Using the OPSWARE Code Workspace

Introduction

The OPSWARE workspace controls all code that may be edited for use in the Mission/Science Operations Center. This includes IDL procedures, shell scripts, compiled programs, and itos resources such as STOL procedures and display pages.

The main workspace is /disks/opsware/integration. Files are edited in subdirectories of the src directory in the workspace. For example, all the archive scripts are edited in

/disks/opsware/integration/src/archive

Executable programs and other useable resources are kept in other directories within the workspace. For instance, all scripts and compiled programs are executed from:

/disks/opsware/integration/SunOS.5.6/bin

As another example, all useable ITOS resources for HESSI are kept in

/disks/opsware/integration/itos_resource/hessi/...

Setup Files

The following files must be "sourced" in the user's .cshrc file:

```
source /usr/local/setup/setup_spro
source /disks/opsware/integration/setup
```

The first sets up the user to use the correct workspace control software. To make sure you're using the right software, run "which" on the two commands "bringover" and "workspace":

```
which bringover
(should print /disks/apollo/export/SUNWspro/bin/bringover)
which workspace
(should print /disks/apollo/export/SUNWspro/bin/workspace)
```

The second configuration file above sets up the user's environment so the software within the workspace is available via environment variables. Check that the OPSHOME environment variable is set:

```
echo $OPSHOME
(/disks/opsware/integration)
```

Note that the second setup file above may be replaced with ~/ows/setup if the user wishes to use the files installed in her own workspace instead of those in the parent workspace. See "Make Your Own Setup File".

Building a Child Workspace

Initial Creation

Decide which source code directories of the workspace you would initially like to acquire. (You may add others later.) The existing directories are:

src/archive - Archiving scripts.
src/fastscripts - Scripts and such specifically for FAST operations.
src/opsUtil - Various utilities for general operations.
src/idlUtil - IDL routines (most imported from the old FAST workspace)
src/itos_resource - ITOS resource files (pages, procs, etc) for each mission.
src/config - Configuration files used by utilities within the workspace.

The last thing to decide is where to put your child workspace. A valid example would be ~/ows, for "Operations WorkSpace". Now you may create your own child workspace and bring over some directories with this command:

```
bringover -w ~/ows -p /disks/opsware/integration  \
   src/Makefile.default  src/config  src/opsUtil
```

The -p option names the parent workspace, and the -w option names the child workspace, which must not yet exist.

The above assumes you've chosen, ~/ows as your child workspace, and you only want to bring over the archive and config directories. Note that you must explicitly name the Makefile.default file in the bringover. (This file is referenced by all other makefiles.)

To install files from your new source directories, change the current working directory to each of the subdirectories of *src* and enter "make install". This will create new directories in the root of the workspace and install files there.

Bringing Over other Workspace Directories

After you have created your child workspace, you may wish to bring over additional directories from the parent workspace. If, for example, you wanted to get the directory tree containing HESSI STOL procedures, you would cd into your workspace directory and enter the commands:

```
bringover -n src/itos_resource/hessi/procs      # To see what would happen
bringover src/itos_resource/hessi/procs         # To actually get the files
```

Adding New Files to the Workspace

Copy your new file into the appropriate subdirectory of the /src/ directory in your workspace. To put it under configuration control, enter the command:

```
sccs create <filename>
```

This will add the edit history file s.filename to the SCCS subdirectory. You probably do not need to edit the Makefile since it should sense the addition automatically. See the section "Makefiles" below.

Checking Out Files for Editing

Files should only be edited in your child workspace. Go to the src subdirectory containing the file and enter the command:

```
sccs edit <filename>
```

Type "sccs info" to see that SCCS has checked out the file for editing. Note also that you now have write permission on the file.

Checking In Files after Edting

After you've saved the file, enter the command:

```
sccs delget <filename>
```

This updates the SCCS history file (the s.file) and restores the file permission to read-only.

Putting Files Back into the Parent Workspace

The *putback* command should always be run with the -n option the first time to see what the workspace *would* do. If you want to put back all the changed files in a directory, /src/archive for example, use the commands:

```
putback -n src/archive              # To see what would happen
putback src/archive
```

If you want to put back specific files instead of a whole directory, you may list them explicitly:

```
putback  src/archive/script1.ksh  src/archive/script2.pl
```

Installing Files for Use

Installing in the Parent Workspace

To install software for everyone else to use, you must become the opsw user because she owns the parent operations workspace. Login as opsw and cd into the main workspace:

```
cd /disks/opsware/integration
```

Next, go to the src subdirectory containing the files you want to install. The "make install" command uses the current directory's Makefile to install files where they may be accessed by other users.

Like the *putback* command, *make* should always be run first with the -n option to see what the command *would* do. Go into the src subdirectory containing the file(s) you wish to install. Then enter:

```
make -n install
(check the output)
make install
```

Installing in Your Own Workspace

You can also do a "make install" in your child workspace. This installs files in directories within your own workspace. This is useful for testing your changes before putting them back into the parent workspace and installing them there. See "Source Your Own Child Workspace".

