



## SeaSonde Radial Site Release 6 **RadialSiteReporter**

Mar 30, 2009

The RadialSiteReporter tool monitors the status of the Radial Site and is an important companion of the RadialWebServer. The RadialWebServer uses the results of this tool for the site status, details, and alerts. The RadialSiteReporter can also email changes in alerts. This tool is scheduled to run once a minute. A Terminal command of "launchctl list" should show a com.codar.RadialSiteReporter in the output; otherwise it is not scheduled to run.

The purpose of this document is to get into the technical aspects of what RadialSiteReporter does and how to fine tune it. RadialSiteReporter is a fairly complicated process since it looks at many files and talks to many applications. This document also discusses some technical details on what it may take to email alerts out from the site.

### **What RadialSiteReporter does**

When RadialSiteReporter runs it does the following in order:

- Read config files: Header.txt, AnalysisOptions.txt, Phases.txt and EllipticalOptions.txt

- Get the computer time, timezone, and runtime.

- Run **PS** unix command to see what processes are running.

- Run **KEXTSTAT** unix command to see what drivers are running.

- Run **PMSET** unix command to check computer power source.

- Run **DF** unix command to check computer disk/volume space.

- Ask **Sentinel** for status to check if about to restart, or just started up, and when the next restart is.

- Search CSQ folders to find latest CSQ spectra.

- Read latest CSQ spectra and perform Bragg/Noise/Interference test.

- Create CSQ Web Status View png plots using **SpectraPlotterMap** tool.

- Search CSS folders to find latest CSS spectra.

- Read latest CSS spectra and perform Bragg/Noise/Interference test.

- Create CSS Web Status View png plots using **SpectraPlotterMap** tool.

- Search radial folders for latest measured pattern radial if enabled.

- Read latest meas radial, make png plots and examine for age, empty, small, short range.

- Search radial folders for latest ideal pattern radial if enabled.

- Read latest ideal radial, make png plots and examine for age, empty, small, short range.

- Run **SeaDisplayTool** to create radial pngs for Web Status View.

- Run **LLUVMerger** tool to create a 25hr average of meas if avail else ideal radial with png plots.

- Search for latest wave model file if enabled.

- Examine latest wave model entry for age and unmeasurable response.

- Search for latest hardware diagnostic .hdt (SeaSondeController hardware stat file)

- Search for latest radial diagnostic .rdt (SpectraProcessing stat file)

- Tell **SeaSondeController** to do a RPT command.

- Parse the RPT response for receiver & transmitter state and status details.

Ask **SeaSondeController** for communication status.

Run **SYSTEM\_PROFILER** unix command to get computer plus operating system information.

Initialize the Alerts. Enabled, Threshold, and Priority values come from Configs/RadialWebConfigs/RadialSiteReporter.plist. If this plist is missing it will be created with default values.

Get the alerts previous state from Data/RadialSiteStatus/info\_alert.plist

Run through all the alert tests to determine trip/clear or leave alone.

Add newly cleared and newly tripped alerts to the alert log (Codar/SeaSonde/Logs/Alert\_XXXX\_YYYYMMDD.log)

Create Data/RadialSiteStatus/info\_alert.plist with the newly cleared alerts and currently tripped alerts. Also add a section about the state of all alerts.

Create Data/RadialSiteStatus/info\_status.plist with status information for Web Status View.

Create Data/RadialSiteStatus/info\_detail.plist with alerts and detail info for the Web Detail View.

Email alert state upon newly cleared or newly tripped alerts.

Site frequency, bandwidth, sweep rate, gps timing, and pulse shaping settings are determined by the last known valid source from hdt diag, CSS, CSQ, and lastly from SeaSondeController. If the receiver is not available, SeaSondeController will report the last stored settings

## Alerts

RadialSiteReporter generates trouble alerts for the site. These alerts are shown by the RadialWebServer, plus are added to an alert log, and they can optionally be emailed when an alert changes state. Email is done by using command line tool sendmail; there is a section below on how the mail system needs to be configured.

Each alert has four states: *newly tripped*, *still tripped*, *newly cleared*, and *still cleared*. Emails alerts happen only when at least one alert is newly tripped or cleared upon which all the newly cleared alerts and all new and old tripped alerts are emailed out.

