The ASGS Developers Guide

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Introduction

This document describes the guidelines that ADCIRC Surge Guidance System developers should use when developing new code for ASGS and its related utilities. It also presents a short tutorial for Subversion (svn), the version control software that we use to coordinate development and ease the task of merging in new features.

2 Guided Tour

The following is a list of directories and files that are provided with the ASGS along with a description of their purpose and function.

2.1 base

The root of the ASGS directory structure contains the scripts that provide the main features of the ASGS.

2.1.1 asgs_main.sh

The actual script that is executed to activate the ASGS is called asgs_main.sh. Its main function is to perform a hindcast if needed, then move to a nowcast/forecast loop. It contains code that controls the preparation of new runs, sets up computational jobs, submits them, and monitors them. It also contains a small section that contains conditionally executed system-specific configuration information.

All the code that actually does the specialized work, e.g., downloading meteorological data, constructing control files, generating visualizations of results, etc., is contained in modular external scripts.

The asgs_main.sh script is the most central and most important one in the ASGS.

2.1.2 control_file_gen.pl

The ADCIRC fort.15 and SWAN fort.26 (run control) files are generated by control_file_gen.pl. This script accepts a wide array of input arguments that describe the sort of run the Operator has configured, including the use of tidal forcing, wave coupling, time step size, desired output file format (ascii vs netcdf), and many others.

Other input arguments to this script provide state information, e.g., whether the model is to be hotstarted, the current simulation time, the advisory number from the latest National Hurricane Center Forecast/Advisory (in the case of hurricane-related runs), etc.

When the script runs, it uses templates that have been made for the control files along with the input arguments and a few external programs (for generating tides, etc) and the already-generated meteorological forcing files (if any) and produces control files for ADCIRC or ADCIRC+SWAN along with metadata in the run.properties file.

This is the 2nd most important script in the ASGS (after asgs_main.sh).

2.1.3 get_atcf.pl

The get_atcf.pl script is used by the ASGS to monitor the RSS feed from the National Hurricane Center for new Forecast/Advisories and to download the latest advisory the moment it becomes available. It does this by parsing the xml file posted on the RSS feed, and when the advisory number has been updated by the NHC, get_atcf.pl follows the link to the html file containing the actual advisory information.

After downloading a new Forecast/Advisory from the National Hurricane Center website, the script then downloads the latest hindcast (BEST track) file for the same storm from NHC's anonymous ftp site. Both these files are required for generating an ADCIRC fort.22 (meteorological forcing) file.

Once a forecast is complete, and the next advisory has not yet been posted, the script waits 60 seconds before checking the RSS site again. It will continue checking every 60 seconds until the next advisory is posted.

The script is called get_atcf.pl because the so-called BEST track file format (that describes the files that the script downloads from the anonymous ftp site) was designed by the Automated Tropical Cyclone Forecasting system (ATCF). Please see the section below that discusses the ADCIRC parametric vortex models below for more details on this format, including a link to the online documentation.

2.1.4 nhc_advisory_bot.pl

The nhc_advisory_bot.pl script takes the html file downloaded by get_atcf.pl script and parses out all the information required to write an ATCF formatted forecast file ("OFCL"). It then generates the ATCF formatted file with the forecast parameters; this file could be used directly by one of ADCIRC's internal parametric vortex models, but the ASGS subjects it to further processing as described below.

2.1.5 storm_track_gen.pl

The storm_track_gen.pl script takes the ATCF-formatted file generated by nhc_advisory_bot.pl (representing the forecast), the ATCF-formatted file downloaded by get_atcf.pl (representing the hindcast), and the currrent simulation state (representing the current hotstart time) and constructs an appropriate ADCIRC fort.22 (meteorological forcing) file as output.

This script accepts optional arguments that can be used to vary the storm's track (expressed as a percentage of the desired distance between the consensus track and the edge of the cone of uncertainty), the radius to maximum winds (expressed as a percentage of the radius to maximum winds determined from the input), the overland speed of the storm (to speed it up or slow it down), and the maximum wind speed. These variations are configured in the overall ASGS configuration file and can be used to explore "what-if" scenarios during a tropical cyclone event.

