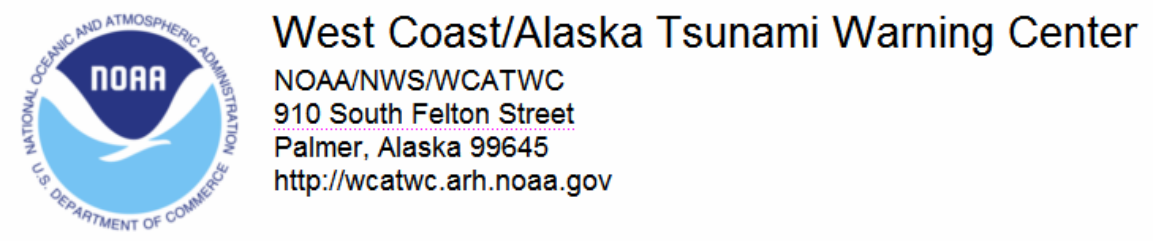
EarlyBird

User Guide

Acknowledgement

EarlyBird seismic data processing modules develo, pick_eb (pick_wcatwc), hypo_display, loc_eb (loc_wcatwc), lpproc, mm, mtinver, disk_eb (disk_wcatwc), analyze, locate and summary were developed by West Coast Alaska Tsunami Warning Center (WC/ATWC). As such, the module and algorithm descriptions contained in Sections 1.1, 3.1, 3.3 and Appendices A, B and C have been taken directly from the WCATWC Operations Manual (http://wcatwc.arh.noaa.gov/DataProcessing/ew-eb.htm).



Part number 16236R3
Release date 2007-11-15

Contents

Chapter 1: Getting Started

1.1 EarlyBird Overview

EarlyBird seismic data processing system was developed by **West Coast Alaska Tsunami Warning Center (WCATWC)**. It is used for both real-time and post-processing of seismic data. EarlyBird is a combination of standard Earthworm modules (Johnson, et al., 1995), WC/ATWC-developed earthworm modules and stand-alone seismic processing software (see Figures 1 and 2).

Several iterations of processing software have been developed by the WC/ATWC (Sokolowski, et al., (1983); Sokolowski, et al., (1990); and Zitek, et al., (1990); Whitmore and Sokolowski, (2002)). EarlyBird automatically locates and sizes (using Mb, Ml, MS, Mw, and Mwp) worldwide, regional, and local earthquakes. Graphical interfaces for the earthworm modules have been created to allow interactive additions and changes to automatically computed parameters during initial earthquake processing or after-the-fact. Real-time data can be monitored and interacted with directly through earthworm modules. Data logged to disk by the system can be analyzed immediately after logging through stand-alone analysis programs.

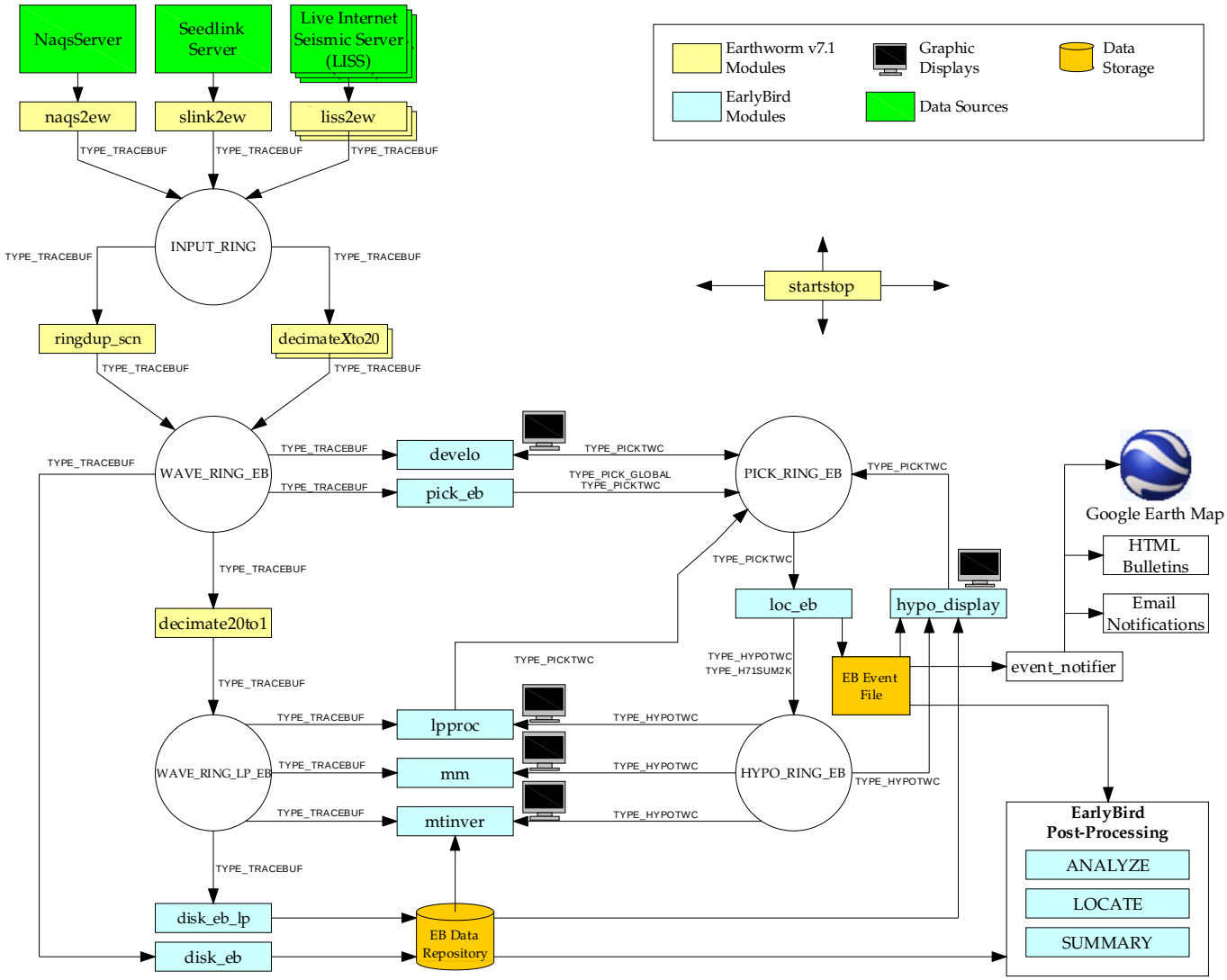


Figure 1: Earlybird Data Processing Modules

Figure 1 shows the data processing flow within EarlyBird, and lists all data processing modules that are included with the default installation. Also shown in Figure 1 is the event_notifier module. This module (written as a Perl script) runs independently of the Earthworm-based modules and monitors EarlyBird output files for events. The script then plots event and station locations in Google Earth, generates html bulletins and sends event notification emails.

All modules are started by startstop module. This module can also be used to stop, start, add or remove individual modules from the system. Earthworm rings are shared memory locations. The windows icons indicate modules which accommodate user interaction and review through graphical user interfaces. Trace data from different sources (Seedlink Server – slink2ew, NaqsServer – naqs2ew, Live Internet Seismic Server – liss2ew) are first placed in the INPUT_RING. Here, the information is either decimated or copied as is into the WAVE_RING_EB. Disk_eb (disk_wcatwc) logs all trace data to disk. This data is immediately available for review by the stand-alone program Analyze, and modules Hypo_display, Lpproc, Mm, and Mtinver. Typically two weeks worth of data is saved on disk (configurable). Large quakes are archived through Analyze. Module Pick_eb (pick_wcatwc) analyzes the signal to determine the onset of an earthquake. Once a pick has been made, the signal is further analyzed to determine Mb, Ml, and Mwp magnitude parameters. Module Develo displays the real-time signal in a similar fashion as an old-fashioned develocorder. Develo displays P-picks made in Pick_eb (pick_wcatwc) and allows the user to add or refine pick data to earthquakes presently being processed. Develo also analyzes the signal to look for strong earthquakes. Alarms are triggered when parameters have been exceeded. The P-picks made in Pick_eb (pick_wcatwc) are sent to the PICK_RING_EB. Module Loc_eb (loc_wcatwc) ingests these P-picks and automatically locates the events. Alarms can be triggered in Loc_eb (loc_wcatwc) based on location and size. The earthquake locations and magnitude information is sent to HYPO_RING_EB. Module Hypo_display summarizes this information and displays a GUI which allows a user to modify P data (which is re-sent to Loc_eb (loc_wcatwc) for refinement).

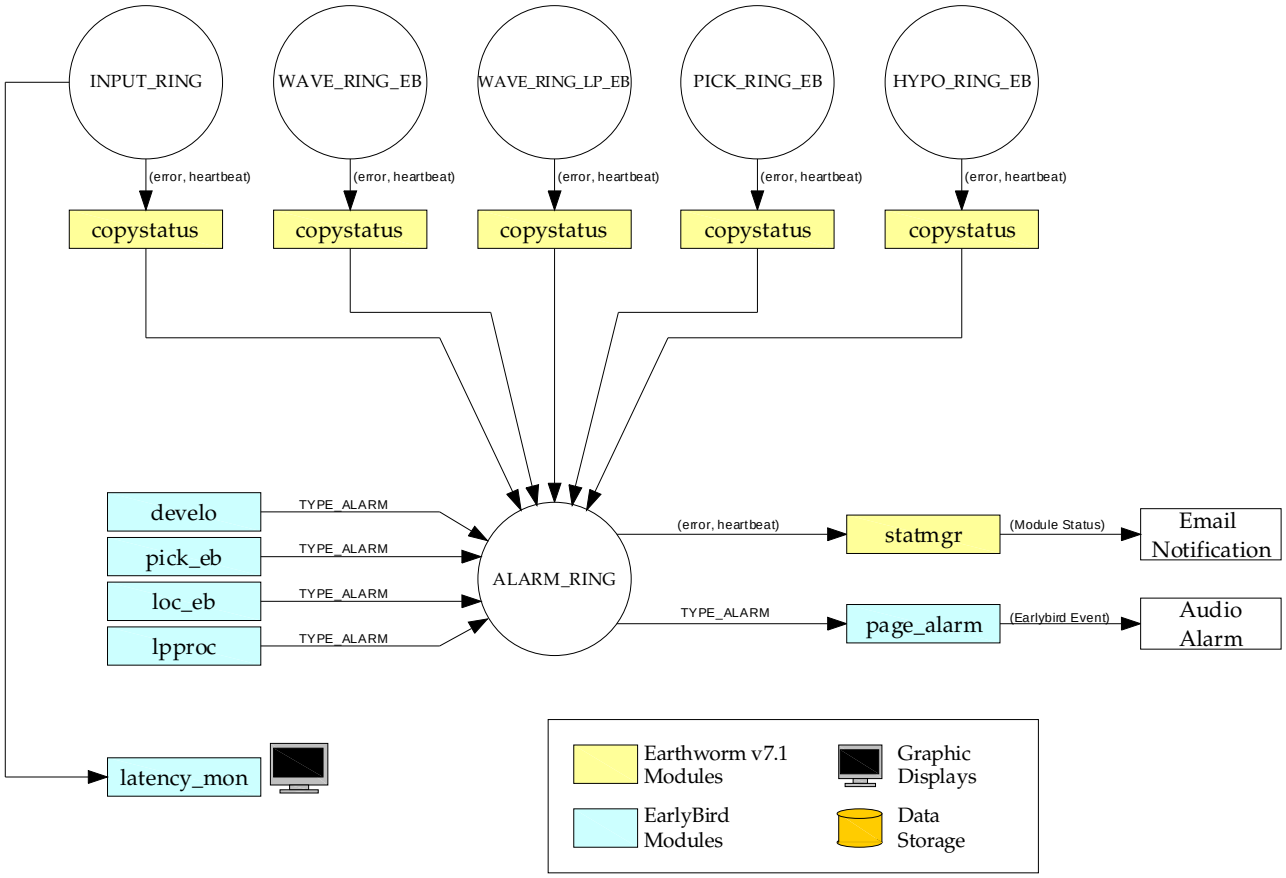


Figure 2: Alarm and Error Reporting Modules

When a large earthquake occurs (M>5), long period and broadband data are processed to refine the magnitude estimate. Broadband data in WAVE_RING_EB is decimated to 1 sample/second and placed in WAVE_RING_EB_LP. Modules Lpproc, Mm, and Mtinver operate on these data to determine Ms, Mw, moment tensor, and fault plane solutions. Each of these modules is described in greater detail in Chapter 3.0.

Figure 2 displays the data flow within alarm and error reporting modules that run in parallel with data processing modules shown in Figure 1. Page_alarm module generates sound alarms based on alarm messages generated by loc_eb (loc_wcatwc), develo, pick_eb (pick_wcatwc) and lpproc modules. Copystatus modules copy heartbeat messages generated by all modules that are part of the Earthworm-based system to HYPO_RING_EB wave ring. These heartbeats are then monitored by Statmgr module which notifies startstop module to restart modules that may not be running any more (watchdog service).

1.2 Core Earthworm Overview

The EarlyBird package includes an option to run a set of Earthworm modules (referred to as CoreEarthworm) in parallel with EarlyBird earthworm modules. These CoreEarthworm modules constitute a second real-time processing system, independent of EarlyBird. CoreEarthworm is used for automatic detection of local events and writes results to an Oracle database. Those events may then be analyzed using the Nanometrics program Atlas. Seismic data is acquired from different sources using naqs2ew, liss2ew and slink2ew common modules and placed in INPUT_RING. From here, data processing follows two independent paths: decimate and pick_eb (pick_wcatwc) modules kick off processing by EarlyBird system (Figure 1) and the pick_ew module kicks off processing by the CoreEarthworm system (Figure 3).

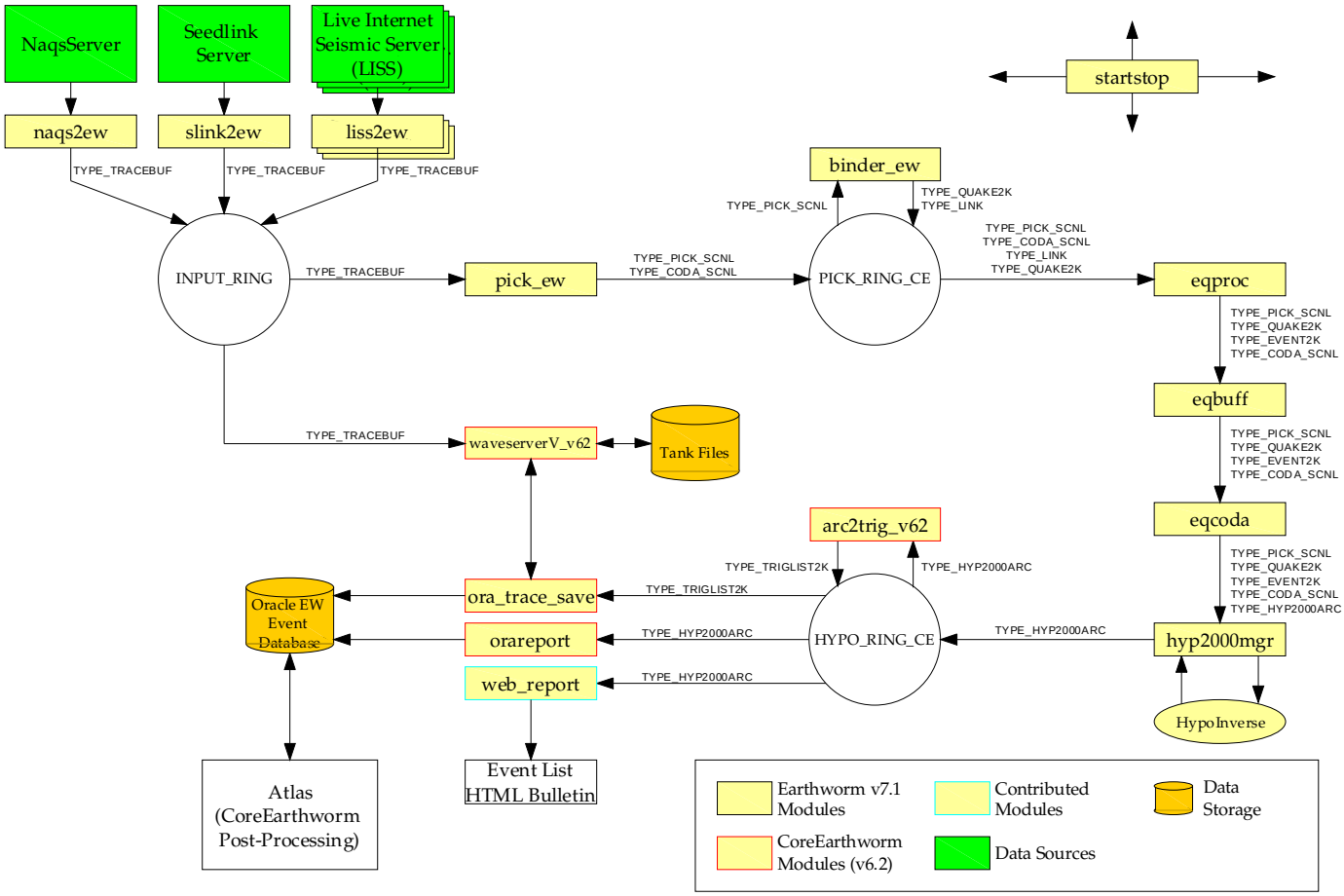


Figure 3: CoreEarthworm Data Processing Modules

CoreEarthworm and EarlyBird systems can be started together or individually, as described in Chapter 5.

As Figure 3 demonstrates, CoreEarthworm supplied as part of the EarlyBird installation is a combination of v7.1 and v6.2 Earthworm modules. The modules that interact with Oracle database (ora_trace_save and ora_report) are part of the Interactive Earthworm package which has not been supported since version 6.2. As ora_trace_save and ora_report are not compatible with version 7.1 of the modules arc2trig and wave_serverV, version 6.2 of these two modules are also included. In all cases where there are multiple versions of the same module present, the executables and configuration files have clearly been marked to distinguish between v6.2 and v7.1 (ex. arc2trig_v62.exe and wave_serverV_v62.d vs arc2trig_v71.exe and wave_serverV_62.d).

The operation and configuration of CoreEarthworm modules is described in more detail in **CoreEarthworm_UserGuide_14639R5.pdf**, which is included in the folder \Earthworm\docs\. Note that this manual corresponds to the CoreEarthworm package assembled using only Earthworm version 6.2 modules and some of the content is no longer applicable.

IMPORTANT NOTE: Prior to starting CoreEarthworm for the first time, it is crucial to pre-populate Oracle database with station coordinate and response information. This can be done in the following way:

1. Install Oracle Server (or Oracle Client if the server already exists) and EW database schema (if not already installed) as per \Earthworm\docs\Atlas_InstallMaint_14390R5.pdf
2. Install Atlas as per \Earthworm\docs\Atlas_InstallMaint_14390R5.pdf
3. Copy UserStations.hinv (assuming that it has been configured with all applicable stations) from \Earthworm\run\params\hypoinverse to \nmx\atlas\user\hypoinverse.
4. Start Atlas and load a segment of data from all stations whose data will be analyzed by CoreEarthworm.
5. Put at least one pick (P, S, Amplitude, Duration) on the vertical trace of each station, and solve for an event (doesn't have to be a real event).
6. Save the event traces and solution to the database. Station coordinate and response information will be saved permanently.
7. Start CoreEarthworm.

This procedure will have to be repeated whenever a new station is added to the system. A new event which includes picks from the newly added station will have to be saved into the database before that station's data gets processed by Earthworm for the first time.

1.3 Installation Requirements

1.3.1 Hardware

Any computer that is sufficient to run NaqsServer, DataServer and Atlas is sufficient to run Earlybird, with these minimum requirements:

- Windows 2000/XP
- 2 Gb of free disk space (depends on the # of stations that are to be processed)
- 1Gb of RAM
- Internet access

1.3.2 Software

Before running EarlyBird, the following software should be installed:

- Active Perl (version 5.8.7.815 or higher – see Section 2.2 for installation instructions)
- Google Earth (version 4.0.2737 or higher – see Section 2.3 for installation instructions)
- Oracle Server (version 9.2.0.1.0 – see **/Earthworm/docs/Atlas_InstallMaint_14390R5.pdf** for installation instructions)
- Oracle Client if Oracle Server is installed on another machine on the network (version 9.2.0.1.0 – see **/Earthworm/docs/Atlas_InstallMaint_14390R5.pdf** for installation instructions)

Note: Oracle Server and Oracle Client are applicable only if CoreEarthworm system is to be utilized.
Note: Prior to running EarlyBird or CoreEarthworm, ensure that the system clock is synchronized with UTC and that UTS time is used instead of local time.

Chapter 2: Installation

2.1 Installing EarlyBird

1. Copy Earthworm directory with all its subdirectories from the EarlyBird CD to the root drive on the destination machine. If the assigned drive letter for the destination drive is something other than c:, edit the \Earthworm**EB.bat**, **CE.bat**, **EB_and_CE.bat**, **EBAnalysis.bat** and **EB_EN.bat** files and change c: to the appropriate drive letter.
2. Alternatively, if installing from a zip file, run the **EarlyBird_x.x.exe** self extracting zip file in the root directory. Directory /Earthworm as well as all subdirectories will be created automatically.
3. Add c:\Earthworm to the Path Environment Variable (in My Computer -> Properties -> Advanced -> Environment Variables -> Path).
4. Configure all modules for the desired network of seismic stations as outlined in Section 4.0, prior to running the software for the first time.

2.2 Installing Active Perl

Active Perl is required to run Event_notifier.pl script, which maps located events and station locations in Google Earth and sends event notification emails.

1. Install Active Perl by double-clicking on **C:\Earthworm\event_notifier\ActivePerl-5.8.7.815-MSWin32-x86-211909.msi** and following the installation instructions.

2.3 Downloading and Installing Google Earth

Google Earth is used by Event_notifier.pl script to map located events and stations.

1. Download the latest version of Google Earth from http://earth.google.com/download-earth.html and install it by double-clicking on the downloaded file and following installation instructions.

2.4 EarlyBird Directory and File Structure

Figure 4 displays the EarlyBird installation directory structure. All software is installed in the root directory /Earthworm. The root directory contains five batch files that start up all EarlyBird-associated software: **EB.bat**, **CE.bat**, **EB_and_CE.bat**, **EBAnalysis and EB_EN.bat** (see Chapter 5: Operation for more details).

All data stored in WCATWC format is stored in the **/Earthworm/EarlyBirdData/ContinuousData** directory. Within this directory, each day, a new subdirectory is created containing data files for that particular day. There are two types of data files in each subdirectory: *.S07 and *.L07. S07 files refer to short period data saved at 20 samples per second and processed by pick_eb (pick_wcatwc) and develo modules. L07 files refer to long period data decimated to 1 sample per second and processed by mtinver, lpproc and mm modules. The length of each data file is configured in the disk_eb.d and disk_eb_lp.d configuration files. In addition to WCATWC format, data may be stored in Tank files by WaveServerV module in **/Earthworm/TankFiles** (by default, this is not included unless running CoreEarthworm).

The directory **/Earthworm/EarlyBirdData/Archive** contains WCATWC data files archived by Analyze. The disk_eb (disk_wcatwc) module has a thread that deletes data files older than a

configurable amount of time from the ContinuousData directory. As such, to preserve event data Analyze includes a mechanism for transferring it to the Archive directory so that it will not be removed. Data files that are applicable to an event that is being saved, are simply copied from ContinuousData directory to the Archive directory by Analyze.