Each alert has a level of Ignore, Success, Notice, Warning, and Failure. Typically when an alert is tripped it will have a level of Warning or Failure and Success when cleared. Tripped alerts are typically shown with a Warning or Failure message while cleared alerts are shown with a Success message.

Each alert has a first tripped time and a cleared time shown in computer timezone. The alert messages show when the alert was first tripped or when it was cleared.

The alert state is saved in Data/RadialSiteStatus/info\_alert.plist. There is a section for newly cleared alerts plus all tripped alerts which is for the Web Status view. There is a second section which holds the current state of all possible alerts. This is used to determine if the alert is newly tripped or newly cleared. Deleting info\_alert.plist will cause the alert state to be cleared and the next time RadialSiteReporter runs it will cause previously tripped alerts to appear as newly tripped; hence you will get an email and the alert tripped timestamp will show the current time.

The alerts are first initialize from /Codar/SeaSonde/Configs/RadialWebConfigs/RadialSiteReporter.plist. With this plist, undesirable alerts can be disabled, threshold can be change for those alerts that use them, priorities can be adjusted and more. Each alerts has it

own settings which can be found using the Alert List below in this document. Here is a list of the keys for the settings and what they do.

Enabled	<b>No</b> to disable alert otherwise <b>Yes</b> (default is Yes)
Duration	A non zero here will hold an alert tripped for this many more minutes even though the cause of the alert has cleared. This allows some hystereses, so that troublesome conditions that might cause an alert to alternate between clearing and tripping every minute could be filtered out by increasing this duration. (default is 0 min)
Threshold1	Some alerts use this as a minimum threshold (like temperature alerts). The application running alerts use this for exact name of the app to look for.
Threshold2	Some alerts use this as a maximum threshold (like computer time)
TripLevel	Sets the number of times that the cause of an alert must happened each check before the alert is considered tripped. Since SeaSondeRadialReport runs every minute this this value is much like Duration except that the alert will not be tripped until TripLevel minutes. (default is 0)
Priority	Sets the priority of the alert. Alerts are listed in order of their priority level. This can be any floating point value. The higher the value the higher the priority. The default values range from AllsWell=101 to ComputerRestarting=25. This plist key does not normally exist and you will have to add it if you wish to change the priority levels.
Class	Sets the class level of the alert. Only these classes are acceptable <b>Ignore, Success, Notice, Warning, or Failure</b> . The default classes are listed below for each alert. You might want to add this key if you want to change a Failure alert into a Warning or Notice and vice versa. This plist key does not normally exist and you will have to add it if you wish to change the class level. Note, this really only effects the output message.

Email alerts are sent only upon newly tripped or newly cleared alerts. Each email will contain all newly cleared and all currently tripped alerts. The email is sent containing both plain text and rich text formats where the rich text shows the alerts in color just like the RadialWebServer.

Example Email:

```
2008-11-29 21:59 UTC Failure: Transmitter Not Found.
2008-11-29 21:59 UTC Failure: CSS has No Bragg.
2008-11-29 21:59 UTC Failure: CSQ has No Bragg.
2008-12-01 00:10 UTC Success: Receiver is Running (14m 21.3s).
2008-11-29 21:59 UTC Failure: Radial Contains No Vectors.
2008-11-30 23:59 UTC Warning: Computer Restarted Recently .
Running for 12 minutes.
```

Alert Time is always shown with the Radial Site **local** time with a label showing its hourly offset from UTC. If it's at UTC then just 'UTC' is shown. If it is a integer hour offset like PDT then it will show UTC-8. If it is a partial hour like IST India, then it will show UTC+5.5 For example a message with '2008-11-30 16:00 UTC-8' will be 2008-12-01 00:00 UTC

## Output files

Most of RadialSiteReporter's output goes to /Codar/SeaSonde/Data/RadialSiteStatus/ folder. In this folder:

