughout the document.

Table of Contents

[Introduction 3](#_Toc509944551)

[Using Templates 4](#_Toc509944552)

[Template Syntax 5](#_Toc509944553)

[Creating a Template 6](#_Toc509944554)

[Specifying which Templates to Use 8](#_Toc509944555)

[Extended Properties 10](#_Toc509944556)

[Standards & Guidelines 13](#_Toc509944557)

[Template Development 13](#_Toc509944558)

[Load scripts 13](#_Toc509944559)

# Introduction

This Training Guide serves as an Introduction to using WhereScape RED Templates. This guide is not meant to be a detailed step-by-step training manual. It introduces the functionality, subjects, topics and concepts, that the reader will need to understand when working with WhereScape RED Templates.

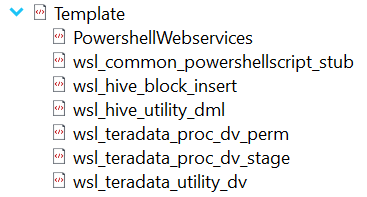
It is assumed that the reader of this guide has completed the WhereScape RED Basics training and has had some experience of building data warehouses using WhereScape RED. You are expected to have a good working knowledge of the RED UI and be able to comfortably navigate your way around the RED UI.

At its starting point this training also assumes that a WhereScape RED metadata repository has been set up and configured as per WhereScape’s standard instructions for creating a metadata repository on a host platform.

# Using Templates

To support all of the various Custom Target platforms that WhereScape RED can now work with, WhereScape provide templates for the code generation. Details on Template use can be found in Section 25 of RED’s Help. There are three main types of Templates that you will use:

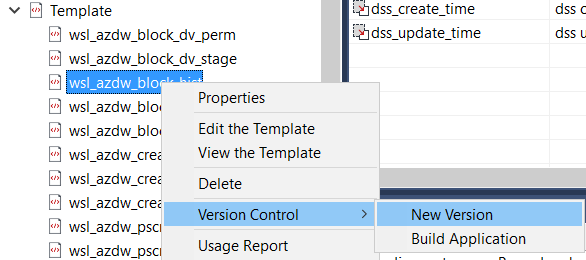
1. Templates for DDL generation e.g. CREATE TABLE or CREATE VIEW.
2. Templates for PowerShell script generation.
3. Templates for SQL code Block or Procedure generation.



You will also find Utility templates that contain common functions that are Imported/referenced by the other types. All of the WhereScape provided Templates have the **wsl\_** prefix in their name. These templates should not be modified by the user, as any upgrade to a new version of RED may replace them. If you require variations on the templates copy the template, give it a name of your own following your naming standards and modify and use the copy.

To copy a template:

* Right click on the Template you wish to copy and choose **Version Control | New Version**



* Give it a new version name of “for Copy” or something similar.
* Right click on the **Template** Object Type header and choose **New Object (from Version)**. Find your “for Copy” version of the Template in the list give it a new name.
* You can then edit the copy by right clicking on it and choosing **Edit the Template.**

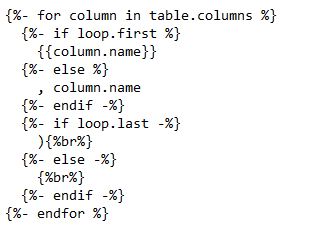
All Templates are written using the open source java Pebble language. See <http://www.mitchellbosecke.com/pebble/home> for more details of its use. The following describes the main syntax features of Pebble.

## Template Syntax

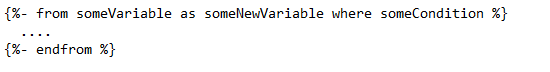
|  |  |  |
| --- | --- | --- |
| Syntax | Description | Examples |
|  |  |  |
| {# #}  # at the start of a line | Comment start and end | {# This is a comment #} |
| {% *command* %} | A Pebble command | {% br %} forces a new line  {% if settings.DeleteBeforeInsert %} |
| {% set variable = 0 %} | Variable declaration and initialization | {% set counter = 0 %} |
| {{ variable }} | Use or print a variable’s value | {{counter}}  {{table.name}}  {{table.loadInfo.sourceConnection.jdbcDriverClass}} |
| {{-variable}}  {%- *command* %} | Trim any spaces from before a variable or command | {{-table.name}} |
| {{variable-}}  {% *command* -%} | Trim any spaces from after a variable or command | {{table.name-}} |
| {{-variable-}}  {%- *command* %} | Trim any spaces from before and after a variable or command | {{-table.name-}} |
| {% if some condition %} …  {% elseif another cond %} …  {% *else* %}  …  {% *endif* %} | Logical if statement |  |

