JStrack

Version 3.0.1

by Jim Graham, N5IAL

Installation and Usage

This is an update of the first cut at real documentation for JStrack, something JStrack has been missing for far too long. It probably isn't complete yet. It's probably got some bugs (most likely things like typos, typesetting errors, etc.) that I haven't caught yet. *PLEASE* send any comments to me via the JStrack e-mail list (see http://www.jstrack.org/jstrack for details).

JStrack users please note: I have gone through two cancers (the first in 2006, and the second in 2011). The first cancer, and its treatment, did some damage to a number of things that impact my memory, typing, reading, and ability to get timely updates out. My typing is the direct result of a tumor in the frontal lobe of my brain, in a part of the brain whose name I'm not likely to remember. I might spell words backwards, mix two different words (e.g., twr woods), and so on. For reading, that was a 2.5 cm tumor in my left occipital lobe, which is the lobe of the brain that processes visual information. So I may read something that LOOKS right, but is, in fact, wrong.

In working on the updates for this documentation, I am trying to make the updates make sense. But just remember, the Downloads page has instructions, too, and if it says one thing, and this file says another, the Downloads page is almost certainly the correct version. Like I said, sometimes what I read is not what's actually on the page. And yes, it is very annoying.

So please, be patient with me. For what it's worth (which, to me, is a LOT), I am still alive; the two cancers are not.

Typeset with TEX on 17 September 2012.

TABLE OF CONTENTS

1. Installing Tcl/Tk (Required to Run JStrack)

Note that this is a general overview for installing Tcl/Tk for the specific purpose of running JStrack. This useful scripting language has many capabilities and many installation options that are not covered here. Installation details may vary depending on your needs and your system.

1.1 For Windows 9x/NT/2k/ME/XP

Windows users, please note that JStrack was not designed for Windows—it was designed to run on Unix systems (e.g., SunOS, Linux, FreeBSD, Mac OS X, and so on). I have done what I can to make using JStrack on Windows as easy as possible, but if you're expecting a single setup.exe, you will be disappointed—it's not quite that simple. The simplest way, in my opinion, is to download the ActiveTcl 8.4.x (8.4.19 is the last of the 8.4.x line) distribution provided on the Downloads page, and work from there. Depending on how long it's been since a major release, you may also have to download and install patches from the Downloads page. Instructions for these will be on the Downloads page. As a general rule, however, just remember that JStrack has a very specific directory structure, and patches are generally just files that replace other files within that directory structure.

1) Download the following from the JStrack downloads page (**http://www.jstrack.org/jstrack/downloads/**) to a directory:
   - EITHER:
     - **ActiveTcl 8.4.15** distribution. This one distribution contains all of the extra packages you will need for JStrack (as of version 3.0.1), minus a few non-Tcl items that are packaged with the JStrack distribution.
   - OR
     - The "Windows EXE Files for JStrack". This package contains executable files containing the Tcl/Tk distribution needed to run JStrack. Note that you still need to download JStrack itself (see next item).
   - The current JStrack distribution (available as either **.rar** or **.tar.gz**).
2) If you have installed Tcl/Tk from the ActiveTcl distribution, Install Tcl/Tk. Run the **ActiveTcl8.4.x exe** file that you downloaded above. Accept the license agreement (it's freeware) and install using the defaults. Otherwise, extract the EXE files package into the same directory you will use for JStrack itself.
3) Decide on a location where you want to install JStrack. For example, you might choose **c:/JStrack**. Assuming this is your choice, put the **.rar** (or **.tar.gz**) file in **c:/** and extract it. It will extract into a subdirectory called **JStrack** (with a number of subdirectories under that).

4) Proceed to JStrack Configuration (see the next section).