2.1.6 get_nam.pl

The get_nam.pl script is used by the ASGS to monitor NCEP's anonymous ftp site for newly released output from their NAM (North American Mesoscale) atmospheric model. The WRF-NMM (Weather Research and Forecasting, Non-hydrostatic Mesoscale Model) is used to produce NAM results.

The script accepts the current time of the ADCIRC simulation as input, and downloads the latest available NAM data, provided that the latest available data is later than the current ADCIRC simulation time. If there are no NAM data available that postdate the current ADCIRC simulation time, the script print and information message and exits. The calling routine (in asgs_main.sh) then retries the download every 60 seconds until new data have been discovered and downloaded.

The get_nam.pl script relies on the "wgrib2" executable to perform basic sanity checks on the grib2 files that it downloads.

2.1.7 NAMtoOWI.pl

The NAM output files in grib2 format that were downloaded by get_nam.pl are converted into a form that ADCIRC (or ADCIRC+SWAN) can use by the NAMtoOWI.pl script. This script uses the wgrib2 utility to extract the sea level u, v, and p from the NAM output files (in grib2 format) and produce ascii data files. The ascii data are then reprojected from the NAM coordinate system (lambert conformal) to the ADCIRC coordinate system (geographic) using the awip_lambert_interp executable. A list of points must be provided where the reprojected and interpolated data should be calculated. Finally, the interpolated data are output in OWI (Ocean Weather Inc) format for use in ADCIRC or ADCIRC+SWAN (NWS=12).

2.1.8 get_flux.pl

If time-varying river flux boundary condition data (ADCIRC's fort.20 file) are available, and the ASGS has been configured to include river flux forcing, the get_flux.pl script retrieves the boundary condition files and formats them for use in ADCIRC. It must reads the ADCIRC mesh file to determine the number of river flux boundary nodes so that it can interpret the data in the incoming flux files, since the file format is not self describing. The script also takes into account the current time in the ADCIRC simulation and the files that are currently available on the remote ftp site when constructing the flux boundary condition file that ADCIRC will actually use.

2.1.9 queue script generators

The ASGS supports many HPC platforms, and it seems that each one is different from the others and has its own idiosyncracies. As a result, there are a number of queue script generation scripts, including erdc.pbs.pl, loadleveler.pl, queenbee.pbs.pl, ranger.serial.pl, ranger.sge.pl, and tezpur.pbs.pl. Although there are different scripts for different platforms, they are broadly similar, in that they all use a template approach . . . a working queue script has "blanks" inserted for parameters that will change for different simulation runs and then a script is written (often cloned from one of the existing ones) that simply fills in the blanks according to the command line arguments.

2.2 config

The "config" subdirectory contains sample configuration files to illustrate the changes to be made for various situations.

2.2.1 config/asgs_config_irene_garnet_ec95d.sh

This is a sample configuration file that sets the ASGS up to run hurricane Irene on the Garnet machine at ERDC using the small ec95d mesh.

2.2.2 config/asgs_config_irene_swan_garnet_r3.sh

This sample configuration file illustrates the settings required for the ASGS to run hurricane Irene with SWAN coupling on the Garnet machine at USACE ERDC using the FEMA R3 mesh for the US mid-Atlantic coast.

2.2.3 config/asgs_config_irene_swan_river_br_ncv6b.sh

During hurricane Irene, this sample configuration file was used in North Carolina to run the ASGS on the Blueridge Dell linux cluster at RENCI with SWAN coupling on the NC v6b mesh.

2.3 doc

This directory contains the following documents: ASGSDevGuide.pdf, ASGSDevGuide.html, ASGSDevGuide.txt, README.asymmet
README.fort.22.meta, ASGSDevGuide.txt, README.asymmetric2, and README.storm_track_gen.

