
spyglass

Release 0.0.1

Spyglass Authors

Jun 18, 2020

CONTENTS

1	Overview	1
1.1	Getting Started	1
1.2	Developer Quickstart Guide	3
1.3	Spyglass CLI	4
1.4	Running Spyglass in a Docker Image	9

OVERVIEW

Spyglass is a data extraction tool which can interface with different input data sources to generate site manifest YAML files. The data sources will provide all the configuration data needed for a site deployment. These site manifest YAML files generated by spyglass will be saved in a Git repository, from where Pegleg can access and aggregate them. This aggregated file can then be fed to Shipyard for site deployment / updates.

Spyglass follows a plugin model to support multiple input data sources. The currently supported plugin is the Spyglass Excel plugin ([spyglass-plugin-xls](#)).

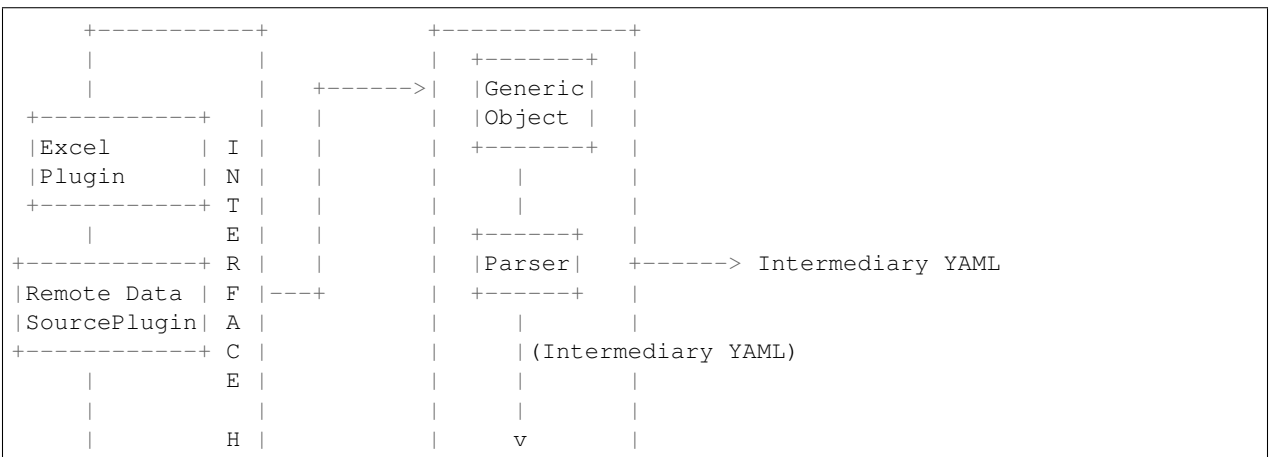
The Spyglass Excel plugin accepts an engineering spec in the form of a spreadsheet and an index file to read the spreadsheet as inputs and generates site level manifests. As an optional step, it can generate an intermediary YAML which contains all the information that will be rendered to generate Airship site manifests. This optional step will help the deployment engineer modify any data if required.

1.1 Getting Started

1.1.1 What is Spyglass?

Spyglass is a data extraction tool which can interface with different input data sources to generate site manifest YAML files. The data sources will provide all the configuration data needed for a site deployment. These site manifest YAML files generated by Spyglass will be saved in a Git repository, from where Pegleg can access and aggregate them. This aggregated file can then be fed to Shipyard for site deployment / updates. Reference: [airship-specs](#)

1.1.2 Architecture



(continues on next page)

(continued from previous page)

	A			+-----+	(templates)	+-----+
	N			Site	+<-----	Repository
	D			Processor	----->	Adapter
	L			+-----+	(Generated	+-----+
	E			^		Site Manifests)
	R			+---+	-----+	
				J2		
				Templates		
				+-----+		
+-----+				+-----+		

—

1.1.3 Supported Features

1. Spyglass Excel Plugin

1.1.4 Future Work

1. Schema based manifest generation instead of Jinja2 templates. It shall be possible to cleanly transition to this schema based generation keeping a unique mapping between schema and generated manifests. Currently this is managed by considering a mapping of j2 templates with schemas and site type.