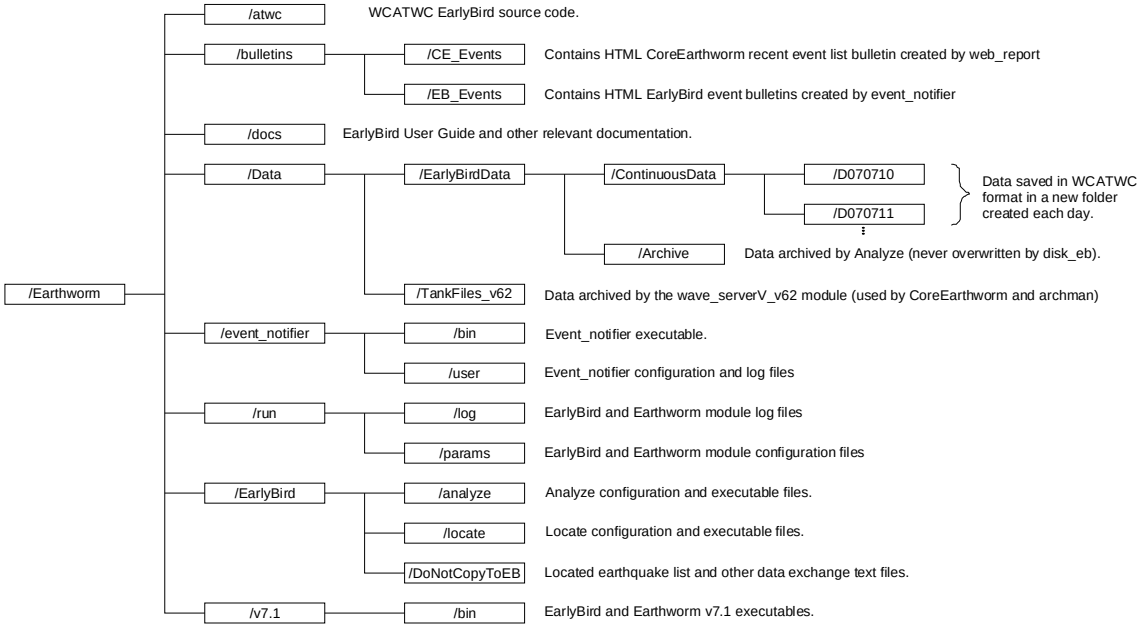


Figure 4: EarlyBird Directory Structure

2.4.1 Important File Locations

1. Directory: **/Earthworm**
   Files: **EB.bat, CE.bat, EB_and_CE.bat, EB_Analysis.bat and EB_NE.bat.**
   Purpose: Batch files used to start up all EarlyBird and CoreEarthworm modules.

2. Directory: **/Earthworm/run/params**
   Files: **alarm.dat, Calibs, decimate_XtoY.d, develo.d, disk_eb.d, disk_eb_lp.d, hypo_display.d, liss2ew.d, lpproc.d, loc_eb.d, mm.d, mtinver.d, naqs2ew.d, page_alarm.d, pick_eb.d, pick_eb.sta, pick_eb_lp.sta, screendisp.ini, slink2ew.d, startstop_nt.d, station.dat** and **statmgr.d**
   Purpose: Configuration files for all Earthworm-based EarlyBird (near) real-time data processing modules.

3. Directory: **/Earthworm/run/log**
   Files: **modulename_date.log**
   Purpose: Daily log files for all Earthworm-based and stand-alone EarlyBird modules.

4. Directory: **/Earthworm/event_notifier/bin**
   Files: **event_notifier.pl**
   Purpose: Event_notifier script executable.

5. Directory: **/Earthworm/event_notifier/user**
   Files: **event_notifier.ini, event_notifier.log**

Purpose: Event_notifier configuration and log files.

6. Directory: **/Earthworm/Earlybird /analyze**
   Files: **analyze.ini, screen.ini** and **analyze.exe**
   Purpose: Analyze data analysis program configuration and executable files.

7. Directory: **/Earthworm/EarlyBird/locate**
   Files: **locate.ini** and **locate.exe**
   Purpose: Locate data analysis program configuration and executable files.

8. Directory: **/Earthworm/EarlyBird/message**
   Files: **summary.ini** and **summary.exe**
   Purpose: Summary earthquake parameter display program configuration and executable files.

9. Directory: **/Earthworm/EarlyBird/DoNotCopyToEB**
   Files: **dummyX.dat** and **oldquakex.dat**
   Purpose: DummyX file contains hypocenter parameters of the last located earthquake and oldquakex.dat file contains a list of hypocenter parameters for the last 40 located events. These files are used by many EarlyBird programs.

10. Directory: **/Earthworm/v7.1/bin**
    Files: **naqs2ew.exe, slink2ew.exe, liss2ew.exe, decimate.exe, disk_eb.exe, pick_eb.exe, mm.exe, lpproc.exe, mtinver.exe, statmgr.exe, startstop.exe, develo.exe, loc_eb.exe** and **hypo_display.exe.**
    Purpose: Executable files for all Earthworm-based EarlyBird modules.

2.5 Upgrading EarlyBird

Upgrading EarlyBird involves some straightforward actions that must be performed with care. The following procedure may be used to upgrade from EarlyBird 2.0 assuming no significant customizations have been implemented and no tuning of EarlyBird or CoreEarthworm processing has been performed.

1. Stop to the currently running EarlyBird.
2. Rename the c:\Earthworm directory to c:\Earthworm_v2.0. This may require temporarily disabling the cleandir service if installed.
3. Run the EarlyBird 3.0 self-extracting zip file in the root directory to create the c:\Earthworm directory structure.
4. Directly copy the following files from the previous installation (c:\Earthworm_v2.0) to the new (c:\Earthworm). These are the configuration files that contain parameters that are network or installation specific (ex. station and channel information, waveserver IP address, Oracle database identifier, etc.).

Source	Destination	
c:\Earthworm_v2.0\Earlybird\analyze\screen.ini	c:\Earthworm\Earlybird\analyze\screen.ini	EB
c:\Earthworm_v2.0\event_notifier\user\event_notifier.ini	c:\Earthworm\event_notifier\user\event_notifier.ini	EB
c:\Earthworm_v2.0\run\params\alarm.dat	c:\Earthworm\run\params\alarm.dat	EB
c:\Earthworm_v2.0\run\params\Calibs	c:\Earthworm\run\params\Calibs	EB
c:\Earthworm_v2.0\run\params\decimate*to20.d	c:\Earthworm\run\params\decimate*to20.d	EB

C:\Earthworm_v2.0\run\params\decimate20to1.d	c:\Earthworm\run\params\decimate20to1.d	EB
c:\Earthworm_v2.0\run\params\mm.d	c:\Earthworm\run\params\mm.d	EB
c:\Earthworm_v2.0\run\params\ora_trace_save.d	c:\Earthworm\run\params\ora_trace_save.d	CE
c:\Earthworm_v2.0\run\params\orareport.d	c:\Earthworm\run\params\orareport.d	CE
c:\Earthworm_v2.0\run\params\pick_eb.sta	c:\Earthworm\run\params\pick_eb.sta	EB
c:\Earthworm_v2.0\run\params\pick_eb_lp.sta	c:\Earthworm\run\params\pick_eb_lp.sta	EB
c:\Earthworm_v2.0\run\params\pick_ew.sta	c:\Earthworm\run\params\pick_ew.sta	CE
c:\Earthworm_v2.0\run\params\ringdup_scn.d	c:\Earthworm\run\params\ringdup_scn.d	EB
c:\Earthworm_v2.0\run\params\screendisplay.ini	c:\Earthworm\run\params\screendisplay.ini	EB
c:\Earthworm_v2.0\run\params\station.dat	c:\Earthworm\run\params\station.dat	EB
c:\Earthworm_v2.0\run\params\wave_serverV_v62.d	c:\Earthworm_v2.0\run\params\wave_serverV_v62.d	CE

5. Set-up the data acquisition modules. This may involve directly copying over naqs2ew.d, slink2ew.d and any liss2ew_*.d / liss2ew_*.desc files. Verify that the startstop_nt_*.d and statmgr_*.d list all appropriate data acquisition modules and the any necessary Liss MODULE types are listed in earthworm.d.
6. If any of the following EarlyBird tuning configuration files have been modified from the 2.0 distribution, the modified parameters will have to be updated manually in the new installation as the format of these files has changed. In most cases, a few parameters have been added or removed and the overall structure remains the same.

c:\Earthworm\run\params\hypo_display.d
c:\Earthworm\run\params\loc_eb.d
c:\Earthworm\run\params\lpproc.d
c:\Earthworm\run\params\mtinver.d
c:\Earthworm\run\params\pick_eb.d

7. All CoreEarthworm tuning configuration files, including the crustal model, that have been customized may be copied directly to the new installation.
8. Move (or copy, disk space permitting) the data files from the original installation (c:\Earthworm_v2.0\Data*) to the new (c:\Earthworm\Data*).
9. Move (or copy, disk space permitting) the log files from the original installation (c:\Earthworm_v2.0\run\log*) to the new (c:\Earthworm\run\log*).
10. Copy the directory c:\Earthworm_v2.0\EarlyBird\DoNotCopyToEB from the original installation to the new c:\Earthworm\ EarlyBird\DoNotCopyToEB. Remove all contents from the directory c:\Earthworm\ EarlyBird\DoNotCopyToEB\LocFiles.
11. Restart EarlyBird (or CoreEarthworm or both).
12. Enable the cleandir service is disabled previously.

Chapter 3: System Modules

3.1 EarlyBird Real-time Data Processing Modules

The modules described in Sections 3.1.1 through 3.1.9 have been developed at the WCATWC and are available as part of the Earthworm Contributed software package. The explanations of the module functionality and controls have been taken from the Operations Manual posted on the WCATWC web page: http://wcatwc.arh.noaa.gov/DataProcessing/ew-eb.htm .

3.1.1 Develo

Date Written: 4/2001
Development Language: MicroSoft C/C++ v7.0 using WinAPI

Necessary Ring:
Input Ring - WAVE_RING_EB; reads TYPE_TRACEBUF format messages
Output Ring - PICK_RING_EB; writes (and reads) TYPE_PICKTWC format messages
Output Ring - ALARM_RING; writes TYPE_ALARM format messages

Necessary Data Files:
StaFile - Picker configuration file
StaDataFile - Seismometer location, elevation, etc.
ScreenFile- Sub-screen control file
DummyFile - Hypo-parameters of last located quake

Initialization File:
develo.d - read on module start-up

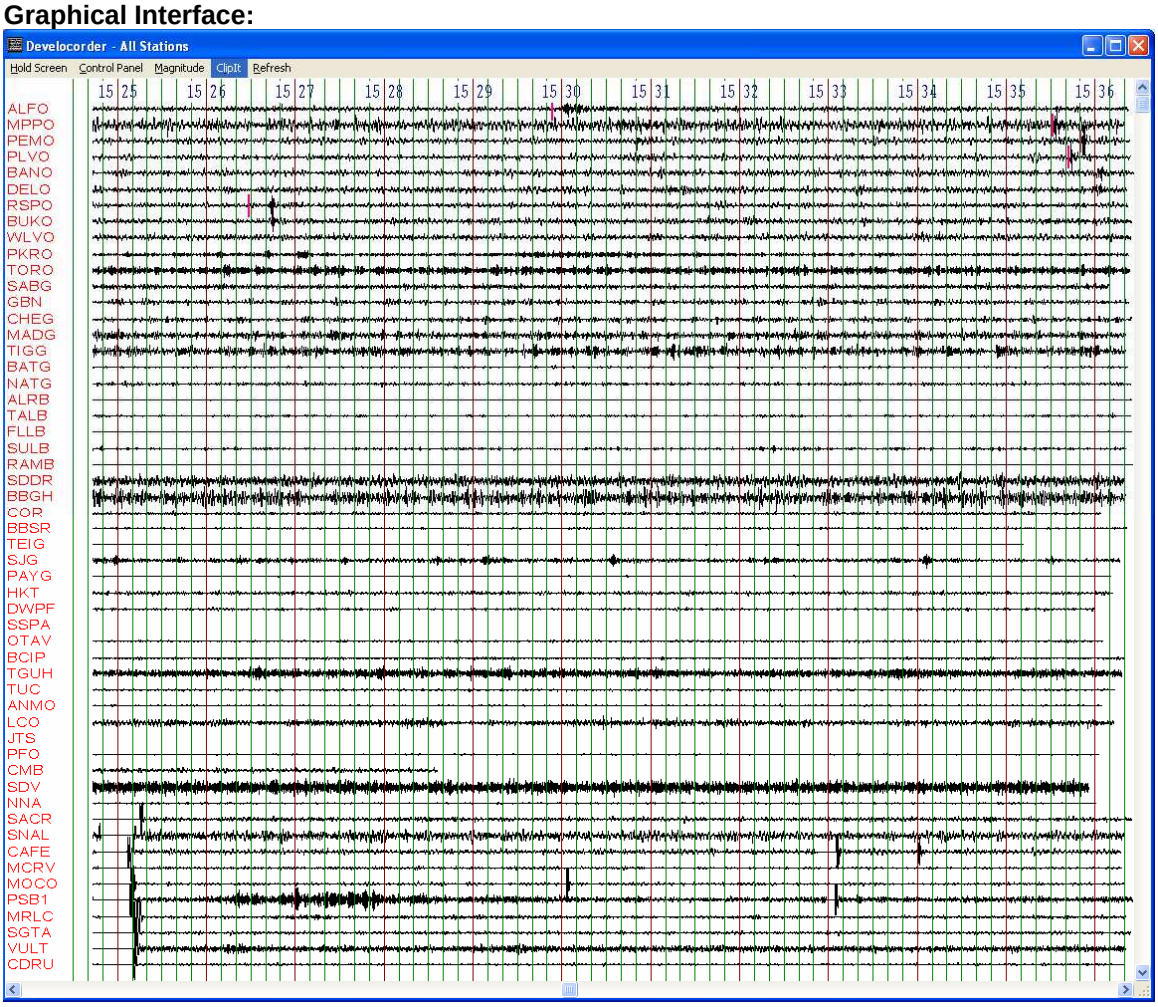
Module Description:
This program takes waveforms from the Input Ring and displays them on a Windows window. The display is similar to an old-style develocorder (with no film delay). Every X seconds the screen scrolls to the left. The most recent data is shown on the right side of the screen. The display can be "held" and manually scrolled through previous times (the time limit of saved data is specified in the .d file).

P-picks entered to a PICK_RING_EB are obtained by this module. The picks are displayed on the scrolling traces. P-picks can also be entered to a pick ring interactively through this module. P-picks are made by left-clicking on a trace. Magnitude information (Mb, Ml, Mwp) are also computed as the pick is made. The incoming signal to display is specified in the pick_eb.sta used by pick_eb (pick_wcatwc). Screendisp.ini can be used by this module. This INI file specifies subsets of stations which are shown on the screen.

This module provides filters which operate on broadband data when specified in pick_eb.sta. Data is short period filtered for display and for magnitude computations (Mb, Ml). Mwp integrations utilize the unfiltered, broadband data only.

As data is displayed, it can also be processed by an alarm function (similar to that in pick_eb). If the signal exceeds some pre-determined thresholds, an alarm is declared and the ALARM_RING notified. The advantage of using the alarm function here instead of in pick_eb (pick_wcatwc) is that alarms can be interactively turned on/off on individual stations through the GUI.

Graphical Interface:



Program Controls:

This module has a Windows graphical interface. Due to this interface, the module will only compile and run under Windows (NT, 2K, or XP). The display shows real-time, short period or broadband data. The most recent time is shown on the right side of the screen. As new data arrives, the screen scrolls left. A "Hold Screen" menu option is available to stop the automatic scrolling. Then, the horizontal scroll bars can be used to manually scroll through all the data which is in memory. P-picks are displayed in red as they are read from the PICK_RING_EB.

Also, green indicators are displayed showing where the P will be expected at each of the sites for the last computed location.

- Mouse buttons:
  - Right button on trace toggles trace display.
  - Right button on station name calls station manipulation menu (for trace scaling, station description, trace display on/off, alarm on/off).
  - Left button on trace sets P-time and Mb, Ml, and Mwp magnitude information are determined when station sensitivity is known.
  - Left button on station name eliminates the station from further processing in loc_eb (loc_wcatwc).

- Menu Options:

- Hold Screen: Prevent automatic scrolling so that scroll bars can be used to move through data.
- Control Panel:
  - Horizontal Scale: Change screen time scale.
  - Vertical Trace Scale - All: Change vertical scale for all stations.
  - Traces Per Screen: Number of traces vertically to show on the screen, rest will scroll.
  - Show DC Offset: Display DC offset in counts to right of Station Name.
  - Show Channel: Display channel name to right of Station Name.
  - Show Network: Display Network ID to right of Station Name.
  - Pick Stations:
  - All Stations
  - User-defined Station Groupings …
  - Display Filtered Data: Check -> Screen shows SP/LP filtered signal for broadband stations.
  - Display Broadband Data: Check -> Screen shows unfiltered traces for all stations.
- Magnitude:
  - Mwp: Set Max Window Time: Set the maximum integration time and S:N manually for broadband stations (used with original Mwp method).
    Display Integrations: Show the integrated waveforms for all stations which could compute an Mwp
- ClipIt: Limit trace amplitudes.
- Refresh: Re-paint the screen.

3.1.2 Disk_eb (Disk_wcatwc)

Date Written: 1/2001
Development Language: MicroSoft C/C++ v7.0

Necessary Rings:
Input Ring - WAVE_RING_EB (or WAVE_RING_EB_LP); reads TYPE_TRACEBUF format messages

Necessary Data Files:
StaFile - picker configuration file
StaDataFile - seismometer location, elevation, etc.
Disk file path and suffix must also be provided for output files.

Initialization File:
disk_eb.d - read on module start-up

Module Description:
This module writes raw trace data to disk files. All data (from the stations listed in pick_eb.sta) will be written to disk. The data is written in a format used by the West Coast/Alaska Tsunami Warning Center's seismic analysis software called ANALYZE, the event simulator atplayer, and by earthworm module hypo_display. The format can be summarized as:

structure DISKHEADER;
structure CHNLHEADER * iNumStations;
chn 0 samp0, chn 0 samp1, ... chn 0 lNumSamps-1;
chn 1 samp0, chn 1 samp1, ... chn 1 lNumSamps-1;
.
.

.
chn NSta-1 samp0, chn NSta-1 samp1, ... chn NSta-1 lNumSamps-1

The DISKHEADER and CHNLHEADER structures are described in Appendix A.

Data from all stations are written to the same file. The file length (in minutes) is specifiable in the .d file. File lengths must be an integer value of minutes. The first sample from the file (Samp0) is the first sample at or after the start time of the file for each channel. This time is encoded in the file name (that is, the nominal start time of the file). The exact start time of each channel is saved in the header. Data is written in binary, 4-byte integer format.
Data is saved at whatever sample rate it is taken off the ring with. So, any decimation must be performed prior to placement on this module's input ring. In practice, more than one instance of this module may be needed. For example, data saved at the full rate may be necessary for P-picking and short period magnitudes. But, this rate may cause disk reads to be too slow for long period data processing where over an hour of data may be needed at once. So, the .d file allows input of the data file name's suffix. This way long period data is discernible by file name from short period data.

Data files are created and initialized with headers and zeroes when the first packet of data arrives that should be placed in the file. Headers are updated as more data arrives and the data is patched in the correct section of file. File names are based on the start time of the data within that file. A new subdirectory is created for each day. The subdirectory form is x:\path\d980522 where 98 is the last two digits of the year, 05 is the month (1=January), and 22 is the day. All files with data for this day are located in this subdirectory. The file names are given in the form x:\path\d980522\s5n2328.s98 where s=seismic, 5=month (a=10, etc.), n=day (a=10, b=11, etc.), 23=hour, 29=minute, s=high frequency data, and 98=the last two digits of the year.
A thread called CircDeleteThread deletes files older than a set number of hours. This prevents disk overload.

NOTE: This module will only work under Windows. The functions necessary in CircDeleteThread are Windows specific.

Graphical Interface:
This module has no graphical interface. Some errors are logged to the Windows console, if created.

3.1.3 Hypo_display

Date Written: 3/2001
Development Language: MicroSoft C/C++ v7.0 using WinAPI

Necessary Rings:
Input Ring – HYPO_RING_EB; reads TYPE_HYPOTWC format messages
Output Ring - PICK_RING_EB; writes TYPE_PICKTWC format messages

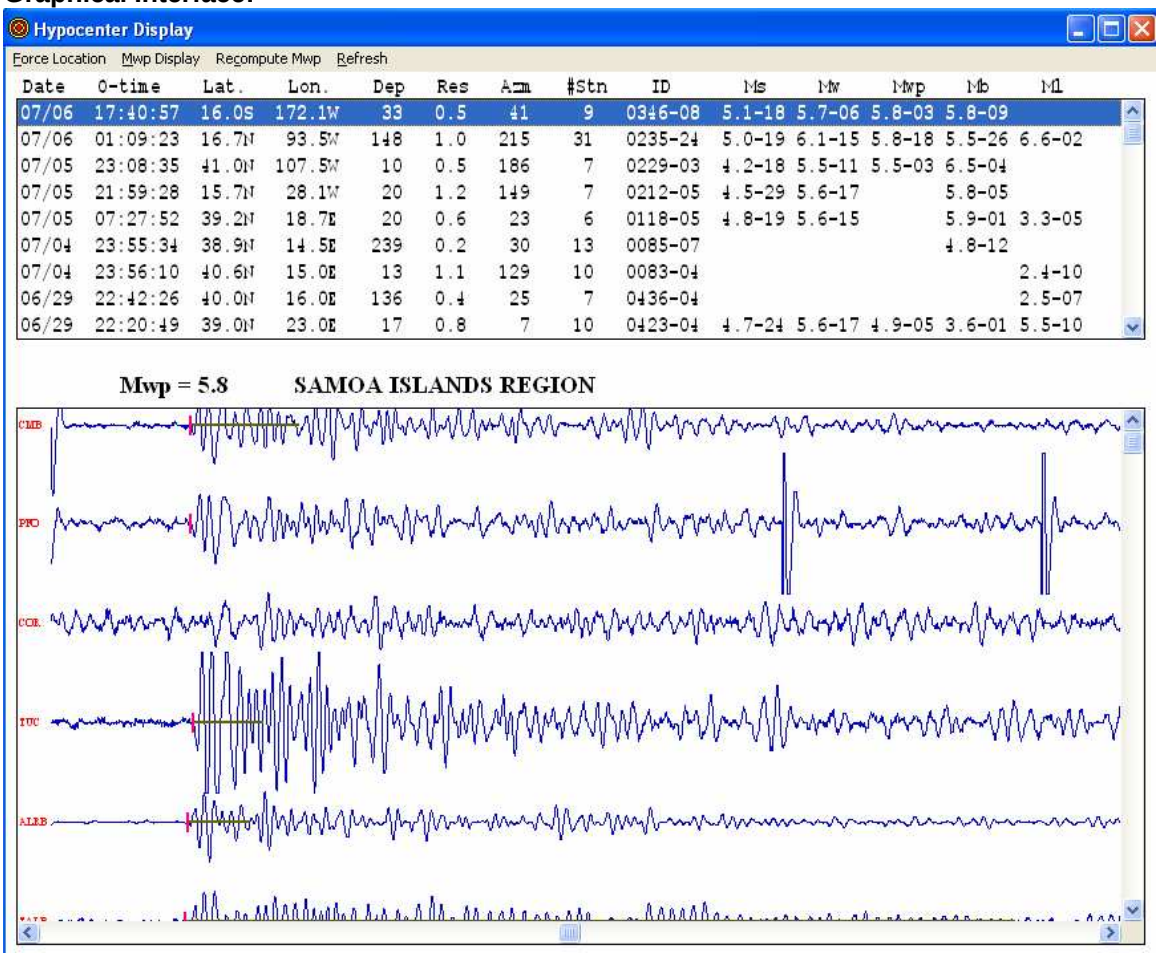
Necessary Data Files:
StaFile- picker configuration file
StaDataFile- seismometer location, elevation, etc.
RTPFile - P-data file to share with WC/ATWC program LOCATE
QuakeFile - List of automatically located quakes
DiskWritePath and FileSuffix: - Disk data path and suffix

Initialization File:
hypo_display.d - read on module start-up

Module Description:
This module takes hypocenter information (in TYPE_HYPOTWC format) from the Input Ring and displays it on a window. The module displays hypocenter information and waveforms on an easy to read and interact with display. Hypocenters computed in loc_eb (loc_wcatwc) are matched with P-data files.