2.4 input

The input subdirectory contains the input files (mesh files, nodal attributes files, etc) and templates for dynamically generated input files as well as dynamically generated queue scripts. It also contains the ptFile_gen.pl script for generating a set of points for reprojecting NAM data into geographic projection.

2.5 logs

This subdirectory is meant for storing the log files that the ASGS generates during execution. The Operator should change to this subdirectory before executing the ASGS.

2.6 output

The output subdirectory contains all the routines that can be used when dealing with output and post processing. The code in this subdirectory can be categorized as follows:

1. Notification via email that new results are available: cera_notify.sh, ncfs_cyclone_notify.sh, ranger_notify.sh, corps_cyclone_notif
   ncfs_nam_notify.sh, ranger_parttrack_notify.sh, corps_nam_notify.sh, notify.sh.

2. File format transformations and conversions: asgsConvertR3ToNETCDF.pl, asgsConvertToNETCDF.pl, station_transpose.pl.
3. Controlling the generation of line plots, contour plots, images, etc.: ncfs_post.sh, part_track_post.sh, ranger_post.sh, post.sh, corps_post.sh, part_track_post_new.sh, r3_post.sh, asgsCreateKMZs.sh.
4. Posting results files to other locations over the network: corps_post.sh, asgsConvertR3ToNETCDF.pl.
5. Archiving of results: ncfs_archive.sh, ranger_archive.sh.

2.6.1 output/PartTrack

This subdirectory contains infrastructure for performing particle tracking post processing.

2.6.2 output/POSTPROC_KMZGIS

This subdirectory contains code for generating Google Earth (kmz) images and GIS shape files. Much of the underlying technology was developed by RENCI.

2.6.3 output/TRACKING_FILES

This subdirectory contains infrastructure for performing particle tracking post processing.

2.7 PERL

The PERL subdirectory contains the DateCalc perl module that is used for date math during execution, particularly for preparation of control files and meteorological forcing files.

2.8 tides

The tides subdirectory contains code and data for dynamically providing nodal factors and equilibrium arguments for simulations that include tidal forcing. It also has infrastructure for setting tidal boundary conditions.

3 Installation

Steps required to get it going on a machine that is already supported.

System requirements. ADCIRC version requirements, etc.

4 Configuration

Steps required to get it going on a machine that is already supported.

5 Execution

Starting the ASGS. Looking at the log file. Making things easier with the *screen* utility.

6 Meteorological Forcing

The ASGS is capable of using gridded met fields from NCEP's NAM model, or two different types of vortex forcing with ADCIRC's asymmetric wind models (NWS9 and NWS19).

6.1 NAM Forcing

The NAM data is downloaded from NCEP and converted to OWI format for use in ADCIRC.

6.2 Storm Track Generation for Vortex Models

README.storm_track_gen

The script storm_track_gen.pl accepts a raw ATCF-format hindcast file (straight from the NHC ftp site) and a raw ATCF-format forecast file (either straight from the NHC ftp site or formed by some other method, e.g., parsed from a text advisory). It produces a fort.22 file for ADCIRC for the NWS=9 (asymmetric wind vortex model) option.

Examples showing the use of command line parameters are as follows:

--dir /path/to/atcf/files : Required. This is used to provide the path to the ATCF formatted hindcast and forecast files.

--storm 01 : Required. The NHC numbers each tropical cyclone of the season, and the storm number is required to determine the names of the input files.

--year 2009 : Required. The year is used in the names of the input files as well.

--coldstartdate 2009090100 : Required. The asymmetric wind model assumes that the first line in the fort.22 file is the one to be used when execution begins (whether the start of execution represents a coldstart or a hotstart). As a result, the correct line in the hindcast and forecast file must be selected to reflect the desired cold start time (YYYYMMDDHH24).

--hotstartseconds 86400.0 : Required if execution begins as a hotstart. Because the asymmetric wind model within ADCIRC will always start at the top of the fort.22 file at the beginning of execution, the fort.22 file must begin at the date and time that matches the hotstart file, if any. If this command line argument is not given, the script assumes a cold start (i.e., that hotstartseconds is zero).

