SemRep for public distribution and production:

The steps to complete a new SemRep release are described below. This set of instructions assumes that you were able to compile and run SemRep locally. These steps mainly involve ensuring that the new compilation can be accessed from the Scheduler (production SemRep) and preparing a SemRep distribution that can be made publicly available. Some of these steps are not completely automated, but can be done so in future releases.
The distribution and production code for the latest release currently resides in *$NLS/specialist/SAW.Linux* (referred to as *$SAW_PROD* henceforth). This directory contains *ARCHIVE* directory where older versions of SemRep are archived. Before completing the production and distribution steps, it is advisable to archive *$SAW_PROD*[1]:

rsync –avz --exclude=ARCHIVE $NLS/specialist/SAW.Linux $NLS/specialist/SAW.Linux/ARCHIVE/ SAW_<version_number>

Distribution:
SemRep public distribution is created using the makefile *$SAW_HOME/Makefile.dist*. This makefile will probably need to be modified for every release. Currently, it includes targets to copy files needed for distribution to a specified distribution directory and build the necessary tar/zip/checksum files.

1- Ensure that distribution-specific files are up-to-date. These include the following (at the time of this writing):
    a. Documentation regarding the new release (*$SAW_HOME/doc* directory)
    b. SemRep output files in *$SAW_HOME/TestSuite* directory must be up-to-date (i.e., they must have been recreated using the new release).
    c. Scripts in *$SAW_HOME/bin* directory (*install.sh* and **.in*, specifically)
    d. *$SAW_HOME/Makefile.dist*
2- Copy the distribution files:
    a. Ensure that $PUBLIC_SEMREP, $PUBLIC_SEMREP_PATH and $PUBLIC_SEMREP_DIST_PATH variables are set correctly in *Makefile.dist*. $PUBLIC_SEMREP_PATH points to the directory under *$SAW_PROD* where the SemRep distribution files will be copied ($PUBLIC_SEMREP is the name of this directory) and $PUBLIC_SEMREP_DIST_PATH points to the directory where tar/zip/checksum files will be copied for distribution.
    b. Use the command
       SAWenv make –f Makefile.dist populate
       to copy all the local SemRep files to the appropriate distribution directories under *$SAW_PROD*[2].

[1] This step is not necessary if version control is used, obviously, but I prefer to archive in both ways.

[2] Note that the target *populate* invokes other sub-targets for copying individual directories and these sub-targets could also be invoked individually, if necessary.

3- Follow the SemRep installation steps to ensure that distribution in $PUBLIC_SEMREP_PATH directory can be installed and run successfully using the Test Suite. [3]

4- Create tar/zip/checksum files for distribution using the command

SAWenv make –f Makefile.dist dist

This command will copy the resulting files to $PUBLIC_SEMREP_DIST_PATH directory. The contents of this directory can then be made available through the website.

Production:

Preparing SemRep for production involves updating the scripts in *$NLS/bin* and ensuring that Scheduler can use these scripts. In addition to SemRep, we make available through Scheduler our modified version of MetaMap (called SemSKR).

Binary files needed for production version should have been already copied into *$PUBLIC_SEMREP_PATH/$PUBLIC_SEMREP/bin* directory, when the distribution files were copied. These files are: *SemRep.sav, MetaMap.sav, semrep.<version>.BINARY.Linux, semskr.<version>.BINARY.Linux*. Furthermore, installation procedure creates the scripts needed to run these binaries in the same directory. These scripts are: *semrep.<version>* and *semskr.<version>.*

The production scripts in *$NLS/bin* call these scripts in *$PUBLIC_SEMREP_PATH/$PUBLIC_SEMREP/bin*. SemRep production script is called *semrep.<version>* and SemSKR production script is called *semskr.<version>*. Furthermore, there are two symbolic links in *$NLS/bin*, *semrep* and *semskr*, which point to *semrep.<version>* and *semskr.<version>* in *$NLS/bin*, respectively. Scheduler Interactive and Batch SemRep point to semrep symbolic link. We do not have a specific page for SemSKR on Scheduler, but it can be run through Batch Generic with Validation.

When SemRep and SemSKR are prepared for Scheduler, the new *semrep.<version>* and *semskr.<version>* scripts need to be created in *$NLS/bin*. The simplest way of doing this is to copy the old scripts for the previous release in the same directory and change the content to point to the correct scripts in *$PUBLIC_SEMREP_PATH/$PUBLIC_SEMREP /bin*. Furthermore, the symbolic links *semrep* and *semskr* should be updated to point to these new scripts.

Once these updates are complete, Jim Mork should be informed so that he can change his scripts if necessary. If SemRep options have changed in the new release and this affects the displayed options on Batch/Interactive SemRep pages, this information should also be provided to him. Furthermore, the SemRep (and SemSKR) functionality should be tested via Scheduler and it should be ensured that the results are the same as when SemRep is run locally with the same data.

Finally, I copy contents of the local SemRep and SKR directories to *$SAW_PROD*. This is not an absolute necessity and is done mostly for historical reasons. It used to be the case that production SemRep and SemSKR scripts were more closely tied to the local directory structure more and it was easier to just copy the local structure over and make the necessary changes; however, this has changed with public distribution version, which reorganizes the local structure and includes only files necessary to run the

[3] This, of course, assumes the presence of MetaMap distribution alongside the SemRep distribution directory. For now, I am assuming that MetaMap distribution has been copied manually to the appropriate directory. In the next release, we can extend *Makefile.dist* to copy MetaMap distribution automatically, as well.

programs . Since source code and other supplementary files are not included in the public distribution, I think it is a good idea to keep copying the local directory structure for the time being. [4]

```
rsync –avz  --exclude=’.svn*’ $SAW_HOME / $SAW_PROD/
```

I also copy *$SKR_HOME* directory to *$SAW_PROD,* so that the entire codebase is under *$SAW_PROD:*

```
rsync –avz  --exclude=’.svn*’ $SKR_HOME  $SAW_PROD/
```

[4] Also note that Local SemRep Compilation document refers to this directory as one method of obtaining the latest version of SemRep, so if someone without access to version control software (Subversion) wants to compile SemRep, this is the preferred method for the time being.