The expected P-time for each channel listed in StaFile is computed, seismic data read from disk, and displayed in order of epicentral distance on a child window. P-picks used in the automatic location are shown on the seismic data display and can be altered interactively. P-picks which are changed or added are then sent back to the PICK_RING to update the P-buffer in loc_eb (loc_wcatwc). Future locations for this quake in loc_eb (loc_wcatwc) will utilize the new and altered P-picks. As P-picks are changed or added here, Mb, Ml, and Mwp parameters are computed from the signal and reported with the P to PICK_RING.

Graphical Interface:



Program Controls:

This module has a Windows graphical interface. Due to this interface, the module will only compile and run under Windows (NT, 2K, and XP). The display is broken into three sections. The top section lists the last MAX_QUAKES locations written to the InRing. Only the latest version of each is shown. Information is listed about each quake, such as: ID, version, O-time, date, lat, lon, depth, residual, azimuth, # stns., and magnitude data. When a quake is left-clicked, the quake's associated waveforms are shown below. This child window displays all waveforms for stations listed in StaFile ordered by epicentral distance with the expected P-time 25% from left to right of

the child window. P-times are shown with red lines on this display. P's can be added or adjusted here by left-clicking on the trace, and the quake re-located either automatically by loc_eb (loc_wcatwc) or interactively in the WC/ATWC program LOCATE. Just above this box is a littoral location and preferred magnitude for the earthquake.

- Mouse Buttons:
  - Left Button on top section list box displays appropriate traces in lower box.
  - Left Button in trace display window adjusts P-picks in file shared with LOCATE and with picks used in loc_eb (loc_wcatwc).
  - Left Button on station name in trace display window removes that station from further locations in loc_eb (loc_wcatwc) and from the file shared with LOCATE.

- Menu Options:
  - Force Location: Drops a “note” in PICK_RING so that loc_eb (loc_wcatwc) will recompute location for quake being evaluated.
  - Mwp Display: Create child window to show integrated broadband signal used in Mwp computations.
  - Recompute Mwp: Specify maximum integration window and signal-to-noise ratio use in Mwp computations.
  - Refresh: Re-paint the screen.

3.1.4 Latency_mon

Necessary Rings:
Input Ring – INPUT_RING; reads TYPE_TRACEBUF format messages

Necessary Data Files:
StaFile- picker configuration file
StaDataFile - seismometer location, elevation, etc.
LogPath – Path to output files which contain latency summaries

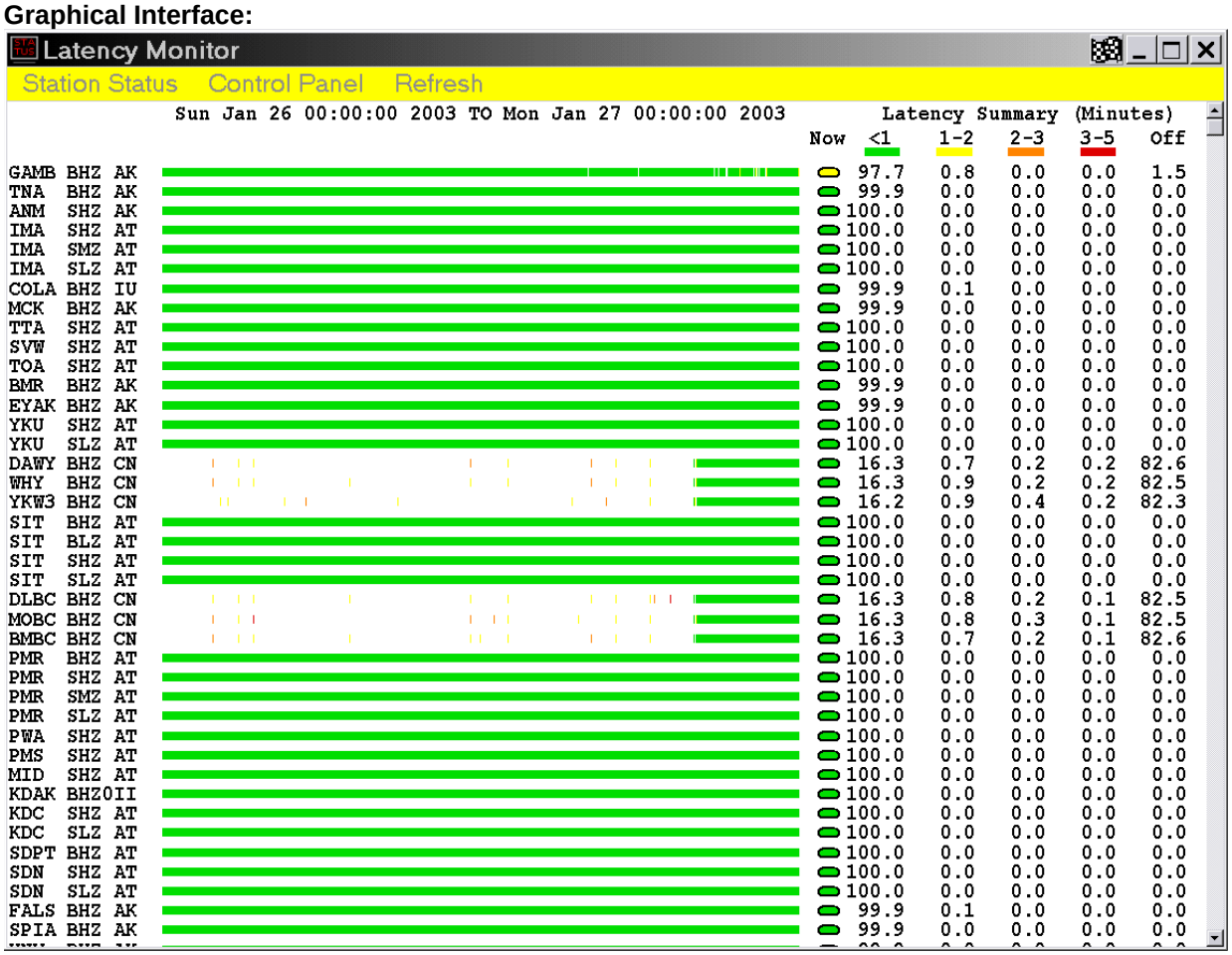
Initialization File:
latency_mon.d - read on module start-up

Module Description:
This module takes waveforms from InRing and logs latency and outages of the data for each trace. Each time a packet arrives, its start time is compared to the end time of the last packet. When there is a gap the outage is logged to a status file for that station. Also, the mid-time of the packet is compared to real-time, and latency is determined. Whenever the latency changes, it is also logged with start and end time to the file. Latencies are grouped generally: <1' = 0; 1'-2' = 1; 2'-3' = 2; 3'-5' = 3; >5' = 4. This module will only give accurate outage values if this module is running all the time the earthworm system is operating. Also, this module is not really relevant to digitized analog data as data packets will be acquired, even if a station is out.

An upgrade was added in 2002 by USGS/NEIC. This allows stations to be specified by an output file from the earthworm utility FindWave as opposed to StaFile. The GUI was also upgraded.

NOTE: The latency log files can become very large when a station flips back and forth often between different latency groups. The files may need to be moved and reinitiated every year.

Graphical Interface:



Program Controls:
This module has a Windows graphical interface. Due to this interface, the module will only compile and run under Windows (NT, 2K, or XP). The graphical output shows a line for each station in the StaFile for a specified length of time. Colors on the line represent the latency of the station over that time interval. If no line is drawn, the data either was not recorded or was later than 5'. To the right of the symbolic lines, a summary is given showing the percentages of time that the signal for each station was in each latency range. This summary can also be printed out.

- Menu Options:
  - Station Status: Specify time range over which to display latencies.
  - Control Panel:
    - *Traces per Screen*: Set number of lines to show on visible section of display.
    - *Print Status Summary*: Print out summary of latency information for last request.
    - *Save Status Summary*: Write latency summary to data file.
  - Refresh: Re-paint the screen.

3.1.5 Loc_eb (Loc_wcatwc)

Necessary Rings:
Input Ring – PICK_RING_EB; reads TYPE_PICKTWC format messages
Output Ring - HYPO_RING_EB; writes TYPE_HYPOTWC and TYPE_H71SUM2K format

messages
Output Ring - ALARM_RING; writes TYPE_ALARM format messages

Necessary Data Files:
StaDataFile - seismometer location, elevation, etc.
BValFile - Richter B-value data for Mb
OldQuakes - List of previously located quakes
AutoLoc - Trigger file for ANALYZE program
DummyFile - Hypo-parameters of last located quake
MapFile - Trigger file for EarthVu
RTPFile - P-data used in last location
QLogFile – Log of all quakes located
MwFile – Mw determined from Mm module

Initialization File:
loc_eb.d - read on module start-up

Module Description:
This program locates earthquakes given TYPE_PICKTWC format P-picks placed in the InputRing. The first task of loc_eb (loc_wcatwc) is to sort Ps into buffers which contain just picks from the same earthquake. This can be a very difficult task for some station geometries. The first sort is just an initial guess and is based on some simple logic. If the time difference between the new P and all other Ps within the buffer is less than the maximum which could be expected for the distance between the Ps, the new P is placed in that buffer. A second check is also made: if the P time is greater than MaxTimBetweenPicks from the last P-pick in the buffer, it will not be included in that buffer. After a buffer has enough P-picks to locate a quake (MinPs), the solution is computed. If a good solution is made, P's from other buffers are compared to this solution and are added back into the buffer if they fit (unless they are in a buffer which has produced a good location). Also, P's which were eliminated by the Bad P discriminator in the locate routine are placed in a different P buffer. This scavenging and removal of Ps after locations are made is the second sorting of Ps throughout the different buffers. As new P-data enters a buffer which has more than MinPs picks, the hypocenter is updated for that buffer. The flowchart illustrating this process in included in Appendix B.

Quake locations are computed using Geiger's method given an initial location. The initial location estimate is first assigned to the location of the first P-time in the buffer. If a solution can not be computed from this initial location, a routine is called to compute the initial location from azimuth and distance determined from a quadripartite of stations. If a location can still not be determined, a bad P-pick discriminator is called. This simply throws out stations one-at-a-time (up to three stations at once) and re-computes the location. Good solutions are verified by total residual.

The IASPEI91 travel times are used as the basis for quake locations in this program. A time/distance/depth table has been created from software provided by the National Earthquake Information Center. Locations with this set of P times have been compared to those made with the Jefferey's-Bullen set of times and were found to be superior in regards to depth discrimination and epicentral location with poor azimuthal control. The P-table is arranged on 10km depth increments and 0.5 degree distance increments.

A routine was added in 2006 which provides better depth control for solutions. The quake depth is fixed to the average depth for the region (based on USGS historical data on a one degree by one degree grid). When enough P control is attained in the buffer, the depth will float, but will be limited by the maximum depth of the region.
A common problem with this associator/locator is that the same quake may be located in two different buffers. This can happen due to phases other than the P (such as the pP or sP) being picked instead of the real P. A routine is called to check for similar quakes in different buffers. Only the location from the buffer with the most Ps is sent to the HYPO_RING_EB.
After a good location has been computed, magnitude is output based on the

amplitude/periods/integrations reported by the P-picker, develo, hypo_display, and/or lpproc and mm. Mb, Ml, MS, Mw, and Mwp magnitudes are computed depending on epicentral distance. The locations/magnitudes are sent to the HYPO_RING_EB. Alarms based on location and magnitude can also be issued to the AlarmRing if desired.

Graphical Interface:
This module has no graphical interface. Locations and some errors are logged to the Windows console, if created.

3.1.6 Lpproc

Date Written: 4/2001
Development Language: MicroSoft C/C++ v6.0 using WinAPI

Necessary Rings:
Input Ring - WAVE_RING_EB_LP; reads TYPE_TRACEBUF format messages
Input Ring - HYPO_RING_EB; reads TYPE_HYPOTWC format messages
Output Ring - PICK_RING_EB; writes TYPE_PICKTWC format messages (with MS parameters)
Output Ring - ALARM_RING; writes TYPE_ALARM format messages

Necessary Data Files:
StaFile – lpproc configuration file
StaDataFile - seismometer location, elevation, etc.

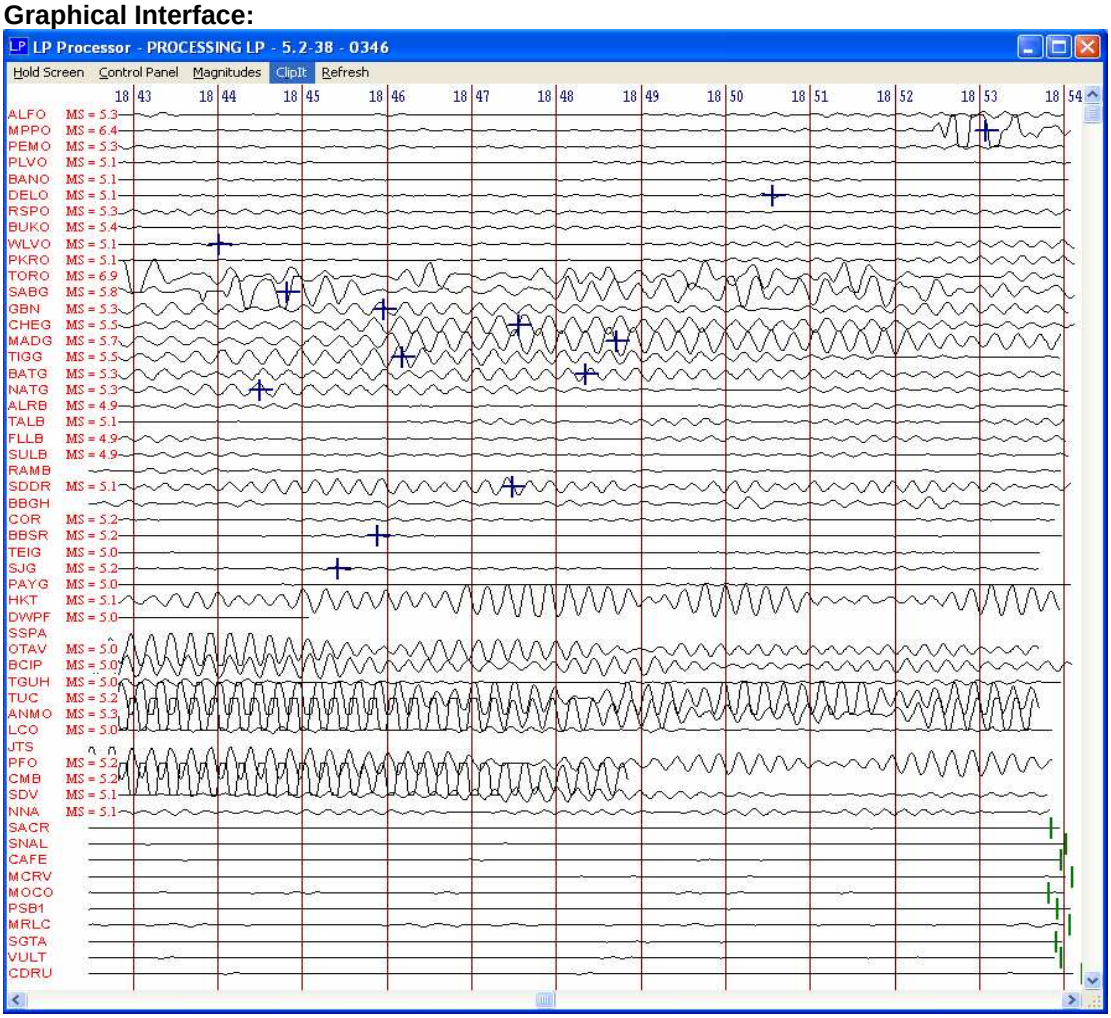
DummyFile - Hypo-parameters of last located quake
LPRTFile - MS data computed for last quake
QuakeFile - List of automatically located quakes

Initialization File:
lpproc.d - read on module start-up

Module Description:
This module takes long period waveforms from the WAVE_RING_EB and displays them on a window. The display is similar to an old-style develocorder (with no film delay). Every X seconds the screen scrolls to the left. The most recent data is shown on the right of the screen. The display can be "held" and manually scrolled through previous times (the time limit of saved data is specified in the .d file). Hypocenters entered to HYPO_RING_EB are obtained by this module. If the hypocenter has a magnitude greater than 5.0 Mb, Ml, or Mwp, this module will compute expected Rayleigh wave times at stations listed in StaFile and will evaluate the data in the window for Ms for up to 30 minutes passed the expected Rayleigh wave arrival time. When an Ms is computed, the data is sent in the TYPE_PICKTWC message format to PICK_RING_EB with P-time=0 and the magnitude is displayed on the lpproc screen. As data is read from the WAVE_RING_EB, it can be processed by an LP alarm with parameters specified in the initialization file and the StaFile. If a large signal is found by the alarm function, an alarm is reported to ALARM_RING.

NOTE: A semaphore is accessed in lpproc so that LP processing can be triggered by program LOCATE. This semaphore should not prevent the program from working if LOCATE is not running.

Graphical Interface:



Program Controls:

This module has a Windows graphical interface. Due to this interface, the module will only compile and run under Windows (NT, 2K, or XP). The display shows real-time, long period data. The most recent time is shown on the right side of the screen. As new data arrives, the screen scrolls left. A "Hold Screen" menu option is available to stop the automatic scrolling. Then, the horizontal scroll bars can be used to manually scroll through all the data which is in memory. Surface wave data processing is triggered automatically via the HYPO_RING_EB. Computed MS is shown near the trace name and an average is shown in the title bar. The MS processing can be started or stopped manually through the magnitude menu option. As magnitudes are computed, the expected Rayleigh wave start time is displayed on each trace, and the cycle which is used to compute MS is marked with a plus sign.

- Mouse Buttons:
  - Right button on trace toggles trace display.
  - Right button on station name calls station manipulation menu (for trace scaling, trace display on/off, alarm on/off).

- Menu Options:
  - Hold Screen: Prevent automatic scrolling so that scroll bars can be used to move through data.
  - Control Panel:

Horizontal Scale: Change screen time scale.
Vertical Trace Scale - All: Change vertical scale for all stations.
Traces Per Screen: Number of traces vertically to show on the screen, rest will scroll.
Display Filtered Data: Check -> screen shows LP filtered signal for broadband stations.
Display Broadband Data: Check -> screen shows unfiltered traces for all stations.

- Magnitude:
  - Compute Ms: Using last location in dummy file, process LP data for MS. After processing, a plus sign is marked on the cycle used to compute MS.
  - Stop Processing: Stop ongoing computations (usually used when automatic location has been updated).
- ClipIt: Limit trace amplitudes.
- Refresh: Re-paint the screen

3.1.7 Mm

Date Written: 1/2002
Development Language: MicroSoft C/C++ v7.0 using WinAPI

Necessary Rings:
Input Ring - WAVE_RING_EB_LP; reads TYPE_TRACEBUF format messages
Input Ring - HYPO_RING_EB; reads TYPE_HYPOTWC format messages

Necessary Data Files:
StaFile - picker configuration file
StaDataFile - seismometer location, elevation, etc.
DummyFile - Hypo-parameters of last located quake
ResponseFile - Broadband instrument response in poles/zeroes format
RegionFile - Mm path correction file
MwFile - Mw results file

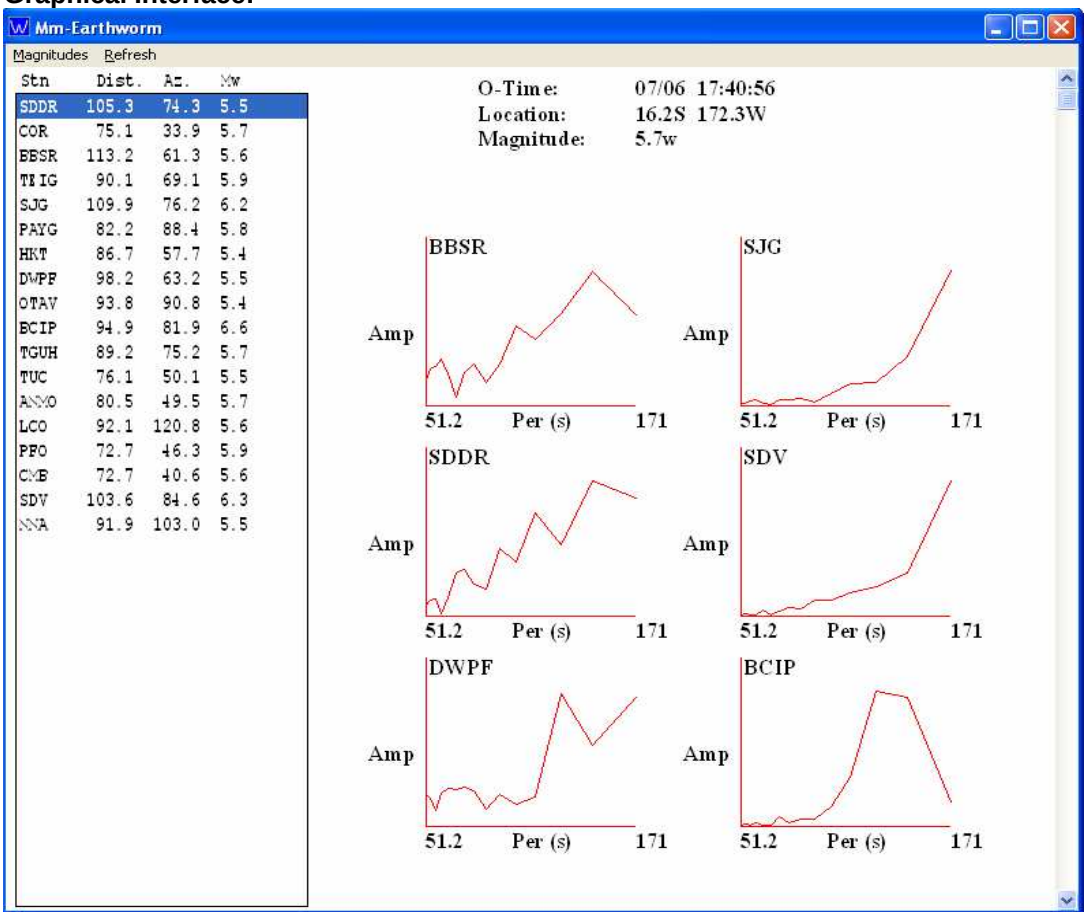
Initialization File:
mm.d - read on module start-up

Module Description:
This module determines moment magnitude from broadband seismic data. The magnitude scheme (Mm) used at the French Polynesian Tsunami Center is used to determine Mm (Mw). This was developed by Okal, Talandier, and others from the French Polynesian center. The original code was provided by Weinstein of PTWC. At the WC/ATWC it was converted to this earthworm module.

Input seismic data is taken from WAVE_RING_EB_LP. The data is buffered by this module. Processing is triggered by a TYPE_HYPOTWC message obtained from the HYPO_RING_EB. If the hypocenter has a magnitude greater than 5.0 Mb, Ml, or Mwp, this module will compute expected Rayleigh wave times at stations listed in StaFile and will evaluate the data in the window for Mw. The Mw is computed 12 minutes after expected Rayleigh wave arrival time based on peak spectral amplitude and Rayleigh wave path. No signal-to-noise checking is performed. Data is saved for up to MinutesInBuffer time, so processing can be performed for a quake so long as the data is still in the buffer. A semaphore is set up here (similar to lpproc) that can be set in the program LOCATE to trigger processing. LOCATE does not have to be running for this code to work. The processing is either triggered automatically through the HYPO_RING_EB, manually in LOCATE, or manually here through a menu option.

Output from this module is written to MwFile and to a window. Module hypo_display and program LOCATE access MwFile.

Graphical Interface:



Program Controls:

This module has a Windows graphical interface. Due to this interface, the module will only compile and run under Windows (NT, 2K, or XP). The display shows a list of computed Mws on the left and spectral displays for chosen stations on the right. As magnitudes are computed, the average Mw will be displayed in the title bar.

Double-clicking a station name in the list box will select that station for spectral display on the right. If no interaction is made, the last 6 stations for which Mw was computed will be displayed on the right.

- Mouse Buttons:
  - Left button on station in list will bring up spectral display for that station.

- Menu Options:
  - Magnitudes:
    Compute Mm: Using last location in dummy file, process broadband data for Mm.
    Stop Processing: Stop ongoing computations (usually used when automatic location has been updated).
  - Refresh: Re-paint the screen.