--name nhcConsensus : Optional. If this argument is not given, the script assumes that the fort.22 for the NHC consensus forecast should be generated. Other possible variations include maxWindSpeed, overlandSpeed, and veer. In the future, variation of rMax will be supported as well. Only one --name argument is allowed per execution.

--percent 50 : Optional. This argument controls the magnitude of variation for the maxWindSpeed, overlandSpeed, and veer variations. For the maxWindSpeed and overlandSpeed variations, a positive percentage increases the speed and a negative percentage decreases the speed. For the veer variation, -100 percent represents the left edge of the cone of uncertainty, and 100 percent represents the right edge of the cone of uncertainty. The default percentage variation is 20 percent for maxWindSpeed, -20 percent for overland speed, and 100 percent for veer.

Selecting the right hindcast or forecast line to start with based on the coldstart time and hotstart seconds, and filling in the date, time, and forecast periods correctly on the following lines requires the bulk of the code in the script. Most of the complexity is due to the fact that ADCIRC may be required to start anywhere in the hindcast and forecast, and it is possible to have a hindcast that is later than the forecast. Creating the storm variations requires a comparatively small amount of code.

In addition to filling in the required forecast period, filling in the date in the forecast portion of the file, and performing the requested variation, the script modifies the data from the NHC in three other ways:

1. The background pressure is changed to 1013 mb.
2. The wind radii, which are required by the asymmetric vortex model, are filled in if they are missing. These fields are populated according to the following procedure: (a) if the wind radius in any of the four quadrants is zero, that quadrant's radius will be filled in with the average of the nonzero radii at that point in time; (b) if all the radii are zero, they will all be set equal to the correpsonding values from the previous time level; (c) if there are no values from the previous time level, they will be set to the then-current hindcast Rmax in the case of a hindcast line, or to the nowcast Rmax in the case of a forecast line.
3. The central pressure is filled in according to an experimental algorithm under development by Jason Fleming. This algorithm uses a family of negatively sloped lines that relate changes in central pressure to changes in max wind speed. This algorithm is being refined for storms in the Gulf. It is a topic of current research and is targeted for publication. As such, it is subject to change and improvement over the course of the season.

The script assumes that the files are named bal[storm][year].dat and al[storm][year].fst. For example, the first tropical cyclone in the Atlantic basin in 2009 will have files named bal012009.dat and al012009.fst.

At the beginning of the script's source code, additional technical assumptions are listed that may be of interest.

6.3 Original Asymmetric Vortex Wind Model (ADCIRC NWS=9)

jgf20090623: Correction: hour 0 in the fort.22 corresponds to the cold start time (ADCIRC time=0) if the simulation is cold started, or to the hot start time if the simulation is hotstarted.

README.asymmetric (NWS=9 in fort.15)

The asymmetric vortex wind model input differs from most other types of wind input in ADCIRC in that it consists of storm parameters, rather than the meteorological data itself. ADCIRC then uses these parameters to generate the actual meteorological data internally at each node at every time step during the simulation. The result is that the fort.22 files are very small (e.g. 20kB), in comparison with fort.22 files that contain actual meteorological data. Furthermore, preprocessing these files is very fast because it consists only of copying the fort.22 file as-is to each subdomain directory.

The format of the fort.22 file for the asymmetric wind model is the ATCF (a.k.a. "BEST track") format. This format was developed by the U.S. Navy, and ATCF stands for Automated Tropical Cyclone Forecast. Historical tracks, real-time hindcast tracks and real-time forecast tracks may be found in this format. The format is documented in detail at the following web site:

http://www.nrlmry.navy.mil/atcf_web/docs/database/new/abrdeck.html

One important thing to remember about this format is that it looks like CSV data, but it is actually fixed column width data. Never change the width of the columns when you edit this data!

