JPL

February 21, 2025

Jet Propulsion Laboratory
California Institute of Technology

AUTHOR: Mike Smyth
Veljko Jovanovic

SUBJECT: SBG-TIR Level 1 V0.01 Delivery Memo

1. Introduction

This is the delivery of version 0.01 of the SBG-TIR Level 1 software to SDS. This is the initial delivery.

This delivers place holder PGEs for l1a_cal, l1b_rad and l1b_geo.

2. Capabilities and Features

The major capabilities and improvements delivered in this build included:

1. To exercise the delivery procedure and SDS interface, deliver placeholder PGEs for l1a_cal, l1b_rad, l1b_geo. This will be an initial first guess, it is expected that SDS will want to change the interface and how the delivery is done. But by having an initial version it helps work out the kinks of how future deliveries should be done.

3. Libraries and Executables Created in this Build

Top level executables are l1a_cal, l1b_rad, l1b_geo.

Software is installed in

```
/home/smyth/sandbox/sbg-install/v0.01
```

4. Docker image

Docker image is available at:

```
/home/smyth/sandbox/sbg-install/sbg-tir-l1-docker-0.01.tar.gz
```

5. Testing

Unit tests are run as part of the build procedures.

In addition, there is test data for a Level 1 end to end test, starting with Level 1A and proceeding through all of the Level 1B processing. This is run in docker. A number of volumes need to be mounted with needed ancillary data.

As a convenience, we have a small script in sbg-test-data – “docker_wrap”. This can be used to run docker commands, handling binding all the volumes. You can edit the script if you need to change the paths where anything is located.

For reference, the script is the following (although you can use the docker_wrap rather than running this directly).

```
docker run \
-v ${AFIDS_DATA_ELEV_ROOT}:/data/sbg_ancillary/srtm_v3_dem_L2:z \
-v ${AFIDS_DATA_LWM}:/data/sbg_ancillary/srtm_v3_lwm:z \
-v ${AFIDS_DATA_SPICEDATA}:/data/sbg_ancillary/cspice:z \
-v ${AFIDS_DATA_ORTHO}:/data/sbg_ancillary/ortho_lsat7:z \
-v ${SBG_OSP_DIR}:/data/sbg_ancillary/l1_osp_dir:z \
-v ${SBG_WORKDIR}:/home/sbg-workdir:z \
sbg-tir-l1/sbg-tir-l1:${SBG_VERSION} \
bash –login -c “<command>”
```

Once docker_wrap has been modified (if needed), you can run the end to end test script. This is done by:

```
cd <working directory>
/home/smyth/sandbox/sbg_ancillary/sbg-test-data/latest/end_to_end.sh
```

Note you may get error messages like:

```
rm: cannot remove './end_to_end_test_l1a_raw': No such file or directory
rm: cannot remove './end_to_end_test_l1a_cal': No such file or directory
rm: cannot remove './end_to_end_test_l1b_rad': No such file or directory
rm: cannot remove './end_to_end_test_l1b_geo': No such file or directory
```

You can ignore these, the script cleans up any output that may already exists and it is harmless if it doesn’t find these directories.

In the future, the end to end test will check against expected results. However now it just checks that the files get generated. If the test is successful, you will see the message:

```
Successful run
```

6. Change Requests and Problem Reports

The following CRs and PRs are closed in this build:

Issue Number	***Short Description***
5	Create placeholder PGEs for l1a_cal, l1b_rad, l1b_geo.
6	Blanket Ticket for V0.01 delivery

7. Liens and Waivers

This does not include the l1a_raw, l1ct_bt or l1cg_bt PGEs.

The PGEs that are delivered are place holders, taking input and generating dummy output.

The PGEs run much faster than the PGEs will once real functionality is put in place, so

these are not sufficient for doing timings.

The tir-node* don't yet have docker/podman, so the docker image was created and tested on the cartlab computers (using podman). The docker image should work without change on tir-node* once docker is available, but we haven't verified that.

The file naming convention is just a placeholder. This conforms to the naming convention in the SBG-TIR Level 1 PPSD, but this may change.

The delivery will evolve as SDS create delivery requirements.

8. Build/Run Requirements

Software runs depends on the following ancillary datasets:

```
/home/smyth/sandbox/sbg_ancillary/cspice
/home/smyth/sandbox/sbg_ancillary/l1_osp_dir
/home/smyth/sandbox/sbg_ancillary/ortho_lsat7
/home/smyth/sandbox/sbg_ancillary/srtm_v3_dem_L2
/home/smyth/sandbox/sbg_ancillary/srtm_v3_lwm
```

Test data for the test run is found at

```
/home/smyth/sandbox/sbg_ancillary/sbg-test-data
```

9. Build Instructions

It is not necessary to build the software. It has been built and is available at /home/smyth/sandbox/sbg-install.

If you do need to build the software, there are two steps involved. First we need to create a conda environment with all the dependencies (e.g., GeoCal). Second, we need to build the actual software. The deliveries are made as docker containers, so for a deliver there is a separate step of creating a docker image.

The software is available at git@github.com:sbg-tir/SBG-TIR-L1.git with the tag V0.01_rc1[1].

In addition, we have a conda build of afids/geocal at git@github.jpl.nasa.gov:Cartography/afids-conda-package.git with the tag sbg-tir-0.01.

Git clone the SBG-TIR-L1 source, and the afids conda packages. There is a makefile in

1 Note, over the years I've developed the convention of delivering an initial version with a -rc1 tag, so that if there are any issues in the delivery we can update to a rc2 etc. Once the delivery has been completed, we can create a final tag without the "rc" extension to mark the final version that was used in the delivery.

the L1 source “env” directory that can be used to install the conda environment. Create a Makefile.local with any updates to the paths in the Makefile[1], and then install with

```
make create-env
```

You can also delete and recreate the environment using:

```
make recreate-env
```

You can then activate the created environment before building and installing in the next step.

The software is built using the normal configure/make cycle:

```
../../sbg-tir-l1/configure –prefix=$CONDA_PREFIX
make -j 20 all && make install && make -j 20 check && \
make end-to-end-check
```

The docker build is a separate process, found in the directory docker-env. Create a Makefile.local[2] with any paths that need to be updated, and then run:

```
make create-base
```

to create the environment with dependencies and then

```
make create-sbg
```

to build and install the SBG environment. We separate the two steps because the base environment will be more stable and needed to be updated less frequently. You can rerun the “create-sbg” step as needed without regenerating the “create-base”. Running “create-base” is only needed if there are changes to the conda/docker base environment.

10.Alternative installation of conda environment

Note that our afids-conda-package uses git LFS for the local conda packages, and needs to live in the JPL github. There is separately a public version of this repository at https://github.com/Cartography-jpl/afids-conda-package. This has all the package build recipes, but not the actually built packages since they are too large for the public github. As an alternative, we attach assets to the releases that can be directly installed (i.e., like miniconda). For example https://github.com/Cartography-jpl/afids-conda-package/releases/download/20240916/afids-20240916-Linux-x86_64.sh. Download the latest of these releases, and install:

1 It is preferable to create a Makefile.local rather than editing the Makefile so that any future updates will appear without issue in a git merge, and your local changes are preserved in the separate Makefile.local file

2 See previous footnote about using Makefile.local rather than editing the Makefile

```
bash afids-sbg-tir-0.01-Linux-x86_64.sh
```

11.Miscellaneous Instructions or Information

None