

---

# **I2tScaffolder Documentation**

**Log2timeline Developers**

**Jun 05, 2023**



---

## Contents

---

<b>1</b>	<b>User documentation</b>	<b>3</b>
1.1	Installing the Tool . . . . .	3
1.1.1	Pip Install . . . . .	3
1.1.2	Install From Sources . . . . .	3
1.2	Use the tool . . . . .	4
1.2.1	Installation . . . . .	4
1.2.2	Preparation . . . . .	4
1.2.3	Using the Tool . . . . .	4
1.3	Setting up L2t scaffolder in a virtualenv . . . . .	5
1.3.1	Fedora Core . . . . .	5
1.3.2	Ubuntu . . . . .	5
1.3.3	Setting up l2t_scaffolder in virtualenv . . . . .	6
1.3.4	Deactivate Virtualenv . . . . .	6
1.3.5	Configuring Git Client . . . . .	6
<b>2</b>	<b>l2tscaffolder package</b>	<b>7</b>
2.1	Subpackages . . . . .	7
2.1.1	l2tscaffolder.definitions package . . . . .	7
2.1.2	l2tscaffolder.frontend package . . . . .	9
2.1.3	l2tscaffolder.helpers package . . . . .	13
2.1.4	l2tscaffolder.lib package . . . . .	15
2.1.5	l2tscaffolder.scaffolders package . . . . .	19
2.1.6	l2tscaffolder.templates package . . . . .	28
2.2	Module contents . . . . .	28
<b>3</b>	<b>Indices and tables</b>	<b>29</b>
	<b>Python Module Index</b>	<b>31</b>
	<b>Index</b>	<b>33</b>



The l2t\_scaffolder is a tool developed to speed up l2t development by automating the generation of plugins and parsers in various tools, such as Plaso and Timesketch.

At the moment there is no documentation to speak of, but it will be provided shortly.

The project's code is available from <https://github.com/log2timeline/l2tscaffolder>

l2tscaffolder is licensed under the Apache license version 2.

Contents:



## 1.1 Installing the Tool

There are two ways to install the tool:

1. Use pip
2. Source from github

Let's cover both ways. But the first recommended step is to setup a virtualenv environment.

Follow the instructions [here](#) or a quick method:

```
$ virtualenv -p /usr/bin/python3 scaffolder  
$ source scaffolder/bin/activate
```

Once the virtual environment is setup you can move on to the next step, either using pip or source installation.

### 1.1.1 Pip Install

To install the latest release of the scaffolder, use:

```
$ pip3 install --upgrade l2tscaffolder
```

### 1.1.2 Install From Sources

First fetch the latest source code from github:

```
$ git clone https://github.com/log2timeline/l2tscaffolder.git
```

Then install dependencies and compile and install the tool:

```
$ cd l2tscaffolder
$ pip3 install -r requirements.txt
$ python3 setup.py build && python3 setup.py install
```

## 1.2 Use the tool

### 1.2.1 Installation

The first step is to install the tool. Follow the instructions [here](#)

### 1.2.2 Preparation

Once the tool is installed you'll need to first fetch the source of whatever project you are adding code to, eg. Plaso. The way log2timeline development is done is that you need to first create a fork of the main project into your own account, clone your fork and then work from that one. Here is an example of fetching and syncing your personal fork:

```
$ git clone https://github.com/kiddinn/plaso.git
$ cd plaso
$ git remote add upstream https://github.com/log2timeline/plaso.git
$ git pull --rebase upstream master
$ git push
```

Once this is ready you can start using the l2t\_scaffolder tool.

### 1.2.3 Using the Tool

The tool will guide you through its use, the parameters are fairly simple:

```
l2t_scaffolder.py [DEFINITION]
```

Where definition is an optional parameter of the name of the project, eg. plaso, timesketch, etc.

The simplest way to run the tool is to run it without any parameters and then follow the questions asked.

```
$ l2t_scaffolder.py
== Starting the scaffolder ==
Gathering all required information.

Available definitions:
[0] plaso
[1] timesketch
Definition choice: 0
plaso chosen.

Path to the project root: plaso
Path [plaso] set as the project path.

Name of the module to be generated. This can be something like "foobar sqlite"
or "event analytics".

This will be used for class name generation and file name prefixes.
```

(continues on next page)



(continued from previous page)

Module Name:

...

After that it is a simple manner of following the instructions given by the tool.

Some notes:

- “*Name of the module*”: this is used to create both the name of the class as well as filenames of the generated files, so if you choose something like: “New Awesome Parser” you’ll end up with a parser/plugin file with the name of new\_awesome\_parser.py and a class name on the lines of NewAwesomeParserParser (depending on the scaffolder some text may be appended to the class name).
- Each scaffolder will determine what questions need to be asked in order to successfully generate files, some may ask more than others, eg. the SQLite plugin will ask for SQL commands, and names of functions. That will be used to generate the skeleton of the code.
- Once the tool has collected all answers to questions it will generate the required files, what it will do is:
  - Create a feature branch inside the git repository
  - Generate all the necessary files
  - Add those files to the git client

Once the tool completes it’s run, you can go to the git repo of the project you just generated the files and start completing them. The tool uses a template, often filled with TODOs or missing parts that need to be completed before the plugin/parser is ready for use. However it should get you started by generating all the necessary files as well as filling out the boiler plate code that is often needed.

## 1.3 Setting up L2t scaffolder in a virtualenv

For development purposes, l2tscaffolder can be installed using virtualenv (preferred method).

### 1.3.1 Fedora Core

#### Install virtualenv

To install virtualenv on Fedora Core (or equivalent) run:

```
$ sudo dnf install python3-virtualenv
```

#### Installing build dependencies

TODO add more text

### 1.3.2 Ubuntu

#### Installing virtualenv

To install virtualenv on Ubuntu (or equivalent) run:

```
$ sudo apt-get install python-virtualenv python3-virtualenv
```

### Installing build dependencies

**TODO add more text**

```
$ sudo apt-get install libyaml-dev liblzma-dev
```

### 1.3.3 Setting up I2t\_scaffolder in virtualenv

To create a virtualenv:

```
virtualenv -p PATH_TO_PYTHON3 scaffolderoenv
```

eg:

```
$ virtualenv -p /usr/bin/python3 scaffolderoenv
```

To activate the virtualenv:

```
$ source ./scaffolderenv/bin/activate
```

**Note that using pip outside virtualenv is not recommended since it ignores your systems package manager.**

Make sure that pip is up-to-date:

```
$ pip3 install --upgrade pip
```

### 1.3.4 Deactivate Virtualenv

To deactivate the virtualenv run:

```
$ deactivate
```

### 1.3.5 Configuring Git Client

Before submitting code to the project, make sure that you have created a fork of the I2tscaffolder project, and check out your personal fork:

```
$ git clone https://github.com/USERNAME/I2tscaffolder.git
```

Add the upstream repo:

```
$ git remote add upstream https://github.com/log2timeline/I2tscaffolder.git
```

And then you can create a feature branch to work on.

```
$ git checkout -b my_feature
```

## 2.1 Subpackages

### 2.1.1 l2tscaffolder.definitions package

#### Submodules

##### l2tscaffolder.definitions.interface module

Interface defining how a project class looks like.

**class** l2tscaffolder.definitions.interface.ScaffolderDefinition

Bases: object

Scaffolder definition interface.

**NAME** = 'undefined'

**ValidatePath** (*root\_path: str*) → bool

Validates the path to the root directory of the project.

**Parameters** **root\_path** (*str*) – the path to the root of the project directory.

**Returns** whether the given path is the correct root path of the project.

**Return type** bool

##### l2tscaffolder.definitions.manager module

The definition manager.

**class** l2tscaffolder.definitions.manager.DefinitionManager

Bases: object

The definition manager.

**classmethod DeregisterDefinition** (*definition\_class*: *Type[l2tscaffolder.definitions.interface.ScaffolderDefinition]*)  
Deregisters a definition class.

Definition classes are identified by their NAME attribute.

**Parameters** *definition\_class* (*type*) – definition class (subclass of ScaffolderDefinition).

**Raises** *KeyError* – if definition class is not set for the corresponding name.

**classmethod GetDefinitionByName** (*name*: *str*) → *Type[l2tscaffolder.definitions.interface.ScaffolderDefinition]*  
Returns a definition class based on registered name.

**Parameters** *name* (*str*) – name of the definition.

**Returns**

**definition class or None** if name is not registered.

**Return type** *interface.ScaffolderDefinition*

**classmethod GetDefinitionNames** () → *Iterator[str]*  
Yields all names of registered definition classes.

**Yields** *str* – definition names.

**classmethod GetDefinitionObjects** () → *Iterator[l2tscaffolder.definitions.interface.ScaffolderDefinition]*  
Yields instances of each registered definition class.

