



materialise

innovators you can count on

Template Script for Magics Automation Module

User Guide

1 Introduction

To make it easier to get started, a template smart script is provided which can be copied for new smart scripts. It contains example for common approaches like passing messages to Magics Automation Module or for localization.

This documentation should give an overview of this template script.

2 Meta data

Each smart script should be accompanied by an additional json file that describes the following properties about the script

- Name
- Description
- Version
- CompatibilityVersion (compatible Magics Automation Module version)
- FileName
- CreatedBy
- DocumentationUrl
- Id (unique GUID to identify the smart script)
- SchemaVersion
- Contract
 - Expected input files
 - Expected output files
 - Expected input parameters as json schema

The required fields are:

- Name
- Version
- CompatibilityVersion
- FileName
- Id
- SchemaVersion
- Contract

The meta data json of the template script contains all necessary information and parameters of each types and an example for localization of parameters.

2.1. Contract

There are 3 different types of parameters

- Input files: These are needed as input for the algorithm
- Output files: These are the outcome of the algorithm
- Script parameters: These are extra parameters for the algorithm

The template scripts has multiple input files, one output files and multiple script parameters. They are examples to describe different types of parameter and files with different properties.

2.1.1 Input/Output files

To describe the input or output files, the following properties are used:

- Name of the parameter
- Expected format of the files
- The type of path expected (file or folder)

One input and one output file must be defined and are not shown to the user. Additionally, the user can be asked to give some more files or a folder as input. Here the file selection

dialogue can get more options. `format` and `FileTypes` can be used both and will be combined, they define which file types will be allowed for selection. By enabling `MultiSelect` the parameters in the python command will change from a single file to multiple files

2.1.2 Script parameters

The script parameters are defined in a json schema. Also input files which should be visible in the parameter window of Magics Automation Module should be defined here. For the template script, all input files except `'input_file'` are defined in the schema and thus are visible in the parameter window.

The supported parameter types are the following simple types: string, integer, number, boolean

For each parameter a parameter title, description, type and default value can be set. Depending on the parameter type other properties, like minimum or maximum value, can be added, too. The meta data json of the template script provides such examples. If a parameter needs to be required, the parameter key have to be added to the `'required'` property of the schema.

2.2. Localization

Localization of the meta data is supported and is done using additional json files. The string for the smart script name and description as well as the strings for parameter titles and descriptions are extracted to separated json files. For each language a separated json file have to be created. The pattern for the json file name have be `'[SmartScriptName]/languages[LanguageCode].json'` and the keys for the same strings has to be equal across all json files.

Inside the meta data json the keys will be used as placeholders. The Magics Automation Module will automatically select the correct string for the selected language according to the key.

The template script provides a parameter with translation as an example for this approach.

3 Script

The python script of the template script contains the basic setup of a smart script. The script doesn't change the input file but copies it to the output. A licensed Mtls API package is required to edit the matamx platform. In order to keep the template as uncomplicated as possible, bundling with such a package has therefore been omitted. But the script provides examples for other common tasks, like parsing arguments or display messages or progress in Magics Automation Module. Each method in the script has also a description about its usage.

3.1. Parsing arguments

When executed, the Magics Automation Module will pass multiple arguments to the script. The first two arguments are the input and output file path for the matamx files, they are passed without keys. The other parameters which are described in the meta data json are passed with key, e.g. `--parameter_name value`. Besides the parameters from the meta data json, there will be additional parameters passed by the Magics Automation Module, e.g. the Magics version or selected language.

It's recommended to use a argument parser of the `argparse` python library. The template script defined such a parser for all parameters. This parser have to be updated accordingly to changes in the meta data json.

3.2. Display messages and progress

During the execution the Magics Automation Module checks the standard output and error outputs of the smart script for specific strings. Depending on the string, the line with this string will be displayed in a message dialog after the execution or will update the current progressbar.

The template script provides a method and an enum to simplify the display of those messages and gives some examples in the main method.

The following message types are supported:

- Info
- Warning
- Error
- Progress

3.3. Localization

The localization of strings in python using babel and gettext is supported and recommended. Babel is used to generate the localization files and gettext is used to load the generated .mo files to translate the extracted strings.

The python script of the template script provides the initialization and setup of the localization. The setup method 'setup_localization' also provides step for how to generate and use the localization.