1.2 For Unix

You will need the following (all but netpbm available from the JStrack downloads page—filenames given are from the JStrack site):

- Tcl 8.4.x: **tcl8.4.15-src.tar.gz**
- Tk 8.4.x: **tk8.4.15-src.tar.gz**
- The standard Tcl/Tk library: **tcllib-1.3.tar.gz**
- The Img package: **img1.2.4.tar.gz**
- The netpbm (formerly pbmplus) package (pre-built packages available for many systems from **http://netpbm.sourceforge.net**

NOTE: If you are using FreeBSD, use the versions from the ports or packages for FreeBSD and save yourself some configuration, etc..... If you are using another Unix, check for available pre-packaged releases for your Unix...they are probably available on its ftp site.

1) Install Tcl. Refer to the installation instructions (typically README and/or INSTALL). Normally, configuration and installation is a simple matter of three commands (the last typically requiring root access)—note that the '$' and '#' are prompts:
   $ ./configure
   $ make
   (su to root)
   # make install
   It may not be that simple, however, for your system (but that would be very, very strange!).
2) Repeat instructions/comments from the previous step to install Tk.
3) Install Tcllib:
   - extract **tcllib-1.3.tar.gz**
   - Run **install.sh**
4) Decide on a location where you want to install JStrack. For example, you might choose $HOME/JStrack-3.0.1. Assuming this is your choice, put the .tar.gz file in $HOME and extract it:
   $ zcat jstrack-3.0.1.tar.gz | tar xvf -
   OR
   $ gzcat jstrack-3.0.1.tar.gz | tar xvf -
   OR
   $ gunzip -c -d jstrack-3.0.1.tar.gz | tar xvf -

It will extract into a subdirectory called JStrack-3.0.1 (with a number of subdirectories under that).

5) Proceed to JStrack Configuration (see the next section).

2. Installing JStrack

All releases of JStrack version 2.0.0 and beyond have a completely revised installation procedure, so read this first!

Important notes:

1) The structure of the configuration files has changed. The location where some variables is no longer the same (some moved to **paths.tcl**).
2) There is now a configuration script. It's not perfect...far from it, in fact, but it's a good start.
3) *Very Important*: If you are upgrading from a previous version (3.0.1 or beyond) and have edited the list of city names to display, you should make a backup of your city definition files before doing the update. The updates are, by default, written into a different directory (by version number), but if you override that, or move files around, without backing up your changes, you will lose them.

2.1 Installation Steps

First, cd into the JStrack directory. You will see several files and directories (e.g., **filt**, **tracker**, **autoconfig.tcl**, **HISTORY**, etc.). For Windows, you can use Windows Explorer (not to be confused with MS Internet Explorer) to get there.

2.1.1 autoconfig.tcl

Run **autoconfig.tcl**. Unix users and Windows users can both run it from a shell prompt, but the command for the Tcl (text-only) shell and the Tcl/Tk (graphical) shell will probably be different. These are the typical commands:

- Unix (text-only): **tclsh8.4**
- Windows (text-only): **tclsh84**
- Unix (graphics): **wish8.4**
- Windows (graphics): **wish84**

Note that the difference between the Unix and Windows version of each is the missing '.' from the Windows command ("84" instead of "8.4").

When you run **autoconfig.tcl** from a shell prompt, you'd run it as shown (replacing **command** with the appropriate command from above (for a text-only or graphical interface). As before, the "$" is your prompt:

$ **command** autoconfig.tcl

Windows users can also run **autoconfig.tcl** from Windows Explorer.

Double-click on **autoconfig.tcl**. If Windows asks you to select what program to run it with, select "Wish Application" and select the option to use this application every time for this type of file. If "Wish Application" doesn't show up, click on Browse and find **C:/Tcl/bin/wish84.exe** and select it. Again, select the option to run **.tcl** files with **wish84** every time.

You may also do the same as above by right-clocking on **autoconfig.tcl** and using "Open With..." and selecting "Wish".

2.1.2 Setting JStrack Options

If you're using the normal (graphical) interface for JStrack (**tracker/jstrack.tcl**), just run JStrack and check the various options to see if you want to change anything. For configuring the plotting (or not) of city names, see the instructions below.