1.1.5 List of Generated Site Manifests:

The Spyglass uses the plugin data source to generate the following site manifests:

- site-definition.yaml
- profile/
- profile/region.yaml
- baremetal/
- baremetal/nodes.yaml
- networks/
- networks/common_addresses.yaml
- networks/control-plane-addresses.yaml
- networks/physical/
- networks/physical/networks.yaml
- software/
- software/charts/
- software/charts/osh/
- software/charts/osh/openstack-tenant-ceph/
- software/charts/osh/openstack-tenant-ceph/ceph-client.yaml
- software/charts/ucp/

- software/charts/ucp/divingbell/
- software/charts/ucp/divingbell/divingbell.yaml
- software/config/
- software/config/corridor.yaml
- software/config/common-software-config.yaml
- deployment/
- deployment/deployment-strategy.yaml
- pki/
- pki/kubelet-node-pkicatalog.yaml

Spyglass maintains corresponding J2 templates for these files and then those are processed with site information obtained from plugin data source.

In some cases, the site might require additional site manifests containing static information independent of the plugin data received. In such cases one can just place the corresponding J2 templates in the appropriate folder.

1.1.6 Basic Usage

Before using Spyglass you must:

1. Clone the Spyglass repository:

```
git clone https://opendev.org/airship/spyglass.git
```

2. Install the required packages in Spyglass:

```
pip3 install pipenv && pipenv install
```

3. Launch the pipenv from your Spyglass directory:

```
pipenv shell
```

4. Install Spyglass into the pipenv:

```
pip3 install .
```

1.2 Developer Quickstart Guide

1. Clone the Spyglass directory. (Perform the following steps from inside the spyglass directory)

```
git clone https://opendev.org/airship/spyglass.git
```

2. Install external dependencies if not already installed.

```
sudo apt install -y python3-pip
sudo apt install -y tox
```

3. Install Pipenv.

```
pip3 install pipenv
```

4. Set up an environment with Pipenv

```
pipenv install
```

5. Enter the Pipenv environment.

```
pipenv shell
```

6. Install spyglass in the tox environment.

```
pip3 install .
```

7. Run spyglass on the example files to generate an intermediate document.

```
mkdir intermediate
spyglass excel documents -s airship-seaworthy -d intermediate -i \
  --excel-spec ../spyglass-plugin-xls/spyglass_plugin_xls/examples/excel_spec.
↪yaml \
  --excel-file ../spyglass-plugin-xls/spyglass_plugin_xls/examples/
↪SiteDesignSpec_v0.1.xlsx \
  --site-configuration spyglass/examples/site_config.yaml \
  --template-dir spyglass/examples/templates/
```

8. Run spyglass on the intermediate document to generate manifests.

```
mkdir manifest_dir
spyglass mi intermediate/airship-seaworthy_intermediary.yaml \
  -m manifest_dir/ -t spyglass/examples/templates/
```

1.3 Spyglass CLI

The Spyglass CLI is used in conjunction with the script `tools/spyglass.sh`.

Note: The default workspace for the `spyglass.sh` script is `/workspace`. The examples below require that this workspace be used.

1.3.1 CLI Options

-v / --verbose (Optional). False by default.

Enable debug logging.

Excel Plugin

Commands available under the Excel plugin package.

Generate Intermediary

Generates an intermediary file from passed Excel data.

```
./spyglass.sh excel intermediary -x <engineering_excel_file> \  
    -e <excel_spec> \  
    -c <additional_site_config> \  
    -s <site_name>
```

Options

-d / --intermediary-dir (Optional).

Path where the intermediary file will be created. Must be a writeable directory.

-x / --excel-file (Required for Excel plugin).