Makefiles

The makefiles are designed to need minimal upkeep, even when adding new software to the workspace. The Makefiles automatically sense the targets that must be built in the current directory. For instance, if you've created script.ksh in the src/archive directory and you enter "make install", the Makefile automatically instructs the make utility to create a copy of the file without the .ksh suffix, and then copy this file to the *bin* directory. Below is a description of the Makefiles in the various workspace directories.

src/archive/Makefile

This Makefile automatically handles all scripts ending in ".ksh", ".csh", and ".pl". During a make install, the files are copied to the same name without the suffix and then installed in the bin directory. The full path of the bin directory is kept in the OPSBIN environment variable. In addition, any ".pm" files in this directory are installed in the Perl Module directory, accessible via the OPSPLMOD environment variable. This directory is meant to contain only those scripts specific to archiving.

src/opsUtil/Makefile

This Makefile is exactly the same as the archive Makefile above. This directory contains scripts and programs specific to mission operations.

src/idlUtil/Makefile

Simply installs all ".pro" files in the IDL directory that is included in the IDL_PATH environment variable.

src/config/Makefile

Copies all files to the *cfg* directory of the current workspace. These files are then accessible via the OPSCONFIG environment variable.

src/itos_resource/<mission_name>/pages/Makefile

The Makefiles in all the ITOS pages directories handle all files ending in ".page", ".sprt", and ".plot". These files are simply installed in a directory in the current workspace that is identical to the current one, exluding the "/src/" section of the path.

src/itos_resource/<mission_name>/procs/Makefile

The Makefiles in all the ITOS pages directories handle all files ending in ".proc". These files are simply installed in a directory in the current workspace that is identical to the current one, exluding the "/src/" section of the path.

src/itos_resource/<mission_name>/cfgmon/Makefile

The Makefiles in all the ITOS pages directories handle all files ending in ".cfg". These files are simply installed in a directory in the current workspace that is identical to the current one, exluding the "/src/" section of the path.

Miscellaneous Workspace Commands

Renaming or Moving Files

If you must change the name of a file in your workspace, go to the directory where it exists and enter:

```
workspace filemv <newfile> <oldfile>
```

After you change the name in your workspace, you perform a putback into the parent workspace to propagate the new name to other users' workspaces.

Deleting Files

To remove a file from the workspace, enter the following command from the directory in which the file exists:

```
workspace filerm <filename>
```

The file is then moved to the deleted_files subdirectory of the workspace. The next time you do a putback, the change will be recorded in the parent workspace also.

Creating Directories

Creating new directories involves adding a new Makefile and possibly editing the OPSWARE setup script. Ask the person in charge of the workspace to do this.

Source Your *Own* Child Workspace

To test changes you've made in your own workspace, you may wish to "source" your own workspace instead of the parent. This means all programs et cetera come from the installation in your own workspace, and none from /disks/opsware/integration.

Making Your Own Setup File

You must first bringover some setup files from the parent workspace. The following command executed in your own workspace will get all necessary files:

```
bringover src/Makefile src/FAST.flp.sh src/FASTpwd.sh \
            src/setup.bourne.sh src/setup.sh src/updaterel.sh
```

These files are necessary to create the setup file that references your own workspace instead of the parent. In your ~/ows/src directory, run the following command:

```
make install
```

You will notice that a file called *setup* has been created in the root directory of your workspace. The principal difference between this file and the parent's is that $OPSHOME is set to your own workspace instead of /disks/opsware/integration.

Sourcing Your Setup File

To reference your own workspace in all subsequently started shells, make the following substitution in your .cshrc file:

```
# source /disks/opsware/integration/setup
source $HOME/ows/setup
```

To change the environment of the current shell only, run the following commands:

```
source $HOME/ows/setup
source $OPSCONFIG/setup_hessi_itos
```

The above assumes your workspace is in ~/ows, and you wish to configure ITOS for the HESSI mission.

Potential Problems

Since you have disconnected yourself from the parent workspace, your own workspace must contain all the operations software you plan to run, and it must have been *installed* in your workspace. For example, notice that the OPSCONFIG environment variable is now set to a directory in your own workspace. You will be unable to **source $OPSCONFIG/setup_hessi_itos** unless you have already brought over the *src/config* directory, and run **make install** in that directory.

Furthermore, you will not be able to run ITOS unless you have brought over *src/itos_resource/<mission>/dbx* and run **make install** there. This installs the mission operational database, without which ITOS will not start. (To circumvent this particular problem without compiling your own database, simply set ITOS_DBDIR to /disks/opsware/integration/itos_resource/<mission>/odb in your ~/.itosrc.)

Example: Updating a FAST Script for the Isolated Ops Network (ION)

```
rlogin <localhost>                          # Login to a machine on the open network
cd ~/ows                                    # Go to your child workspace
bringover src/fastscripts                   # Update workspace from parent if necessary
cd ~/ows/src/fastscripts                    # Go to directory containing file to be edited
sccs info                                   # See what's being edited in this directory
sccs edit tcwfast.ksh                       # Check out file for editing
emacs tcwfast.ksh                           # Edit the file (and save it)
sccs diffs tcwfast.ksh                      # See what changes you've made
sccs delget tcwfast.ksh                     # Check file back in
make install                                # Install all edited files into your personal
                                            # executable area.
                                            # In this case, the file will install to:
                                            # ~/ows/fastopsbin/tcwfast
putback src/fastscripts                     # Check any edited files into parent workspace
rlogin -l opsw <localhost>                  # Login as owner of OPSWARE workspace
cd /disks/opsware/integration/src/fastscripts # Go to corresponding directory in parent ws.
make install                                # Install all edited files into the world executable
                                            # area.  This particular file will install to:
                                            # /disks/opsware/integration/fastopsbin/tcwfast
ssh ravi                                    # Secure Shell to machine on isolated network
                                            # (may have to be from juneau)
rlogin -l opsw ravi                         # Login as owner of OPSWARE workspace
cd /disks/opsware/integration/fastopsbin    # Go to directory where executable will reside
rcp surya:/disks/opsware/integration/fastopsbin/tcwfast .      # Remote copy the file to ION
rlogin -l fastops ravi                      # Login as user that will run script
which tcwfast                               # Make sure script is in PATH
```