3.1.8 Mtinver

Date Written: 8/2004
Development Language: MicroSoft C/C++ v7.0 using WinAPI

Necessary Rings:
Input Ring - WAVE_RING_EB_LP; reads TYPE_TRACEBUF format messages
Input Ring - HYPO_RING_EB; reads TYPE_HYPOTWC format messages

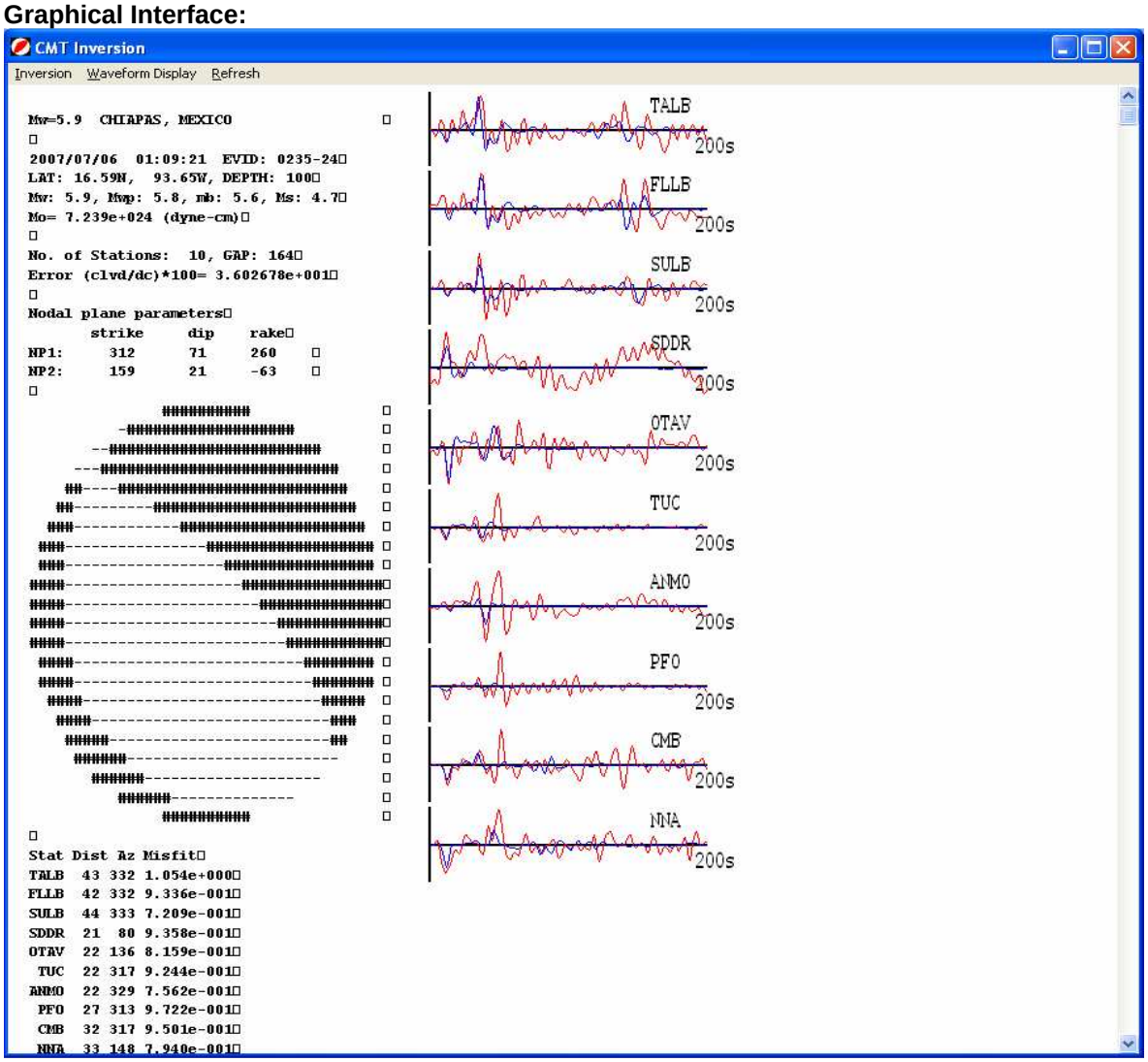
Necessary Data Files:
StaFile - picker configuration file
StaDataFile - seismometer location, elevation, etc.
DummyFile - Hypo-parameters of last located quake
ResponseFile - Broadband instrument response in poles/zeroes format
MTFile – Results file
EmailFile – Results file to send
DepthFile – Maximum expected quake depth file
EBPTimeFile – Observed P-Times file

Initialization File:
mtinver.d - read on module start-up

Module Description:
This module determines moment tensor solutions from teleseismic broadband seismic data by inverting the first 200s of data. The technique was provided to the WC/ATWC by USGS, and converted to an earthworm module which reads data from a WAVE_RING_EB, or from files created by disk_eb (disk_wcatwc). Observed data are compared to synthetic data to determine the most likely moment tensor. Green's functions provided by USGS are used to generate the synthetic seismograms. Trace data are only used for stations with observed p-times (taken from automatic p-picks or interactive p-picks). The synthetics are shifted up to a specified amount of time to account for bad p-picks. Synthetic seismograms are created for depths up to the maximum expected depth of the epicentral region. Moment tensors, scalar moment magnitude, focal plane solution, and depth are output. An epicenter must be provided to the module. The data and synthetics are filtered based on the first seismic moment estimate. Lower frequency filters are used for larger earthquakes.

Input seismic data is taken from WAVE_RING_LP. The data is buffered by this module. Processing is triggered by a TYPE_HYPOTWC message obtained from the HYPO_RING_EB. If the hypocenter has a magnitude greater than 5.0 Mb, Ml, or Mwp, this module will compute a moment tensor every one minute from 11 minutes after the origin to 25 minutes after. Several checks are made on the data, including signal-to-noise, gaps checks, and DC offset removal. Data are saved for up to MinutesInBuff time, so processing can be performed for a quake so long as the data are still in the buffer. The processing is either triggered automatically through the HYPO_RING_EB, or manually in this module through a menu option.

Graphical Interface:



Program Controls:

This module has a Windows graphical interface. Due to this interface, the module will only compile and run under Windows (NT, 2K, or XP). The display shows a beachball and source parameters on the left, and comparisons between the observed signal and synthetic on the right.

- Mouse Buttons:
  - o No effect.

- Menu Options:
  - o Inversion:
    - Compute MT: Using last location in dummy file, process broadband data for moment tensor inversion. Several control parameters can be interactively set in the follow-up dialog box.
    - Stop Processing: Stop ongoing computations.
  - o Waveform Display:
    - Green's functions: Display original Green's functions for a distance/depth.
    - Synthetics: Display the results for the last compute moment tensor.
  - o Refresh: Re-paint the screen.

3.1.9 Page_alarm

Date Written: 1/2001
Development Language: MicroSoft C/C++ v7.0

Necessary Rings:
Input Ring - ALARM_RING; reads TYPE_ALARM format messages

Necessary Data Files:
AlarmTextFile – Text to send over cel phones
AlarmFile – Cell phones to call
AlarmBatFile – Batch file to send cell phone messages
PagerFile – Text to send to pagers.

Initialization File:
page_alarm.d - read on module start-up

Module Description:
This module takes a message from an ALARM_RING, and if the message is a valid alarm message, it will send it out to different output devices dependent on flag settings. Alarm messages can:

- Be sent out a serial port (to activate a pager system - the serial port baud rate, etc., are set up in the initialization file),
- Activate a .wav file which is broadcast over the computers speakers, or
- In conjunction with an email sending program, send cell phone text messages.

Filtering of alarm messages to send over the various options must be performed by the modules sending the alarms.

Graphical Interface:
This module has no graphical interface. Pager messages are logged to the Windows console, if created.

3.1.10 Pick_eb (Pick_wcatwc)

Date Written: 12/2000
Development Language: MicroSoft C/C++ v7.0

Necessary Rings:
Input Ring - WAVE_RING_EB; reads TYPE_TRACEBUF format messages
Output Ring - PICK_RING_EB; writes TYPE_PICKTWC and TYPE_PICK_GLOBAL format messages
Output Ring - ALARM_RING; writes TYPE_ALARM format messages

Necessary Data Files:
StaFile - picker configuration file
StaDataFile - seismometer location, elevation, etc.
AlarmFile – multi-station alarm parameters

Initialization File:
pick_eb.d - read on module start-up

Module Description:
This earthworm module is the latest re-write of the P-picker used at the West Coast/Alaska Tsunami Warning Center since the early 1980's. The P-picker was developed by Veith at Teledyne GeoTech. Several additions have been made regarding magnitude determinations, automatic calibrations, and alarming, but the P-pick algorithm itself is basically the same as first proposed. See Appendix C for a more detailed description of the P-pick algorithm.
Short period filtering is provided by this module for broadband data when specified in the initialization file. Response for the default filter has been computed and is used to convert amplitude to ground motion. If other filter parameters are used, the response must be calculated and used in the ground motion calculations. The raw broadband data is still used, though, for Mwp computations.

Two alarm features were added in this implementation of the Veith picker. This first looks for potentially large quakes by comparing the signal amplitude at a single station to a set threshold. The amplitude threshold must be passed along with a duration threshold. When the amplitude threshold is surpassed throughout the duration threshold in a cyclical manner, an alarm is declared and fed to the alarm ring. Thresholds are set in StaFile. A second alarm feature was added in 2004. This alarm is activated when strong P-arrivals are recorded at a specified number of stations within a specified time period and distance range. Parameters for the second type of alarm are set in the AlarmFile.

A no data alarm was also added to pick_wcatwc in 8/01. An alarm time is specified in the .d file. If this time passes without data arriving in the ring, an alarm message will be issued to the alarm ring. The picker attempts first motion determination.

Graphical Interface:
This module has no graphical interface. Picks and some errors are logged to the Windows console, if created.

3.2 Earthworm Modules

The modules described in Sections 3.2.1 through 3.2.6 are part of the standard Earthworm software package. Additional information on operation and configuration of these modules can be obtained at the following Earthworm web site: http://folkworm.ceri.memphis.edu/ew-doc/modules.html .

3.2.1 Copystatus

Necessary Rings:
All rings created as part of the system (INPUT_RING, WAVE_RING_EB, WAVE_RING_EB_LP, PICK_RING_EB, HYPO_RING_EB, ALARM_RING).

Necessary Data Files:
None.

Initialization File:
None

Module Description:
The status manager ("statmgr") program reads all error and heartbeat messages from a single transport ring (shared-memory region). However, client programs to the status manager may write their error and heartbeat messages to any transport ring. In order to make all error and heartbeat messages accessible to the status manager, the copystatus program is used to copy messages from one transport ring to another.

Graphical Interface:
None

3.2.2 Decimate

Necessary Rings:
Input Ring – INPUT_RING; reads TYPE_TRACEBUF messages
Output Ring – WAVE_RING_EB; writes TYPE_TRACEBUF messages

Necessary Data Files:
None

Initialization File:
decimate_XtoY.d - read on module start-up

Module Description:

The Decimate module implements multi-stage filter and decimation routines to process wave data (TYPE_TRACEBUF) coming from the InRing defined in decimate.d. Decimated trace messages are written to the OutRing with their SCNs changed as specified in decimate.d. The decimated trace message retains its TYPE_TRACEBUF type, but it (normally) has a new SCN name.

Decimate uses FIR filters to reduce the incoming wave frequency content down to the Nyquist frequency of the output wave. FIR filters have the advantage of being strictly causal and stable. However, there is considerable delay introduced by a FIR filter. Fortunately FIR filters have linear phase so the delay does not change the shape of the wave. The delay will be removed from the data by adjusting their timestamps. This adjustment makes the filter acausal: small precursory artifacts will be seen in front of impulsive wave arrivals. This is an inherent characteristic of FIR filters. See "Of Poles and Zeros" by Frank Scherbaum for a method of removing these artifacts.

The decimation is performed as part of the filter routine for efficiency. The FIR filter coefficients are determined from the specified decimation rate, using the Remez Exchange algorithm. This produces an equi-ripple filter. Currently the ripple limits are hard coded to 0.005 in the pass band and 0.0031 in the stop band (see decimate.h in the source directory.) The coefficients zeroes of each filter, as well as the effective delay, are logged on startup. FIR filters have no poles.

Graphical Interface:
None

3.2.3 Liss2ew

Necessary Rings:
Output Ring – INPUT_RING; writes TYPE_TRACEBUF messages

Necessary Data Files:
None

Initialization File:
Liss2ew_stationname.d – read on module startup

Module Description:

Liss2ew is the earthworm module for receiving trace data from the Live Internet Seismic Server. The LISS is a very simple service that provides trace data in miniSEED format in near-real-time

over the Internet. The liss2ew module will connect to one LISS, read selected streams of trace data, and write that trace data into the earthworm transport ring.

One LISS normally has data for a single station. One or more components are sent together in the data stream. In some cases, one component is provided in more than one sample-rate. The ‘location code’ is used in this case to label the different kinds of data.

The miniSEED format may contain a few other types of blockettes besides the Data-Only blockette. (You can read the PDF version of the SEED manual if you want to know more.) Currently liss2ew ignores these non-trace-data blockettes.

Graphical Interface:
None

3.2.4 Naqs2ew

Necessary Rings:
Output Ring – INPUT_RING; write TYPE_TRACEBUF messages

Necessary Data Files:
None

Initialization File:
Naqs2ew.d – read on module startup

Module Description:

Naqs2ew is an interface through which waveform data collected by the Nanometrics data acquisition software, NaqsServer, can be fed into an Earthworm system in near-real-time. Naqs2ew is a client of NaqsServer's "Online Data Streams" service which allows client programs to open a socket connection, subscribe to only the data they want, and receive the requested data over the same socket. The "online data streams" protocol and data formats are described in the Nanometrics Reference Manuals. Currently the only NaqsServer packet types that naqs2ew can process are: channel list messages, decompressed data packets, compressed data packets, and error messages. Naqs2ew is not required to run on the same computer as NaqsServer.

Naqs2ew is a single-threaded program. After reading its configuration file, allocating working buffers, and attaching to a transport ring, naqs2ew attempts to open a socket connection to NaqsServer, using a timeout of CONNECT_TIMEOUT seconds (currently 5 s). If the connection attempt fails, naqs2ew will attempt to connect every CONNECT_RETRY_DT seconds (currently 10 sec) forever. Naqs2ew will continue beating its own heart while trying to establish a NaqsServer connection.

Once it establishes a socket connection with NaqsServer, naqs2ew initiates the "online data streams" protocol. It sends a "connect message," reads and logs the list of available channels, and requests only the channels listed in its own configuration file. If some channels in the configuration file are not available, naqs2ew logs that fact and continues. While the socket connection remains open, naqs2ew reads a complete packet (blocking), converts it to one or more Earthworm TYPE_TRACEBUF messages, and writes the message(s) to the transport ring. Naqs2ew makes no attempt to reorder time series packets. It does attempt to produce one-second TYPE_TRACEBUF messages for each channel, buffering any fractional seconds between packets from NaqsServer if necessary. Time tears between packets will cause naqs2ew to produce TYPE_TRACEBUF packets shorter than one-second long. Naqs2ew also logs any error messages it receives from NaqsServer. After each packet is processed, naqs2ew enters its heartbeat function, and issues a heartbeat if its heartbeat timer has expired.

If naqs2ew experiences any socket errors or if NaqsServer breaks the connection, naqs2ew closes the socket and issues a TYPE_ERROR message stating the reason the connection was dropped. It then goes back to looping on reconnection attempts. When the connection is reestablished, naqs2ew will issue another TYPE_ERROR message (really an "un-error" message), stating that everything's OK again.

Graphical Interface:
None.

3.2.5 Ringdup_scn

Necessary Rings:
Input Ring – INPUT_RING; reads TYPE_TRACEBUF messages
Output Ring – WAVE_RING_EB; writes TYPE_TRACEBUF messages

Initialization File:
ringdup_scn.d - read on module start-up

Module Description:

The ringdup_scn module is used to copy TYPE_TRACEBUF messages from one ring into another. In EarlyBird, it is used to transfer TYPE_TRACEBUF message from the INPUT_RING to the WAVE_RING_EB for channels that are received from the source at 20sps and do not require decimation.

Graphical Interface:
None

3.2.6 Slink2ew

Necessary Rings:
Output Ring – INPUT_RING; writes TYPE_TRACEBUF messages

Necessary Data Files:
None

Initialization File:
slink2ew.d - read on module start-up

Module Description:

Slink2ew is a SeedLink client module for Earthworm. SeedLink is the data server component of the Seismological Communication Processor, or SeisComP, originally developed at GEOFON. The SeedLink protocol can be summarized as a simple negotiation scheme followed by the streaming of data packets from the server. SeedLink packets are composed of a small header followed by a 512-byte Mini-SEED record (data only SEED). The negotiation phase allows the client to request only specified data types from the server for each data stream. A data stream is defined by a network and station code pair.

By utilizing sequence numbers for each packet in a data stream the SeedLink protocol allows for connections to be resumed, eliminating most data gaps. The ability to resume data streams is primarily dependant on how much data, time-wise, the remote SeedLink has in it's buffer. The protocol allows for two different modes of data transmission, uni-station and multi-station modes. Uni-station mode operates by transmitting a single data stream (data from a single

station) through one network connection. In this mode the data stream does not need to be specified by the client as it is implied by the internet address and port. Multi-station mode operates by transmitting multiplexed data streams (data from multiple stations) through a single network connection.

The maximum size of the TRACEBUF messages created by slink2ew is limited to the maximum number of samples (using Steim2 compression) that can fit into a 512-byte Mini-SEED record (721 samples = 2884 bytes) plus the size of the TRACEBUF header (64 bytes). So 2948 bytes is a good estimate of the maximum size of TRACEBUF messages created by slink2ew.

Version 1.1 and greater of slink2ew supports the creation of either TRACEBUF or TRACEBUF2 messages depending on which version of Earthworm the module was built for. By default the module will create TRACEBUF2 messages if the system supports them and TRACEBUF messages otherwise. Additionally there is an optional command ForceTraceBuf1 that will force the use of TRACEBUF messages on systems that support TRACEBUF2.

Graphical Interface:
None

3.2.7 Statmgr

Necessary Rings:
Input Ring – HYPO_RING_EB; reads TYPE_STOP and heartbeat messages
Output Ring – HYPO_RING_EB; writes TYPE_RESTART and TYPE_PAGE messages

Necessary Data Files:
None

Initialization File:
statmgr.d - read on module start-up

Module Description:

Statmgr is tool to monitor the health of all the Earthworm modules. It reports on the health by email, and it may automatically issue a restart request for a dead module, if the module's .desc file configures statmgr to do so.

Statmgr works by monitoring error messages which are produced by other Earthworm modules, and determines whether to report and how to report an error. Errors are reported by sending email or generating TYPE_PAGE messages. User-provided software can then pick up the TYPE_PAGE message and hand it to paging software to transmits these messages via modem to a pager service. Statmgr also monitors heartbeats of client modules, and if heartbeats are not received, an email and/or pager message is produced.

Statmgr has a restart feature which allows the system to recover if any module hangs by restarting only the hung module. Any module can request to be restarted if it's heartbeat stops. Otherwise, no restart attempt will be made. If statmgr detects that heartbeats from the module have stopped, statmgr will send a message of type TYPE_RESTART to the startstop program. Startstop will then kill the module process and restart the module.

Statmgr monitors for TYPE_STOP messages. If it sees one, it will not attempt to restart the stopped module, assuming it's been intentionally stopped with the "stopmodule" command line utility, or the "stopmodule" command in startstop. It also monitors for TYPE_RESTART messages. If it sees a restart of a stopped module, it'll assume that it's been started again, and will resume monitoring of it.

By default, Statmgr only monitors for heartbeat messages the RingName specified in the statmgr.d config file. Typically modules only send heartbeat messages to the ring they're active

on. Thus if one wants to have statmgr monitor modules which aren't on the ring that RingName specifies, one needs to do one of two things. The first option is to set CheckAllRings to 1 in statmgr.d. Statmgr will make a status request to startstop when it starts up and monitor all the rings that startstop knows about. This works fine on many systems, but some systems with large amounts of information moving through a single ring may overload statmgr's ability to keep up. The second option is to set up a 'copystatus' module to copy the status from every ring with an active module, the ring specified by RingName which statmgr is monitoring. It clutters up your status screen a bit, but does the job.

For each module monitored by statmgr, a descriptor file must exist and be specified in the statmgr configuration file. The earthworm convention has been to use the suffix '.desc' to indicate a descriptor file. In the descriptor file, the user may specify the following:

How often the statmgr should check for the modules heartbeat and if email and/or pagers messages should be sent in case of missing heartbeats.

Who should pager messages (pagegroup command overrides same command in statmgr configuration file).

For each error reported by a module, should email and/or pagers messages be sent and how often should the messages be sent.

Graphical Interface:
None

3.3 EarlyBird Post-processing Modules

The stand-alone programs described in Sections 3.3.1 through 3.3.3 have been developed at the WCATWC and are available as part of the Earthworm Contributed software package. The explanations of the module functionality and controls have been taken from the Operations Manual posted on the WCATWC web site: http://wcatwc.arh.noaa.gov/DataProcessing/ew-eb.htm.

3.3.1 Analyze

Date Written: 12/1997
Development Language: MicroSoft C/C++ v7.0 using WinAPI

Initialization File:
analyze.ini - read on program start-up

Program Description:

Program ANALYZE is a non-Earthworm-based program which runs independent of any other seismic processing program. Data files and semaphores are shared between ANALYZE and other programs, but it is not dependent on any other program. Its main function is to evaluate seismic data previously written to disk file. disk_eb (disk_wcatwc) logs data to disk in the format expected by ANALYZE. File reads and writes are protected in both disk_eb (disk_wcatwc) and ANALYZE so that ANALYZE can access data files while disk_eb (disk_wcatwc) is filling them with data. Data files are named such that the file's time coverage can be determined from the file's path and name. The file name contains the start time minute for all data in the file. The first sample for each channel is the first sample at or after the start minute listed in the file name. File length in time must be in whole minutes (two minutes is normally used). Configurable options in ANALYZE are set in file analyze.ini.

This disk logging approach works well where data comes from many different sources. Most data is continuous, though some may be event-triggered. Some data arrives with seconds latency,

while others may be delayed several minutes. Some data are saved at high sample rates while others are saved at low rates. Regardless of how or when the data arrives, it is patched into the proper location of a pre-created file. At any time after file creation, ANALYZE has access to the data already logged in the file.

Program Controls:
The graphical interface and program controls are very similar to module develo. Scrolling through the data is performed with scroll bars or the mouse roller. As data from different files are needed, the file is opened and read. When file time is near present, data is checked for zeroes. If it appears that a channel has not been recorded yet, (possibly due to a high latency) the file is re-read on the next scroll. In this way the latest set of data should be read into memory.