If, on the other hand, you're using JStrack's text-only interface (**tracker/jstrack-text_only.tcl**), edit **sysconfig.tcl** and **options.tcl** (both under the **tracker** directory) just like you had to before. Note that the text-only interface has not been updated since, ummmmm, well, a long, long time ago. But then, there hasn't really been much to add to it, so.....

2.1.3 Preparing the HTTP Update Script

Change directories into the **httpfilt** directory and run the **initial_timestamps.tcl** script—this sets the timestamp files to the time of the most recent update (actually, the initial time is the time that I packaged up the distribution, but either way, it's a reference that indicates that anything on the NHC's web site that is newer than the its associated timestamp file is to be downloaded and processed).

2.1.4 Support for Windows Systems

I no longer have access to anything other than Mac OS X Lion (a Unix variant). My last Windows system finally ate itself. As a result, my ability to support Windows systems is now somewhat limited. As long as the problem is not the .exe files I created for Windows users who don't want to install Tcl/Tk, this should not be a problem. If those .exe files break, I'll need someone else to help with that.

2.2 A Few (Hopefully) Useful Comments

All satellite imagery features require the Img library for Tk. If you are using the ActiveTcl distribution of Tcl/Tk, you have all of the libs you need (except for the windows binaries for a couple of image sizing/conversion programs from pbmplus, which are in a windows binaries directory).

Unix users will need to make sure that the Img library, tcllib, and pbmplus are all installed and available to JStrack (that is, if you want the features that require those libs to work).

2.3 Configuring the Display of City/Location Names

This is a very new feature, and is very much in its infancy. However, I think it's helpful to have these, so I'm including what I've got. You can turn it off by commenting out everything in **config_user_citylist.tcl** (in the **tracker** directory).

If you want to display a given city/location name, find its entry (if there is one) in this file, and if it's commented out, uncomment it (remove the leading "# ").

If you want to display a city/location that is *not* already in this file, you will first need to determine its lat/long coordinates. Note that JStrack uses positive numbers only for both (e.g., for Pensacola, FL: 30.47 and 87.20). Then, you will need to edit the file, **config_citydefs.tcl**, in the **tracker** directory and add the location name and coordinates. The format for this file is explained at the top of the file. Then you'll need to edit **config_user_citylist.tcl** (also in the tracker directory, and also explained in comments at the top of the file) to add the location to those plotted.

To plot/refresh city names, or to remove them from the chart, use the appropriate options in the Options menu. Note that these changes are *not* saved to

your config file. If you want more permanent changes, make changes to the **config_user_citylist.tcl** file.

When you make changes to **config_user_citylist.tcl**, you do *not* have to exit and restart JStrack. Just reload the city names via the Options menu.

Plotted locations for city/location names *should* be correct (I got most of them from the NHC's breakpoints file, i.e., their coordinates). But this was a big file, requiring some reformatting to get it into JStrack, so the possibility is always there that I messed something up somewhere. If you find any such problems, please let me know so I can fix them.... Thanks.

2.4 Finally, a Note on Satellite Imagery and Saving Screen Images

If you have satellite imagery displayed on the tracking chart, you may or may not be able to use features within (or outside of) JStrack to save the screen image to an image file. It may work on one system and not another. I've only just discovered this problem with my setup here at home—on my laptop, running JStrack running under Windows XP *and* displayed remotely from my FreeBSD Unix system using X11 (and Cygwin). Saving images in some cases will work under Windows or directly on my FreeBSD machine (i.e., sitting at the console, not remotely displayed) that will not work when remotely displayed via X11. This is, as far as I can tell, a problem with the X server under Cygwin, and there's nothing I can do about that. Even xwd, run from the FreeBSD side and from the Windows+Cygwin side fails with the same error—it's not JStrack.

Note, however, that just last night...about 12 hours after releasing JStrack 3.0.1 with the above comments, I discovered that, on my XP laptop, running XFree86 (part of Cygwin) in one virtual desktop in JSpager—cool program, btw...I can't survive without multiple desktops on Unix (ctwm) or Windows (JSpager ... no releation to JStrack)—I *could* use **Shift** **Print Scrn** and then switch to another desktop and dump the full-screen image into Paint Shop, where I was able to crop, resize, etc., as needed.