**Yields** *ScaffolderDefinition* – definition object.

**classmethod RegisterDefinition** (*definition\_class*: *Type[l2tscaffolder.definitions.interface.ScaffolderDefinition]*)  
Registers a definition class.

Definition classes are identified by their NAME attribute.

**Parameters** *definition\_class* (*ScaffolderDefinition*) – definition class.

**Raises** *KeyError* – if definition class is already set for the corresponding name.

## I2tscaffolder.definitions.plaso module

The plaso definition class.

**class** *l2tscaffolder.definitions.plaso.PlasoProject*  
Bases: *l2tscaffolder.definitions.interface.ScaffolderDefinition*

Plaso project definition.

**NAME** = 'plaso'

**ValidatePath** (*root\_path*: *str*) → *bool*  
Validates the path to a Plaso development tree.

**Parameters** *root\_path* (*str*) – the path to the root of the project directory.

**Returns**

**whether the given path is the correct root path to a Plaso** development tree.

**Return type** *bool*

## I2tscaffolder.definitions.timesketch module

The Timesketch definition class.

```
class I2tscaffolder.definitions.timesketch.TimesketchProject
    Bases: I2tscaffolder.definitions.interface.ScaffolderDefinition
    Timesketch project definition.
    NAME = 'timesketch'
    ValidatePath (root_path: str) → bool
        Validates the path to a Timesketch development tree.
        Parameters root_path (str) – the path to the root of the project directory.
        Returns
            whether the given path is the correct root path to a Timesketch development tree.
        Return type bool
```

## I2tscaffolder.definitions.turbinia module

The Turbinia definition class.

```
class I2tscaffolder.definitions.turbinia.TurbiniaProject
    Bases: I2tscaffolder.definitions.interface.ScaffolderDefinition
    Turbinia project definition.
    NAME = 'turbinia'
    ValidatePath (root_path: str) → bool
        Validates the path to a Timesketch development tree.
        Parameters root_path (str) – the path to the root of the project directory.
        Returns
            whether the given path is the correct root path to a Timesketch development tree.
        Return type bool
```

## Module contents

This file imports Python modules that registers definitions.

## 2.1.2 I2tscaffolder.frontend package

### Submodules

#### I2tscaffolder.frontend.cli\_output\_handler module

The output file handler for click

```
class I2tscaffolder.frontend.cli_output_handler.OutputHandlerClick
    Bases: I2tscaffolder.frontend.output_handler.BaseOutputHandler
    Output handler for click.
```

**Confirm** (*text: str, default=True, abort=True*)

Returns a bool from a yes/no question presented to the end user.

**Parameters**

- **text** (*str*) – the question presented to the end user.
- **default** (*bool*) – the default for the confirmation answer. If True the default is Y(es), if False the default is N(o)
- **abort** (*bool*) – if the program should abort if the user answer to the confirm prompt is no. The default is an abort.

**Returns** False if the user entered no, True if the user entered yes

**Return type** bool

**PrintError** (*text: str*)

Presents an error message.

**Parameters** **text** (*str*) – the error message to present.

**PrintInfo** (*text: str*)

Presents the user with an informational text.

**Parameters** **text** (*str*) – the text to present.

**PrintNewLine** ()

Adds a new or blank line to the output.

**PrintOutput** (*text: str*)

Presents the user with output from the tool.

**Parameters** **text** (*str*) – the text to present the user with.

**PromptError** (*text: str*) → str

Presents the user with an error message and return back the answer.

**Parameters** **text** (*str*) – the text to prompt

**Returns** the user input

**Return type** str

**PromptInfo** (*text: str*) → str

Presents the user with a message prompt and return back the answer.

**Parameters** **text** (*str*) – the text to prompt

**Returns** the user input

**Return type** str

**PromptInfoWithDefault** (*text: str, input\_type: type, default: object*) → object

Presents the user with a prompt with a default return value and a type.

The prompt can have a default value to be chosen as well as a defined type of the returned data.

**Parameters**

- **text** (*str*) – the text to prompt
- **input\_type** (*type*) – the type of the input
- **default** (*object*) – the default value

**Returns** the user input, using the supplied input type.

**Return type** object

## l2tscaffolder.frontend.frontend module

The scaffolder frontend.

**class** `l2tscaffolder.frontend.frontend.ScaffolderFrontend` (*output\_handler:*  
*l2tscaffolder.frontend.output\_handler.BaseOutput*)

Bases: object

A frontend implementation for the scaffolder project.

**CreateGitFeatureBranch** (*project\_path: str, module\_name: str*)

Creates a feature branch inside the git project.

Creates a feature branch inside the git project path to store all the generated files in.

### Parameters

- **project\_path** (*str*) – path to the git project folder.
- **module\_name** (*str*) – name of the output module.

**GatherScaffolderAnswers** (*scaffolder, scaffolder\_engine*)

Asks all questions that scaffolder requires and store the results in it.

### Parameters

- **scaffolder** (*scaffolder\_interface.Scaffolder*) – the scaffolder that stores all required questions and stores all results as well.
- **scaffolder\_engine** (*scaffolder\_engine.ScaffolderEngine*) – the scaffolder engine object, needed to store answers from questions asked.

**Raises** `UnableToConfigure` – if the answer causes the scaffolder not to be configured properly.

**GetDefinition** (*definition\_string: str*) → `l2tscaffolder.definitions.interface.ScaffolderDefinition`

Returns the definition object as chosen by the user.

**Parameters** **definition\_string** (*str*) – definition name, read from user input.

**Returns** the chosen definition object.

**Return type** `definition_interface.ScaffolderDefinition`

**GetModuleName** () → `str`

Returns the module name as chosen by the user.

**GetProjectPath** (*definition: l2tscaffolder.definitions.interface.ScaffolderDefinition*) → `str`

Returns the path to the project's root folder as chosen by the user.

**Parameters** **definition** (*definition\_interface.ScaffolderDefinition*) – the chosen definition. Used to validate the project path.

**Returns** the path to the project's root folder.

**Return type** `str`

**Raises** `errors.WrongCliInput` – when no valid project path has been provided.

**GetScaffolder** (*definition: l2tscaffolder.definitions.interface.ScaffolderDefinition*) →

`l2tscaffolder.scaffolders.interface.Scaffolder`  
Returns the scaffolder as chosen by the user.

**Parameters definition** (*definition\_interface.ScaffolderDefinition*) – the chosen definition. Used to determine available scaffolders.

**Returns** the chosen scaffolder object.

**Return type** *scaffolder\_interface.ScaffolderCli*

**Start** (*definition\_value*)

Start the CLI.

**Parameters definition\_value** (*str*) – the definition string chosen by UI.

## I2tscaffolder.frontend.output\_handler module

The output file handler.

This file defines the interface of how an output handler should operate. An output handler is used as a UI element, for two things: 1. Relay information back to the user. 2. Gather input from an end user and presenting it back to the tool.

**class** *I2tscaffolder.frontend.output\_handler.BaseOutputHandler*

Bases: *object*

Interface for the output handler.

**Confirm** (*text: str, default=True, abort=True*)

Returns a bool from a yes/no question presented to the end user.

**Parameters**

- **text** (*str*) – the question presented to the end user.
- **default** (*bool*) – the default for the confirmation answer. If True the default is Y(es), if False the default is N(o)
- **abort** (*bool*) – if the program should abort if the user answer to the confirm prompt is no. The default is an abort.

**Returns** False if the user entered no, True if the user entered yes

**Return type** *bool*

**PrintError** (*text: str*)

Presents an error message.

**Parameters text** (*str*) – the error message to present.

**PrintInfo** (*text: str*)

Presents the user with an informational text.

**Parameters text** (*str*) – the text to present.

**PrintNewLine** ()

Adds a new or blank line to the output.

**PrintOutput** (*text: str*)

Presents the user with output from the tool.

**Parameters text** (*str*) – the text to present the user with.

**PromptError** (*text: str*) → *str*

Presents the user with an error message prompt and returns the answer.

**Parameters text** (*str*) – the text to prompt

**Returns** the user input.



**Return type** `str`

**PromptInfo** (*text: str*) → `str`

Presents the user with a message prompt and return back the answer.

**Parameters** **text** (*str*) – the text to prompt

**Returns** the user input.

**Return type** `str`

**PromptInfoWithDefault** (*text: str, input\_type: type, default: object*) → `object`

Presents the user with a prompt with a default return value and a type.

The prompt can have a default value to be chosen as well as a defined type of the returned data.

**Parameters**

- **text** (*str*) – the text to prompt
- **input\_type** (*type*) – the type of the input
- **default** (*object*) – the default value

**Returns** the user input, using the supplied input type.

**Return type** `object`

## Module contents

### 2.1.3 I2tscaffolder.helpers package

#### Submodules

#### I2tscaffolder.helpers.cli module

Helper for command line functions.