It is assumed by the asymmetric wind code within ADCIRC that the first entry in the fort.22 file corresponds to the cold start time (ADCIRC time=0) if the simulation is cold started, or to the hotstart time if the simulation is hotstarted. Therefore, the forecast period (column #6) needs to be edited by the workflow scripts to reflect the time of the forecast/nowcast for each track location (each line) in hours from the start of the simulation (0, 6, 12, 18, etc). The original data in that column depends on what type of best track format data is being used. The original data might have 0 or other numbers in that column.

For more details on how to prepare the input files for the asymmetric wind model, see the documentation in the ADCIRC manual under NWS=9:

http://adcirc.org/documentv48/parameter_defs.html#NWS

Real time hindcast and forecast advisories are made available by the NHC in ATCF format via their ftp site every six hours when there is an active tropical cyclone. These files are downloaded by the workflow and used to generate fort.22 files for ADCIRC.

Since NHC forecast advisories do not contain the radius of maximum winds, this option uses the standard radii at specific wind speeds (34, 50, 64, 100 knots) reported in the four quadrants (NE, SE, SW, NW) of the storm to calculate the radius of maximum winds as a function of the azimuthal angle.

The asymmetric wind model is activated by setting NWS=9 in the fort.15 file and adding a WTIMINC line in the appropriate place in this file. The WTIMINC line is not used, but is maintained for file-format compatilbility with the symmetric vortex wind model, and for possible future expansion.

The asymmetric vortex wind model was developed and coded by Craig Mattocks (cmattock@email.unc.edu) and implemented in ADCIRC by Cristina Forbes (cforbes@email.unc.edu).

Reference:

Mattocks C. and C. Forbes, 2008: A real-time, event-triggered storm surge forecasting system for the state of North Carolina, Ocean Modelling, 25, 95-119, doi:10.1016/j.ocemod.2008.06.008

6.4 "What-if" Asymmetric Vortex Wind Model (ADCIRC NWS=19)

README.asymmetric2 (NWS=19 in fort.15)

The extended asymmetric vortex wind model is selected by setting NWS in the ASGS configuration script to 19.

This wind model is based on the code that generates the original asymmetric vortex wind data (NWS=9), and its behavior is similar in many ways to that model. As a result, this documentation will focus mainly on the differences between the two wind models.

The extended asymmetric vortex wind model was developed for several reasons: (1) to allow more visibility into the parameters, such as Rmax, that control a storm's size and shape and are calculated within the wind model code; (2) to allow the user to control parameters such as Rmax so that they may be adjusted by the user; and (3) to allow the user to deterministically compensate for input data that are missing or nonexistent (such as wind radii in various quadrants for a particular isotach).

The mechanism for achieving the goals described above is a preprocessing program called the asymmetric wind input preprocessor, or aswip, that takes the ATCF formatted input data that would normally be used for the NWS8 or NWS9 and adds columns to it that describe the following things:

1. The Rmax from the hindcast (i.e., BEST lines) has been persisted from the Rmax column (described as MRD in the ATCF documentation) from the value in the forecast (i.e., OFCL lines).
2. The storm direction DIR and speed SPEED in the ATCF file have been replaced with the calculated direction and speed to be used by ADCIRC. The values are provided in the same format as the ATCF file to provide compatibility between methods. The speed in given in knots and the direction is given in compass coordinates, with zero degrees indicating North and values increasing clockwise.
3. In the 2nd column after the storm name, the cycle number is provided. A *cycle* is an entry or set of entries in the file that all have the same storm time or forecast period. For cycles that have more than one isotach, this value will be repeated for each isotach (starting from 1 for the first cycle in the file).
4. The 3rd column after the storm name contains the number of isotachs that are reported for that particular cycle. This value is also repeated on each line for each isotach that is reported per cycle. For example, if the cycle has a 34kt and a 50kt isotach entry then this column will contain a *2* for both entries in that cycle.
5. The following 4 columns contain the flags that tell the ADCIRC NWS 19 code whether or not to use a particular wind radius from the isotach under consideration. There is a flag for each quadrant. A *0* indicates that the wind radius for that isotach and quadrant will not be used. A *1* indicates that the wind radius for that isotach and quadrant will be used.