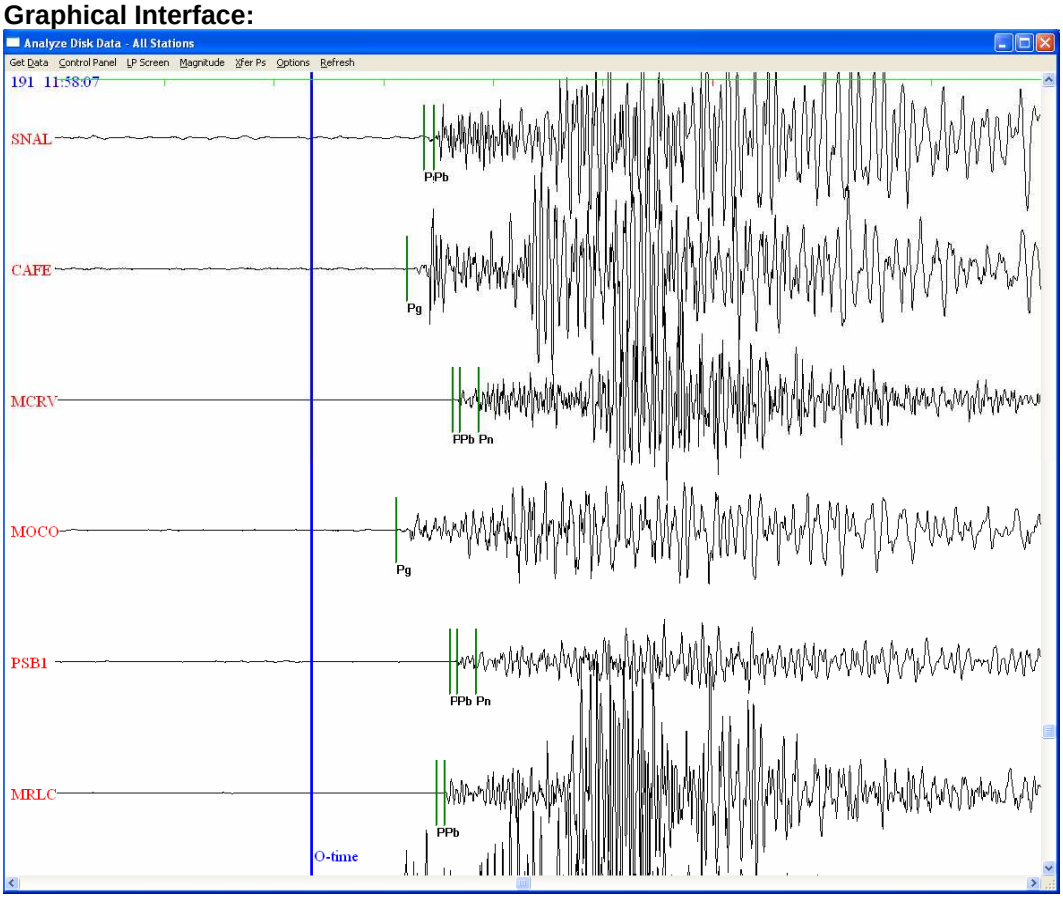
- Mouse buttons:
  - Right button on trace toggles trace display.
  - Right button on station name calls station manipulation menu (for trace scaling, station description, trace display on/off, P-picks on/off, show Spectrogram/FFT).
  - Left button on trace sets P-time and Mb, Ml, and Mwp magnitude information are determined when station sensitivity is known. LOCATE accesses this data through a shared file.
  - Left button on station name eliminates the station from the shared file with LOCATE.

- Menu Options:
  - Get Data:
    Most Recent: Display data with the left side of the screen set four minutes prior to the present time.
    New Quake: Call a dialog box which prompts for day and time to display on left side of screen
    Use Auto Location: Display the time of the first P arrival of the station set shown on the screen for the last located earthquake (i.e., the quake shown in the SUMMARY screen).
    Previous Quakes: Call up a list box which gives the last X automatic locations performed by loc_wcatwc.
    Archive Event: Copy disk files which cover the last located event to an archive. Data is saved from 2 to 4 minutes prior to the first P wave arrival to 30 minutes after the last Rayleigh wave arrival time.
    Read Archived Data: Retrieve previously saved data from an archive.
  - Control Panel:
    Horizontal Scale: Change screen time scale.
    Vertical Trace Scale - All: Change vertical scale for all stations.
    Traces Per Screen: Number of traces vertically to show on the screen, rest will scroll.
    Set Filter Parameters: Set cutoffs for SP/LP filters (defaults are normal values used within program - changing parameters will cause incorrect amplitude-to-ground motion conversions).
    All Stations:
    Other User-defined Station Groupings …
    Display Filtered Data: Check -> screen shows SP/LP filtered signal for broadband stations.
    Display Broadband Data: Check -> screen shows unfiltered traces for all stations.
    Show Velocity Data: Trace is seismometer output.
    Show Acceleration Data: Trace is differential of raw data.
    Show Displacement Data: Trace is integral of raw data.
  - LP Screen: Toggle screen display with long period filtered data.

- Magnitude:
  - Mwp: Set Max Window Time: Set the time of integration and S:N manually for broadband stations (used with original Mwp method).
  - Display Integrations: Show the integrated waveforms for all stations which could compute an Mwp
  - Original Method: Window length set by program or above menu option
  - Knight method: Window length set by frequency content. Cannot be manually overridden.
  - Mb:/Ml:/Ms | After any of these are clicked, rubber-band the cycle of interest for magnitude computations
  - Process LP: Using last location in dummy file, process LP disk data for MB/MS/Mw. After processing, the long period station set is displayed with boxes around cycles used to compute MS. The Mw spectral data is shown in a child window. The MS and Mw results are written to a file shared with LOCATE.
  - Calibrate: Use this option to calibrate analog WC/ATWC stations. Calibrations made on sine wave cals. Rubber-band the cals as in interactive magnitude computations.
- Xfer Ps: Update screen and P-data with picks made in hypo_display or associated in loc_wcatwc.
- Options:
  - Seismo: When flagged, manual P-picks bring up a phase window.
  - ClipIt: Limit trace amplitudes.
  - No Phase: Do not overlay seismic phase data on screen
  - Show P Phase: Overlay P, Pdiff, or PKP on data using IASPEI91 tables
  - Show Depth Phases: Overlay P Phases along with pP and sP phases
  - Show All Phases: Overlay all phases selected from IASPEI91 tables
  - Picks: Make P-picks on initial data read
  - No P-picks: Do Not make P-picks on initial data read (default)
- Refresh: Re-paint the screen

ANALYZE is signaled from loc_wcatwc to re-draw the traces whenever loc_wcatwc has computed an automatic 6-station location. The time of the first P arrival is sent to ANALYZE as the signal. ANALYZE then displays this time in the middle of the screen.

Graphical Interface:



3.3.2 Locate

Date Written: 11/1997
By: Whitmore - WC/ATWC
Development Language: MicroSoft C/C++ v7.0

Initialization File:
locate.ini - read on program start-up

Program Description:

Program LOCATE is a non-Earthworm-based program which runs independent of any other seismic processing program. Data files and semaphores are shared between LOCATE and other programs, but it is not dependent on any other program. Its main function is to interactively locate and size earthquakes based on data obtained from loc_wcatwc/hypo_display or ANALYZE. (Though, P-picks and magnitude information can be added through a menu option.) Locations computed within this program are not forced or triggered by any other module or program.

Locations are determined by a two-step process. First an initial guess is made at the epicenter. This is either the location of the station with the first P arrival, or a more sophisticated method using station tripartites. Whichever type of initial locator is used, it produces the starting point for an implementation of Geiger's location method. The location are based on either the IASPEI91 travel times (Buland and Chapman, 1983; Kennet, 1991) or the Jeffereys-Bullen travel times. The IASPEI91 P wave travel times (Kennet, 1991) table was created using the National Earthquake Information Center's software (Buland and Chapman, 1983) after conversion to C programming

language. The table contains P wave travel times at 0.5 degree increments of distance and 10km increments of depth. The Jeffereys-Bullen P wave travel time table can also be used, though the new IASPEI91 times produce better locations with data of poor azimuthal control and produce more accurate hypocenter depth solutions.

A routine was added by Dave Nyland in 2006 which provides better depth control for solutions. The quake depth is fixed to the average depth for the region (based on NEIC historical data on a one degree by one degree grid). When the location is allowed to float, the depth will be limited by the maximum depth of the region.

Program Controls:

A LOCATE screen display for an earthquake is shown in the Graphical Interface section. On this screeb capture, notice four buttons on the right side of the screen named RT P's, Disk P's, RT LP, and Disk LP. These push buttons are used to obtain data from lpproc, loc_eb/hypo_display, and ANALYZE. The RT P's button copies P-pick data, along with Mb, Ml, and Mwp magnitude information from loc_eb/hypo_display. The Disk P's button copies the same type of data from ANALYZE. The RT LP button forces long period data processing to start (or re-start) in lpproc and mm. The MS and Mw results are then copied into LOCATE. Similarly, the Disk LP button forces long period data processing in ANALYZE and the MS and Mw results are copied into LOCATE.

The three buttons below the LP buttons are used to compute hypocenter solutions for the P data listed in the P-pick/magnitude list box (left side of the screen). The Solve button forces a location using all the P data which has not been eliminated from computations. Bad Ps also produces a location given the P data, but it utilizes the bad P-pick discriminator to remove up to three unassociated picks. The Use All button forces a location using all P data in the list box including those which were previously eliminated. The Depth Control buttons are used in solving hypocentral depth. Initially the depth is set to the average for the region. The Float option is used to let the PC solve for the depth which produces the lowest average residual in the quake solution (constrained by the maximum for the region). The Enter Below option allows a geophysicist to specify a depth based on pP-P times or known information about the source zone. The Delete Station check box causes a double-clicked P-time to be removed from the list when the box is checked.

The upper-right sections of the screen gives a short summary of the last location produced by LOCATE. The large P-pick/magnitude list box lists all P times to use in the solution. After the location has been computed, residual, epicentral distance, azimuth, and magnitudes are listed for each channel. At the bottom of the box, the average of each type of magnitude is listed. (Modified averages are actually computed. That is, values significantly different from the overall average are eliminated.)

The Add P/Mag menu item calls dialog boxes which prompt for P-times and magnitude data to add to the P-pick/magnitude list box.

- Mouse buttons:
  - Left button double-click on station list eliminates it from the computation (if Delete Station check box is clicked, the P is deleted).

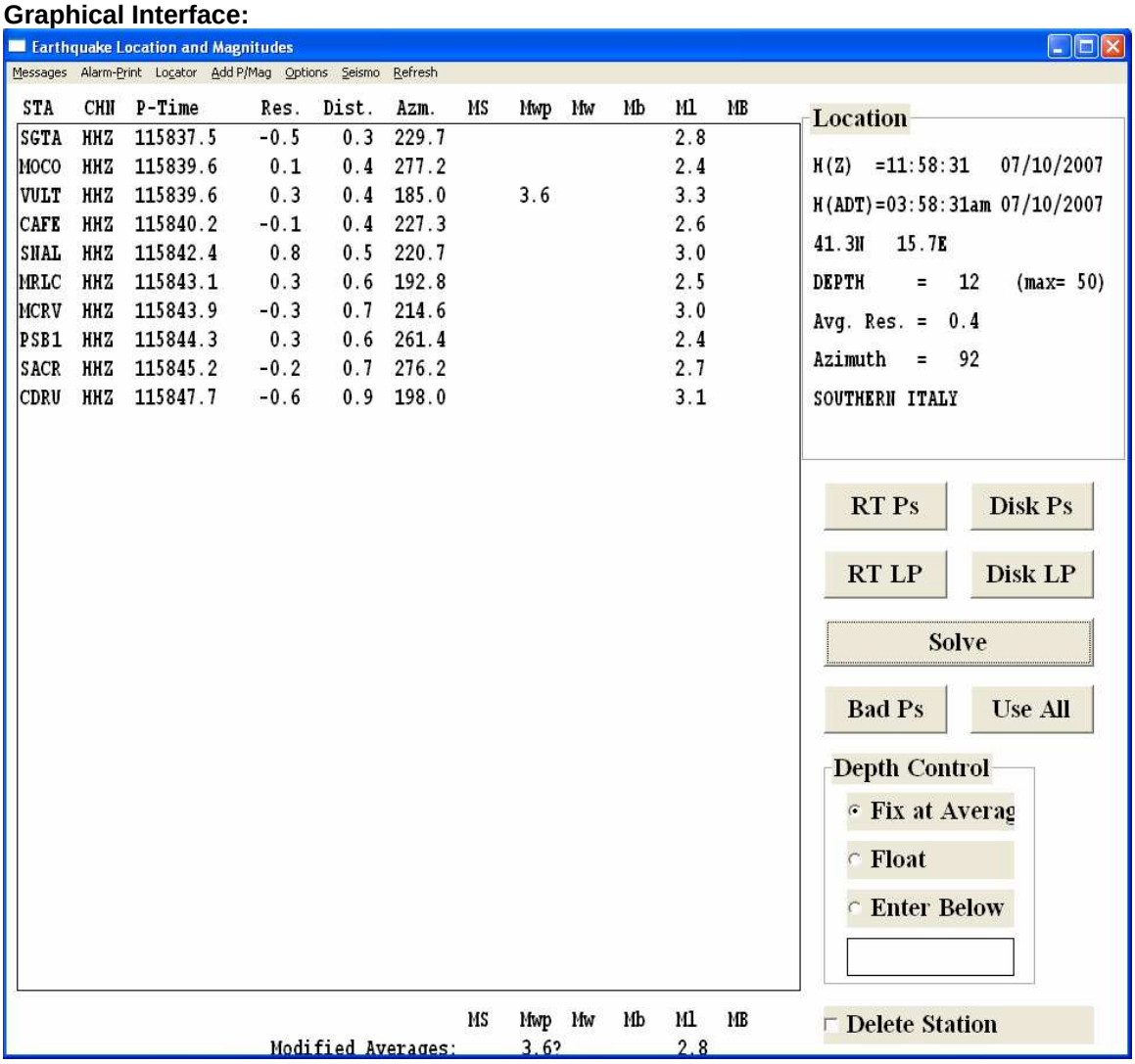
- Menu Options:
  - Messages: Call program Message2 to create bulletin.
  - Alarm Print: Send alarm form to laser printer for archive and records.
  - Locator:

    Auto Initial: (Default) First try locating earthquake using the station with the first arrival time as the initial location. If not a good solution, re-compute initial location with tripartite method.
    Nearest Station: Use location of station with earliest P-arrival as initial location.

Compute Initial: Compute the initial location using the tripartite technique.
Fix Initial: Take initial location from entry field on main screen.
Force Location: Set location for locator so that it will show residuals.
IASPEI91 Travel Times: When checked, use IASPEI91 travel times for locations.
J/B Travel Times: When checked, use Jeffereys/Bullen P-time tables to compute locations.

- Add P/Mag:
  Add P-Time: Enter a station/P-time.
  Add Magnitude: Enter period/amplitude/gain information for magnitudes.
- Options:
  Print Location: Print last location on shared printer.
  pP Depth: Compute depth from pP-P time.
  Compute Distances: Given two lat/lons, a distance between them will be output. Also, the geocentric lat/lon of point 1 will be listed.
  Voice: Turn PC voice on/off.
  Quake Summary File: Provide a file name and save the quake information to that file.
  Locate on P Copy: Turn on/off location immediately after RT P's or Disk P's chosen.
  Force Re-Process LP: Checked means that when RT LP or Disk LP are selected, the data in the appropriate program is reprocessed for MS/MB. Unchecked indicates just the MS/MB data files are copied over, the data is not re-processed. This may be necessary to uncheck when you have to get a magnitude by rubber-banding an MS or MB cycle. This rubber-banded T/A will be erased by the automatic re-processing of the LP data.
- Seismo: (Used for daily seismogram interpretations)
  Add Comments: Add felt or tsunami information to a WC/ATWC station with a P-pick.
  Save Quake: Save P-data, comments, and hypocenter data for output in daily files.
  Send Seismo: E-mail SEISMO data to NEIC and others and log day's quakes to disk.
  Delete Seismo Files: Remove daily files for start of next day's records.
- Refresh: Re-paint the screen

Graphical Interface:



3.3.3 Summary

Date Written: 1997
Development Language: MicroSoft C/C++ v7.0using WinAPI

Initialization File:
summary.ini - read on program start-up

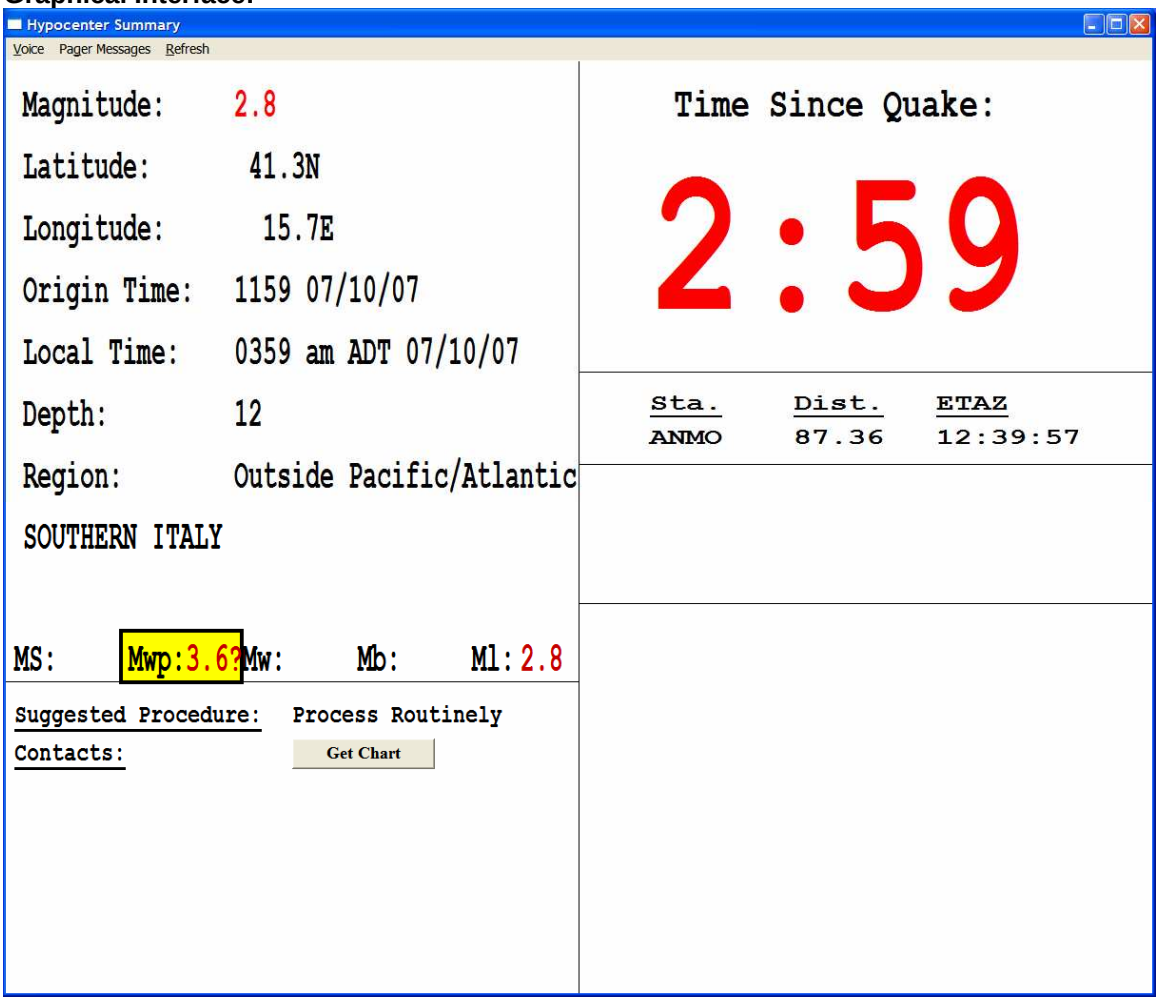
Program Description:

Program SUMMARY is a non-Earthworm-based program which runs independently of any other seismic processing program. Data files are shared between SUMMARY and other programs, but it is not dependent on any other program. Its purpose is to clearly display earthquake parameters. In addition to this, the program can generate WCATWC-specific alarms and procedures. These features of the program are not used in general distribution.
Summary monitors a file which contains the earthquake parameters from the last located earthquake. Based on these parameters, information is displayed on the screen. The **display** is divided into six sections. The upper-left quadrant lists the quake's location, magnitude, depth, and origin time. The upper-right quadrant shows the time in minutes since the earthquake's origin.

The section below that shows Rayleigh wave arrival times for nearby seismometers. The section below that is reserved for Alarm Messages (not used). The lower right section is reserved for future use. The lower-left section suggests a course-of-action and a list of contacts for this earthquake (not used). A new addition to the quadrant is the Get Chart button which calls a procedural flow chart for this quake when pressed (not used).

Program Controls:
All program controls are WCATWC specific and are not used in the general release.

Graphical Interface:



3.4 Event_notifier

Date Written: 2007
Development Language: Perl

Initialization File:
Event_notifier.ini – read on program start-up

Module Description:

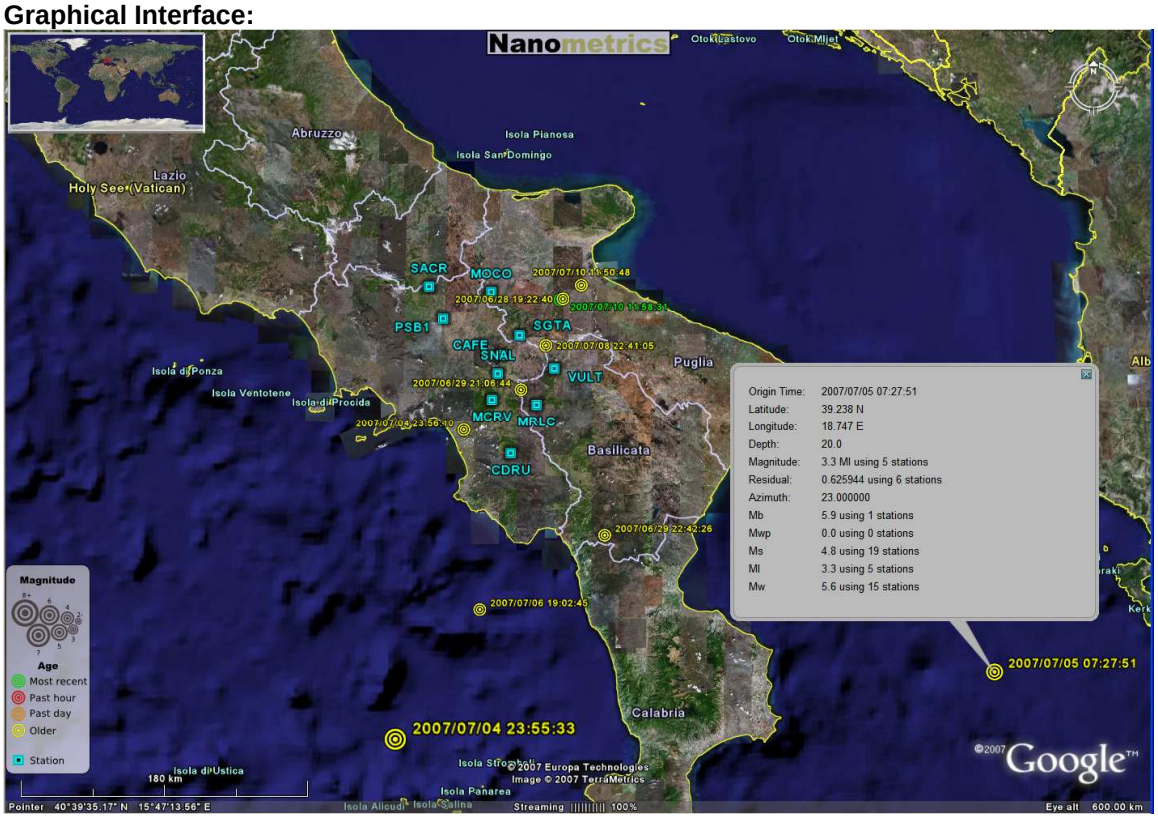
Event_notifier script was written by Nanometrics as a stand-alone module. The module can perform three tasks: display station and event locations in Google Earth, send notification emails

for all new events and generate event html bulletins. Each of these tasks can be individually enabled or disabled in the configuration file.

The script monitors the following two files (which are part of the standard Earlybird release) for changes: **station.dat** and **oldquakex.dat**. **Station.dat** file is used to obtain a list of stations to display in Google Earth. For each station, the following information is displayed (when station icon is clicked on): station name, network name, channels, geographic name, latitude, longitude and elevation. The event information is extracted from **oldquakex.dat** file. For each event, the following information is displayed (when event icon is clicked on): origin time, latitude, longitude, depth, magnitude, residual, azimuth, Mb, Mwp, Ms, Ml and Mw magnitudes as well as the number of stations used to calculate each magnitude.

The configuration file enables the user to specify the number of events to display on the map, which logos to display in Google Earth, the sizes of event icons, the notification email event magnitude threshold, event titles and whether event maps should be included in the html bulletins.