- "info\_alert.plist" contains a "alert\_info" list of all tripped and recently cleared alerts. It also contains a "alert\_state" list which is the status of all possible alerts.
- "info\_report.plist" contains a list of what's to be displayed in the RadialWebServer Details View.
- "info\_status.plist" contains a list of what's to be displayed in the RadialWebServer Status View.
- "MergeRadial.ruv" is average of the last 25hrs of measured pattern radials if enabled otherwise ideal.
- "rads1full.png" is radial vector plot of latest measured pattern radial if enabled otherwise latest ideal.
- "rads2full.png" is radial vector plot of latest ideal pattern radials if measured is enabled otherwise is 2nd to latest ideal pattern radials.
- "rads3full.png" is radial vector plot of the 25hr averaged radial
- "rads1thumb.png", "rads2thumb.png", "rads3thumb.png" are smaller plots of the full size ones (158 by 158 pixels)
- "spectra1full.png" is a power color map of the latest CSS
- "spectra2full.png" is a power color map of the latest CSQ
- "spectra1thumb.png", "spectra2thumb" are smaller plots of the full size ones. (240 by 135 pixels.)

Newly tripped or cleared alerts are appended to a an alert log in /Codar/SeaSonde/Logs/Alerts/Alert\_XXXX\_YYYYMMDD.log where XXXX is the site code, and the YYYYMMDD year, month, day for the Sunday that the weekly logs start on.

## Email Alerts

Currently RadialSiteReporter only supports the email on changed alert status. It does not send report log periodically even though there's a plist and RadialWebServer UI support for it.

Whom the emails are sent to comes from /Codar/SeaSonde/Configs/RadialWebConfigs/RadialSiteAlertMail.plist. This plist contains a Enabled key which if disabled, disables all emailing. The MailTo array can contains a list of up to 256 different destinations. Each destination may have a single or multiple email address, enabled, and other options. This plist is managed by the RadialWebServer. A RadialWebServer user with an admin account can manage who gets sent alert emails.

Emails are done through the command line tool sendmail. Note, that this has no connection or relation to Apple's Mail.app or its preferences. Sendmail buffers the mail and a background process sends it out, which means that RadialSiteReporter does not know if the email was sent successfully. Your internet service provider may reject email coming from the radial site. There's a fair chance that you Internet Service Provider will bounce the mail. Bounced mail will accumulate in the file /var/mail/<user> (where <user> is mostly likely codar). SeaSondeStartupExtras which is run once each time Sentinel starts up the software, currently will copy this file to /Codar/SeaSonde/Logs/PreviousMail.txt and then delete it to keep from

piling up too many bounced mails. To tell if your alert emails are being bounce you will have to take a look at the PreviousMail.txt or use the Terminal and enter 'mail' and return. Also use Console.app to take a look at /var/log/mail.log

*If your ISP is bouncing the alert emails, the follow technical steps can be tried. This instructions require some familiarity with unix command line and information on what your internet service provider requires for email.*

If your ISP is bouncing the emails , you can edit /etc/postfix/main.cf. You can use Finder menu Go -> Go to Folder and enter /etc/postfix to view this normally hidden folder and locate main.cf and edit it with TextEdit. Try adding the following line to main.cf and make sure there is not already a line that starts with relayhost (#relayhost is doesn't count). Change **yourisp** with your actual internet service provider name, for example 'smtp.comcast.net'.

```
relayhost = smtp.yourisp.com
```

or if your provider uses mail on another port like 587 use smtp.yourisp.com:587

```
relayhost = smtp.yourisp.com:port
```

Save main.cf and give it a try. (See below for instructions on sending test mail)

If your ISP requires a password login to send mail, you can try the following

Add these lines to main.cf

```
smtp_sasl_auth_enable = yes
```

```
smtp_use_tls = yes
```

```
smtp_enforce_tls = yes
```

```
smtp_sasl_security_options =
```

```
smtp_sasl_tls_security_options =
```

```
smtp_sasl_tls_verified_security_options =
```

```
smtp_tls_loglevel = 2
```

```
smtp_sasl_password_maps = hash:/etc/postfix/smtp_sasl_passwords
```

```
smtp_tls_per_site = hash:/etc/postfix/smtp_tls_sites
```

```
tls_random_source = dev:/dev/urandom
```

Create the file /etc/postfix/smtp\_tls\_sites with the following contents:

```
smtp.yourisp.com MUST_NOPEERMATCH
```

Then run the following commands from a Terminal Window

```
cd /etc/postfix
```

```
chmod go-rx smtp_sasl_passwords
```

```
postmap smtp_sasl_passwords
```

```
postmap smtp_tls_sites
```