These Logical structures are most commonly used:

FOR loop

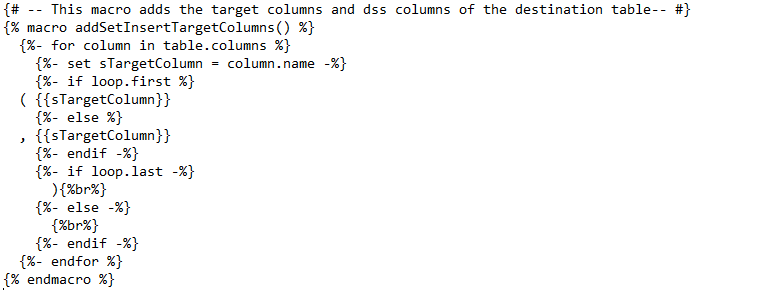


FROM loop

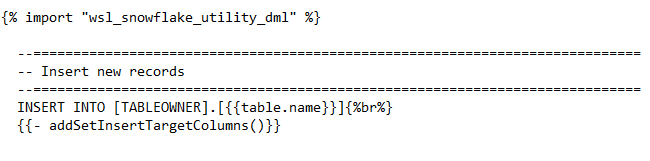


Macros

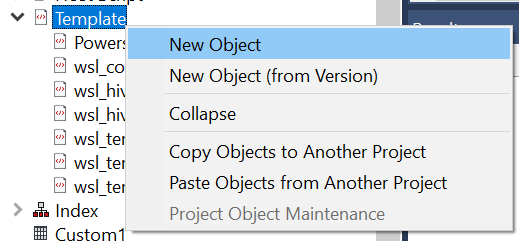
Macros are re-usable functions that generate code. Often used as Utility functions.



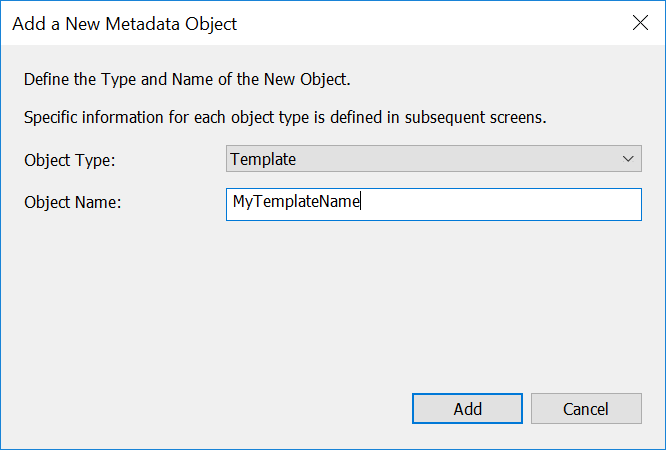
A macro is declared and called as shown here:



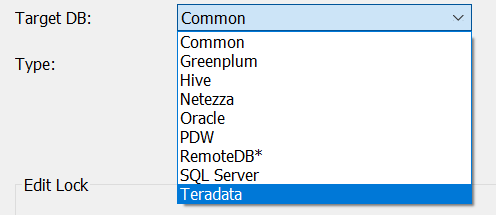
## Creating a Template



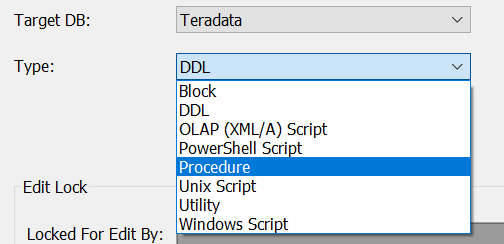
Give your template a name



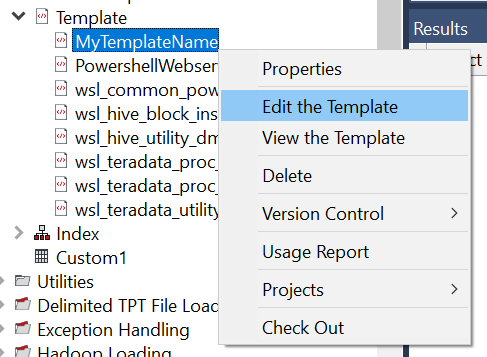
Fill in the Purpose of the Template and choose the Target DB it is designed to be used on.