On Windows, I've seen problems saving screen images from within JStrack if the satellite imagery is loaded on the tracking chart. I've gotten around this by using **Shift** **Print Scrn** to copy the entire display to the buffer, and then paste it into an imagery program like Paint Shop Pro, Adobe Photoshop, etc., and then crop/resize as needed.

3. Using JStrack

3.1 JStrack's Different Parts

(at least, those you care about as a user)

JStrack has many different parts. Normally, you'll only need to worry about the tracking part (and, for Windows users, the HTTP script for grabbing data from the NHC's HTTP site). But first, you need to get a few things set up, and knowing what's in the various directories in the distribution.

The first directory is the top-level directory (by default, **JStrack-3.0.0**). In this directory, you'll find (for now, at least) the older versions of the installation docs, the **HISTORY** file, etc., along with a couple of Tcl/Tk (**.tcl**) files and a few subdirectories.

By now, you've already seen and run **autoconfig.tcl**. What you may or may not know is that it made updates to another file in that directory called **paths.tcl** (as well as a number of other **.tcl** files that we'll get to in a bit). The **paths.tcl** file tells JStrack's programs where to find things they need.

There are three key parts to JStrack: the filter script, the httpfilt HTTP-based data input (and its associated utils), and the tracking program itself.

3.1.1 The "filt.tcl" Filter Script

In order to run, JStrack must (obviously) have data on the storms that are out there. This data comes from the National Hurricane Center/Tropical Prediction Center (NHC/TPC, or simply NHC). JStrack gets its data from the following NHC/TPC products:

- the "Marine Forecast" (aka "Forecast/Advisory")
- the "Public Advisory" (aka "Advisory")
- the "Discussion"
- the "Strike Probabilities" (if any)

The Marine Forecast is where JStrack gets its most critical data. Without this data, you're stuck with entering it by hand...and JStrack's manual data entry isn't pretty.

These products are "read" by the filter script (in the **filt** directory) called **filt.tcl** (which, in turn, uses another script in that same diretory). The data is extracted and formatted for JStrack. This filter gets its data from any of four sources:

- via direct, automatic data feed using the WX-ATLAN e-mail list
- via HTTP using **get_http.tcl** and the NHC's web server
- via cut/paste into filt/winfilt.tk
- by hand from within JStrack...definitely not the greatest....

The first, via e-mail, relies on the WX-ATLAN e-mail list, which pushes out copies of NHC/TPC products. Unix users (and possibly Windows users who use more recent versions of Cygwin than I have) then use an e-mail filter program, like procmail, to pipe a copy of the incoming NHC/TPC products to **filt.tcl**. This has the advantage of being completely automatic (you don't even have to be logged in)—when e-mail arrives, it is processed by procmail on its way to your incoming mail according to rules you define. This is, by far, the preferred method, and is how JStrack was originally designed to work. Unix users should use this method (if you don't have procmail, you'll need to install it).

The second method, grabbing data via HTTP, is provided for those (e.g., Windows users) who don't have procmail or anything that provides the same functionality, and thus lack the ability to use the e-mail option. This option uses the **get_http.tcl** script (located in the **httpfile** directory). This script connects to the NHC's web site and downloads all data newer than the timestamp files in the same directory (the files' modification times are used to indicate the most recently downloaded data for each type of product used).

The third method, grabbing the data via cut/paste (e.g., from a Windows-based e-mail program while reaidng NHC products via WX-ATLAN, from the NHC web page, or whatever). This was what I'd originally hacked together for Windows users to get their data inputs, before I wrote the FTP (now HTTP after the NHC switched) script.

In the **filt** directory, there is an old script called **winfilt.tk** (which I should really rename to **winfilt.tcl**). Just start it up (either by associating it with **wish84.exe**, renaming it to **winfilt.tcl** and running it, or by running it from the command line: **wish84 winfilt.tk**. Then, copy the full text of each of the four product types (or, at least, what's available) that JStrack uses.