**class** `I2tscaffolder.helpers.cli.CLIHelper` (*mock\_responses=None*)

Bases: `object`

Command line interface (CLI) helper.

**mock\_responses**

mappings of commands to responses.

**Type** `dict[str, str]`

**preferred\_encoding**

preferred encoding of output.

**Type** `str`

**RunCommand** (*command: str*)

Runs a command.

**Parameters** **command** (*str*) – command to run.

**Returns**

**exit code, output that was written to stdout** and `stderr`.

**Return type** `tuple[int, str, str]`

**Raises** `AttributeError` – if the command is not recognized.

## I2tscaffolder.helpers.git module

Git helper for the scaffolder project.

This file provides a class to assist with git operations.

**class** `I2tscaffolder.helpers.git.GitHelper` (*project\_path: str*)

Bases: `I2tscaffolder.helpers.cli.CLIHelper`

Helper class for git operations.

**project\_path**

path to the git project folder.

**AddFileToTrack** (*file\_path: str*)

Add a file to those that are tracked by the git repo.

**Parameters** `file_path` (*str*) – path to the file to be added to tracked files by this git repo.

**Raises** `errors.UnableToConfigure` – when the tool is not able to add newly added files to the git repo.

**CreateBranch** (*branch: str*) → int

Creates a git branch and returns the exit code of the command.

**Parameters** `branch` (*str*) – the name of the git branch.

**Returns** the exit code from the git command.

**Return type** int

**GenerateBranchName** (*module\_name: str*) → str

Generates a git branch name.

**Parameters** `module_name` (*str*) – module name to generate a git branch name from.

**Returns** git branch name.

**Return type** str

**GetActiveBranch** () → str

Determines the active branch of the git project.

**Returns** the active branch of the git project.

**Return type** str

**Raises** `errors.UnableToConfigure` – when the tool is not able to get the active branch of the git project.

**HasBranch** (*branch\_name: str*) → bool

Tests for the existence of a specific branch.

**Parameters** `branch_name` (*str*) – the name of the branch to test for.

**Returns** True if the branch exists.

**Return type** bool

**RunCommand** (*command: str*) → Tuple[int, str, str]

Runs a command.

**Parameters** `command` (*str*) – command to run.

**Returns**

exit code, output that was written to stdout and stderr.

**Return type** tuple[int, str, str]

**SwitchToBranch** (*branch: str*) → int

Switches the git branch and returns the exit code of the command.

**Parameters** **branch** (*str*) – the name of the git branch.

**Returns** the exit code from the git command.

**Return type** int

## Module contents

### 2.1.4 I2tscaffolder.lib package

#### Submodules

##### I2tscaffolder.lib.code\_formatter module

Formatter for generated code.

**class** I2tscaffolder.lib.code\_formatter.**CodeFormatter** (*yapf\_path: str*)

Bases: object

Formats code in files.

**Format** (*code: str*) → str

Formats the code.

**Parameters** **code** (*str*) – code to format

**Returns** the formatted code

**Return type** str

##### I2tscaffolder.lib.definitions module

The format specification classes.

##### I2tscaffolder.lib.engine module

The scaffolder engine.

**class** I2tscaffolder.lib.engine.**ScaffolderEngine**

Bases: object

The engine, responsible for file handling and setting up scaffolders.

**GenerateFiles** () → Iterator[str]

Generates needed files.

**Raises** errors.EngineNotConfigured – when not all attributes have been configured.

**Yields** *str* – the full path to a file that was generated and written to disk.

**SetModuleName** (*module\_name: str*)

Sets the module name as chosen by the user.

**Parameters** **module\_name** (*str*) – name of the module to be generated by the scaffolder.

**SetProjectRootPath** (*root\_path: str*)

Sets the path to the root of the project tree.

**Raises** `errors.NoValidDefinition` – when root path is not identified as a valid definition path.

**SetScaffolder** (*scaffolder: l2tscaffolder.scaffolders.interface.Scaffolder*)

Stores and initializes the scaffolder object in the engine.

**Parameters** **scaffolder** (*scaffolder\_interface.Scaffolder*) – the scaffolder class that the engine will use to generate files.

**StoreScaffolderAttribute** (*name: str, value: object, value\_type: Type[CT\_co]*)

Stores an attribute read from the CLI.

### Parameters

- **name** (*str*) – the attribute name.
- **value** (*object*) – the attribute value.
- **value\_type** (*type*) – the attribute type.

### Raises

- `KeyError` – if the attribute name is already defined.
- `ScaffolderNotConfigured` – if the scaffolder has not yet been set.
- `ValueError` – if the value is not of the correct type.

## I2tscaffolder.lib.errors module

This file contains the error classes.

**exception** `l2tscaffolder.lib.errors.EngineNotConfigured`

Bases: `l2tscaffolder.lib.errors.Error`

Raised when the scaffolder engine has not been configured correctly.

**exception** `l2tscaffolder.lib.errors.Error`

Bases: `Exception`

Base error class.

**exception** `l2tscaffolder.lib.errors.FileHandlingError`

Bases: `l2tscaffolder.lib.errors.Error`

Raised when the file handler is unable to do file operation.

**exception** `l2tscaffolder.lib.errors.NoValidDefinition`

Bases: `l2tscaffolder.lib.errors.Error`

Raised when no valid project definition has been identified.

**exception** `l2tscaffolder.lib.errors.ScaffolderNotConfigured`

Bases: `l2tscaffolder.lib.errors.Error`

Raised when the scaffolder has not been configured correctly.

**exception** `l2tscaffolder.lib.errors.UnableToConfigure`

Bases: `l2tscaffolder.lib.errors.Error`

Raised when the scaffolder tool has issues with configuration.

**exception** `l2tscaffolder.lib.errors.WrongCliInput`

Bases: `l2tscaffolder.lib.errors.Error`

Raised when wrong input is entered into the CLI.

## I2tscaffolder.lib.file\_handler module

The file handler.

**class** `l2tscaffolder.lib.file_handler.FileHandler`

Bases: `object`

Handles the creation of files.

**AddContent** (*source: str, content: str*) → `str`

Adds content to a file and create file if non existing.

### Parameters

- **source** (*str*) – path of the file to edit.
- **content** (*str*) – content to append to the file.

**Returns** path of the edited file.

**Return type** `str`

**AddImportToInit** (*path: str, entry: str*)

Adds an import into an init file in the correct order.

### Parameters

- **path** (*str*) – path to the `__init__` file.
- **entry** (*str*) – the import statement.

**CopyFile** (*source: str, destination: str*) → `str`

Copies a file.

### Parameters

- **source** (*str*) – path of the file to copy
- **destination** (*str*) – path to copy the file to.

**Returns** the path of the copied file

**Return type** `str`

**Raises** `errors.FileHandlingError` – when file copy operation fails.

**CreateFile** (*directory\_path: str, file\_name: str, filename\_extension: str*) → `str`

Creates a empty file.

### Parameters

- **directory\_path** (*str*) – path to the directory the file should be created in.
- **file\_name** (*str*) – name of the new file.
- **filename\_extension** (*str*) – extension of the new file.

**Returns** path of the created file

**Return type** `str`

**CreateFileFromPath** (*file\_path: str*) → str

Creates a empty file.

**Parameters** **file\_path** (*str*) – path to the file.

**Returns** the path of the created file

**Return type** str

**classmethod CreateFilePath** (*path: str, name: str, extension: str*) → str

Creates the file path from the directory path, filename and suffix.

**Parameters**

- **path** (*str*) – path to the file directory.
- **name** (*str*) – filename.
- **extension** (*str*) – file extension.

**Returns** the path to the file.

**Return type** str

**CreateFolderForFilePathIfNotExist** (*file\_path: str*) → str

Creates folders for the given file if it does not exist.

**Parameters** **file\_path** (*str*) – path to the file

**Returns** directory path of the created directory

**Return type** str

**CreateOrModifyFileWithContent** (*source: str, content: str*)

Adds content to a file and create the file and path if non existing.

**Parameters**

- **source** (*str*) – path of the file to edit.
- **content** (*str*) – content to append to the file.

**Returns** path of the edited file.

**Return type** str

### I2tscaffolder.lib.mapping\_helper module

Helper methods for mapping.

**class** `l2tscaffolder.lib.mapping_helper.MappingHelper` (*template\_path: str = "", formatter\_path: str = ""*)

Bases: object

Mapping helper for scaffolders.

**GenerateClassName** (*scaffolder\_name: str*) → str

Generates a class name from the scaffolder name for file generation.