Path to the engineering Excel file. Multiple files can be included, provided they follow the same specification. Must be readable file(s) in a Microsoft Excel supported format (.xls, .xlsx, etc...).

-e / --excel-spec (Required for Excel plugin).

Path to the specification YAML that defines the content of the provided engineering Excel files. Must be a readable file in YAML format.

-c / --site-configuration (Optional).

Path to site specific configuration YAML. Must be a readable file.

-r / --rule-configuration (Optional).

Path to rules configuration YAML file. This file defines the rules used for data manipulation. Default rules are used if no rules YAML is entered.

--intermediary-schema (Optional).

Path to the intermediary schema to be used for validation.

--no-validation (Optional).

Skips validation on generated intermediary data.

-s / --site-name (Optional).

Name of the site for which the intermediary is generated.

Generate Manifests

Generates manifests from intermediary file created from passed Excel data. Intermediary data is always generated, but will not be saved unless specified.

```
./spyglass.sh excel documents -x <engineering_excel_file> \  
    -e <excel_spec> -c <additional_site_config> \  
    -s <site_name> -t <j2_template_directory>
```

Options

-i / --generate-intermediary (Optional). False by default.

Saves the intermediary file used to make the manifests created by the command.

-d / --intermediary-dir (Optional).

Path where the intermediary file will be created. Must be a writeable directory.

-x / --excel-file (Required for Excel plugin).

Path to the engineering Excel file. Multiple files can be included, provided they follow the same specification. Must be readable file(s) in a Microsoft Excel supported format (.xls, .xlsx, etc. ...).

-e / --excel-spec (Required for Excel plugin).

Path to the specification YAML that defines the content of the provided engineering Excel files. Must be a readable file in YAML format.

-c / --site-configuration (Optional).

Path to site specific configuration YAML. Must be a readable file.

-r / --rule-configuration (Optional).

Path to rules configuration YAML file. This file defines the rules used for data manipulation. Default rules are used if no rules YAML is entered.

--intermediary-schema (Optional).

Path to the intermediary schema to be used for validation.

--no-validation (Optional).

Skips validation on generated intermediary data.

-s / --site-name (Optional).

Name of the site for which the intermediary is generated.

-t / --template-dir (Required).

Path to the Jinja2 template files that will be used to generate manifest files. Must be a readable directory with Jinja2 files using the .j2 extension.

-m / --manifest-dir (Optional).

Path where generated manifest files should be written. Must be a writeable directory.

General

Generate Manifests from Intermediary

Generates manifests using an existing intermediary file. This is a shortcut to skip intermediary generation if it has already been completed.

```
./spyglass.sh mi <intermediary_file> -t <j2_template_directory>
```

Arguments

INTERMEDIARY_FILE (Required).

Path to an existing intermediary YAML file that can be used to generate manifests.

Options

-t / --template-dir (Required).

Path to the Jinja2 template files that will be used to generate manifest files. Must be a readable directory with Jinja2 files using the .j2 extension.

-m / --manifest-dir (Optional).

Path where generated manifest files should be written. Must be a writeable directory.

--force (Optional).

Forces manifests to be written, regardless of undefined data.

Validate Documents

Validates pegleg documents against their schema.

```
spyglass validate -d <DOCUMENT_PATH> -p <SCHEMA_PATH>
```

Options

-d / --document-path

Path to the document(s) to validate.

-p / --schema-path

Path to a schema or directory of schema files used to validate documents in document path.