3.1.2 The get_http.tcl HTTP-Based Data-Fetching Script

As mentioned in the previous section, the **get_http.tcl** script is one of the ways to get the NHC data that JStrack needs in order to function. If you're running JStrack under Windows, you may want to create a shortcut to your desktop (or other location) from **get_http.tcl** and **autoupdate.tcl**. The **autoupdate.tcl** script is a scheduler that runs the **get_http.tcl** script periodically, thus (hopefully) keeping your data up-to-date without having to do the updates manually. Whether or not you use this is entirely up to you.

Before you run the HTTP script for the first time after installing JStrack, you should edit (to set the date) and run (or, if the distribution is more recent than your last data update, re-run) the **initial_timestamps.tcl** script to tell the HTTP script how recent your data is (so it won't download it all over again). Note that this script is run using **tclsh84.exe**, not **wish84.exe**.

Note that the HTTP scripts rely on the National Hurricane Center's web server to be up and functioning. If it's down, or if it's too busy to accept (or properly handle) additional incoming connections, you may get cut off. If you get disconnected, all I can suggest is that you try again a few minutes later. There's nothing I can do about it, though.

3.2 Running JStrack

First, if you are running JStrack from a menu (or, for Windows users, an icon), you can probably skip this section and move on to 3c (Using JStrack), as the only thing you're likely to do is select a menu item (or click on an icon).

If you're running JStrack from the command line, there are two basic ways you can start it up. For the remainder of this section, we will assume that the '$' is your shell prompt, and the Tcl/Tk **wish** interpreter is **wish8.4** (for Windows users, it will be **wish84**—note "84" instead of "8.4"). We also assume that you have run **autoconfig.tcl**, and that, for Windows users, **.tcl** files have been associated with **wish84.exe** (Unix users don't have to worry about this, as the name and location of **wish8.4** is configured on the first line of each of the executable script files.

To start JStrack without any command line options, you may use either of the following (from the **JStrack-3.0.1/tracker** directory):

```
$ wish8.4 jstrack.tcl
    OR
```

```
$ ./jstrack.tcl
```

There is no significant difference between these two. From this point forward, we will use the latter notation...but you can always use the former. With that in mind, the command line usage is:

```
./jstrack.tcl [-map map_name] [storm 1 name] .... [storm n name]
```

The three maps you may choose from are:

- The "overall" map: map_name is "overall" or "full"
- The "East coast" map: map_name is "east"
- The "Gulf coast" map: map_name is "gulf".
- The "Caribbean" map: map_name is "carib".

Storm names, if used on the command line, are the name (or number, for unnamed tropical depressions) in lower case, e.g.,

```
$ ./jstrack.tcl danielle
$ ./jstrack.tcl danielle erin six seven
```

If you wish to load a map other than your default map right from startup (for example, I normally startup with the Gulf coast map...but if I'm looking at a storm that I know is not within the area that map covers, I would start right off with a more appropriate map):

```
$ ./jstrack.tcl -map overall lisa
$ ./jstrack.tcl -map east dianne
```

And as far as the command line goes, that's basically it, at least, that's it as of JStrack version 3.0.1...who knows what I'll add in future releases.....

3.3 Using JStrack

Once you've started JStrack, there are two ways you can control it. You can use the mouse/menu based commands, or you can use the console (which, depending on your OS, will vary in both method of accessing it and in its command line editing capabilities). The command line interface assumes a knowledge of JStrack's internals, so for now, we'll limit the discussion to the "normal" user interface.

3.3.1 The "File" Menu

- Console
- Saffir-Simpson Scale
- Save main window to image
- PostScript output
- Update Storm Data (From File)
- Merge Tropical Depression/Named Storm Data
- Exit

Availability of some of these options (the console, saving the main window to an image file, and PS output) will depend on your system.