Choose the type of template:



Choose to Edit the Template to open the Template Editor

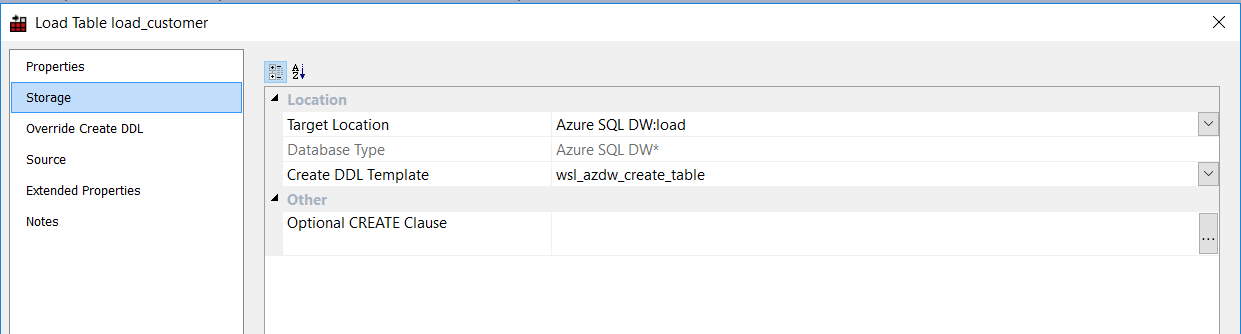


You could now author your template from scratch or cut and paste from other templates.

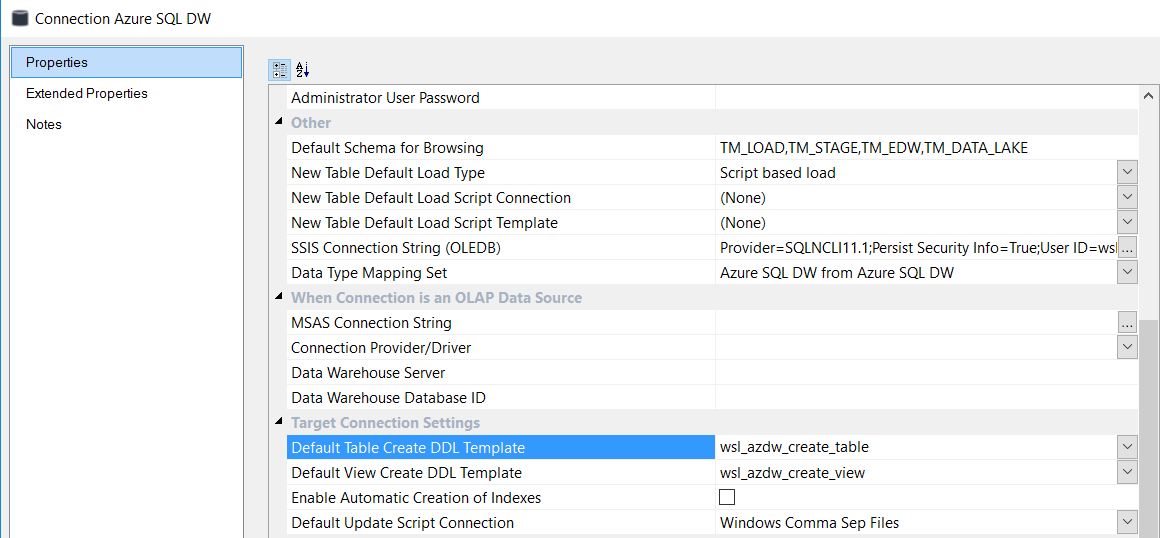
**Note** you could use the **Tools | Create API Example Template** to dump all the pre-declared variables for a template type.

## Specifying which Templates to Use

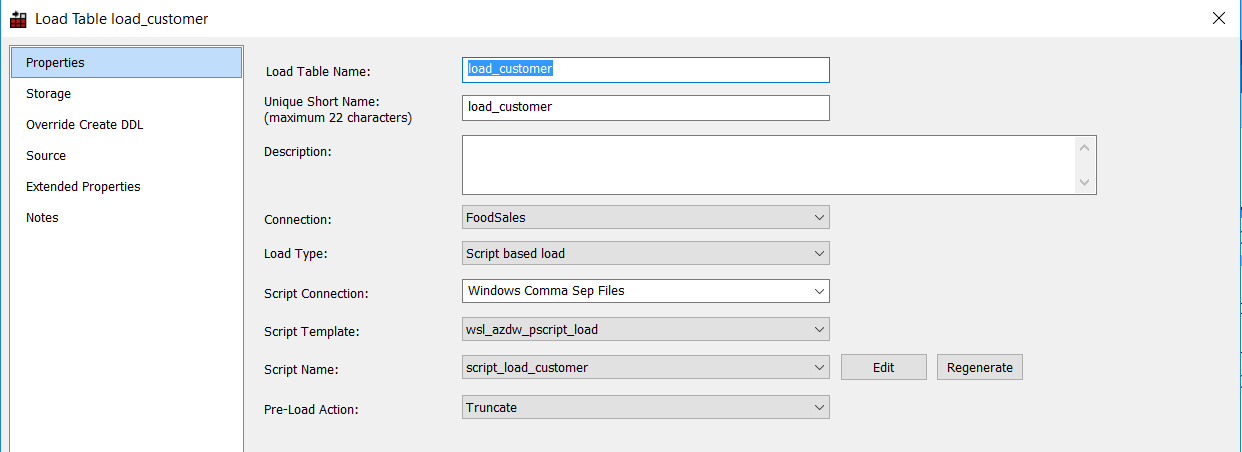
A DDL Template is specified in the Storage Property of the Table or View object.