**Parameters** **scaffolder\_name** (*str*) – name of the scaffolder

**Returns** name of the class

**Return type** str

**RenderTemplate** (*template\_filename: str, context: dict*) → str

Renders the template with the context to return a string.

**Parameters**

- **template\_filename** (*str*) – the name of the template
- **context** (*dict*) – the context of the template as a dictionary

**Returns** the rendered template as a string

**Return type** str

**Module contents**

Library code for I2tscaffolder.

**2.1.5 I2tscaffolder.scaffolders package****Submodules****I2tscaffolder.scaffolders.interface module**

The scaffolder interface classes.

```
class I2tscaffolder.scaffolders.interface.BaseQuestion (attribute: str, prompt: str)
    Bases: object

    Scaffolder question.

    attribute
        the name of the attribute the question prompts for.

        Type str

    prompt
        help string that is displayed before the question is asked.

        Type str

    TYPE = None

    ValidateAnswer (answer: object)
        Validate an answer to a question.

        Parameters answer (object) – the answer to the question asked.

        Raises errors.UnableToConfigure – if the answer is invalid.

class I2tscaffolder.scaffolders.interface.DictQuestion (attribute, prompt,
                                                         key_prompt, value_prompt)
    Bases: I2tscaffolder.scaffolders.interface.BaseQuestion

    Scaffolder dict question.

    attribute
        the name of the attribute the question prompts for.

        Type str

    prompt
        help string that is displayed before the question is asked.

        Type str
```

**key\_prompt**  
the help string that is displayed before asking for each key.

**Type** str

**value\_prompt**  
the help string that is displayed before asking for each value in the dict.

**Type** str

**TYPE**  
alias of `builtins.dict`

**class** `l2tscaffolder.scaffolders.interface.IntQuestion` (*attribute: str, prompt: str*)  
Bases: `l2tscaffolder.scaffolders.interface.BaseQuestion`

Scaffolder integer question.

**TYPE**  
alias of `builtins.int`

**class** `l2tscaffolder.scaffolders.interface.ListQuestion` (*attribute: str, prompt: str*)  
Bases: `l2tscaffolder.scaffolders.interface.BaseQuestion`

Scaffolder list question.

**TYPE**  
alias of `builtins.list`

**class** `l2tscaffolder.scaffolders.interface.Scaffolder`  
Bases: `object`

The scaffolder interface.

**DESCRIPTION** = ''

**GenerateFiles** () → `Iterator[Tuple[str, str]]`  
Generates files this scaffolder provides.

**Yields** *list* – file name and content of the file to be written to disk.

**GetFilesToCopy** () → `Iterator[Tuple[str, str]]`  
Return a list of files that need to be copied.

If not overwritten this will return an empty iterator.

**Yields** *tuple (str, str)* – file name of source and destination.

**GetInitFileChanges** () → `Iterator[Tuple[str, str]]`  
Generate a list of init files that need changing and the changes to them.

**Yields** *tuple (str, str)* – path to the init file and the entry to add to it.

**GetJinjaContext** () → `Dict[str, object]`  
Returns a dict that can be used as a context for Jinja2 templates.

**Returns**

**containing:** str: name of Jinja argument. object: Jinja argument value.

**Return type** dict

**GetQuestions** () → `List[l2tscaffolder.scaffolders.interface.BaseQuestion]`  
Returns scaffolder questions.

**Returns** questions to prompt the user with.



**Return type** list[*BaseQuestion*]

**NAME** = 'base\_parser'

**PROJECT** = 'plaso'

**QUESTIONS** = []

**RaiseIfNotReady** ()

Checks to see if all attributes are set to start generating files.

By default this function only checks to see if all attributes defined in questions and Jinja2 context have values and are not empty.

**Raises** ScaffolderNotConfigured – if the scaffolder is not fully configured.

**SetAttribute** (*name: str, value: object, value\_type: type*)

Stores an attribute read from the CLI.

**Parameters**

- **name** (*str*) – the attribute name.
- **value** (*object*) – the attribute value.
- **value\_type** (*type*) – the attribute type.

**Raises**

- *ValueError* – if the value is not of the correct type.
- *KeyError* – If the attribute is not configured for this scaffolder.

**SetOutputName** (*output\_name: str*)

Sets the name of the output module.

This is the name of the generated output module this scaffolder implements.

**Parameters** **output\_name** (*str*) – the name of the output that the scaffolder generates, whether that is an output module, plugin, parser, analyzer or something else.

**class** `l2tscaffolder.scaffolders.interface.StringQuestion` (*attribute: str, prompt: str*)

Bases: `l2tscaffolder.scaffolders.interface.BaseQuestion`

Scaffolder string question.

**TYPE**

alias of `builtins.str`

## I2tscaffolder.scaffolders.manager module

The scaffolder manager.

**class** `l2tscaffolder.scaffolders.manager.ScaffolderManager`

Bases: `object`

The scaffolder manager.

**classmethod** **DeregisterScaffolder** (*scaffolder\_class: Type[l2tscaffolder.scaffolders.interface.Scaffolder]*)

Deregisters a scaffolder class.

The scaffolder classes are identified based on their lower case name.

**Parameters** **scaffolder\_class** (*type*) – scaffolder class (subclass of `Scaffolder`).

**Raises** `KeyError` – if scaffolder class is not set for the corresponding name.

**classmethod** `GetScaffolderClasses ()` → `Iterator[Type[I2tscaffolder.scaffolders.interface.Scaffolder]]`  
Generates a list of all registered scaffolder classes.

**classmethod** `GetScaffolderInformation ()` → `Iterator[Tuple[str, str]]`  
Retrieves the scaffolder information.

**Yields** `tuple[str, str]` – pairs of scaffolder names and descriptions.

**classmethod** `GetScaffolderNames ()` → `Iterator[str]`  
Retrieves the scaffolder names.

**Yields** `str` – scaffolder names.

**classmethod** `GetScaffolderObjectByName (scaffolder_name)` → `Optional[I2tscaffolder.scaffolders.interface.Scaffolder]`  
Retrieves a specific scaffolder object by its name.

**Parameters** `scaffolder_name (str)` – name of the scaffolder.

**Returns** scaffolder object or `None`.

**Return type** `Scaffolder`

**classmethod** `GetScaffolderObjects ()` → `Dict[str, I2tscaffolder.scaffolders.interface.Scaffolder]`  
Retrieves the scaffolder objects.

**Returns** scaffolders per name.

**Return type** `dict[str, Scaffolder]`

**classmethod** `GetScaffolderQuestionByName (scaffolder_name: str)` → `List[I2tscaffolder.scaffolders.interface.BaseQuestion]`  
Retrieve a list of questions asked by a scaffolder based on name.

**Parameters** `scaffolder_name (str)` – name of the scaffolder.

**Returns**

**a list with all the questions needed to setup the scaffolder.** If `scaffolder_name` is not registered an empty list will be returned.

**Return type** `list`

**classmethod** `GetScaffolderQuestions ()` → `List[I2tscaffolder.scaffolders.interface.BaseQuestion]`  
Retrieves all the questions asked by scaffolders.

**Returns** questions asked by all scaffolders.

**Return type** `list[interface.BaseQuestion]`

**classmethod** `GetScaffolders ()` → `Iterator[Tuple[str, Type[I2tscaffolder.scaffolders.interface.Scaffolder]]]`  
Retrieves the registered scaffolders.

Retrieves a dictionary of all registered scaffolders.

**Yields** `tuple` – contains:

- `str`: name of the scaffolder:
- `type`: scaffolder class (subclass of `Scaffolder`).

**classmethod** `RegisterScaffolder (scaffolder_class: Type[I2tscaffolder.scaffolders.interface.Scaffolder])`  
Registers a scaffolder class.

The scaffolder classes are identified based on their lower case name.

**Parameters** `scaffolder_class` (*type*) – scaffolder class (subclass of Scaffolder).

**Raises** `KeyError` – if scaffolder class is already set for the corresponding name.

**classmethod** **RegisterScaffolders** (*scaffolder\_classes: List[Type[I2tscaffolder.scaffolders.interface.Scaffolder]]*)  
Registers scaffolder classes.

The scaffolder classes are identified based on their lower case name.

**Parameters** `scaffolder_classes` (*list[type]*) – scaffolders classes (subclasses of Scaffolder).

**Raises** `KeyError` – if scaffolder class is already set for the corresponding name.

## I2tscaffolder.scaffolders.plaso module

Plaso scaffolder that generates plaso parser and plugins.

**class** `I2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder`

Bases: `I2tscaffolder.scaffolders.interface.Scaffolder`

The plaso base scaffolder interface.

**class\_name**

class name of the plaso parser or plugin to be generated.

**Type** `str`

**test\_file**

name of the file used for testing the parser or plugin.

**Type** `str`

**test\_file\_path**

path to the test file.