The "Console" item is for Windows users, who are not left with a command line after running JStrack (Tcl/Tk under Windows does not drop the user into a command prompt, but instead provides the console window option). This item will only appear on Windows systems.

The next item, "Saffir-Simpson Scale" displays the Saffir-Simpson Scale, with descriptions of each strength (category) hurricane and its ranges of winds, atmospheric pressures, storm surge levels, and a text description of typical damage patterns for each.

The "Save main window to image" item uses the Img extension's capabilities to save the main JStrack window to an image file. After repeated failures when trying to create PNG, JPG, or GIF images (various errors, usually claiming "too many colors", etc.) I have changed this to only create TIFF images. To convert a TIFF to a PNG, etc., there are a number of options, including using the netpbm utilities which compile cleanly on Unix and Windows systems (and should be available for Windows on the Internet).

The "PostScript output" item uses the Tk canvas widget's PS output capability to produce a PS file from the canvas contents (i.e., the map and displayed track data). Your results may vary.

The "Update Storm Data (From File)" item is used to update the storm data from a text file containing one of the NHC products (Advisory, Marine Forecast, Discussion, and Probabilities, which the NHC does not appear to use anymore).

The "Merge Tropical Depression/Named Storm Data" item basically does what the **merge** shell script in the **storms** directory does: it takes the track data files for

a tropical depression (e.g., **one.trk**) and merges the data with that of the named storm that the tropical depression later evolved into (e.g., **alice.trk**). This avoids having both "TD ONE" and "TS ALICE" appear as if they were two different storms, when in fact they are just one.

Again, this is a duplicate of the functionality of the shell script (merge) in the **storms** directory. It makes no difference which you use.

The "Exit" item exits JStrack.

3.3.2 The "Options" Menu

The "Options" menu contains the following options settings:

- Plot City Names/Update Plotted Names
- Remove City Names
- Default Map
  - Overall picture
  - East coast (north of FL)
  - Gulf of Mexico, etc.
- Miles/MPH
- NM/KT
- Always plot latest wind radii
- Always plot latest forecast positions
- Watch for storm updates in background
- Automatically load NEW storms
- Home Coordinates
- Warnings (for when a storm gets too close)
- Display coordinates of mouse pointer
- Save Config (user-specific)
- Save Config (GLOBAL config)

Plotting and Removing City/Location Names

The first two plot and remove city/location names, respectively. Display (or not) of these names is controlled via two files (which, at present, are edited by hand). WARNING: Keep a backup copy of any changes you make to the city/location name/definition files. Extracting a new version will over-write any existing copy. Backups are good.....

The two files (both in the **tracker** directory) that control the display of city and location names are:

- **config_user_citylist.tcl** - select which names to display
- **config_citydefs.tcl** - defines coordinates for cities/locations

The format for these files is described in each file.

Selecting the Default Map

Thanks to Ron Murphy, NWS Birmingham, many versions ago, JStrack got a nice, new set of maps that drastically improved its appearance. As of way back then, there are three maps to choose from. This option selects the default map to load (assuming you don't override it from the command line) when JStrack starts up. The options are:

- Overall picture
- East coast (north of FL)
- Gulf of Mexico, etc.

Note that, if a storm's position is off of the current map, a big, fat arrow is displayed pointing (more or less) to its off-display location.

Miles/MPH vs NM/KT: This is a simple either/or selection of which units you want to use for distance and speed.

Yes/No Options:

- Always plot latest wind radii
- Always plot latest forecast positions
- Watch for storm updates in background
- Automatically load NEW storms
- Display coordinates of mouse pointer

Wind radii: The wind radii plot is important, and should not be turned off. Keep in mind that you need to look at more than just the center of the storm—you need to look at the extent of the winds. This is the purpose behind the wind radii. Wind radii information is provided by the NHC in the Forecast/Advisory (aka Marine Forecast).

Forecast positions: Forecast position information is provided by the NHC in the Forecast/Advisory.