1.3.2 Examples

Running Spyglass with Excel Plugin

```
spyglass excel documents -i -x <Excel File> -e <Excel Spec> \  
    -c <Site Config> -s <Site Name> -t <j2 template dir>
```

Generating intermediary and manifests

```
spyglass excel documents -i \  
    -x ../spyglass-plugin-xls/spyglass_plugin_xls/examples/SiteDesignSpec_v0.1.  
↪xlsx \  
    -e ../spyglass-plugin-xls/spyglass_plugin_xls/examples/excel_spec.yaml \  
    -c spyglass/examples/site_config.yaml \  
    -s airship-seaworthy -t spyglass/examples/templates/
```

Generating intermediary without manifests

```
spyglass excel intermediary \  
    -x ../spyglass-plugin-xls/spyglass_plugin_xls/examples/SiteDesignSpec_v0.1.  
↪xlsx \  
    -e ../spyglass-plugin-xls/spyglass_plugin_xls/examples/excel_spec.yaml \  
    -c spyglass/examples/site_config.yaml \  
    -s airship-seaworthy
```

Generating manifests without intermediary

```
spyglass excel documents \  
    -x ../spyglass-plugin-xls/spyglass_plugin_xls/examples/SiteDesignSpec_v0.1.  
↪xlsx \  
    -e ../spyglass-plugin-xls/spyglass_plugin_xls/examples/excel_spec.yaml \  
    -c spyglass/examples/site_config.yaml \  
    -s airship-seaworthy -t spyglass/examples/templates/
```

Generating manifests using intermediary

```
spyglass mi <intermediary.yaml> -t <j2 template dir>
```

Where sample *excel_spec.yaml* and *SiteDesignSpec_v0.1.xlsx* can be found in *spyglass-plugin-xls* in the *spyglass_plugin_xls/examples* folder. The Jinja2 templates and *site_config.yaml* can be found in the *spyglass/examples* folder.

Validate Documents

```
spyglass validate -d <DOCUMENT_PATH> -p <SCHEMA_PATH>
```

1.4 Running Spyglass in a Docker Image

This is a guide to creating a docker image locally and running Spyglass in the docker image.

1. Before creating the Spyglass image, ensure you have cloned [Spyglass](#), [Spyglass Excel plugin](#), and that you have installed [Docker CE](#).

2. Update the Makefile with desired Linux distribution:

```
DISTRO=(ubuntu_xenial or ubuntu_bionic or opensuse_15)
```

3. If necessary, change the PROXY variable to your proxy and change the USE_PROXY to true in the Makefile.

4. Go to the Spyglass directory in Terminal and run `make images`. When the build is done, you will see `Successfully built <DOCKER_IMAGE>`

5. From your home directory, create a new directory to store the output from docker with the following command:

```
mkdir -p ~/tmp/spyglass
```

6. Run Spyglass with the following command. The `-v` tag is used to mount the necessary local folders to the docker image. You must use the full path to the local folders you are mounting to. The path following the `‘:’` is a path inside the docker container. These folders will be made automatically by docker. (The directory you are mounting to must be `‘mnt’` or another directory owned by `airship` user in the docker image)

```
docker run -v /path/to/input/files:/mnt/input \
-v ~/tmp/spyglass:/mnt/output \
<DOCKER_IMAGE> spyglass excel documents -i \
-x /mnt/input/<excel_file_name> \
-e /mnt/input/<excel_spec_name> \
-c /mnt/input/<site_config_name> \
-s <site_name> -t /mnt/input/<templates_folder> \
-d /mnt/output -m /mnt/output
```

1.4.1 Example

The following shows the command used to create the site manifest and intermediary in the docker container using the example files provided in Spyglass and the Spyglass Excel Plugin. (`/root/path/` is the path to spyglass in your directory)

```
docker run \
-v /root/path/spyglass/spyglass/examples:/mnt/examples \
-v /root/path/spyglass-plugin-xls/spyglass_plugin_xls/examples:/mnt/examples_xls \
-v ~/tmp/spyglass:/mnt/output \
<DOCKER_IMAGE> spyglass excel documents -i \
-x /mnt/examples_xls/SiteDesignSpec_v0.1.xlsx \
-e /mnt/examples_xls/excel_spec.yaml \
-c /mnt/examples/site_config.yaml \
-s airship-seaworthy \
-t /mnt/examples/templates \
```

(continues on next page)

(continued from previous page)

```
-d /mnt/output/ \
-m /mnt/output/
```