If you ISP requires a security certificate to send mail, you can try the following: (Note the 2 lines here are really one long line; be sure to copy both.

```
security find-certificate -a -p /System/Library/Keychains/
```

```
SystemRootCertificates.keychain > /System/Library/OpenSSL/certs/cert.pem
```

Include the following in /etc/postfix/main.cf:

```
smtp_tls_CAfile = /System/Library/OpenSSL/certs/cert.pem
```

Lastly, if your internet service provider requires the email to come from a valid email user, edit /Codar/SeaSonde/Configs/RadialWebConfigs/RadialSiteAlertMail.plist and enter your email address "youremail@yourisp.com" into the value for the **MailFrom** key. Note, you can add extra info here if you use this email format "anything<youremail@yourisp.com>" replacing anything with whatever info you want, example "Site\_XXXX<me@isp.com>". If you leave the MailFrom key value empty, then the alert emails will be from "user@machine.local" where user is the computer user name (probably codar) and machine is the computer name from system preferences Sharing.

To send test mail to all RadialWebServer alert users, in a Terminal.app window run the following commands:

```
cd /Codar/SeaSonde/Apps/Bin
./RadialSiteReporter -testmail -noreport
```

To test the mail to any email destination, in a Terminal.app window run the following commands replacing you@isp.com with the desired email address.

```
cd /Codar/SeaSonde/Apps/Bin
./RadialSiteReporter -testmail=you@isp.com -noreport
```

Or you can do your own email test by entering this into a Terminal.app window.

```
printf "To: you@isp.com\nSubject: Test\nHello" | sendmail -t -f you@isp.com
```

## Radial Site Alerts

Here's a list of all the possible alerts. Between the {} is the plist keyword used for RadialSiteReporter.plist to id each alert. The bold text is the alert tripped message followed by a description of what tripped the alert.

{AllsWell}

### **Nothing To Report**

**Success** when no other Alerts are tripped.

This is the only alert that shows success when tripped.

This alert's trip time is updated each time while date after 'Since' shows how long there was nothing to report.

{CompPower}

### **Computer Power is Running From / Battery / UPS / AC Power**

**Failure** when not AC Power

{CompTime}

### **Computer Time Invalid (Year is #)**

**Failure** when computer current date year is <2000 or >2030 (thresholds set in plist)

{CompTemperature}

### **Computer Temperature is High**

*Not Yet Implemented!*

{CompRestarted}

**Computer Restarted Recently** (Running for # Minutes)

**Warning** when mach clock\_get\_time is <15minutes

{CompDiskSpaceLow}

**Computer Disk Space is Getting Low** (#GB avail out of #GB)

**Warning** when boot volume is less than 1GB but >100MB (settable)

{CompDiskLow}

**Computer Disk is Full** (#GB avail out of #GB)

**Failure** when boot volume is <100MB (settable)

{CompRestarting}

**Computer is About to Restart** (In # Minutes)

**Warning** while Sentinel is going to restart the computer within 5 minutes.

{SentStart}

**Sentinel is Starting Up**

**Warning** while Sentinel is starting up all the SeaSonde applications in order.

{AppSent}

**Sentinel Not Running**

**Failure** when Sentinel not found in Unix PS response

{AppSSCtrl}

**SeaSondeController is Not Running.**

**Failure** when SeaSondeController not found in Unix PS response

{AppSSAcq}

**SeaSondeAcquisition is Not Running.**

**Failure** when SeaSondeAcquisition not found in Unix PS response

{AppCSPro}

**CSPRO is Not Running.**

**Failure** when CSPro not found in Unix PS response

{AppArch}

**Archivalist is Not Running.**

**Failure** when Archivalist not found in Unix PS response

{TermSpectraProc}

**AnalyzeSpectra is Not Running.**

**Failure** when AnalyzeSpectra not found in Unix PS response

{RcvrAppFailure}

**Unable to Get Receiver Status**

**Failure** when SeaSondeController did not respond or applescript command somehow failed.

{RcvrNotFound}

**Receiver Not Found**

**Failure** when SeaSondeController returns an openErr in response to a RPT command meaning that SeaSondeController cannot find any receiver.