Storm updates: Unless you turn this off (not recommended) JStrack will silently watch for updates to the storm track information (**.trk**) files. If new storm data arrives (e.g., via e-mail/procmail, autoupdate, etc.) for a storm that is currently displayed, JStrack will detect this and reload the storm's track.

Auto-load NEW storms: If a NEW storm (i.e., one for which there was no data in the storms directory) appears and this option is set, JStrack will automatically load the new storm's data.

Display coordinates of mouse pointer: If set, will display the coordinates (latitude and longitude) of the location that the mouse is currently pointing at.

Home Coordinates: If you look at the sample images of JStrack, you'll see a big 'X' over NW FL. That's where I live. This is where you enter your home coordinates so that 'X' is over your location instead of mine.

Warnings (for when a storm gets too close): This section allows you to define a set of external programs/commands to run to sound alarms, make lots of noise to wake you up if a storm is in a threatening position, and so on. This can also be used to automatically update a web page (e.g., for continuously displaying the latest plots).

The configuration window for this was (in/before 3.0.0) too big for smaller displays. As of 3.0.1, the code for this is totally revised. (For what it's worth, when I wrote the original, I didn't know any other way).

3.3.3 The "Maps" Menu

From this menu, you can change maps, overlay current GOES East satellite imagery, and select a version of the maps with a more printer-friendly color set—this doesn't look as nice, but it will save you from wasting a whole ink cartridge (well, ok, perhaps it's not QUITE that bad) on one printout.

3.3.4 The "Storm Data" Menu

The "Storm Data" menu contains the following options settings:

- Load/reload storm(s)
- Remove storm(s) from chart
- Input storm data manually
- History of storm
- Plot all wind radii for storm
- Remove all wind radii for storm
- Clean chart

Load/reload storm(s): From here, you get a list of storms available to load (or reload, e.g., if new data has arrived and you do not have JStrack configured to reload automatically). JStrack is designed to have storm data in the **storms** subdirectory under the **tracker** directory. Storm data for the current year is stored in **storms**.

Storm data for previous years is stored in **storms/old/YYYY** where "YYYY" is the four-digit number for the year. JStrack will sort the data from current to oldest (current data first, old data displayed by year *if* you keep the data arranged in the right format).

The dual-listbox selection works (as of 3.0.1...a typo in the 3.0.0 and earlier versions—exactly how far back, I'm not sure—prevented selected items from being de-selected) this way: available storm names are on the left. Selected storm names are on the right. Click on a name on the left to add it to the selected names on the right. If you decide not to load one of the ones you've added, click on its name in the right window.

Remove storm(s) from chart: This gives you a list of storms currently displayed and asks which ones you want to remove. The dual-listbox selection works as described above.

Input storm data manually: This provides an extremely crude interface for adding new storm data when no network option is available (e.g., you're at a shelter, using your laptop and getting updates via radio, shelter personnel, etc.). It's not pretty, but it works. Some of the entry fields have a '?' button next to them—use this if you aren't sure what format the data should be in for these fields. For example, the '?' for the timestamp will check the current time (using

your computer's time) and attempt to guess which NHC update it would most likely be.

All times are given in 24-hour time (so midnight would be 0000, 6 AM would be 0600, noon would be 1200, 6 PM would be 1800, and so on).

Coordinates are entered using decimal degrees (positive numbers only).

Course direction is entered in degrees, with North being 0 degrees and South being 180 degrees. A '?' is provided if you need it.

History of storm: After selecting a storm (as in examples above), you are shown a text display of the storm's (data) history.

Plot all wind radii for storm: Normally, only the current and forecast wind radii are plotted. This lets you plot *all* of the wind radii, past, present, and future.

"Remove all wind radii for storm" removes the wind radii.

Clean chart: Removes all storms from the map.

3.3.5 Update Storm Data from NHC/TPC Web Server

This is a simple command button which runs the **get_http.tcl** update script to grab new storm data (if there is any new storm data available when you run it) from the National Hurricane Center's web server.