For example: if only the 34kt isotach is provided, then then all four wind radii must be used, and the columns will all be set to *1* : … 34 … 1 1 1 1 1 …

Another example: if 3 isotachs are provided then the columns may look like the following: … 34 … 3 0 0 0 0 … … 50 … 3 0 0 1 1 … … 64 … 3 1 1 0 0 … this indicates - use NO radii from the 34 kt isotach use the 3 & 4 quadrant radii from the 50 kt isotach use the 1 & 2 quadrant radii from the 64 kt isotach

Users could potentially modify these flags in the input file to manually select which radii to use for each cycle.

1. In the next 4 columns, the calculated Rmax for each quadrant is listed in the following order: NE SE SW NW.
2. The next column contains the overall Holland B value.

Finally, one valuable aspect of this file format is that it can be used by NWS8, NWS9, or NWS19, since the original data have not been modified. The extra columns are used as input by NWS19, and as a result, they provide both metadata and control over these parameters.

7 Metadata Generation

When the ASGS Workflow creates a fort.22 file, it is difficult to tell how that file has been modified from the original, just by looking at it. The Workflow (specifically, storm_track_gen.pl) is capable of generating variations in track, overland speed, and maximum wind speed. Modifications to radius to max wind will be provided for in a future ASGS release (beyond v1.2).

In order to clearly indicate whether the track has been varied, and if so, how the track has been varied, two metadata mechanisms are provided. First, a file with a fixed name, fort.22.meta, is created that contains a description of the wind file. In addition, a symbolic link is made to the fort.22 file that contains encoded information about the data in the file. These mechanisms will now be described in turn.

The fort.22.meta file uses the # symbol to denote comments, so lines starting with # are ignored. The other lines are key value pairs using the delimiter *:*. Everything on one line before the first delimiter is considered to be the key, everything after that until the end of the line (not including the linebreak) is the value.

The order of the key/value pairs is arbitrary, and the names of the keys are subject to extension and/or change. In order to manage any changes, and make it future-safe, a version specification in the first line is a requirement. This would make it possible to read in the files after changes in the format/key names easily later.

An example of a fort.22.meta file is as follows:

# Example fort.22 meta file version:1 year:2008 storm:07 advisory:01 hostname:tezpur.hpc.lsu.edu directory advisory:/work/jgflemin/07
directory storm:/work/jgflemin/072008/01/nowcast modified:y variation veer:+40

The shorthand mechanism for fort.22 metadata consists of a symbolic link to the fort.22 file. The name of the link encodes the track variations according to the following format:

YYYYSSAA[c|m]_w[|-]PPo[|-]PPv[|-]PPPr[|-]PP

where YYYY is the four digit year SS is the number of that storm for that year AA is the advisory number c means consensus or m means modified w[|-]PP is the 2 digit, positive or negative percent variation in max wind speed o[|-]PP is the 2 digit, positive or negative percent variation in over land speed v[|-]PPP is the 3 digit, positive or negative percent variation in veer r[|-]PP is the 2 digit, positive or negative percent variation in rmax (not supported by storm_track_gen.pl as of ASGS v1.2).

As a concrete example of the above format that provides the same track information as the fort.22.meta file described above:

20080701m_w+00o+00v+040r+00

This symbolic link can be found by looking for a filename that follows this format; there will not be another file in the directory whose name matches this format.

8 Development Strategy

The ASGS development strategy consists of the maintenance of two separate versions at any given time: the stable version and the development version. The stable version only receives bug fixes, while the development version receives bug fixes as well as new features.

At some point, the development version becomes stable. At that time, a stable branch is created from it, and then new features are added to the previosly stable code, creating a new development version. At that time, the major version number will be incremented, e.g., stable v48.xx will become the next development version as v49.00.

The minor version number is changed each time there is a change in the code and the modified code is committed back to the repository. Details of the implementation of this strategy with Subversion are provided in the section entitled "Subversion".