{RcvrCommFailure}

**Receiver Communication Failure / No Response / Cannot Find the Receiver / Invalid Status Response / # Comm Error**

**Failure** when SeaSondeController status reports a communication problem.

{RcvrTempHot}

**Receiver Temperature Too Hot**

**Failure** when RPT chassis or awg3 module temp is over 45 C (settable)

{RcvrRestarted}

**Receiver Restarted Recently** (Running for #m #s)

**Warning** when RPT receiver runtime less than 15minutes.

{RcvrVoltageLow}

**Receiver Supply Voltage is Low** (supply is #V)

**Failure** when RPT +5V<4.75, -5V>-4.5, +12V<10.5, or 24VDC if avail <19.

{RcvrVoltageHigh}

**Receiver Supply Voltage is High** (supply is #V)

**Failure** when RPT +5V>5.5, -5V>-5.9, +12V>12.9, or 24VDC if avail >28

{RcvrPLLUnlock}

**Receiver PLL is Unlocked**

**Failure** when RPT PLL is unlocked.

{RcvrGPSTimeBad}

**Receiver GPS Time is Not Valid**

**Failure** when RPT GPS Time flags are not valid.

{RcvrGPSAntBad}

**Receiver GPS Antenna** (short/open)

**Failure** when RPT GPS antenna reports short or open.

{RcvrGPSNoSats}

**Receiver GPS Not Enough Satellites** (no gps time, pdop, not usable, 1, 2, 3, bad sat, TRAIM)

**Failure** when RPT GPS satellite status is not 0(normal)

{RcvrGPSSurvey}

**Receiver GPS Survey in Progress** ( #% Completed)

**Failure** when RPT GPS satellite status is not 0(normal)

{RcvrGPSLocBad}

**Receiver GPS Location** (Bad or Not Stored)

**Failure** when RPT GPS status not accurate or corrupted

**Warning** when RPT GPS status not stored

{RcvrPPSBad}

**Receiver GPS PPS is Not Valid**

*Not Yet Implemented!*

{RcvrGPSTimingBad}

**Receiver GPS Timing Not Aligned** (off by #us)

**Failure** when RPT shows that GPS Synchronized Timing does not match set timing value)

{RcvrGPSTimingSync}

**Receiver GPS Timing was Realigned**

*Not Yet Implemented!*

{RcvrGPSDisciplineBad}

**Receiver GPS Timing is Not Disciplined** (phaselocking,osc warming,freq locking,placing,init,comp vcxo, inactive,recovery)

**Failure** when RPT GPS disp status is not zero

{RcvrFreqBad}

**Receiver Frequency is Not Set**

**Failure** when SeaSondeController center freq <1MHz

{RcvrBWBad}

**Receiver Bandwidth is Not Set**

**Failure** when SeaSondeController bandwidth <1kHz

{RcvrNotStored}

**Receiver Settings / are Not Stored / Have Changed / Shown From Last Stored / Match Stored**

**Warning** when SeaSondeController status indicates one of / **are Not Stored / Have Changed / Shown From Last Stored.**

**Success** when SeaSondeController status indicates settings match last stored.

**Have Changed** shows only briefly in SeaSondeController, which next automatic static check will switch to **are Not Stored.**

**Shown From Last Stored** from stored happend when SeaSondeController is launched and cannot find the receiver or bad comm.

{RcvrNoSamples}

**Receiver No Sample Data Available**

**Failure** when when SeaSondeAcquisition status indicates that there is no sample data available from the receiver.

{RcvrBlankingOn}

**Receiver Blanking is Set On**

**Failure** when RPT Blanking is set On.

{RcvrBlankingOff}

**Receiver Blanking is Set Off**

**Failure** when RPT Blanking is set Off.

{TxNotFound}

**Transmitter Not Found.**

**Failure** when RPT does not contain a XRPT response

{TxCommBad}

## Transmitter Communication Failure

*Not Yet Implemented!*

(Covered by Transmitter Not Found.)

{TxMeasBad}

### Transmitter Measurement Invalid

**Failure** when measurements on transmitter front panel board do not appear to be valid. (This checked through the A/D Reference measurement.)