Graphical Interface:



3.5 Contributed Earthworm Modules

The modules described in this section have been created by members on the Earthworm community and are not necessarily part of the standard Earthworm software package. Additional information on operation and configuration of these modules may be obtained at the following Earthworm web site: http://folkworm.ceri.memphis.edu/ew-doc/Contributed/Contributed.htm

3.5.1 archman

Necessary Rings:
Input Ring – INPUT_RING; reads TYPE_TRACEBUF messages
Output Ring – INPUT_RING; writes TYPE_HEARTBEAT messages

Necessary Data Files:
archman.dat – State information stored between invocations

Initialization File:
archman.d

Module Description:
The archive manager ("archman") program may be used to automatically convert real-time seismic data streams into other formats for storage. Available formats include SUDS, SAC, AH, Seisan, GSE_INT and Tank. archman performs the conversions by invoking the Earthworm module waveman2disk. waveman2disk retrieves data Note that the version of archman included with the EarlyBird distribution is compatible with wave_serverV v6.2 and uses waveman2disk v6.2. Both of these are included with the distribution.

Implementation Note:
Archman retrieves data from tank files via the waveserver. Ensure that the tank files are large enough to hold the amount of data required for the archive file interval. If you find that the beginning portions of the archive files are missing, either increase the tank file size in waveserverV_v62.d or decrease the file interval size in archman.d.

Graphical Interface:
None

3.5.2 cleandir

Initialization File:
cleandir.d

Module Description:
The cleandir module may be used to automatically remove old files from select directories. This is convenient for cleaning up old log files or obsolete data files. Note that cleandir is not run as and Earthworm module and is not controlled by startstop. It must be installed and run as a Windows service. cleandir operates completely independent of Earthworm. Installation may be performed using the following steps as a user with Administrator privileges:

1. Ensure that the cleandir.d file is configured as desired. Take care to ensure this is done correctly as this service does perform automatic file deletion.

2. Select *Start -> Control Panel -> System -> Advanced -> Environment Variables* and add the following new System Variables:

Variable	Value
EW_INSTALLATION	INST_WILDCARD
EW_LOG	c:\Earthworm\run\log
EW_PARAMS	c:\Earthworm\run\params

   Click OK to close the Environment Variables window. Click OK to close the System Properties window.
3. Open a DOS prompt and create the cleandir service using the following command:

```
c:\Earthworm>sc create cleandir binPath= c:\Earthworm\v7.1\bin\cleandir.exe start= auto
```

4. Select *Start -> Control Panel -> Administrative Tools -> Services* to verify that the service has been create.
5. Double-click to open the service and verify the Startup type is Automatic.
6. Start the service.

Graphical Interface:
None

3.5.3 web_report

Necessary Rings:
Input Ring – HYPO_RING_CE; reads TYPE_HYP2000ARC messages

Initialization File:
web_report.d

Module Description:
The web_report module may be used to maintain a html page with a list of the most recent events detected by CoreEarthworm. It monitors the configured ring for TYPE_HYP2000ARC messages and updates the html files accordingly. This module is used with CoreEarthworm by default as a convenient, though limited, alternative to the Earthworm database for capturing the results of automatic processing. It is not applicable to EarlyBird. Note that the web_report magnitude field will be 0.00 because it corresponds to local magnitude, which is not calculated by CoreEarthworm.

Graphical Interface:
None

Chapter 4: Configuration

Chapter 4 describes how to configure individual EarlyBird modules for a particular network of seismic stations. The default configuration files have entries for three stations: ANMO, SDV and TORO. These three entries are only used to illustrate the format in which station parameters are to be entered, and should be replaced with actual data sources in order to avoid error messages.

EarlyBird module configuration files contain three types of configurable parameters:

1. Paths and names of data exchange and library files used for data processing
   - All library and data exchange file names as well as their paths have been configured appropriately as part of the default set of parameters. None of these settings have to be edited during the initial setup. These types of settings will only have to be changed if the names of particular files or their locations are changed.
   - Example: *RegionFile "\earthworm\run\params\regions" # Mm path correction file*
2. Data processing configuration parameters
   - Data processing configuration parameters affect the magnitude and location computations directly, and are set to defaults used by the West Coast Alaska Tsunami Warning Center (WCATWC). These parameters should be left unchanged during the initial installation and possibly fine tuned once the system has located a number of earthquakes.
   - Example: *MwpSigNoise 2.5 # Signal-to-noise ratio necessary for Mwp comps.*
3. Station-specific configuration parameters
   - All parameters that are network and station specific involve such settings as station locations, response information, sampling rates and data source modules. These parameters have to be configured before the initial start up of the system according to the instructions provided in Sections 2.1 to 2.x.
   - Example: *RequestChannel ALRB HHE PO 2002 30 0 0*

All EarlyBird (and Earthworm) configuration files are located in c:\Earthworm\run\params directory. Each module has at least two configuration files: module.d (main configuration file for the module) and module.desc (module error and restart description file monitored by Statmgr). In addition to these standard configuration files, some of the modules require configuration of additional files as specified in Sections 4.1 through 4.5.

4.1 Data Acquisition and Decimation Modules

Note: EarlyBird's picker, data storage, Ms magnitude, Mw magnitude and moment tensor modules operate on TYPE_TRACEBUF messages. Consequently, any Earthworm module capable of producing TYPE_TRACEBUF messages can be used as a data source. This section provides instructions on how to configure three data source modules that are supplied with EarlyBird installation: naqs2ew, liss2ew and slink2ew.

Naqs2ew Module

Files to edit: **naqs2ew.d**

1. Add the IP address (or the hostname) of the source NaqsServer and TCP/IP port used for the client connection as follows:
   NaqsServer 10.10.1.10 28000 # IP address and Port of NaqsServer
2. Add one line for each channel that is to be imported from NaqsServer, and processed by EarlyBird (naqs2ew.d file contains explanations for each argument on the RequestChanSCNL line):
   RequestChanSCNL STN1 HHZ NT -- 2001 30 0 0

Liss2ew Module

Files to edit: **liss2ew_staname.d** and **liss2ew_staname.desc**

1. If using liss_2ew to acquire data from the Live Internet Seismic Server (LISS), create a new module for each station that is to be used as a data source as described in Steps 2 to 8 for sample station stn1.
2. Edit liss2ew_anmo.d (default liss2ew file), and add the IP address (or the hostname) of the liss server for station STN1 to the following line (where stn1 is the name of the station and nt is the name of the network):
   LISSaddr stn1.nt.liss.org # IP address of the LISS (server)
3. Enter the name of the new module under MyModId:
   MyModId MOD_LISS2EW_STN1
4. Add the names of the STN1 channels to be receiving from with multiple AcceptSCNL commands:
   AcceptSCNL STN1 BHZ NT 00 1317
   AcceptSCNL STN1 BHN NT 00 1317
   AcceptSCNL STN1 BHE NT 00 1317
5. Save the modified file as liss_stn1.d.
6. Edit liss2ew_anmo.desc (default liss2ew description file), and change the following section (replace anmo with stn1 and INST_WILDCARD with the installation ID assigned by the Earthworm Development Team – if applicable):
   modName liss2ew_stn1
   modId MOD_LISS2EW_STN1
   instId INST_WILDCARD
7. Save the modified file as liss_stn1.desc.
8. Add the newly created module to the "Data Acquisition Modules" section in startstop_nt.d file:
   Process "liss2ew liss2ew_stn1.d"
   PriorityClass Normal
   ThreadPriority Normal
   Display NoNewConsole
9. Ensure that the Statmgr monitors the newly-created module by adding the following line to the end of the statmgr.d file:
   Descriptor liss2ew_stn1.desc

Slink2ew Module

Files to edit: **slink2ew.d**

1. Add the IP address (or the hostname) of the data source SeedLink server:
   SLhost 128.95.166.130 # Host address of the SeedLink server
   SLport 18000 # Port number of the SeedLink server
2. Add a list of stations and channels to connect to using the Stream command (slink2ew.d file contains explanations on the format of the Stream command arguments):
   Stream NT_STN1 "00BHZ.D"

Decimate Module

Note: Standard sampling rate for EarlyBird short period data processing is 20 samples/s. Data sampled at higher sampling rates has to be decimated. Default EarlyBird configuration includes two data decimation modules: decimate_100to20 and decimate_20to1. Decimate_100to20 is used to decimate data sampled at 100 sps down to 20 sps for EarlyBird processing. As some of the EarlyBird modules (mm, lpproc and mtinver) need to process large amounts of long period data, 20 sps data is further decimated down to 1 sps with decimate_20to1. Only data from broadband stations should be decimated down to 1 sps for processing by mm, lpproc and mtinver. The user can create other decimate modules for input data sampling rates other than 20 or 100 samples per second.

Files to edit: decimate_100to20.d and decimate_20to1.d

1. Add the station and the channel name that is to be decimated to the Input SCNL/Output SCNL table at the bottom of the decimate_100to20.d file (if applicable):
   GetSCNL STN1 BHZ TN -- STN1 BHZ TN --
2. Add the station and the channel name that is to be decimated to the Input SCNL/Output SCNL table at the bottom of the decimate_20to1.d file (if applicable):
   GetSCNL STN1 BHZ TN -- STN1 BHZ TN –
3. Generate any additional decimate_XtoY modules (if applicable) by following Steps 2 to 8 from the Liss2ew Module section. In the outlined procedure, replace liss2ew_anmo with decimate_XtoY and replace liss2ew.d configuration parameters with decimate.d configuration parameters.

Ringdup_scn Module

Files to edit: ringdup_scn.d

1. Add Send_scnl commands for each channel that is to be transferred without decimation to the bottom of the ringdup_scn.d file (if applicable):
   *Send_scnl ANMO BHZ IU **

4.2 Data Storage Modules

Disk_eb (disk_wcatwc) Module

Note: Station.dat file should contain all stations whose data will be processed by EarlyBird modules including the stations not sending real time data. The stations have to be listed in alphabetical order. Pick_eb.sta file should only contain stations that are sending real time data and are processed by the (near) real-time analysis component of the system.

Files to edit: **pick_eb.sta**, **station.dat** and **disk_eb.d**

1. Edit pick_eb.sta and add a new line for each channel whose traces are to be written to disk files. Configure the attributes for each channel as per attribute descriptions provided below:
   STN1 BHZ TN 1 1 5 3 0.000012 20.0 0.333 1 20.0 0.1

   The following are explanations for all attributes:
   - Column 1 (Station) – station code
   - Column 2 (Channel) – channel identifier
   - Column 3 (Network) – network identifier
   - Column 4 (Pick Status) – analyze data from this channel for picks (1) or not (0)
   - Column 5 (Filter) – filter data from this channel before picking (1) or not (0)
   - Column 6 (Signal:Noise) – signal to noise ratio that must be exceeded in order for station to trigger
   - Column 7 (Alarm) – output alarms to pager (1 – not supported), output alarms to speaker (2), output alarm to both (3) or no alarm (0)
   - Column 8 (Alarm Amplitude) – amplitude to be exceeded for SP (short period) alarm
   - Column 9 (Alarm Duration) – duration of the signal with amplitude from Column 8 for SP alarm activation
   - Column 10 (Alarm Min. Frequency) – min frequency for SP alarm activation
   - Column 11 (Compute Mwp) – compute Mwp magnitude for this channel (1) or not (0)
   - Column 12 (Sample Rate) – sampling rate of the incoming data for this channel
   - Column 13 (Screen Scaling) –parameter which determines screen scaling for the waveforms displayed in develo module

2. Edit station.dat and add a new line for each channel whose traces are to be written to disk files. Configure the attributes for each channel as per attribute descriptions provided below:
   STN1 BHZ TN 924400000.000000 0.000000 34.946100 -106.456600 1740.000000 4194304.000000 0.000000 6 5 Albuquerque, New Mexico

   The following are explanations for all attributes:
   - Column 1 (ANMO) – station name
   - Column 2 (BHZ) – channel name
   - Column 3 (IU) – network name
   - Column 4 (924400000.000000) – channel gain (in $^{counts}/_{m/s}$ – S_{GV} in Appendix D3)
   - Column 5 (0.000000) – gain calibration (can be left as 0)
   - Column 6 (34.946100) – station location latitude (east positive)
   - Column 7 (-106.456600) – station location longitude (north positive)
   - Column 8 (1740.000000) – station elevation
   - Column 9 (4194304.000000) – clip level (in counts)
   - Column 10 (0.000000) – time correction (delay in seconds)
   - Column 11 (6) – station type (see Appendix D1 for a list)
   - Column 12 (5) – agency running the station (see Appendix D2 for list) – not used
   - Column 13 (Albuquerque, New Mexico) – geographical name of the station

3. Edit disk_eb.d file and modify the CircDeleteHoursd parameter. This parameter triggers a thread that which will delete data files older than the configured number of hours. If one does not wish to delete any data and keep recording, set this parameter to 0.
   CircDeleteHours 420 #

Disk_eb_lp Module

Note: If the station.dat file contains a complete list of data source channels, no changes to this file have to be made for any subsequently listed modules (including disk_eb_lp).

Files to edit: **pick_eb_lp.sta** and **disk_eb_lp.d**

1. Edit pick_eb_lp.sta, and add a new line for each broadband and long period seismometer trace that is decimated down to 1 sps and processed by mm, lpproc and mtinver modules. Pick_eb_lp.sta file has the same format as pick_eb.sta file with alarm settings (Columns 8, 9 and 10 in Step 1 of pick_eb.sta configuration procedure) applying to LP alarms instead of SP, and sample rate (Column 12) being 1.0 instead of 20.0.
2. Edit disk_eb_lp.d file and modify the CircDeleteHours parameter. This parameter triggers a thread that which will delete LP data files older than the configured number of hours. If you do not wish to delete any data and keep recording, set this parameter to 0.
   CircDeleteHours 420 #

4.3 Data Processing Modules

Develo Module

Files to edit: **develo.d** and **screendisp.ini**

1. Edit develo.d file and modify the number of channels to be displayed on one screen. This number should usually match the number of channels being processed and may also depend on the size of the display.
   NumTracePerScreen 3 # Number of traces to show on visible screen
   # (the rest will scroll)
2. Edit screendisp.ini file to group stations by their geographical location or other criteria (such as the agencies operating the stations). Screendisp.ini file enables the user to display trace data from pre-defined clusters of stations in develo. Alternatively, if the

number of data stations is not large, one can define only one station screen named All Stations. First, define total number of station screens.
Number of Station Screens: 3

3. For each station screen, define the name of the screen and then list all station SCNLs followed by arguments specifying which screens that particular station is to appear on. The number of columns of 1s and 0s has to match the number trace screens.
Screen Name: All Stations
Screen Name: North America
Screen Name: Central America
ALFO HHZ PO 1 1 0
ALRB HHZ PO 1 1 0
BANO HHZ PO 1 0 1
BATG HHZ PO 1 0 1

Pick_eb (pick_wcatwc) Module

Files to edit: **alarm.dat**

1. Edit alarm.dat file and group stations according to their geographical position. Alarm.dat is used by pick_eb module to sound alarms when more than a configured number of stations within a configured region trigger. Start by defining the number of multiple trigger regions.
iNumRegions 1 # This must be first parameter in file
2. For each region, define the following: region name, number of stations to trigger in order to generate an alarm, the signal to noise ratio to be exceeded for the pick to be valid, the maximum time between the first and the last pick (to ensure that the picks are part of the same event) and the total number of stations in the region.
Region "Ontario" # Regional name
AlarmThresh 2 # No. stations needed for alarm
PStrength 1. # S:N to exceed for P-pick
MaxTime 200. # Max time (s) between first and last alarm stn.
NumStnInReg 5 # Number of stations in this region
Station ANMO
Station TORO
Station PICO
Station WLMO
Station ALFO

Loc_eb (loc_wcatwc) Module

Files to edit: None

Hypo_display Module

Files to edit: **hypo_display.d**

1. Edit hypo_display.d and configure the duration of trace data to be displayed prior to the P pick for located earthquakes.
PreEventTime 20. # Time (seconds) to display before P
2. Modify the number of traces to display on the visible screen. The rest of the traces will scroll.
NumTracePerScreen 5 # Number of traces to show on visible screen
3. Modify the total duration of data to be displayed as part of the solution in Hypo_display GUI.
TimePerScreen 150. # Seconds of data to display on screen

Lpproc Module

Files to edit: lpproc.d

1. Edit lpproc.d and specify the number of traces to be displayed on the visible screen.
   NumTracePerScreen 3 # Number of traces to show on visible screen
2. Specify the duration of the long period trace data to be displayed on the visible screen.
   TimePerScreen 720. # Seconds of data to display on screen

Mtinver Module

Note: In most cases, the response information required by the Calibs file can be obtained by sending an email to autodrm@gldfs.cr.usgs.gov (USGS AutoDRM service) in the following format:

```
BEGIN GSE2.0
MSG_TYPE REQUEST
MSG_ID  id_0001 Data Request
E-MAIL emailaddress@xyz.com
AUX_LIST PO
STA_LIST ALFO
CHAN_LIST HHZ
RESPONSE GSE2.0
STOP
```

In this example, the response information for channel HHZ from station ALFO that is part of the network PO in GSE2.0 format is requested. The response received should be almost identical to the one listed in step 1 below, with the following modifications: insert 1 at the end of the first line and insert a new line below the line that begins with CAL2 containing latitude, longitude and elevation of the station.

Files to edit: **Calibs**

1. Edit Calibs to configure response information for each data source channel. All channels have to be listed in alphabetical order. The following is an example of the channel response format, together with clarifications for each attribute:

```
CAL2 STN1  BHZ NT   STS-1    1.45E-01   1.000   20.00000 2004/03/11 00:00 1
  8.883870 -70.634050 1619
PAZ2  1 C  4.16022389E+03          4   3 STS-1
 -1.23399999E-02  1.23399999E-02
 -1.23399999E-02 -1.23399999E-02
 -3.91800003E+01  4.91199989E+01
 -3.91800003E+01 -4.91199989E+01
  0.00000000E+00  0.00000000E+00
  0.00000000E+00  0.00000000E+00
  0.00000000E+00  0.00000000E+00
```

   Row 1
   - Column 1 (CAL2) – leave as default
   - Column 2 (STN1) – station name
   - Column 3 (BHZ) – channel name
   - Column 4 (NT) – network name
   - Column 5 (STS-1) – seismometer type
   - Column 6 (1.45E-01) – channel sensitivity (in $^{nm}/_{count}$ - S_{SysD} in Appendix D3)
   - Column 7 (1.000) – calibration period (where seismometer velocity response is flat)
   - Column 8 (20.00000) – sample rate (of the processed data, not necessarily the original sampling rate)
   - Column 9 (2004/03/11) – date when the current response became valid
   - Column 10 (00:00) – time when the current response became valid
   - Column 11 (1) – process data from this station (1) or not (0)

   Row 2
   - Column 1 (8.883870) – station latitude (N is positive)
   - Column 2 (-70.634050) – station longitude (E is positive)
   - Column 3 (1619) – station elevation

   Row 3

- Column 1 (PAZ2) – leave as default
- Column 2 (1) – leave as default
- Column 3 (C) – leave as default
- Column 4 (4.16022389E+03) – normalization factor (in displacement – A_{0D} in Appendix D3)
- Column 5 (4) – number of poles in seismometer response
- Column 6 (3) – number of zeros in seismometer response*****
- Column 7 (STS-1) – seismometer name

***** Seismometer response is to be specified in displacement so in order to convert from velocity response to displacement response, add one zero at 0(Re) 0(Im). For acceleration, add two zeros at 0 (Re) 0(Im).

Row 4 – Row 10

- Seismometer response poles and zeros. Poles are listed first, followed by zeros. The first column represents real part and the second column imaginary part.

Mm Module
Files to edit: None

4.4 Post Processing Standalone Programs

Analyze

Note: Analyze.ini is the main configuration file for Analyze software. This configuration file contains paths to various files used by Analyze. Unless the locations of some of these files have changed, there is no need to edit analyze.ini.

Files to edit: **screen.ini**

1. Edit screen.ini and enter the number of station screens to be listed under Control Panel in Analyze. The stations are usually divided into clusters according to their geographical locations.
   Number of Station Screens: 1
2. Define each cluster of stations screen by giving it a name and defining its boundaries in latitude and longitude.
   Screen Name: Core Stations
   S. Lat. (+/-): -20.
   N. Lat. (+/-): 20.
   W. Lon. (+): 120.
   E. Lon. (+): 130.
3. For each station screen, list all station SCNLs followed by arguments specifying which screens (clusters) that particular station is part of. The number of columns of 1s and 0s has to match the number station screens (clusters).
   ALFO HHZ PO 1 1 0
   ALRB HHZ PO 1 1 0
   BANO HHZ PO 1 0 1
   BATG HHZ PO 1 0 1

Locate

Files to edit: **locate.ini**

1. Enter the printer that the workstation is connected to for earthquake summary printing and alarm printing (if any).
   Printer port assignment (PRN, COMx, \\cpu\printername, or NONE): NONE
   Alarm Printer (PRN, COMx, \\cpu\printername, or NONE): NONE

Summary

Files to edit: None

4.5 Event_notifier Script

Files to edit: event_notifier.ini

1. Enter the subject line for event notification emails.
   mail_subject = Earlybird Event $EVENT_ID$ - $MAGNITUDE$ M$MAGNITUDE_TYPE$ @ $ORIGIN_TIME$
2. Enter the event magnitude threshold to trigger notification emails.
   notification_magnitude = 5.0
3. Modify the mail server to be used and enter sender and recipient email addresses. Multiple recipient emails can be separated by commas.
   mail_server = mail.xyz.com
   mail_sender = sender@xyz.com
   mail_recipient = recipient@xyz.com
4. Modify the name of the logo file to be displayed in the upper portion of Google Earth map display screen (default nmx.png displays Nanometrics logo).
   logo_file_name=nmx.png
5. Modify the amount of time event_notifier waits for the new event before refreshing the existing Google Earth display.
   refresh_period = 30
6. Modify the number of events to be displayed on the map (maximum is 40).
   maximum_map_events = 5
7. Modify event icon sizes. The sizes listed in the configuration file correspond to earthquake magnitudes 0-10.
   event_icon_sizes=0.75, 0.75, 0.75, 1.0, 1.25, 1.50, 1.75, 2.0, 2.25, 2.5, 2.5
8. Modify the altitude of the initial view of the current event.
   initial_view_altitude=600
9. Enter whether you wish to include event maps in html bulletins.
   write_bulletins = 1

Chapter 5: Operation

5.1 Start up Earlybird

Earlybird programs are invoked using five batch files located in Earthworm directory: **EB.bat, CE.bat, EB_and_CE.bat, EBAnalysis and EB_EN.bat. EB.bat** is used to start up Summary stand-alone program and all real-time data processing EarlyBird modules (Figure 5). **CE.bat** is used to start up CoreEarthworm real-time data processing modules (Figure 6). **EB_and_CE.bat** is used to start up both EarlyBird and CoreEarthworm real-time data processing modules in parallel (Figure 7). **EBAnalysis.bat** is used to start up Earlybird Analyze and Locate stand-alone programs (Figure 8). **EB_EN.bat** is used to start up Event_notifier script (Figure 9).

