IVS GSFC Technology Development Center

Ed Himwich, John Gipson, Dave Horsley, Mario Bérubé

Abstract This report summarizes the activities of the GSFC Technology Development Center (TDC) and describes plans for the future. The GSFC TDC develops station software including the Field System (FS), Monitoring and Archiving System (MAS), IVS session Web page software, and scheduling software (*sked*); hardware including tools for station timing and meteorology; scheduling algorithms, and operational procedures. It provides a pool of individuals to assist with station implementation, check-out, upgrades, and training.

1 General Information

The IVS GSFC Technology Development Center (TDC) develops hardware, software, algorithms, and operational procedures. It provides manpower for station visits for training and upgrades. Other technology development areas at GSFC are covered by other IVS components such as the GSFC Analysis Center. The current staff of the GSFC TDC consists of John Gipson, Ed Himwich, Dave Horsley, and Mario Bérubé. The remainder of this report covers the status of the main areas supported by the TDC.

2 Field System

The GSFC TDC is responsible for the development, maintenance, and documentation of the Field System (FS) software package. The FS provides equipment control at VLBI stations. It interprets the *.snp* schedule and *.prc* procedure files (both as prepared by *drudg* from the *.skd* schedule file). The FS controls the antenna, data acquisition hardware, and related ancillary equipment needed for making VLBI measurements. All major VLBI data acquisition backends are supported. The FS is customizable to allow it to control station-specific equipment. It is used at almost all of the IVS Network Stations (more than 35) and also at many stations that perform VLBI only for astronomical observations. The only major VLBI facilities not using the FS are the DSN, LBA, VLBA, and VERA.

The main work on the FS for the period was:

- Conversion of the source to be 32- and 64-bit compatible. The key point for this conversion was to avoid the use of `long` variables, particularly in fixed length data structures. A utility *unlongify* was written to convert occurrences of `long` variables to `int` variables. Afterward it was necessary to change variables back if they were used in system calls that required `long` variables. Changes also had to made manually for the use of pointers and `time_t` variables, both of which are different lengths for 32- and 64-bit systems. The *unlongify* utility is available for stations needing to convert their station code for use on a 64-bit platform.
- Placing the source under *git* and distributing it using *github*. This effort included importing the existent history of FS source code. It includes over 130 FS9 versions (under Linux), 17 FS8 versions (under

NVI, Inc.

GSFC Technology Development Center

IVS 2019+2020 Biennial Report

VENIX), and two older versions (under HP RTE-1000/A) going back to version 5.5 in 1988. Having the source code in git greatly simplified the task of merging the main and VGOS branches. Having the historical code in *git* is a great resource for understanding its evolution.
- Merging the “main” and “VGOS” branches. The branches were merged using *git*, which made it more manageable and tracked the detailed changes that were made.
- Supporting Jonathan Quick (HartRAO) for development of FSL10, based on Debian *Stretch*. FSL10 is the current standard Linux distribution for the FS. It is expected to be under long-term support (LTS) until June 2022. The installation instructions can be found at: https://nvi-inc.github.io/fsl10/

The new FS was available for beta testing in 2020, The official release was in February 2021, and the update notes can be found at:

https://nvi-inc.github.io/fs/releases/10/0/10.0.0.html

2.1 Plans for the Future

Several other improvements are expected in future releases, including:

- Merging EVN-developed support for the DBBC3 rack;
- Adding support for R2DBE racks;
- Adding support for chopper-wheel and hot/cold load calibration methods;
- A complete update to the documentation and conversion to a more modern format that will be easier to use and maintain;
- *chekr* support for Mark 5A and Mark 5B systems;
- Support for periodic firing of the noise diode during observations;
- Completion of the VEX2 standard and implementation of it;
- Further unification of the Patriot 12-m (GGAO) and ISI 12-m (Kokee Park and McDonald) antenna interface code. This will allow a common code base to be used for the two very similar Antenna Control Units (ACUs), which are also used at other locations;
- FS Linux 11, based on Debian *Bullseye*.

3 Monitor and Archiving System (MAS)

The GSFC TDC is also responsible for development, maintenance, and documentation of the Monitoring and Archiving System (MAS) software package—formerly named TIG after its components: Telegraf, InfluxDB, and Grafana—and hardware specification. The MAS provides a system for collecting, storing, processing, and visualizing time-series data collected from various components of a VLBI station.

The software suite is comprised of several open-source packages along with some custom software specific for VLBI stations. The system is capable of collecting data from the Field System and PC diagnostic subsystems as well as certain meteorological devices, backends, and antennas. The suite can easily be expanded to include site-specific data. Currently the system is deployed at the NASA-managed stations, and the hardware specification and software are available to the community. There were no significant changes during this period.