{TxDrive}

### Transmit Drive is / Set On / Set Off / is Forced Off

**[GPSNotReady,RcvrTooHot,NoStatus,PLL Unlock,GPS UnSync,ExternHalt,Set Off]**

**Failure** when Tx Watch is tripped which forces tx drive off.

**Warning** when Tx set off.

**Failure** when Tx set On

{TxTempHigh}

### Transmitter Temperature Too High

**Failure** when chassis or amplifier temperature is over 45 C (settable)

{TxVoltageLow}

### Transmitter Voltage Low (supply is #V)

**Failure** when  $5V < 4.75$  or  $28V < 23$  (settable)

{TxVoltageHigh}

### Transmitter Voltage High (supply is #V)

**Failure** when  $5V > 5.5$  or  $28V > 30$  (settable)

{TxPowerLow}

### Transmit Power Low (at # watts)

**Failure** when RPT forward power is  $< 30$  watts (settable)

{TxReflectHigh}

### Transmit Reflected Power High (at # watts)

**Failure** when RPT reflected power is  $> 10$  watts (settable)

{CfgMissing}

### Config. Missing File

**Failure** if RadialConfigs AnalysisOptions, Header, Phases, IdealPattern, or AngSeg files are missing.

{CfgSiteCodeNotSet}

### Config. Site Code Not Set

**Failure** if config sitecode is 0, XXXX, or ????

{CfgLocationNotSet}

### Config. Location Not Set

**Failure** if config receiver location is 0,0

{CfgPhasesNotSet}

### Config. Phases Not Set

Warning if config phases are 0,0

{CfgBearingNotSet}

**Config. Receive Antenna Bearing Not Set**

**Failure** if config bearing is 0.

{CfgCoastlineNotSet}

**Config. Coastline Angles Not Set**

**Failure** if wave processing is enabled and coastline angles are 0,0

{CfgPatternNotInstalled}

**Config. Measured Pattern is Not Installed**

**Failure** if measured pattern radials enabled and RadialConfigs MeasPattern.txt file is missing.

{CSQNotFound}

**No CSQ CrossSpectra Files**

**Failure** if no CSQ files found.

{CSQTooOld}

**CSQ is Not Up-to-Date**

**Failure** if CSQ time stamp is older than twice its coverage time.

{CSQNoBragg}

**CSQ is Has No Bragg** (Loop1/Loop2/Monopole)

**Failure** if CSQ test on each antenna has Bragg widths <4 doppler bins average on rc 3 to 16

{CSQHighNoise}

**CSQ is Has High Noise or Interference** (Loop1/Loop2/Monopole)

**Failure** if CSQ test on each antenna has Bragg widths >4 doppler bins and non Bragg energy is twice the Bragg Energy.

{CSQNoBistatic}

**CSQ is Has No Bistatic Bragg** (Loop1/Loop2/Monopole)

*Not Yet Implemented!*

{CSQAmpIBad}

**CSQ Amplitudes don't Match the Pattern**

*Not Yet Implemented!*

{CSQPhasesBad}

**CSQ Phases don't Match the Pattern**

*Not Yet Implemented!*

{CSSNotFound}

**No CSS CrossSpectra Files**

**Failure** if no CSS files found.

{CSSTooOld}

**CSS is Not Up-to-Date**

**Failure** if CSS time stamp is older than twice its coverage time.

{CSSNoBragg}

**CSS is Has No Bragg** (Loop1/Loop2/Monopole)

**Failure** if CSS test on each antenna has Bragg widths <4 doppler bins average on rc 3 to 16

{CSSHighNoise}

**CSS is Has High Noise or Interference** (Loop1/Loop2/Monopole)

**Failure** if CSS test on each antenna has Bragg widths >4 doppler bins and non Bragg energy is twice the Bragg Energy.

{CSSNoBitstatic}

**CSS is Has No Bistatic Bragg** (Loop1/Loop2/Monopole)

*Not Yet Implemented!*

{CSSAmpIBad}

**CSS Amplitudes don't Match the Pattern**

*Not Yet Implemented!*

{CSSPhasesBad}

**CSS Phases don't Match the Pattern**

*Not Yet Implemented!*

{RadNotFound}

**No Radial Files**

**Failure** if (doing Ideal and no Ideal Rads) or (doing Meas and no Meas Rads) or (doing both and no Rads).

{RadTooOld}

**Radial is Not Up-to-Date**

(Latest Radial is selected by radial antenna type for RDLi or RDLm radial)

**Failure** if radial time stamp is older than twice its coverage time.

{RadNoVectors}

**Radial Contains No Vectors**

(Latest Radial is selected by radial antenna type for RDLi or RDLm radial)