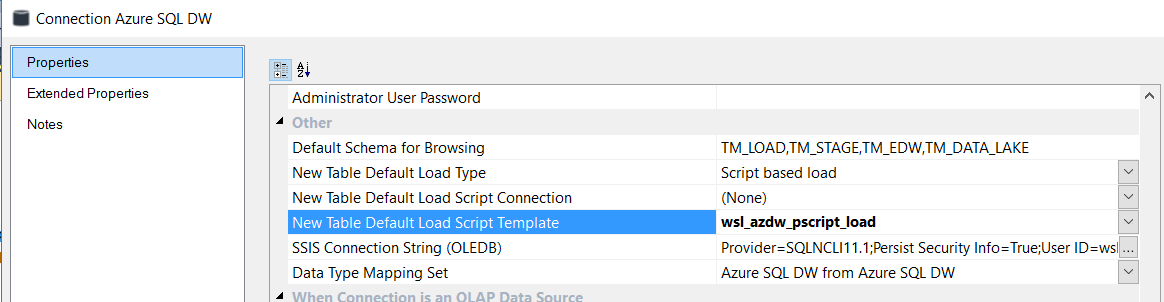
The default DDL Template is specified in the Properties of the Target Connection



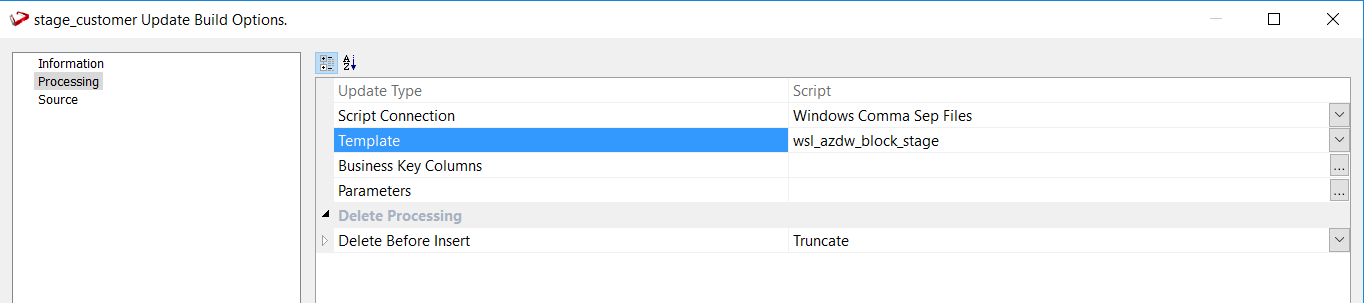
A PowerShell script Template is specified in the Script Template property of the Table or View



The default PowerShell Script template to use for data loading is specified in the Properties of the Target Connection



The Template to use when generating a SQL code block is specified when you **Build** or **Rebuild** the procedure code.

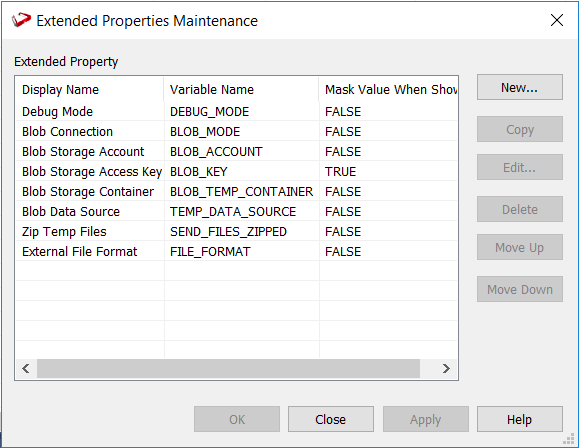


There is no place to specify a default SQL code block Template in RED.