**Type** `str`

**DESCRIPTION** = 'This is a scaffolder for plaso parsers and/or plugins'

**GenerateFiles** () → `Iterator[Tuple[str, str]]`

Generates all the files required for a plaso parser or a plugin.

**Yields** *list[tuple]* –

**containing:** `str`: file name. `str`: file content.

**GetFilesToCopy** () → `Iterator[Tuple[str, str]]`

Return a list of files that need to be copied.

**Raises** `IOError` – when the test file does not exist.

**Yields** *tuple* –

**containing:** `str`: file name of source. `str`: file name of destination.

**GetInitFileChanges** () → `Iterator[Tuple[str, str]]`

Generate a list of init files that need changing and the changes to them.

**Yields** *Tuple[str, str]* – path to the init file and the entry to add to it.

**GetJinjaContext** () → `Dict[str, object]`

Returns a dict that can be used as a context for Jinja2 templates.

**Returns**

**containing:** str: name of Jinja argument. object: Jinja argument value.

**Return type** dict

**GetQuestions** () → List[l2tscaffolder.scaffolders.interface.BaseQuestion]

Returns scaffolder questions as well as adding plaso related ones.

**Returns** questions to prompt the user with.

**Return type** list[*interface.BaseQuestion*]

**NAME** = 'plaso\_base'

**PROJECT** = 'plaso'

**QUESTIONS** = []

**RaiseIfNotReady** ()

Checks to see if all attributes are set to start generating files.

**Raises** ScaffolderNotConfigured – if the scaffolder is not fully configured.

**TEMPLATE\_FORMATTER\_FILE** = 'generic\_plaso\_formatter.jinja2'

**TEMPLATE\_FORMATTER\_TEST** = 'generic\_plaso\_formatter\_test.jinja2'

**TEMPLATE\_PARSER\_FILE** = 'generic\_plaso\_parser.jinja2'

**TEMPLATE\_PARSER\_TEST** = 'generic\_plaso\_parser\_test.jinja2'

**class** l2tscaffolder.scaffolders.plaso.PlasoParserScaffolder

Bases: *l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder*

Scaffolder for generating plaso parsers.

**parser\_name**

name of the parser to be generated.

**Type** str

**GetJinjaContext** () → Dict[str, object]

Returns a dict that can be used as a context for Jinja2 templates.

**Returns**

**containing:** str: name of Jinja argument. object: Jinja argument value.

**Return type** dict

**class** l2tscaffolder.scaffolders.plaso.PlasoPluginScaffolder

Bases: *l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder*

Scaffolder for generating plaso plugins.

**GetJinjaContext** () → Dict[str, object]

Returns a dict that can be used as a context for Jinja2 templates.

**Returns**

**containing:** str: name of Jinja argument. object: Jinja argument value.

**Return type** dict

**class** l2tscaffolder.scaffolders.plaso.TestFileQuestion (*attribute: str, prompt: str*)

Bases: *l2tscaffolder.scaffolders.interface.StringQuestion*

Test file question.

**ValidateAnswer** (*answer: str*)

Validates the answer to the test file question.

**Parameters** *answer* (*str*) – path to a test file.

**Raises** `errors.UnableToConfigure` – if the answer is invalid.

## I2tscaffolder.scaffolders.plaso\_sqlite module

The scaffolder interface classes.

**class** `I2tscaffolder.scaffolders.plaso_sqlite.PlasoSQLiteScaffolder`

Bases: `I2tscaffolder.scaffolders.plaso.PlasoPluginScaffolder`

The plaso SQLite plugin scaffolder.

**database\_name**

name of the test SQLite database for the plugin.

**Type** `str`

**database\_schema**

a dict containing all table names (keys) and the SQL statement used to create the table (value), derived from the test database.

**Type** `dict`

**data\_types**

a dict containing all the data types generated for the parser, the key is the name for each SQL statement run against the database and the value is the data type used for each generated event resulting from that SQL statement.

**Type** `dict`

**queries**

a dict containing query name and SQL statements or queries run against the database.

**Type** `dict`

**query\_columns**

for each SQL statement run against the database, with the key being query name and value being a list of all SQL column names that are returned for each query.

**Type** `dict`

**required\_tables**

a list of all required tables needed for the plugin to parse this particular database.

**Type** `list`

**timestamp\_columns**

a dict containing a list of all columns with timestamp values, with query names as the key.

**Type** `dict`

**DESCRIPTION** = 'Provides a scaffolder to generate a plaso SQLite plugin.'

**GenerateFiles** () → `Iterator[Tuple[str, str]]`

Generates files required for the SQLite plugin.

**Yields** *tuple* – file name and content of the file to be written to disk.

**Raises** `errors.UnableToConfigure` – if it is not possible to generate the files.

**GetJinjaContext** () → Dict[str, object]

Returns a dict that can be used as a context for Jinja2 templates.

**Returns**

**containing:** str: name of Jinja argument. object: Jinja argument value.

**Return type** dict

**NAME** = 'sqlite'

**QUESTIONS** = [<l2tscaffolder.scaffolders.plaso\_sqlite.SQLQuestion object>, <l2tscaffolder.scaffolders.plaso\_sqlite.SQLQuestion object>]

**SCHEMA\_QUERY** = 'SELECT tbl\_name, sql FROM sqlite\_master WHERE type = "table" AND tbl\_name = ?'

**TEMPLATE\_FORMATTER\_FILE** = 'sqlite\_plugin\_formatter.jinja2'

**TEMPLATE\_FORMATTER\_TEST** = 'sqlite\_plugin\_formatter\_test.jinja2'

**TEMPLATE\_PARSER\_FILE** = 'sqlite\_plugin.jinja2'

**TEMPLATE\_PARSER\_TEST** = 'sqlite\_plugin\_test.jinja2'

```
class l2tscaffolder.scaffolders.plaso_sqlite.SQLQuestion(attribute, prompt,
                                                         key_prompt,
                                                         value_prompt)
```

Bases: *l2tscaffolder.scaffolders.interface.DictQuestion*

SQL Query question.

**ValidateAnswer** (answer: dict)

Validates the answer to the SQL query question.

The answer should be a dict that has query names as key values and valid SQLite commands as values. This function attempts to verify that the SQL commands do not have syntax errors in them by attempting to run it against an empty SQLite database stored in memory.

The function also makes sure the key value confirms to the style guide of plaso, to be in the form of CamelCase, eg. BookmarkRow.

**Parameters** **answer** (dict) – the answer to the question asked.

**Raises** errors.UnableToConfigure – if the answer is invalid.

## I2tscaffolder.scaffolders.timesketch module

Timesketch scaffolder that generates analyzer plugins.

**class** l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder

Bases: *l2tscaffolder.scaffolders.interface.Scaffolder*

The Timesketch base scaffolder interface.

**class\_name**

class name of the Timesketch analyzer to be generated.

**Type** str

**DESCRIPTION** = 'This is a scaffolder for Timesketch analyzers'

**GenerateFiles** () → Iterator[Tuple[str, str]]

Generates all the files required for a Timesketch analyzer plugin.

**Yields** list[tuple] –

**containing:** str: file name. str: file content.

**GetInitFileChanges** () → Iterator[Tuple[str, str]]

Generate a list of init files that need changing and the changes to them.

**Yields** *Tuple[str, str]* – path to the init file and the entry to add to it.

**GetJinjaContext** () → Dict[str, object]

Returns a dict that can be used as a context for Jinja2 templates.

**Returns**

**containing:** str: name of Jinja argument. object: Jinja argument value.

**Return type** dict

**NAME** = 'timesketch\_base'

**PROJECT** = 'timesketch'

**QUESTIONS** = []

**TEMPLATE\_PLUGIN\_FILE** = ''

**TEMPLATE\_PLUGIN\_TEST** = ''

## l2tscaffolder.scaffolders.timesketch\_index module

Timesketch index analyzer scaffolder.

**class** l2tscaffolder.scaffolders.timesketch\_index.**TimesketchIndexScaffolder**

Bases: *l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder*

The Timesketch index analyzer plugin scaffolder.

**DESCRIPTION** = 'Provides a scaffolder to generate a Timesketch index analyzer plugin.'

**NAME** = 'index\_analyzer'

**TEMPLATE\_PLUGIN\_FILE** = 'ts\_index\_analyzer.jinja2'

**TEMPLATE\_PLUGIN\_TEST** = 'ts\_index\_analyzer\_test.jinja2'

## l2tscaffolder.scaffolders.timesketch\_sketch module

Timesketch sketch analyzer scaffolder.

**class** l2tscaffolder.scaffolders.timesketch\_sketch.**TimesketchSketchScaffolder**

Bases: *l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder*

The Timesketch sketch analyzer plugin scaffolder.