**Failure** if radial doesn't contain any vectors.

{RadFewVectors}

**Radial Contains Few Vectors**

(Latest Radial is selected by radial antenna type for RDLi or RDLm radial)

**Failure** if radial contains < 20% vectors of the expected angle width/ angle res \* half the processing ranges.

{RadLimitedRange}

**Radial Has Limited Range**

(Latest Radial is selected by radial antenna type for RDLi or RDLm radial)

*Not Yet Implemented!*

{RadOverLand}

**Radial Has Many Vectors Overland**

(Latest Radial is selected by radial antenna type for RDLi or RDLm radial)

*Not Yet Implemented!*

{WavNotFound}

**No Wave Files**

**Failure** if no wave files and wave processing enabled

{WavTooOld}

**Wave Entry is Not Up-to-Date**

**Failure** if last wave entry time is older than current time - 2 x coverage time.

{WavNoResult}

**Wave Entry was not Determinable**

**Warning** if last wave entry is all 999s except wind direction.

{ElpConfigMissing}

**Elliptical Configuration Missing File**

*Not Yet Implemented!*

{ElpNotFound}

**No Elliptical Files**

*Not Yet Implemented!*

{ElpTooOld}

**Elliptical is Not Up-to-Date**

*Not Yet Implemented!*

{ElpNoVectors}

**Elliptical Contains No Vectors**

*Not Yet Implemented!*

{ElpFewVectors}

**Elliptical Contains Few Vectors**

*Not Yet Implemented!*

{ElpLimitedRange}

**Elliptical Has Limited Range**

*Not Yet Implemented!*

{ElpOverland}

**Elliptical Has Many Vectors Overland**

*Not Yet Implemented!*

{AppTsuArch}

**ArchivalistTsunami Not Running**

**Failure** if TsunamiConfigs exists and PS ArchivalistTsunami not found.

{AppCSProBkgnd}

**CSProBackground Not Running**

**Failure** if TsunamiConfigs exists and PS CSProBackground not found.

{AppCSProFrgnd}

### **CSPProForeground Not Running**

**Failure** if TsunamiConfigs exists and PS CSPProForeground not found.

{TermTsuBkgndProc}

### **AnalyzeBkgndSpectra Not Running**

**Failure** if TsunamiConfigs exists and PS AnalyzeBkgndSpectra not found.

{TermTsuFrgndProc}

### **AnalyzeFrgndSpectra Not Running**

**Failure** if TsunamiConfigs exists and PS AnalyzeFrgndSpectra not found.

{TermTsuDetProc}

### **AnalyzeTsunamiRadials Not Running**

**Failure** if TsunamiConfigs exists and PS AnalyzeTsunamiRadials not found.

{TsuCfgMissing}

### **Tsunami Detection Configuration Missing File**

*Not Yet Implemented!*

{TsuBkRadialNotFound}

### **No Tsunami Background Radials**

*Not Yet Implemented!*

{TsuFrRadialNotFound}

### **No Tsunami Foreground Radials**

*Not Yet Implemented!*

{TsuBkRadialTooOld}

### **Tsunami Background Radial is Not Up-to-Date**

*Not Yet Implemented!*

{TsuBkRadialNoVectors}

### **Tsunami Background Radial Contains no Vectors**

*Not Yet Implemented!*

{TsuBkRadialFewVectors}

### **Tsunami Background Radial Contains Few Vectors**

*Not Yet Implemented!*

{TsuFrRadialTooOld}

### **Tsunami Foreground Radial is Not Up-to-Date**

*Not Yet Implemented!*

{TsuFrRadialNoVectors}

### **Tsunami Foreground Radial Contains no Vectors**

*Not Yet Implemented!*

{TsuFrRadialFewVectors}

### **Tsunami Foreground Radial Contains Few Vectors**

*Not Yet Implemented!*

{TsuDetMapNotFound}

**No Tsunami Detection Map Files**

*Not Yet Implemented!*

{TsuDetStripNotFound}

**No Tsunami Detection Strip Files**

*Not Yet Implemented!*

{TsuSimOn}

**Tsunami Simulation is Enabled (mode=#)**

*Not Yet Implemented!*

{TsuDetect}

**Tsunami Detection is Tripped (QF=#, TOS=#)**

*Not Yet Implemented!*