An I&T’s Guide to Building the SIRTF Downlink Sub-system

F. J. Masci (09/15/04)

This document outlines the steps required to successfully deploy and build the downlink software system and associated SDM dependencies for use in automated pipelines. It is primarily intended for the Integration and Testing team (I&T). It is generic enough for building on either a Solaris (FORTE) or Linux (Intel) machine.

1. In the Configuration-Management CVS repository (used by I&T):
   ../cm/tools/build, three files are required for a Solaris build:
   build_downlink_IandT.csh, twoPassBuild.csh, and downlink_IandT.env.
   For a Linux build, the three required files are: build_downlink_IandT.csh, twoPassBuild.csh, and downlink_IandT_linux.env.

2. Ensure the directory path specified by the SOS_VERSION environment variable in either downlink_IandT.env (Solaris) or downlink_IandT_linux.env (Linux) exists and if so, cd into this directory. If this directory does not exist, you must make it before-hand.

3. From within the directory specified by SOS_VERSION, export or check-out a delivery of “/downlink” and concurrent deliveries of “/sdm” and “/common” from CVS. For the S11.0 delivery, the associated tag for downlink is BugFixTagS11:
   % cvs checkout –r BugFixTagS11 –P downlink
   % cvs checkout –P sdm
   % cvs checkout –P common
   The “–P” option removes (or “prunes”) empty directories.

4. The cspice library under “/common” must be built.
   On a Solaris machine, type:
   % cd /common/cspice/VN52a
   % ./buildCspiceVn52a.csh
   On a Linux machine, type:
   % cd /common/cspice/VN52a
   % ./buildCspiceVn52a_linux.csh

5. Copy the three files (either build_downlink_IandT.csh, twoPassBuild.csh, and downlink_IandT.env for Solaris or build_downlink_IandT.csh, twoPassBuild.csh, and downlink_IandT_linux.env for Linux) to the /downlink sub-directory.

6. On the very first build, execute the following in your /downlink directory:
   % ./build_downlink_IandT.csh
   This will perform the two-pass build automatically.

7. If you wish to perform a complete build again in the same /downlink directory, you only need to execute the second pass build by executing:
   % ./twoPassBuild.csh 2
8. The entire build process takes \( \approx 4 \) hours on a 500 Mhz Solaris machine or \( \approx 1 \) hour on a 2.2 GHz Linux machine. A log file called build_log is generated under /downlink which contains a log of all environment variables and the build process. For the Solaris build, \( 209 \) binaries and \( 27 \) libraries (*.a) are expected under /downlink/bin and /downlink/lib respectively. For the Linux build, \( \approx 126 \) binaries and \( \approx 24 \) libraries (*.a) are expected. These Linux numbers are approximate since porting of downlink software for Linux compliance is incomplete and currently progressing.

9. If there is a need to rename hard-coded environment variable paths in files/scripts for deployment on different systems (e.g., I&T versus OPS), here’s the procedure:
   i. cd to the directory which contains your files/scripts.
   ii. execute: 
       
       \$SOS_VERSION/downlink/perltools/replaceword 'old_string' 'new_string' *
       
       where \$SOS_VERSION is defined in downlink_IandT.env above. This operation will replace ‘old_string’ with ‘new_string’ globally in all files.

10. Also, depending on whether deployment is being done on I&T or OPS, there are special deployment instructions for the QA CGI scripts which reside in the following directory: \$SOS_VERSION/downlink/infrastructure/apache/cgi-bin/qa. Please follow instructions in the header of the file “deployQaCgiScripts.csh”, located in this same directory.

11. Procedure for “patch builds”:
   i. If re-building downlink software with dependency on SDM libraries, you must ensure that the SDM library: “libsdm_sodb.so” is up to date. If not, you must first build it:
      - Update/synchronize the sdm repository under /SOS_VERSION/sdm.
      - cd /SOS_VERSION/downlink
      - copy into this directory the scripts: /cm/tools/build/sdm_downlink_dep_build.csh and /cm/tools/build/downlink_IandT.env (sdm software is only Solaris compliant).
      - From within /SOS_VERSION/downlink, execute.
        ./sdm_downlink_dep_build.csh . This will build libsdm_sodb.so and place it in its required place.
   ii. If re-building individual C/C++/Fortran or Java modules (or anything where a binary is expected):
        - source downlink_IandT.env (for Solaris) or source downlink_IandT_linux.env (for Linux)
        - cd /SOS_VERSION/downlink/<desired_dir>, where <desired_dir> is the directory containing the source code you wish to build.
        - cvs update
        - type “make clean”
        - type “make” and the binary will be made under /SOS_VERSION/downlink/bin
   iii. If deploying wrapper and perl scripts which reside under /SOS_VERSION/downlink/plexec/wrappers/<dir>,
- cd /SOS_VERSION/downlink/plexec/wrappers/<dir> where <dir> is the subdirectory containing the script(s) you wish to deploy.
- cvs update
- copy new/changed files to the following directory level: /SOS_VERSION/downlink/plexec/wrappers
- chmod a+x ../wrappers/*.pl

iv. If deploying anything to do with INGEST which resides under /SOS_VERSION/downlink/infrastructure/ingest,
- cd /SOS_VERSION/downlink/infrastructure/ingest,
- If re-deploying any perl script, cvs update and copy the script to the directory /SOS_VERSION/downlink/scripts
- If re-building any of the ingest binaries: Ingest, Subscribe, ReSubscribe or Sleep, cvs update, type “make clean” and then “make”. The binaries will be made under /SOS_VERSION/downlink/bin and the perl scripts will be automatically copied to /SOS_VERSION/downlink/scripts