9 Pioneering

How to put the ASGS on a machine where it has never run before.

10 Subversion

Subversion is a version control system. That is, it provides a means for many software developers to collaborate on a single software project.

A subversion repository is a storehouse of code for a particular project. Subversion repositories consist of a trunk, which is the mainline code that everyone shares, and any number of branches that are created by and for individual users.

A branch may be created by a developer that has a lot of changes to make over a longer period of time. This lone developer can work on a branch without affecting the mainline code. Once the changes in the branch are satisfactory, the branch can be merged back to the trunk, making the changes available to all.

A further advantage of Subversion is that it automates the process of merging new features into ASGS. For example, in the past, without Subversion, developers modified their local copies of ASGS and wanted to merge the modifications into the mainline code. This required hours or days of painstakingly examining each change they had made and cutting and pasting the changes one by one into the up-to-date version of ASGS. With Subversion, the software examines the changes that were made locally and makes corresponding changes to the mainline version in an automated way.

10.1 Policies

Working with subversion requires greater coordination among ASGS developers. For example, if two developers change the same line of code in two different branches in two different ways, this will create a conflict that will have to be resolved manually.

Furthermore, if one developer makes changes to trunk that prevent the code from compiling, it will be difficult for other developers to continue working if they update to the latest code. As a result, ASGS development with Subversion will adhere to the following policies:

- Communicate. Jason Fleming is responsible for making sure ASGS development is smooth, pain-free and productive---when in doubt, email him (jason.fleming@seahorsecoastal.com).
- Make a branch to develop new features, rather than making changes to trunk.
- Don't commit changes to trunk that prevent the code from compiling, at least on your platform. You should also confirm that your code compiles with and without NetCDF.

10.2 Instructions

Comprehensive documentation for Subversion is available: http://svnbook.red-bean.com/.

The ASGS code is hosted at the following Subversion repository location:

```
http://www.seahorsecoastal.com/svn/asgs
```

To check out code from the trunk, please type

```
svn checkout http://www.seahorsecoastal.com/svn/asgs/trunk
```

Here are a few examples of things that we commonly do with svn.

10.2.1 Compare Revisions

In order to see the changes between revisions of the repository, type (for example)

```
svn diff -r 15:16 control_file_gen.pl
```

10.2.2 Package Up Code From Repository

In order to create a tar file that contains just the ASGS code and excludes the .svn directories, the first step is to export the code to a new subdirectory. For example:

```
svn export ./trunk ./asgs_today
```

Would copy the local files under version control from the trunk subdir to the asgs_today subdir, excluding the .svn files. Particular revisions of the svn repository can also be extracted, see the svn documentation for details on how to do this.

10.2.3 Pull Old Version from Repository

One advantage of using svn is that it is easy to go back to an older version of the code, if necessary. The first thing to do in this case is usually to look at the svn log to see what changes were made, so that you can select the right version of the repository to extract. Go to the subdirectory of the code tree that you want an older version of and issue the command

```
svn log
```

Look at the log entries, and note the revision that you'd like to extract. Let's say you want to retrieve asgs/trunk at revision 65 of the repository (for example). Issue the following command:

```
svn checkout -r 65 http://www.seahorsecoastal.com/svn/asgs/trunk
```

10.2.4 Make a Branch

To make a branch off trunk to add a new feature and call it the myfeature branch:

```
svn copy http://www.seahorsecoastal.com/svn/asgs/trunk http://www.seahorsecoastal.com/svn/
```

Then to start using the newly created branch:

```
svn checkout http://www.seahorsecoastal.com/svn/asgs/branches/myfeature
```

A This Document

This document was prepared from the text file ASGSDevGuide.txt using software called asciidoc (http://www.methods.co.nz/-asciidoc/). The document can be formatted as an html page with the command

```
asciidoc ASGSDevGuide.txt
```

or formatted as a pdf with the command

```
a2x --format=pdf ASGSDevGuide.txt
```