4 Automation

The GSFC TDC is responsible for maintaining the IVS session Web pages, maintaining the master schedule, and providing analysis and scheduling of IVS sessions. These activities require finding, downloading, and validating many files stored on IVS or correlator data centers. To avoid duplication of efforts, an integrated approach has been developed to support automation of some activities at GSFC. The automation system is built around a database for rapid access to information and a message broker (https://en.wikipedia.org/wiki/Message_broker) for controlling data flow between processes. The approach is also using small processes controlled by the message broker. This is easier for maintenance than a large application. Also, adding new functionalities does not require changes to existing code but only re-routing the data flow using the message broker. Independent watchdog processes monitor the automated system, ensuring continuous operation.

For rapid access to available information, a database has been developed to store information about sessions, catalogs, available files, and some emails. This approach avoids reading numerous text files many times to find and validate any information. Master files and catalogs have been transferred to the database that is maintained by an automated process that validates any new master or catalog files. This database is also used for storing inventory of files on IVS Data Centers. This is mainly used for keeping the IVS session Web pages but also for synchronizing GSFC data files with IVS Data Centers.

Building the inventory of IVS Data Centers is done using ftp and http crawlers, or scanners, that are serving multiple purposes. When new files are detected, the scanners inform internal processes using the message broker. Using the same information, one process will update the IVS session Web pages while another one will download and pre-process the file.

One of the main objectives of this automated system was to support the analysis effort by ensuring that all files required for processing are available. A special scanner looks for any new vgosDB files on correlator data repositories. Using the message broker, the scanner initiates a special process that downloads, extracts, and pre-processes the vgosDB files. The analysts are then informed that a new database is ready for analysis. Post analysis processes have been improved by developing a new application for generating and submitting analysis reports and products.

After each observing session, GSFC also provides a network performance report that is used by correlators and analysts. This activity uses stations' emails and logs to generate a report on possible problems at stations. An automated process reads emails to extract session and station information and store it in a database. Logs are used to detect any antenna or recording problems. A small report of detected problems and emails are combined in a "sumops" file that is submitted to IVS Data Centers for each session.

IVS Operation Centers are required to submit observing sessions' schedule files seven days prior to the sessions' observing dates. Emails were sent in the past regarding missing schedules, but these emails disappeared after upgrades at CDDIS. A new automated process has been implemented to send emails directly to the IVS Operation Center that has not submitted a schedule in time.

The automated system has been operational for many months, and new functionalities are being developed to make it more robust and improve day-to-day operations.

5 *sked* and *drudg*

The GSFC TDC is responsible for the development, maintenance, and documentation of *sked* and *drudg*. These two programs are very closely related, and they operate as a pair for the preparation of the detailed observing schedule for a VLBI session and its proper execution in the field. In the normal data flow for a geodetic schedule that is written with *sked*, first *sked* is run at an Operation Center to generate the *.skd* file that contains the full network observing schedule. Then each station uses the *.skd* file as input to *drudg* to make the FS schedule and procedures for that station. Catalogs are used to define the equipment, stations, sources, and observing modes that are selected when writing a schedule with *sked*.

Changes to *sked* and *drudg* are driven by changes to equipment and by feedback from the users. The following sub-section summarizes some of the important changes to these programs during the report period. This summary includes only the most important bugs that were found and fixed over this period. A more complete summary of the changes can be found in the `change_log.txt` files associated with *sked* and *drudg*.

5.1 *sked* and *drudg* Changes

Overall the changes to *sked* and *drudg* were modest during this period, and no major features were added. The changes can be summarized as follows:

- The `implicit none` statement was added at the top of the files of almost all of the subroutines. The only exception was a set of externally written subroutines for which the change would have required a lot of work.
- The removal of obsolete features continued. Examples include support for S2 recorders, headstack, and passes. Inactive parts of code that refer to these features might still exist, but if schedules do not refer to headstacks or passes, they should run fine.
- The VGOS and S/X versions of *drudg* were merged. This involved changes to approximately 30 subroutines.

5.2 Catalog Changes

The *sked* catalogs were updated as new VGOS stations became operational, the equipment at stations changed, stations were added to existing observing modes, or new observing modes were requested.

5.3 Plans for the Future

Plans for the future include the following:

- Make VEX the native format for *sked*, so that there will be no more `.skd` files.
- With the finalization of VEX2, sked and drudg will be modified to read (and, in the case of *sked*, write) VEX2 schedules.