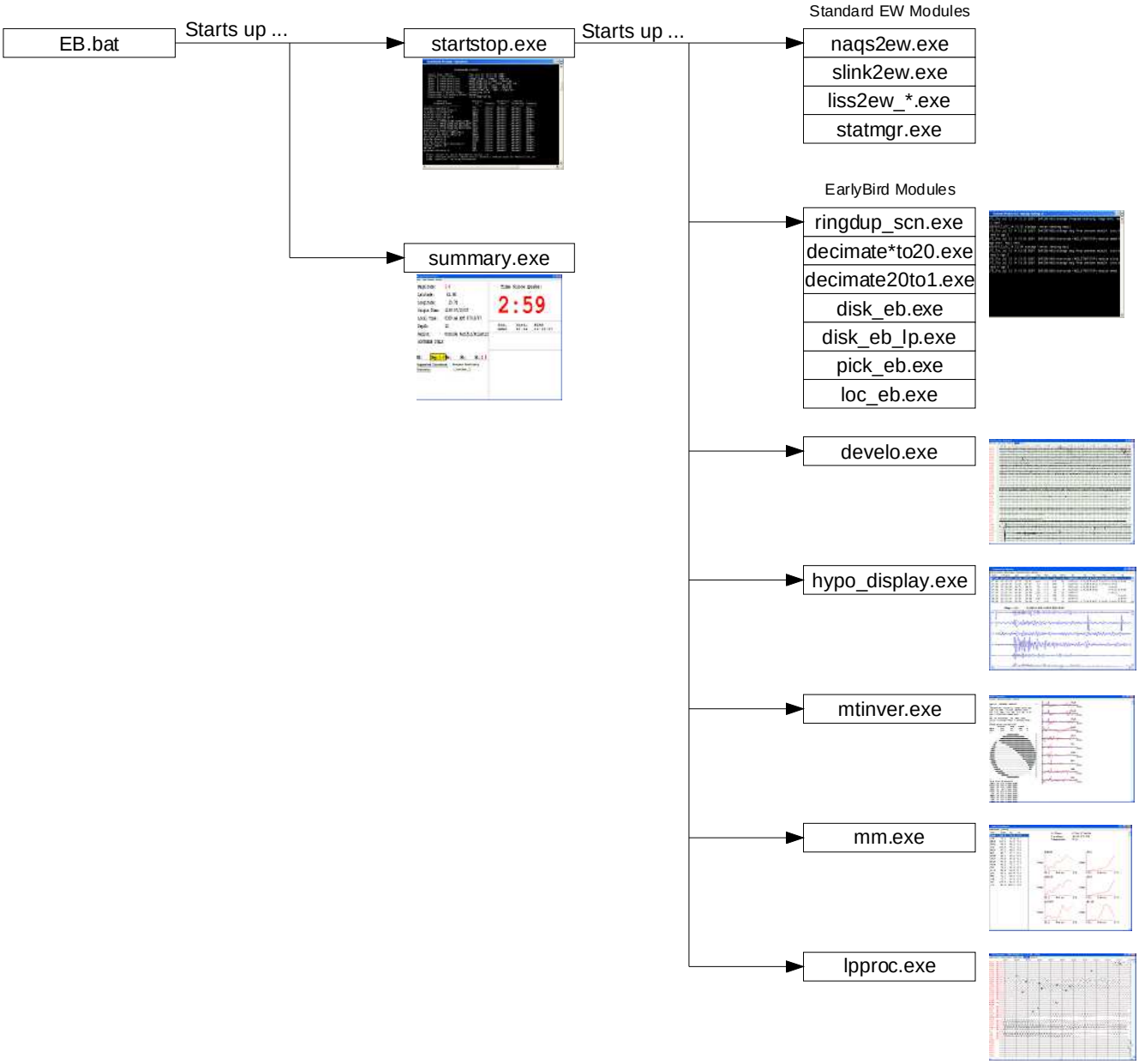


Figure 5: EB Batch File

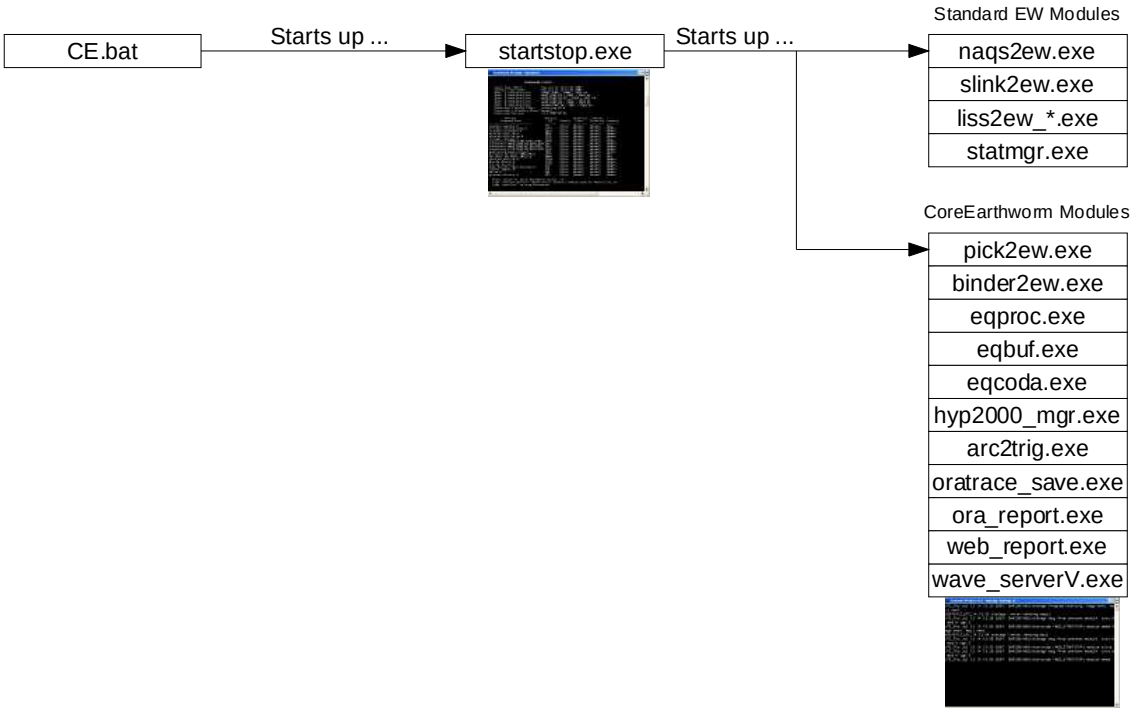


Figure 6: CE Batch File

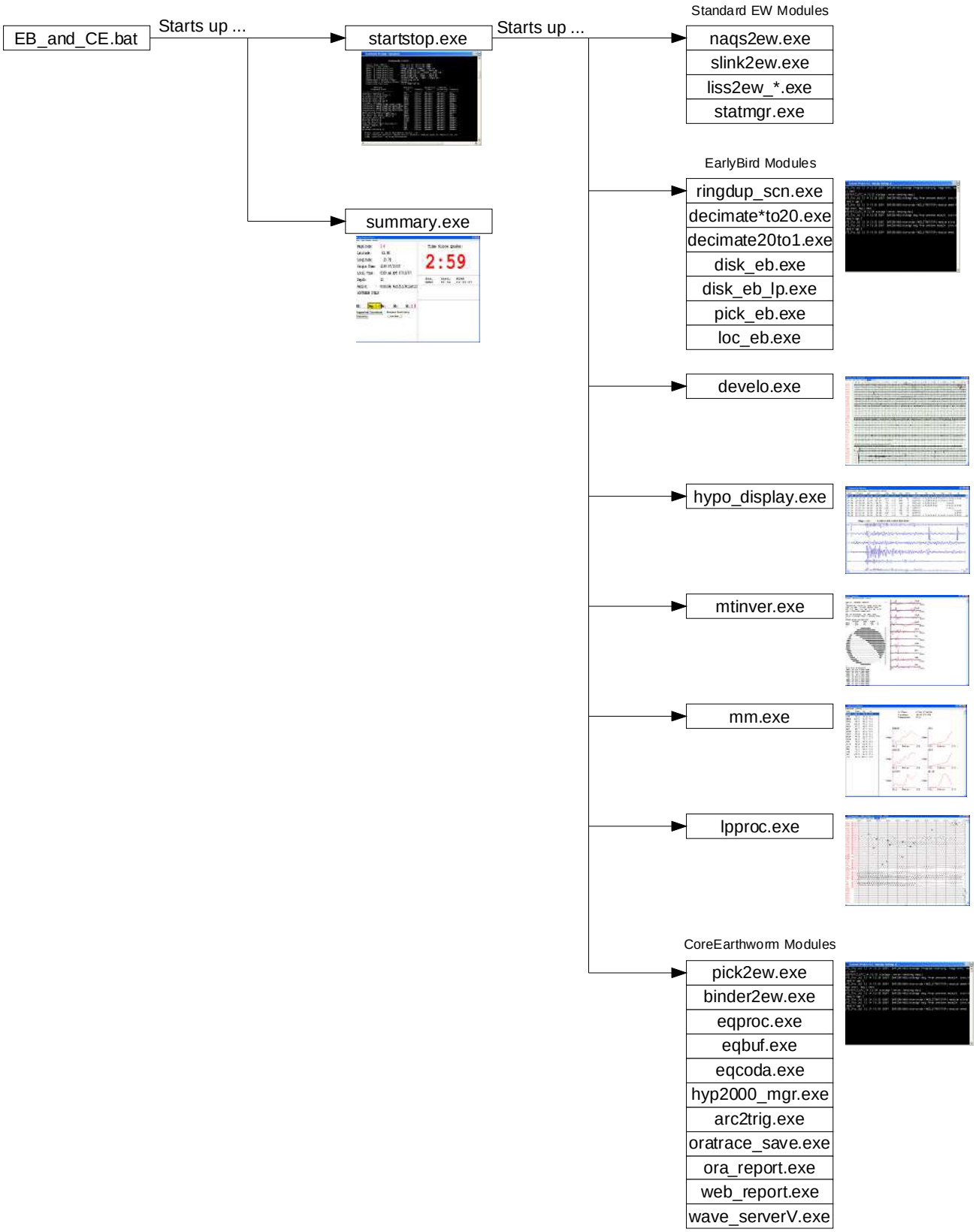


Figure 7: EB_and_CE Batch File

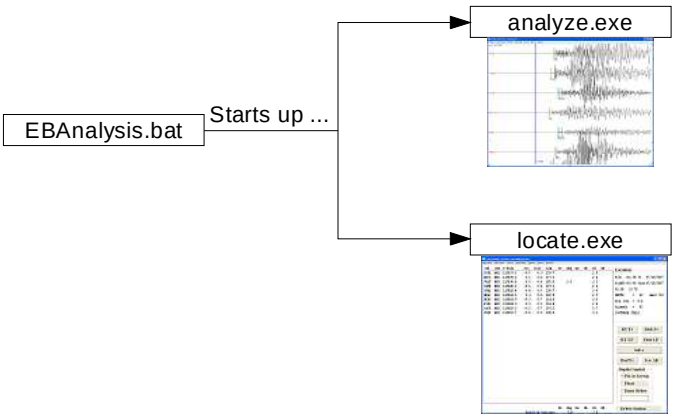


Figure 8: EBAnalysis Batch File

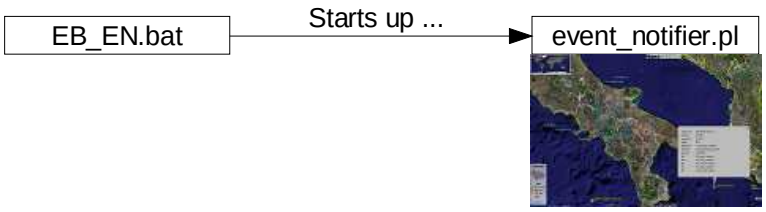


Figure 9: EB_EN Batch File

As shown in Figures 5 through 9, the modules naqs2ew, slink2ew, liss2ew, statmgr, ringdup_scn, decimate*to20, decimate20to1, disk_eb, disk_eb_lp, pick_eb (pick_wcatwc), loc_eb (loc_wcatwc), pick2ew, binder2ew, eqproc, eqbuf, eqcoda, hyp2000_mgr, ora_trace_save, ora_report, web_report and waveserverV modules do not have graphical user interfaces. Some of these modules have their own console windows and the others output their log messages to the startstop console window.

Note: Before starting EarlyBird:

1. Ensure that all software was installed as described in Chapter 2: Installation.
2. Ensure that all software was configured as explained in Chapter 4: Configuration.
3. Ensure that the host workstation has access to Internet and seismic data sources.
4. Ensure that the system clock is synchronized with UTC and that UTC time is used instead of local time.

To start Earlybird:

1. Open a command window (startstop window).
2. Type **EB.bat** (in any directory) to start EarlyBird modules.
   OR
   Type **CE.bat** (in any directory) to start CoreEarthworm modules.
   OR
   Type **EB_and_CE.bat** (in any directory) to start EarlyBird and CoreEarthworm modules.
3. Open a second command window.
4. Type **EB_EN.bat** to start Event_notifier (only applicable if EarlyBird is running).
5. Open a third command window.

6. Type **EBAnalysis.bat** to start Analyze and Locate programs (only applicable if EarlyBird is running).

5.1.1 Check Earlybird and CoreEarthworm Startup

To check that all EarlyBird components started properly:

1. Go to Command Prompt window marked EB, CE or EB_and_CE and hit Enter. The summary of all the memory rings that have been created as well as all of the modules that are supposed to be running should be displayed as shown in Figure 10.

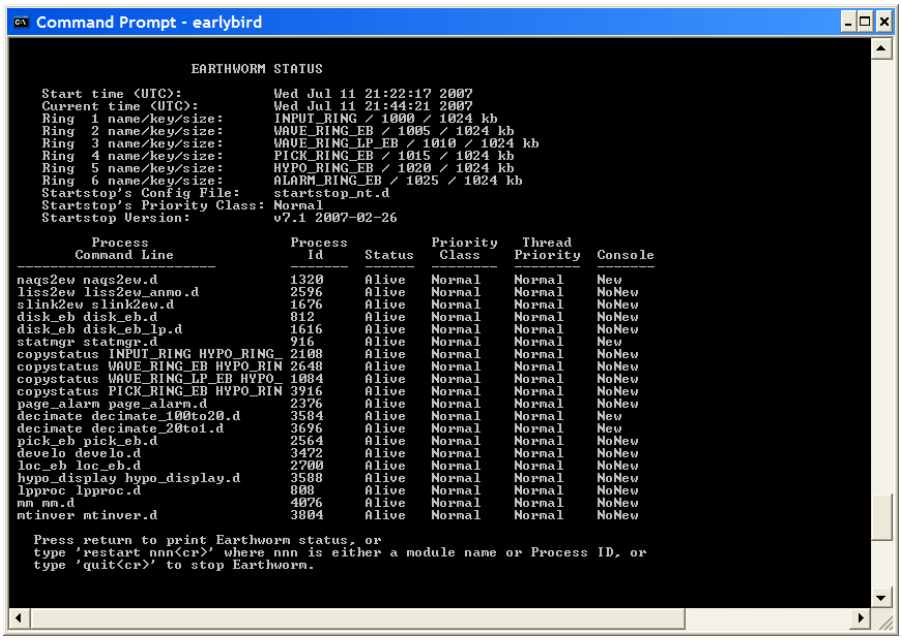


Figure 10: Startstop Module Summary Window

2. Check that the Status column indicates **Alive** for all Earthworm-based modules that are started with startstop module. If the Status column indicates **Dead** for any particular module, check its log file (**/Earthworm/run/log**) to investigate why the module did not start properly.
3. Check that there is a window named Hypocenter Summary. This window represents Summary stand-alone program which is independent of all the Earthworm-based modules.

Note: Steps 4 and 5 are applicable only if EB_EN file was invoked.

4. Check that a Command Window named c:\Perl\bin\perl.exe exists (see Figure 11) and that no error messages are displayed in it. This window shows that Event_notifier script is running. This script is independent of all Earthworm-based modules controlled by startstop.

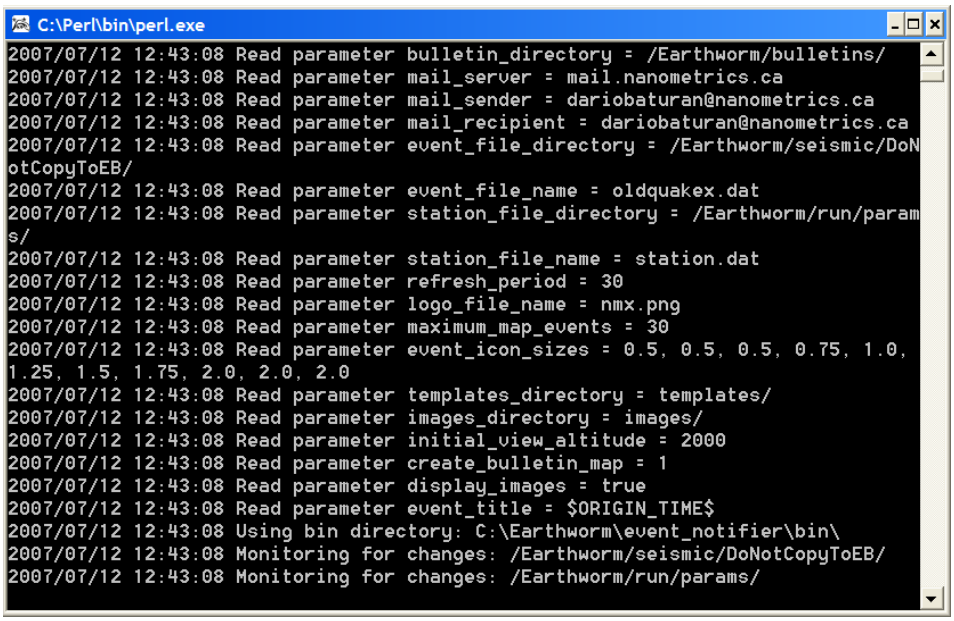


Figure 11: Event_notifier Script Window

5. Check that Google Earth has been started and that it is displaying all station and event locations. Alternatively, the display may be centered on Ottawa, Canada by default, the first time the script is started. To ensure that **station.dat** and **oldquakex.dat** have been properly read by Event Notifier, open either one of the files and save it without making any changes. Event Notifier should interpret this as a change to either station or event location and center the display on the location of the last recorded event.

Note: Steps 6 and 7 are applicable only if EBAnalysis.bat file was invoked.

6. Check that there is a window named Analyze Disk Data. This window indicates that Analyze program was started successfully.
7. Check that there is a window named Earthquake Location and Magnitudes. This window indicates that Locate program was started successfully.

5.2 Stopping EarlyBird Programs

To stop all programs started with **EB, CE, EB_and_CE, EBAnalysis** and **EB_EN** batch files:

1. Type quit in the startstop window shown in Figure 10. Startstop will shutdown all Earthworm-based modules within 60 seconds and release the shared memory regions being used for message rings.
2. Manually close HypoCenter Summary window.

Note: Steps 3 and 4 are applicable only if EB_EN.bat file was invoked.

3. Manually close perl.exe (Event_notifier) window shown in Figure 6.
4. Manually close Google Earth. Select No when prompted to save unsaved items to “My Places” folder.

Note: Steps 5 and 6 are applicable only if EBAnalysis.bat file was invoked.

5. Manually close Analyze window.
6. Manually close Locate window.

5.2.1 Stopping and Restarting Individual Earthworm-based Modules

All EarlyBird Earthworm-based modules that are started with startstop, read their configuration files on start up. In order for changes made to individual module configuration files to take effect, affected modules have to be restarted.

To stop and restart individual modules without shutting down the rest of the EarlyBird programs:

1. In startstop window (Figure 10), type **stop pid**. Pid is the process id of the module you are trying to stop as listed in Process Id column (see Figure 5). Wait for at least 30 seconds.
2. Hit Enter in startstop window and the Status column for the process that was to be stopped should read Stop instead of Alive.
3. In order to restart the module that was just stopped, type **restart Pid** and hit Enter.

5.2.2 Starting Analyze, Locate and Summary Programs Individually

If any changes are made to Analyze, Locate, Summary or Event_notifier configuration files, these programs will have to be restarted individually. This is in order for changes to take effect and so that other system components continue running uninterrupted.

To start Analyze individually:

1. Open a Command Prompt and go to /Earthworm/seismic/analyze.
2. Type ew_nt. This will set up environment variables.
3. Type Analyze.

To start Locate individually:

1. Open a Command Prompt and go to /Earthworm/seismic/locate.
2. Type ew_nt.
3. Type Locate.

To start Summary individually:

1. Open a Command Prompt and go to /Earthworm/seismic/messages.
2. Type ew_nt.
3. Type Summary.

5.3 Adding Earthworm Modules

To add other Earthworm modules to Earthworm-based part of EarlyBird software:

1. Ensure that the executable file for the new module has been copied to **/Earthworm/v7.1/bin** directory.
2. Ensure that the configuration files (module.d and module.desc) of the new module have been copied to **/Earthworm/run/params** directory.
3. Edit **earthworm.d** file in **/Earthworm/run/params**, and add the name (in MOD_MODULE_NAME format) of the new module to the list of Module Ids. If necessary, add any new message types (in TYPE_MESSAGE format), used by the new module, to Message Types list.
4. Edit the appropriate startstop_nt_*.d files (depending on whether or not you are running EarlyBird, CoreEarthworm or both) and add a section that will tell startstop to start the new module in the following format:
   ```
   Process "module_name module.d"
   PriorityClass       Normal
   ```

```
ThreadPriority      Normal
Display             NewConsole
```

5. Edit the appropriate statmgr_*.d files (depending on whether or not you are running EarlyBird, CoreEarthworm or both) and add an entry at the end of the file for the *.desc file for the new module:
   Module.desc
6. Type **reconfigure** in startstop window. This command will cause startstop module to re-read its configuration file and start up any newly listed modules without stopping the existing installation.
7. Check that the new module has been added to the list of running modules by hitting Enter in startstop window.

Appendix A: WCATWC Data Format

Figure A1 illustrates the byte stream of a typical WC/ATWC data file. The WC/ATWC file structure includes a disk header, a channel header block, and a data block. As shown in Table A2, the disk header consists of the first 24 bytes of the file. This data is immediately followed by a series of N 200-byte channel headers. The specific information contained in the channel header is listed in table A1. The data block follows the last channel header and the exact number of bytes in each sequential data channel is determined from the information provided in the corresponding channel header.
A quick way to obtain disk and channel header information is to simply perform a hexadecimal data dump of the file. This is illustrated in Figure A2. The bytes are in little endian format and should be read from right to left. For example, the hexadecimal value for the year is found in bytes 1-2 of the disk header. When converting the hexadecimal value to a decimal number, the value D5 07 should be read as 07 D5.

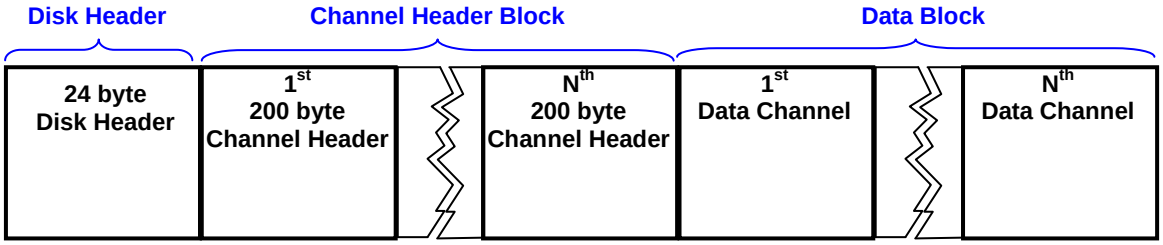


Figure A1: Byte stream structure for a WC/ATWC data file.