**DESCRIPTION** = 'Provides a scaffolder to generate a Timesketch sketch analyzer plugin.'

**NAME** = 'sketch\_analyzer'

**TEMPLATE\_PLUGIN\_FILE** = 'ts\_sketch\_analyzer.jinja2'

**TEMPLATE\_PLUGIN\_TEST** = 'ts\_sketch\_analyzer\_test.jinja2'

### I2tscaffolder.scaffolders.turbinia module

Turbinia component scaffolder.

**class** `l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder`

Bases: `l2tscaffolder.scaffolders.interface.Scaffolder`

The Turbinia base scaffolder interface.

**class\_name**

class name of the Turbinia job and task to be generated.

**Type** str

**DESCRIPTION** = 'Provides a scaffolder to generate a Turbinia job and task plugins.'

**GenerateFiles** () → Iterator[Tuple[str, str]]

Generates all the files required for a Turbinia component.

**Yields** *list[tuple]* –

**containing:** str: file name. str: file content.

**GetInitFileChanges** () → Iterator[Tuple[str, str]]

Generate a list of init files that need changing and the changes to them.

**Yields** *Tuple[str, str]* – path to the init file and the entry to add to it.

**GetJinjaContext** () → Dict[str, object]

Returns a dict that can be used as a context for Jinja2 templates.

**Returns**

**containing:** str: name of Jinja argument. object: Jinja argument value.

**Return type** dict

**NAME** = 'turbinia\_job\_and\_task'

**PROJECT** = 'turbinia'

**TEMPLATE\_JOB\_FILE** = 'turbinia\_job.jinja2'

**TEMPLATE\_TASK\_FILE** = 'turbinia\_task.jinja2'

### Module contents

This file imports Python modules that registers scaffolders.

### 2.1.6 I2tscaffolder.templates package

#### Module contents

## 2.2 Module contents

defining the version



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### I

- [l2tscaffolder](#), [28](#)
- [l2tscaffolder.definitions](#), [9](#)
- [l2tscaffolder.definitions.interface](#), [7](#)
- [l2tscaffolder.definitions.manager](#), [7](#)
- [l2tscaffolder.definitions.plaso](#), [8](#)
- [l2tscaffolder.definitions.timesketch](#), [9](#)
- [l2tscaffolder.definitions.turbinia](#), [9](#)
- [l2tscaffolder.frontend](#), [13](#)
- [l2tscaffolder.frontend.cli\\_output\\_handler](#),  
[9](#)
- [l2tscaffolder.frontend.frontend](#), [11](#)
- [l2tscaffolder.frontend.output\\_handler](#),  
[12](#)
- [l2tscaffolder.helpers](#), [15](#)
- [l2tscaffolder.helpers.cli](#), [13](#)
- [l2tscaffolder.helpers.git](#), [14](#)
- [l2tscaffolder.lib](#), [19](#)
- [l2tscaffolder.lib.code\\_formatter](#), [15](#)
- [l2tscaffolder.lib.definitions](#), [15](#)
- [l2tscaffolder.lib.engine](#), [15](#)
- [l2tscaffolder.lib.errors](#), [16](#)
- [l2tscaffolder.lib.file\\_handler](#), [17](#)
- [l2tscaffolder.lib.mapping\\_helper](#), [18](#)
- [l2tscaffolder.scaffolders](#), [28](#)
- [l2tscaffolder.scaffolders.interface](#), [19](#)
- [l2tscaffolder.scaffolders.manager](#), [21](#)
- [l2tscaffolder.scaffolders.plaso](#), [23](#)
- [l2tscaffolder.scaffolders.plaso\\_sqlite](#),  
[25](#)
- [l2tscaffolder.scaffolders.timesketch](#),  
[26](#)
- [l2tscaffolder.scaffolders.timesketch\\_index](#),  
[27](#)
- [l2tscaffolder.scaffolders.timesketch\\_sketch](#),  
[27](#)
- [l2tscaffolder.scaffolders.turbinia](#), [28](#)
- [l2tscaffolder.templates](#), [28](#)



## A

[AddContent\(\)](#) ([l2tscaffolder.lib.file\\_handler.FileHandler](#) [method](#)), 17  
[AddFileToTrack\(\)](#) ([l2tscaffolder.helpers.git.GitHelper](#) [method](#)), 14  
[AddImportToInit\(\)](#) ([l2tscaffolder.lib.file\\_handler.FileHandler](#) [method](#)), 17  
[attribute](#) ([l2tscaffolder.scaffolders.interface.BaseQuestion](#) [attribute](#)), 19  
[attribute](#) ([l2tscaffolder.scaffolders.interface.DictQuestion](#) [attribute](#)), 19  
[CreateFileFromPath\(\)](#) ([l2tscaffolder.lib.file\\_handler.FileHandler](#) [method](#)), 17  
[CreateFilePath\(\)](#) ([l2tscaffolder.lib.file\\_handler.FileHandler](#) [class method](#)), 18  
[CreateFolderForFilePathIfNotExist\(\)](#) ([l2tscaffolder.lib.file\\_handler.FileHandler](#) [method](#)), 18  
[CreateGitFeatureBranch\(\)](#) ([l2tscaffolder.frontend.frontend.ScaffolderFrontend](#) [method](#)), 11  
[CreateOrModifyFileWithContent\(\)](#) ([l2tscaffolder.lib.file\\_handler.FileHandler](#) [method](#)), 18

## B

[BaseOutputHandler](#) ([class](#) [l2tscaffolder.frontend.output\\_handler](#)), 12  
[BaseQuestion](#) ([class](#) [l2tscaffolder.scaffolders.interface](#)), 19

## C

[class\\_name](#) ([l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder](#) [attribute](#)), 23  
[class\\_name](#) ([l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder](#) [attribute](#)), 26  
[class\\_name](#) ([l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder](#) [attribute](#)), 28  
[CLIHelper](#) ([class](#) [in](#) [l2tscaffolder.helpers.cli](#)), 13  
[CodeFormatter](#) ([class](#) [in](#) [l2tscaffolder.lib.code\\_formatter](#)), 15  
[Confirm\(\)](#) ([l2tscaffolder.frontend.cli\\_output\\_handler.OutputHandlerClick](#) [method](#)), 9  
[Confirm\(\)](#) ([l2tscaffolder.frontend.output\\_handler.BaseOutputHandler](#) [method](#)), 12  
[CopyFile\(\)](#) ([l2tscaffolder.lib.file\\_handler.FileHandler](#) [method](#)), 17  
[CreateBranch\(\)](#) ([l2tscaffolder.helpers.git.GitHelper](#) [method](#)), 14  
[CreateFile\(\)](#) ([l2tscaffolder.lib.file\\_handler.FileHandler](#) [method](#)), 17  
[data\\_types](#) ([l2tscaffolder.scaffolders.plaso\\_sqlite.PlasoSQLiteScaffolder](#) [attribute](#)), 25  
[database\\_name](#) ([l2tscaffolder.scaffolders.plaso\\_sqlite.PlasoSQLiteScaffolder](#) [attribute](#)), 25  
[database\\_schema](#) ([l2tscaffolder.scaffolders.plaso\\_sqlite.PlasoSQLiteScaffolder](#) [attribute](#)), 25  
[DefinitionManager](#) ([class](#) [in](#) [l2tscaffolder.definitions.manager](#)), 7  
[DeregisterDefinition\(\)](#) ([l2tscaffolder.definitions.manager.DefinitionManager](#) [class method](#)), 7  
[DeregisterScaffolder\(\)](#) ([l2tscaffolder.scaffolders.manager.ScaffolderManager](#) [class method](#)), 21  
[DESCRIPTION](#) ([l2tscaffolder.scaffolders.interface.Scaffolder](#) [attribute](#)), 20  
[DESCRIPTION](#) ([l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder](#) [attribute](#)), 23  
[DESCRIPTION](#) ([l2tscaffolder.scaffolders.plaso\\_sqlite.PlasoSQLiteScaffolder](#) [attribute](#)), 25  
[DESCRIPTION](#) ([l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder](#) [attribute](#)), 26  
[DESCRIPTION](#) ([l2tscaffolder.scaffolders.timesketch\\_index.TimesketchIndex](#) [attribute](#)), 27