3.3.6 The "Satellite Imagery" Menu

From this menu, you can select GOES East satellite imagery to view in a window, as opposed to the overlay on the tracking chart. From this menu you can also view the satellite imagery loops, showing the recent progression of the storm as seen from space.

3.4 Position-Based Menus & Displayed Data (mouse/location based)

With your mouse, you can get additional data and options directly from the track display:

- coordinates of the location the mouse is at. This is turned on/off via the options menu.
- mouse-overs: move the mouse over displayed data to get a description of what it is
- left-click on current/previous storm plot point: plot point menu from which you can get:
  - Detailed Info - storm data listing for that update
  - NHC Advisory Data
  - Plot Forecast Positions
  - Plot Wind/Seas Radii
  - Plot Forecast Radii
  - Plot Probabilities (if the data is available)
  - Storm History (as described above)
  - Remove From Map (removes the storm)
- right-click on map area: select from plot points to load menu (selects a region from the screen...if the most recent data point is within that region, it will be identified as such.) This is very useful in cases where a storm is moving very slowly (or not at all) and you want to select the latest plot point out of a jumble of overlapping data. Just right-click, select the item identified as current, and click on the "Get Selected Data" button (bottom-right).

Appendix A: Getting WX-ATLAN

WX-ATLAN is a mailing list that sends NHC/TPC products (which JStrack uses for its data input) via an e-mail server called a listserv. You can subscribe to it by sending e-mail to listserv@po.uiuc.edu, with the following line in the body of the e-mail:

sub wx-atlan [your full name]

The listserv will confirm that you really want to be subscribed to this list, and will give you instructions to follow to complete the process. Once the listserv confirms that it has subscribed you to the list, you can choose from various topics (by default, you only get the updates, which are a one-paragraph or so note listing the storms, if any, that the NHC is tracking). I would suggest setting this to ALL, and then trimming it down later. To set the topics to ALL, send another e-mail to listserv@po.uiuc.edu, with the following line in the body of the e-mail:

set wx-atlan topics: all

At a minimum, for JStrack (assuming you're filtering incoming e-mail via something like procmail, and passing NHC/TPC products to filt.tcl), you should set the following:

set wx-atlan topics: outlook,forecast,advisory,strmdisc,strike

The outlook topic is not used by JStrack, but it's good for keeping up with what the NHC is watching, even if it's not named yet.

You can also go to the WX-ATLAN Site on the UIUC listserv for archives, to join/leave the list, or to change your settings for the list.

Appendix B: Configuring procmail For JStrack

In order to actually use the WX-ATLAN stuff automatically, you need an e-mail filter program that will recognize the NHC/TPC products and pipe a copy to the standard input of JStrack's filter program (**filt/filt.tcl**). The freeware program, procmail is ideal for this purpose (as well as filtering of other e-mail).

Here's how you set it up:

First, download the source for procmail from www.procmail.org. Configure and compile procmail following their instructions for the Unix variant that you're running (Or, if your system has a pre-configured package, install it.)

Assuming you're root, it's a good idea to configure procmail to serve as your MDA (Mail Delivery Agent). Look in the procmail docs for how to do this with the MTA (Mail Transport Agent) you're using (e.g., sendmail, postfix, smail, etc.). (This is not required for JStrack, but it is a good idea.)

If you're not root, you'll need to use procmail from your **$HOME/.forward** file (again, look in the procmail docs—they tell you how to do this).

Then, create a file called **$HOME/.procmailrc** from the procmail examples (if nothing else, "man procmailex"), and include the following (note that the directories given are for my machine—edit as needed):

```
# the 'w' says keep the lockfile until the filter finishes!
:0 cw:
* ^To.*WX-ATLAN
* ^Subject.*(ADVISORY|FORECAST|STRMDISC|STRIKE):
|/home/jim/src/ht/filt/filt.tcl
```

After that, as advisories arrive via e-mail (from WX-ATLAN), they're automatically processed by JStrack's filter program (whether the tracking portion of the code is running or not, and, obviously, regardless of whether or not you're logged on).