CHANNEL HEADER

Byte	Data Type	# Bytes	Parameter	C Structure	Example
1-6	char	6	szStation[6]	CHNLHEADER	BILL
7-12	char	6	szChannel[6]	CHNLHEADER	BHZ
13-16	char	4	szNetID[4]	CHNLHEADER	IU
17-32	SYSTEMTIME	16	stStartTime	CHNLHEADER	As in disk header
33-40	double	8	dSampRate	CHNLHEADER	20
41-44	long	4	lNumSamps	CHNLHEADER	2400
45-48	int	4	iBytePerSamp	CHNLHEADER	4
49-52	int	4	iTrigger	CHNLHEADER	0 or 1
53-56	int	4	iSignalToNoise	CHNLHEADER	5
57-60	int	4	iPickStatus	CHNLHEADER	1
61-64	int	4	iStationType	CHNLHEADER	1
65-72	double	8	dLat	CHNLHEADER	68.0651
73-80	double	8	dLong	CHNLHEADER	166.4524
81-88	double	8	dElevation	CHNLHEADER	299.0
89-96	double	8	dGain	CHNLHEADER	1027600000.0
97-104	double	8	dGainCalibration	CHNLHEADER	0
105-112	double	8	dClipLevel	CHNLHEADER	4194304.0
113-120	double	8	dTimeCorrection	CHNLHEADER	0.0
121-128	double	8	dScaleFactor	CHNLHEADER	0.03
129-132	long	4	lUnused1	CHNLHEADER	NA
133-136	int	4	iUnused1	CHNLHEADER	NA
137-140	int	4	iUnused2	CHNLHEADER	NA
141-148	double	8	dUnused1	CHNLHEADER	NA
149-156	double	8	dUnused2	CHNLHEADER	NA
157-164	double	8	dUnused3	CHNLHEADER	NA
165-196	char	32	szUnused1[32]	CHNLHEADER	NA

Table A1: The channel header format used at the WCATWC

DISK HEADER					
Byte	**Data Type**	**# Bytes**	**Parameter**	**C Structure**	**Example**
1-2	2 byte integer	2	wYear	SYSTEMTIME	1980-2099
3-4	2 byte integer	2	wMonth	SYSTEMTIME	1-12
5-6	2 byte integer	2	wDayOfWeek	SYSTEMTIME	0-6
7-8	2 byte integer	2	wDay	SYSTEMTIME	1-31
9-10	2 byte integer	2	wHour	SYSTEMTIME	0-23
11-12	2 byte integer	2	wMinute	SYSTEMTIME	0-59
13-14	2 byte integer	2	wSecond	SYSTEMTIME	0-60
15-16	2 byte integer	2	wMilliseconds	SYSTEMTIME	0-999
17-20	4 byte integer	4	iNumChans	DISKHEADER	currently set at 200
21-24	4 byte integer	4	iChanSize	DISKHEADER	currently set at 200

Table A2: The disk header format used at WCATWC

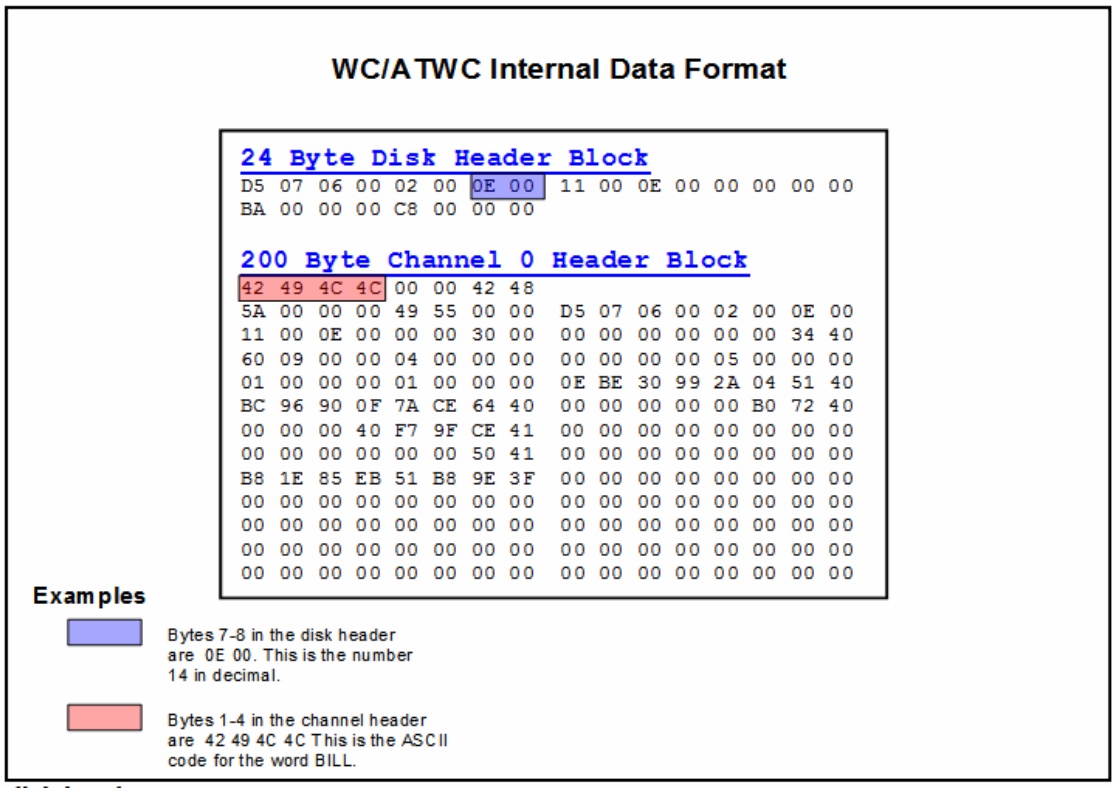
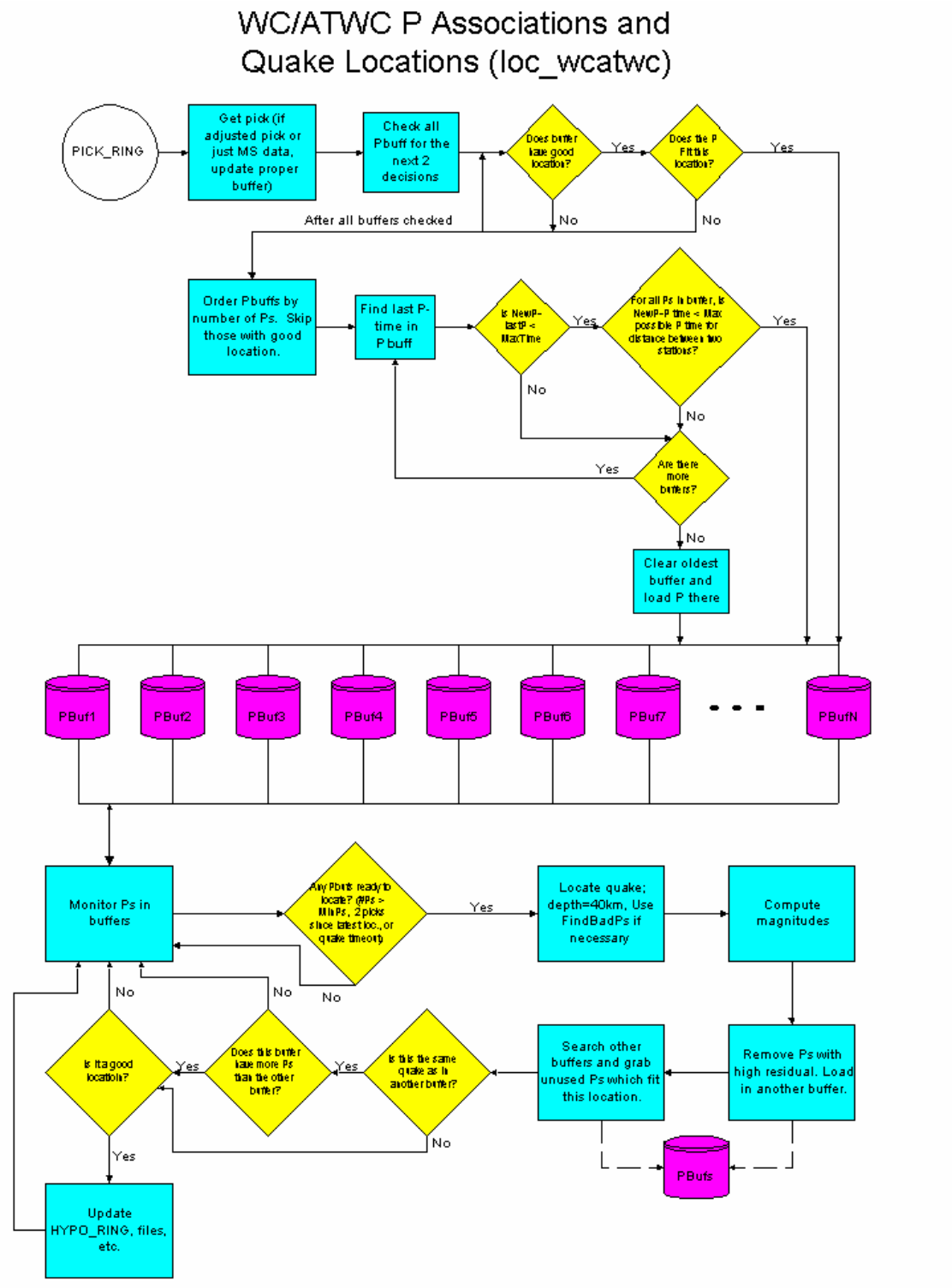


Figure A2: Hexadecimal data dump of a typical channel and disk header

Appendix B: Loc_eb (Loc_wcatwc) Association and Location Flowchart



Appendix C: EarlyBird P-picker Algorithm

The P-pick algorithm in use at the WC/ATWC was developed by Veith (1978), and the initial implementation was described by Sokolowski, et al. (1983). Figure C1 shows a flowchart for the P-picker routine. In general, the signal must pass through three phases before a pick is declared. The first stage is a simple test which looks for higher than normal signal amplitude. A running average of the accumulated amplitude differences between the samples (MDF) is updated every LTASeconds. The average MDF over LTASeconds is averaged with the previous average MDF to obtain the new average MDF. If the present MDF is greater than half the specified signal-to-noise ratio for the station times the average MDF (the trigger threshold), phase 1 is passed. The time of this sample is saved as the P-time in case all phases pass.

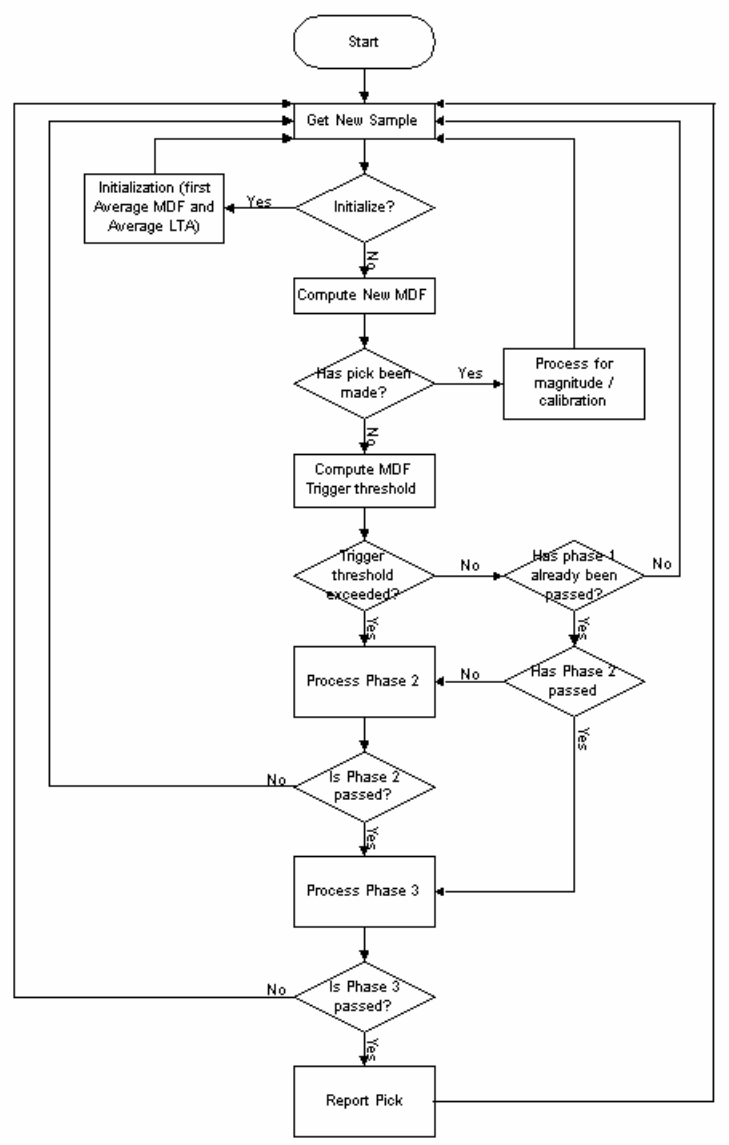


Figure C1: Flowchart for P-picker Algorithm

The three tests of phase 2 are all conducted simultaneously. Test 1 requires energetic signal by testing for multiple passing of phase 1. That is, the trigger threshold in phase 1 must be exceeded lNumConsec times before 3*lNumConsec samples pass (lNumConsec = Sample Rate / (2 *

Minimum Frequency of interest)). Test 2 verifies that the signal strength exceeds the set signal-to-noise ratio times the average signal level sometime during phase 2. Test 3 checks if the trigger threshold was not exceeded more than half of the time at any point during phase 2. If test 3 is true, all variables are reset and phase 1 must be passed again
Phase 3 requires the signal to be oscillatory. This phase must be completed before 12 * lNumConsec samples pass. The signal's MDF must exceed the trigger threshold six times in opposite directions (which requires three full cycles of signal). If the MDF is below the trigger threshold more than half the time it takes to pass the oscillation test and is at least four seconds under the threshold, phase 3 fails. If phase 3 passes, a pick is declared and the time of phase 1 passage is declared the P-time.
Veith's (1978) algorithm also has a procedure for terminating the event based on signal strength. In pick_wcatwc, this is neglected since the signal is processed for Mb, Ml and Mwp magnitudes for a set time after the P onset. The pick is terminated and variables reinitialized when all magnitude information has been gathered for that P-pick.
A few processes have been added to the original P-pick algorithm of Veith. Real-time magnitude information is processed as soon as phase 1 is passed. The maximum peak-to-trough amplitude (MDF) in the first MbCycles half-cycles of signal after the P onset is used to determine Mb. The number of samples is doubled to compute period. The signal for a total of LGSeconds, after the first MbCycles half-cycles, is processed to determine the maximum amplitude for Ml. One hundred fifty seconds allows Ml computations for up to approximately nine degrees epicentral distance. Automatic Mwp computations (Tsuboi, et al., 1995; Tsuboi, et al., 1999; Whitmore, et al., 2002) are performed MwpSeconds after the P arrival on broadband seismic data. One other function added to the original algorithm is automatic station calibration for those WC/ATWC network short period stations equipped with a sine-wave calibrator. The 60s, 1 Hz sine wave calibrations are automatically triggered twice a day. The P-picker will pick these as a P wave. If the first six full cycles of signal have the characteristics of a calibration pulse (that is, 1 Hz frequency and constant peak MDF) the pick is declared a calibration and the amplitude can be used to set the station's magnification.
After a pick is made, the pick information is logged and reported to the PICK_RING_EB in the PICKTWC and in the PICK_GLOBAL Earthworm message formats. Magnitude information is added as it is computed. The pick can be altered interactively using modules develo or hypo_display. Pick information (time and magnitude data) are re-reported to the PICK_RING as pick_wcatwc updates the magnitude information or it is interactively changed in another module. Typically, the larger the earthquake, the more accurate the P-picks. Picks are normally determined within a few samples for distinct arrivals; whether from local earthquakes or teleseisms. Very emergent P waves can produce erroneous automatic P-picks. For this reason an easy method of adjusting picks, as provided by modules develo and hypo_display, is necessary. The P-pick algorithm works best on short period data or short period filtered, broadband data. It was tested unsuccessfully on raw, broadband data.

References

Sokolowski, T.J., G.W. Fuller, M.E. Blackford, and W.J. Jorgensen (1983). The Alaska Tsunami Warning Center's automatic earthquake processing system, in *Proceedings, 1983 Tsunami Symposium, Hamburg, FRG, August, 1983*, 131-147.
Tsuboi, S., K. Abe, K. Takano, and Y. Yamanaka (1995). Rapid determination of **M**w from broadband P waveforms, *Bull. Seism. Soc. Am.*, **85**, 606-613.
Tsuboi, S., P.M. Whitmore, T.J. Sokolowski (1999). Application of **M**wp to deep and teleseismic earthquakes, *Bull. Seism. Soc. Am.*, **89**, 1345-1351.
Veith, K.F. (1978). Seismic signal detection algorithm, Teledyne Geotech Technical Note 1/78, 10 pp.
Whitmore, P.M., S. Tsuboi, B. Hirshorn, and T.J. Sokolowski (2002). Magnitude-dependent correction for **M**wp, *Science of Tsunami Hazards Journal*, **20**, 187-192.

Appendix D: Station and Calib File Parameters

Appendix D1: Station Type List (Station.dat File)

The following list maps integers, that are to be entered in column 11 (channel type) of the station.dat file for each channel listed in that file, with corresponding seismometer that is installed at each station. This information is used to display the correct station Instrument information when Station Description is selected in the develo GUI (right click on station name).

The seismometer types listed below are hard coded and if the list is to be added to, the software must be recompiled.

1 = STS1 360s
2 = STS2 130s
3 = CMG-3NSN 30s
4 = CMG-3T 100s
5 = KS360i 360s
6 = KS5400 350s
7 = CMG-3 30s
8 = CMG-40T 60s
9 = CMG3TNSN 30s
10 = KS-10 20s
11 = CMG3ESP_30 30s
12 = CMG3ESP_60 60s
13 = Trillium
14 = CMG3ESP_120 120s
15 = CMG40T_20 20s
16 = CMG3T_360 360s
17 = KS2000_120 120s
20 = unknown broadband (no cal)
30 = EpiSensor FBA ES-T
40 = GT_S13
41 = MP_L4
50 = generic LP (no cal)
51 = ATWC LP Response (Hi gain)
52 = ATWC LP Response (Low gain)
100 = Generic SP (no cal)
101 = ATWC SP Response (Hi gain)
102 = ATWC SP Response (Medium gain)
103 = ATWC SP Response (Low gain)

Appendix D2: Operating Agency List (Station.dat File)

The following list maps integers, that are to be entered in column 12 (agency) of the station.dat file for each channel listed in that file, with corresponding agency that is running the station. This setting is not used anywhere and should only be set for GSN stations so that a correct station description is displayed in develo GUI.

The agencies listed below are hard coded and if the list is to be added to, the software must be recompiled.

1=atwc
2=ptwc
3=neic
4=aeic,

5=iris/asl
6=iris/ida
7=usgs menlo
8=ucb
9=cal tech
10=uw
11=pgc
12=avo
13=gtsn
14=jma
15=livermore
16=hong kong
17=korea
18=st.louis u.
19=HVO
20=Canada CNSN
21=LDGO
22=U Utah
23=Boston C.

Appendix D3: Channel Sensitivity and Normalization Factor Calculations (Station.dat and Calib Files)

Displacement Channel Sensitivity S_{SysD} Calculation (row 1 column 6 of each channel entry in the Calib file):

1. Calculate system gain in velocity S_{GV}:

   $$S_{GV} = S_{Dig} \left(^{counts}/_{V}\right) \times S_{Seis} \left(^{V}/_{m/s}\right)$$

   Ex. Trident digitizer with sensitivity of 0.4 $^{counts}/_{\mu V}$ and Trillium 40 seismometer with sensitivity of 1500 $^{V}/_{m/s}$:

   $$S_{GV} = 400000 \left(^{counts}/_{V}\right) \times 1500 \left(^{V}/_{m/s}\right) = \boxed{0.6\ ^{counts}/_{nm/s}*}$$

2. Calculate system sensitivity in velocity S_{SysV}:

   $$S_{SysV} = 1 / S_{GV}$$

   Ex. Using the above example:

   $$S_{SysV} = 1/0.6 \left(^{counts}/_{nmx/s}\right) = 1.66667\ ^{nm/s}/_{count}$$

3. Calculate system sensitivity in displacement S_{SysD}:

   $S_{SysD} = S_{SysV} \left(^{nm/s}/_{counts}\right) / (2 \times \pi \times f(Hz))$ where f = normalization frequency

   Ex. Using the above example:

   $$S_{SysD} = 1.666667 \left(^{nm/s}/_{count}\right) / (2 \times \pi \times 1 \left(^{1}/_{s}\right)) = \boxed{0.265258\ ^{nm}/_{count}**}$$

* The value (in $^{counts}/_{m/s}$) that goes into column 4 for each channel listed in station.dat file
** The value that goes into column 6 row 1 for each channel entry in the Calib file

Displacement Normalization Factor A_{0D} Calculation (row 3 column 4 of each channel entry in the Calib file):

1. Obtain the seismometer normalization factor (in velocity) from the seismometer manual. The normalization factor is usually given in seismometer response section together with poles and zeros.

   Ex. Trillium 40 normalization factor A_{0V} = 1333310

2. Calculate normalization factor in displacement A0D:

   $A_{0D} = A_{0V} / (2 \times \pi \times f\ (Hz))$ where f = normalization frequency

   Ex. Using the above Trillium 40 example:

   $A_{0D} = 1333310 / (2 \times \pi \times 1\ (^{1}/_{s})) =$ 212202.88***

*** The value that goes into row 3 column 4 for each channel entry in the Calib file.

Appendix E: Adding Stations to Earlybird

The following procedure outlines which files have to be editesd in order to add a station to the list of stations that are processed by EarlyBird software (station TORO is used as an example).

1. Configure data source module
    a. If the station data is acquired through NaqsServer, add the station entry to naqs2ew.d file as shown here (one line entry per channel):
    Ex. RequestChanSCNL TORO HHZ PO -- 2002 30 0 0
    b. If the station data is acquired through Live Internet Seismic Server, create a new liss2ew.d configuration file with the name of the station (ex. liss2ew_TORO.d). Add the IP address of the liss server for the station being added.
    c. If the station data is acquired though IRIS/IDA SeedLink server, add the station to slink2ew.d file as shown here (one line entry per channel):
    Ex. Stream II_TORO "00HHZ.D"
2. Add the station information to pick_eb.sta file (one line per channel):
    Ex. ALFO HHZ PO 1 1 5 3 0.000012 20.0 0.333 1 20.0 0.03
3. Add the station information to pick_eb_lp.sta file (one line per channel):
    Ex. ALFO HHZ PO 1 1 5 3 0.000012 20.0 0.333 1 1.0 0.03
4. Add the station parameters to stations.dat file (one line per channel):
    Ex. TORO HHZ PO 792000000.000000 0.000000 52.510300 -125.084400 1237.000000 4194304.000000 0.000000 4 20 Anahim Lake, BC
5. Add the station to appropriate decimation configuration files (decim_100to20.d, decim_20to1.d and/or decim_40to20.d) depending on the sampling rate at which the data is acquired (one entry per channel):
    Ex. GetSCNL TORO BHZ CU -- TORO BHZ CU --
6. Add the station channel response information to Calib file in the following format (vertical channel only):
    Ex. CAL2 TORO HHZ PO CMG-3E 1.26E-01 1.000 100.00000 2007/02/09 00:00 1 52.5103 -125.0844 1237
    PAZ2 1 C 7.21599665E+08 5 3 CMG-3E
    -5.02654800E+02 0.00000000E+00
    -1.00530960E+03 0.00000000E+00
    -1.13097340E+03 0.00000000E+00
    -4.44200000E-02 4.44200000E-02
    -4.44200000E-02 -4.44200000E-02
    0.00000000E+00 0.00000000E+00
    0.00000000E+00 0.00000000E+00
    0.00000000E+00 0.00000000E+00
7. Adjust the number of traces displayed per screen in develo.d and lpproc.d configuration files :
    Ex. NumTracePerScreen = 54
8. Add the new station to the appropriate station grouping in screendisp.ini file (develo module):
    Ex. TORO HHZ PO 1 1 0 0 0
9. Add the new station entry to the appropriate station grouping in screen.ini file (Analyze module):
    Ex. TORO HHZ PO 1
10. Add the new station entry to the appropriate station regional alarm groupings in alarm.dat file (page_alarm module).
    Ex. Station TORO

Appendix F: UserStations.hinv File Format

Hypoinverse uses UserStations.hinv file to get information such as delay, location, amplitude and duration magnitude correction, weight code, calibration factor etc. for stations whose data it processes. The Hypoinverse station file has a specific format with data being sorted and parsed in assigned columns.

Figure F1 shows a typical line from the UserStation file together with column numbers. The line marked Good shows the correct format read by the Hypoinverse which is released as part of Earthworm v7.1. The line marked Bad shows the incorrect format that was used by earlier versions of Hypoinverse and is produced if one uses the EarthwormConfig utility to configure CoreEarthworm modules.

```
Col:  0        1         2         3         4         5         6         7  7
      1        0         0         0         0         0         0         0  3
Bad:  DDN   GH  HHZ  30 19.6799N 78  0.8580E 6191.0    0.00  0.00  0.00  0.00 0  1.00
Good: DDN   GH  HHZ  30 19.6799N 78  0.8580E 6191.0     0.00  0.00  0.00  0.00 0  1.00--
```

Figure F1: UserStations.hinv File Format

To ensure that the UserStations file is correct format: insert space at column 46 and add -- to the end of the line in the file created by EarthwormConfig utility. In addition to this, all SOH channels should be removed.