DESCRIPTION (l2tscaffolder.scaffolders.timesketch_sketch.TimesketchSketchScaffolder attribute), 27	GetInitFileChanges () (l2tscaffolder.scaffolders.interface.Scaffolder method), 20
DESCRIPTION (l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder attribute), 28	GetInitFileChanges () (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder method), 23
DictQuestion (class in l2tscaffolder.scaffolders.interface), 19	GetInitFileChanges () (l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder method), 26
<b>E</b>	GetInitFileChanges () (l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder method), 28
EngineNotConfigured, 16	GetJinjaContext () (l2tscaffolder.scaffolders.interface.Scaffolder method), 20
Error, 16	GetJinjaContext () (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder method), 23
<b>F</b>	GetJinjaContext () (l2tscaffolder.scaffolders.plaso.PlasoParserScaffolder method), 24
FileHandler (class in l2tscaffolder.lib.file_handler), 17	GetJinjaContext () (l2tscaffolder.scaffolders.plaso.PlasoPluginScaffolder method), 24
FileHandlingError, 16	GetJinjaContext () (l2tscaffolder.scaffolders.plaso_sqlite.PlasoSQLiteScaffolder method), 25
Format () (l2tscaffolder.lib.code_formatter.CodeFormatter method), 15	GetJinjaContext () (l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder method), 27
<b>G</b>	GetJinjaContext () (l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder method), 28
GatherScaffolderAnswers () (l2tscaffolder.frontend.frontend.ScaffolderFrontend method), 11	GetModuleName () (l2tscaffolder.frontend.frontend.ScaffolderFrontend method), 11
GenerateBranchName () (l2tscaffolder.helpers.git.GitHelper method), 14	GetProjectPath () (l2tscaffolder.frontend.frontend.ScaffolderFrontend method), 11
GenerateClassName () (l2tscaffolder.lib.mapping_helper.MappingHelper method), 18	GetQuestions () (l2tscaffolder.scaffolders.interface.Scaffolder method), 20
GenerateFiles () (l2tscaffolder.lib.engine.ScaffolderEngine method), 15	GetQuestions () (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder method), 24
GenerateFiles () (l2tscaffolder.scaffolders.interface.Scaffolder method), 20	GetScaffolder () (l2tscaffolder.frontend.frontend.ScaffolderFrontend method), 11
GenerateFiles () (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder method), 23	GetScaffolderClasses () (l2tscaffolder.scaffolders.manager.ScaffolderManager class method), 22
GenerateFiles () (l2tscaffolder.scaffolders.plaso_sqlite.PlasoSQLiteScaffolder method), 25	GetScaffolderInformation () (l2tscaffolder.scaffolders.manager.ScaffolderManager class method), 22
GenerateFiles () (l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder method), 26	GetScaffolderNames () (l2tscaffolder.scaffolders.manager.ScaffolderManager class method), 22
GenerateFiles () (l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder method), 28	
GetActiveBranch () (l2tscaffolder.helpers.git.GitHelper method), 14	
GetDefinition () (l2tscaffolder.frontend.frontend.ScaffolderFrontend method), 11	
GetDefinitionByName () (l2tscaffolderdefinitions.manager.DefinitionManager class method), 8	
GetDefinitionNames () (l2tscaffolderdefinitions.manager.DefinitionManager class method), 8	
GetDefinitionObjects () (l2tscaffolderdefinitions.manager.DefinitionManager class method), 8	

[\(l2tscaffolder.scaffolders.manager.ScaffolderManager class method\), 22](#)  
[l2tscaffolder.lib.code\\_formatter \(module\), 15](#)  
[GetScaffolderObjectByName\(\) \(l2tscaffolder.scaffolders.manager.ScaffolderManager class method\), 22](#)  
[l2tscaffolder.lib.definitions \(module\), 15](#)  
[l2tscaffolder.lib.engine \(module\), 15](#)  
[l2tscaffolder.lib.errors \(module\), 16](#)  
[GetScaffolderObjects\(\) \(l2tscaffolder.scaffolders.manager.ScaffolderManager class method\), 22](#)  
[l2tscaffolder.lib.file\\_handler \(module\), 17](#)  
[l2tscaffolder.lib.mapping\\_helper \(module\), 18](#)  
[GetScaffolderQuestionByName\(\) \(l2tscaffolder.scaffolders.manager.ScaffolderManager class method\), 22](#)  
[l2tscaffolder.scaffolders \(module\), 28](#)  
[l2tscaffolder.scaffolders.interface \(module\), 19](#)  
[GetScaffolderQuestions\(\) \(l2tscaffolder.scaffolders.manager.ScaffolderManager class method\), 22](#)  
[l2tscaffolder.scaffolders.manager \(module\), 21](#)  
[GetScaffolders\(\) \(l2tscaffolder.scaffolders.manager.ScaffolderManager class method\), 22](#)  
[l2tscaffolder.scaffolders.plaso \(module\), 23](#)  
[GitHelper \(class in l2tscaffolder.helpers.git\), 14](#)  
[l2tscaffolder.scaffolders.plaso\\_sqlite \(module\), 25](#)

## H

[HasBranch\(\) \(l2tscaffolder.helpers.git.GitHelper method\), 14](#)  
[l2tscaffolder.scaffolders.timesketch \(module\), 26](#)  
[l2tscaffolder.scaffolders.timesketch\\_index \(module\), 27](#)

## I

[IntQuestion \(class in l2tscaffolder.scaffolders.interface\), 20](#)  
[l2tscaffolder.scaffolders.timesketch\\_sketch \(module\), 27](#)  
[l2tscaffolder.scaffolders.turbinia \(module\), 28](#)

## K

[key\\_prompt \(l2tscaffolder.scaffolders.interface.DictQuestion attribute\), 19](#)  
[ListQuestion \(class in l2tscaffolder.scaffolders.interface\), 20](#)

## L

[l2tscaffolder \(module\), 28](#)  
[l2tscaffolder.definitions \(module\), 9](#)  
[l2tscaffolder.definitions.interface \(module\), 7](#)  
[l2tscaffolder.definitions.manager \(module\), 7](#)  
[l2tscaffolder.definitions.plaso \(module\), 8](#)  
[l2tscaffolder.definitions.timesketch \(module\), 9](#)  
[l2tscaffolder.definitions.turbinia \(module\), 9](#)  
[l2tscaffolder.frontend \(module\), 13](#)  
[l2tscaffolder.frontend.cli\\_output\\_handler \(module\), 9](#)  
[l2tscaffolder.frontend.frontend \(module\), 11](#)  
[l2tscaffolder.frontend.output\\_handler \(module\), 12](#)  
[l2tscaffolder.helpers \(module\), 15](#)  
[l2tscaffolder.helpers.cli \(module\), 13](#)  
[l2tscaffolder.helpers.git \(module\), 14](#)  
[l2tscaffolder.lib \(module\), 19](#)

## M

[MappingHelper \(class in l2tscaffolder.lib.mapping\\_helper\), 18](#)  
[mock\\_responses \(l2tscaffolder.helpers.cli.CLIFHelper attribute\), 13](#)

## N

[NAME \(l2tscaffolder.definitions.interface.ScaffolderDefinition attribute\), 7](#)  
[NAME \(l2tscaffolder.definitions.plaso.PlasoProject attribute\), 8](#)  
[NAME \(l2tscaffolder.definitions.timesketch.TimesketchProject attribute\), 9](#)  
[NAME \(l2tscaffolder.definitions.turbinia.TurbiniaProject attribute\), 9](#)  
[NAME \(l2tscaffolder.scaffolders.interface.Scaffolder attribute\), 21](#)  
[NAME \(l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder attribute\), 24](#)  
[NAME \(l2tscaffolder.scaffolders.plaso\\_sqlite.PlasoSQLiteScaffolder attribute\), 26](#)  
[NAME \(l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder attribute\), 27](#)

NAME (l2tscaffolder.scaffolders.timesketch\_index.TimesketchIndexScaffolder attribute), 27

NAME (l2tscaffolder.scaffolders.timesketch\_sketch.TimesketchSketchScaffolder attribute), 27

NAME (l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder attribute), 28

NoValidDefinition, 16

## O

OutputHandlerClick (class in l2tscaffolder.frontend.cli\_output\_handler), 9

## P

parser\_name (l2tscaffolder.scaffolders.plaso.PlasoParserScaffolder attribute), 24

PlasoBaseScaffolder (class in l2tscaffolder.scaffolders.plaso), 23

PlasoParserScaffolder (class in l2tscaffolder.scaffolders.plaso), 24

PlasoPluginScaffolder (class in l2tscaffolder.scaffolders.plaso), 24

PlasoProject (class in l2tscaffolder.definitions.plaso), 8

PlasoSQLiteScaffolder (class in l2tscaffolder.scaffolders.plaso\_sqlite), 25

preferred\_encoding (l2tscaffolder.helpers.cli.CLIHelper attribute), 13

PrintError () (l2tscaffolder.frontend.cli\_output\_handler.OutputHandlerClick method), 10

PrintError () (l2tscaffolder.frontend.output\_handler.BaseOutputHandler method), 12

PrintInfo () (l2tscaffolder.frontend.cli\_output\_handler.OutputHandlerClick method), 10