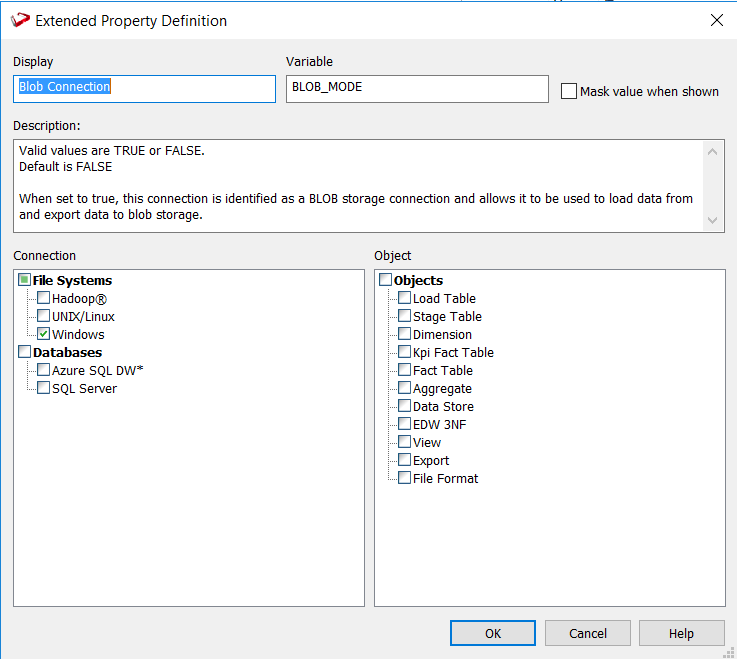
# Extended Properties

Extended Properties are used throughout WhereScape RED. They provide information to further complement the RED metadata and are used by script and code generation Templates to generate the desired scripts and code in RED. Detailed information on Extended Properties usage can be found in section 45 of RED’s Help.

You can add, edit or remove Extended Properties of your own. On RED’s menu bar choose **Tools | Extended Properties | Maintain Extended Properties**



Click on **New** to create a new Extended Property or double click on an existing Extended Property to edit it.



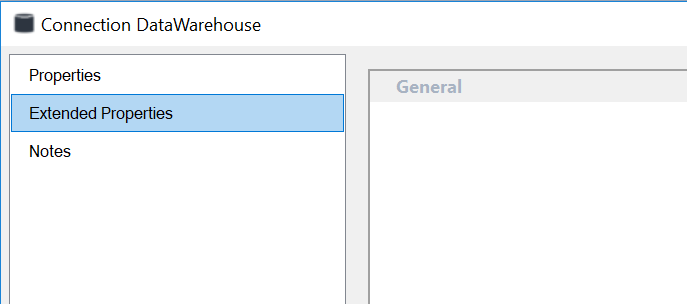
You need to give an Extended Property:

A **Display** Name**:**  the name by which it will be exposed in the Properties.

A **Variable** Name: the name by which it will be identified in a Template.

You also need to identify which RED object types with which the Extended Property is to be related and with which it will be used in conjunction.

The Extended Properties available to the RED developer will appear on the left hand side of the Property Grids to which they have been assigned.



Extended Properties can be referenced in Templates or Powershell scripts by the following syntax:

*<object\_type>.*extendedPropertyValuesByName.*<ExtendedPropertyVariableName>*

For Example:

{% set newLine = "" %}

{%- set list = table.extendedPropertyValuesByName.SourceSystem -%}

{%- for item in (list.Replace(';',newLine)|lines) -%}

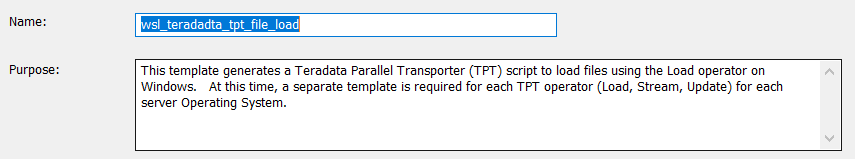
  Do something with this value: {{item}}{%br%}

{%- endfor %}

# Standards & Guidelines

## Template Development

S: Templates should be documented in the purpose window within the properties page:



## Load scripts

S: Use the same script output variable – here shown as %CTL\_FILE% in all templates that generate a script. This will facilitate portability between scripts.

ECHO USING CHARACTER SET {{table.loadInfo.sourceFile.charSet}} > %CTL\_FILE%