PrintInfo () (l2tscaffolder.frontend.output\_handler.BaseOutputHandler method), 12

PrintNewLine () (l2tscaffolder.frontend.cli\_output\_handler.OutputHandlerClick method), 10

PrintNewLine () (l2tscaffolder.frontend.output\_handler.BaseOutputHandler method), 12

PrintOutput () (l2tscaffolder.frontend.cli\_output\_handler.OutputHandlerClick method), 10

PrintOutput () (l2tscaffolder.frontend.output\_handler.BaseOutputHandler method), 12

PROJECT (l2tscaffolder.scaffolders.interface.Scaffolder attribute), 21

PROJECT (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder attribute), 24

PROJECT (l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder attribute), 27

PROJECT (l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder attribute), 28

PromptError () (l2tscaffolder.frontend.cli\_output\_handler.OutputHandlerClick method), 10

PromptError () (l2tscaffolder.frontend.output\_handler.BaseOutputHandler method), 12

PromptInfo () (l2tscaffolder.frontend.cli\_output\_handler.OutputHandlerClick method), 10

PromptInfo () (l2tscaffolder.frontend.output\_handler.BaseOutputHandler method), 13

PromptInfoWithDefault () (l2tscaffolder.frontend.cli\_output\_handler.OutputHandlerClick method), 10

PromptInfoWithDefault () (l2tscaffolder.frontend.output\_handler.BaseOutputHandler method), 13

## Q

queries (l2tscaffolder.scaffolders.plaso\_sqlite.PlasoSQLiteScaffolder attribute), 25

query\_columns (l2tscaffolder.scaffolders.plaso\_sqlite.PlasoSQLiteScaffolder attribute), 25

QUESTIONS (l2tscaffolder.scaffolders.interface.Scaffolder attribute), 21

QUESTIONS (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder attribute), 24

QUESTIONS (l2tscaffolder.scaffolders.plaso\_sqlite.PlasoSQLiteScaffolder attribute), 26

QUESTIONS (l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder attribute), 27

## R

RaiseIfNotReady () (l2tscaffolder.scaffolders.interface.Scaffolder method), 21

RaiseIfNotReady () (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder method), 22

RegisterDefinition () (l2tscaffolder.definitions.manager.DefinitionManager class method), 8

RegisterScaffolder () (l2tscaffolder.scaffolders.manager.ScaffolderManager class method), 22

RegisterScaffolders () (l2tscaffolder.scaffolders.manager.ScaffolderManager class method), 23

RegisterTemplate () (l2tscaffolder.lib.mapping\_helper.MappingHelper method), 18



required\_tables (l2tscaffolder.scaffolders.plaso\_sqlite.PlasoSQLiteScaffolder attribute), 25

RunCommand () (l2tscaffolder.helpers.cli.CLIHelper method), 13

RunCommand () (l2tscaffolder.helpers.git.GitHelper method), 14

## S

Scaffolder (class in l2tscaffolder.scaffolders.interface), 20

ScaffolderDefinition (class in l2tscaffolder.definitions.interface), 7

ScaffolderEngine (class in l2tscaffolder.lib.engine), 15

ScaffolderFrontend (class in l2tscaffolder.frontend.frontend), 11

ScaffolderManager (class in l2tscaffolder.scaffolders.manager), 21

ScaffolderNotConfigured, 16

SCHEMA\_QUERY (l2tscaffolder.scaffolders.plaso\_sqlite.PlasoSQLiteScaffolder attribute), 26

SetAttribute () (l2tscaffolder.scaffolders.interface.Scaffolder method), 21

SetModuleName () (l2tscaffolder.lib.engine.ScaffolderEngine method), 15

SetOutputName () (l2tscaffolder.scaffolders.interface.Scaffolder method), 21

SetProjectRootPath () (l2tscaffolder.lib.engine.ScaffolderEngine method), 15

SetScaffolder () (l2tscaffolder.lib.engine.ScaffolderEngine method), 16

SQLQuestion (class in l2tscaffolder.scaffolders.plaso\_sqlite), 26

Start () (l2tscaffolder.frontend.frontend.ScaffolderFrontend method), 12

StoreScaffolderAttribute () (l2tscaffolder.lib.engine.ScaffolderEngine method), 16

StringQuestion (class in l2tscaffolder.scaffolders.interface), 21

SwitchToBranch () (l2tscaffolder.helpers.git.GitHelper method), 15

## T

TEMPLATE\_FORMATTER\_FILE (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder attribute), 24

TEMPLATE\_FORMATTER\_FILE (l2tscaffolder.scaffolders.plaso\_sqlite.PlasoSQLiteScaffolder attribute), 26

TEMPLATE\_FORMATTER\_TEST (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder attribute), 24

TEMPLATE\_JOB\_FILE (l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder attribute), 28

TEMPLATE\_PARSER\_FILE (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder attribute), 24

TEMPLATE\_PARSER\_FILE (l2tscaffolder.scaffolders.plaso\_sqlite.PlasoSQLiteScaffolder attribute), 26

TEMPLATE\_PARSER\_TEST (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder attribute), 24

TEMPLATE\_PARSER\_TEST (l2tscaffolder.scaffolders.plaso\_sqlite.PlasoSQLiteScaffolder attribute), 26

TEMPLATE\_PLUGIN\_FILE (l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder attribute), 27

TEMPLATE\_PLUGIN\_FILE (l2tscaffolder.scaffolders.timesketch\_index.TimesketchIndexScaffolder attribute), 27

TEMPLATE\_PLUGIN\_FILE (l2tscaffolder.scaffolders.timesketch\_sketch.TimesketchSketchScaffolder attribute), 27

TEMPLATE\_PLUGIN\_TEST (l2tscaffolder.scaffolders.timesketch.TimesketchBaseScaffolder attribute), 27

TEMPLATE\_PLUGIN\_TEST (l2tscaffolder.scaffolders.timesketch\_index.TimesketchIndexScaffolder attribute), 27

TEMPLATE\_PLUGIN\_TEST (l2tscaffolder.scaffolders.timesketch\_sketch.TimesketchSketchScaffolder attribute), 27

TEMPLATE\_TASK\_FILE (l2tscaffolder.scaffolders.turbinia.TurbiniaJobTaskScaffolder attribute), 28

test\_file (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder attribute), 23

test\_file\_path (l2tscaffolder.scaffolders.plaso.PlasoBaseScaffolder attribute), 23

TestFileQuestion (class in l2tscaffolder.scaffolders.plaso), 24

TimesketchBaseScaffolder (class in l2tscaffolder.scaffolders.timesketch), 26

TimesketchIndexScaffolder (class in l2tscaffolder.scaffolders.timesketch\_index), 27

TimesketchProject (class in l2tscaffolder.definitions.timesketch), 9

TimesketchSketchScaffolder (class in l2tscaffolder.scaffolders.timesketch\_sketch), 27

`timestamp_columns`  
(*l2tscaffolder.scaffolders.plaso\_sqlite.PlasoSQLiteScaffolder*  
*attribute*), 25

`TurbiniaJobTaskScaffolder` (class in  
*l2tscaffolder.scaffolders.turbinia*), 28

`TurbiniaProject` (class in  
*l2tscaffolder.definitions.turbinia*), 9

TYPE (*l2tscaffolder.scaffolders.interface.BaseQuestion*  
*attribute*), 19

TYPE (*l2tscaffolder.scaffolders.interface.DictQuestion*  
*attribute*), 20

TYPE (*l2tscaffolder.scaffolders.interface.IntQuestion* at-  
*tribute*), 20

TYPE (*l2tscaffolder.scaffolders.interface.ListQuestion* at-  
*tribute*), 20

TYPE (*l2tscaffolder.scaffolders.interface.StringQuestion*  
*attribute*), 21

## U

`UnableToConfigure`, 16

## V

`ValidateAnswer()` (*l2tscaffolder.scaffolders.interface.BaseQuestion*  
*method*), 19

`ValidateAnswer()` (*l2tscaffolder.scaffolders.plaso.TestFileQuestion*  
*method*), 24

`ValidateAnswer()` (*l2tscaffolder.scaffolders.plaso\_sqlite.SQLiteQuestion*  
*method*), 26

`ValidatePath()` (*l2tscaffolder.definitions.interface.ScaffolderDefinition*  
*method*), 7

`ValidatePath()` (*l2tscaffolder.definitions.plaso.PlasoProject*  
*method*), 8

`ValidatePath()` (*l2tscaffolder.definitions.timesketch.TimesketchProject*  
*method*), 9

`ValidatePath()` (*l2tscaffolder.definitions.turbinia.TurbiniaProject*  
*method*), 9

`value_prompt` (*l2tscaffolder.scaffolders.interface.DictQuestion*  
*attribute*), 20

## W

`WrongCliInput`, 16