

AWIPS Data Monitor/Restart

Overview

AWIPS 4.3.3 includes a redesigned monitor/restart capability. New from the user's perspective is a Restart button at the bottom of the monitor screen. While the monitor part of the interface is unchanged, the underlying scripts and control files are significantly modified.

Data Monitor

The processes to be monitored and restarted are listed in a configuration file, `$FXA_DATA/data/fxa_monitor/monitorProcesses.txt`. Any changes that occur to the system (e.g., moving a process from one machine to another) need to be reflected in this file in order for the monitor and restart procedure to operate properly.

Additional process may be added to this file at any time and they will be monitored by the Process Monitor and potentially stopped or restarted by the restart procedure, provided that the necessary information is included. (Note that it is possible to include a process to be monitored but not stopped/restarted, or to stop and restart processes that are not monitored.)

Changes to `monitorProcess.txt` take effect immediately. There is no need to restart the monitor processes after editing.

The standard 4.3.3 `monitorProcesses.txt` file is shown here.

```
# Monitor Processes
#
# This file contains the list of processes that the process monitor will
# monitor and/or restart through ingProcMon.pl and the restart menu.
# The format is as follows:
#
# Multiple fields separated by the "|" character.
#
# i.e.
#
# Field 1 | Field 2 | Field 3 | Field 4 | Field 5 | Field 6 | Field 7 | Field
8
#
# Field 1
#     Executable name of process [-s stop script -r re-start script].
```

```

# The executable name is the "EXACT" name as used when one starts the
process.
# The name (or the startup script) must be unique within the first 45
# characters in order for it to be uniquely identified via the 'ps' command.
# Field 2
#     Machine on which process runs [as1, as2, ds].
# Field 3
#     Whether or not the process is monitored [Y(es), N(o)].
# Field 4
#     Monitoring Class [g(rid), generalized text(t1), specialized text(t2),
#                       s(atellite), p(oint), S(ystem), A(FOS), generalized
#                       radar processing (r1), dedicated radar(r2), dial
#                       radar (r3), m(essage handling system)].
# Field 5
#     Action [R(estart), S(top)]. (Blank implies that nothing is done.)
# The R or the S is followed by a number from 1 to 3 indicating the order in
# which certain processes are stopped or re-started relative to the other
# processes. Currently three priority levels are supported. Processes
# containing the number 1 following the R or S will be stopped or re-started
# first. Processes with the highest number are stopped or re-started last.
# Field 6
#     Site Identification [w(fo), r(fc), O(COUNS), C(ONUS)]
# Field 7
#     User ID [fxa, oper]
# Field 8
#     Directory where executable (or start-up script) resides.
# Required only if the process needs to be restarted or has a stop script
# associated with it. (Don't forget the '/' at the end!)
#
# e.g.
#
# MaritimeDecoder | as1 | Y | p | S2 |wrOC | fxa | /awips/fxa/bin/
#
# The above line means the following:
#     The Maritime Decoder runs on the as1;
#     It is monitored by the process monitoring software;
#     It is part of the point subsystem;
#     It is only to be stopped and is NOT to be restarted. In addition, it
is
#     stopped only after all S1 processes have been stopped;
#     The process runs at all sites; and
#     It runs under the fxa ID,
#     From the /awips/fxa/bin/ directory.
#
# Any time a process is moved to a different machine, this file will need
# to reflect this change in order for the restart to work.
#
acqserver -r startAcqServer |ds
|Y|ptlgs|S2R2|wrOC|fxa|/awips/fxa/bin/
afoscomsrv -r startAFOS |as1|Y|A
|S2R2|wrOC|fxa|/awips/fxa/bin/
asyncScheduler -s stopAsyncScheduler -r startAsyncScheduler |as1|Y|t2
|S2R2|wrOC|fxa|/awips/fxa/bin/
binLightningDecoder |as1|Y|p |S2 |wrC
|fxa|
caseArchiveServer |ds |Y|r1
|S2R2|wrOC|fxa|/awips/fxa/bin/

```

CollDBDecoder	as2 Y t1	S2
wrOC fxa		
CommsRouter COMMS_ROUTER	ds	Y pt1s
S2R1 wrOC fxa /awips/fxa/bin/		
CommsRouter GRID_ROUTER	ds	Y g
S2R1 wrOC fxa /awips/fxa/bin/		
DataController COMMS_ROUTER RadarController.config	as1 Y r1	
S1R2 wrOC fxa /awips/fxa/bin/		
DataController COMMS_ROUTER SatelliteController.config	ds	Y s
S1R2 wrOC fxa /awips/fxa/bin/		
DataController COMMS_ROUTER TextCont.config	as1 Y p	
S1R2 wrOC fxa /awips/fxa/bin/		
DataController COMMS_ROUTER TextCont2.config	as1 Y p	
S1R2 wrOC fxa /awips/fxa/bin/		
### DataController COMMS_ROUTER TextCont3.config		ds Y p
S1R2 wrC fxa /awips/fxa/bin/		
DataController COMMS_ROUTER TextDB_Controller.config	as2 Y t1	
S1R2 wrOC fxa /awips/fxa/bin/		
DataController COMMS_ROUTER WarnDB_Controller.config	as2 Y t1	
S1R2 wrOC fxa /awips/fxa/bin/		
DataController COMMS_ROUTER tStormController.config	as1 Y p	
S1R2 wrOC fxa /awips/fxa/bin/		
### DataController COMMS_ROUTER SCANcontroller.config	as1 Y p	
wrOC fxa /awips/fxa/bin/		
### DataController COMMS_ROUTER FFMPcontroller.config	as1 Y p	
wrOC fxa /awips/fxa/bin/		
DataController GRID_ROUTER GribController.config	ds	Y g
S1R2 wrOC fxa /awips/fxa/bin/		
dialRadar	ds	N r3 S1
wrOC fxa		
DialServer	ds	Y r3
S2R2 wrOC fxa /awips/fxa/bin/		
GribDecoder	ds	Y g S2
wrOC fxa		
MaritimeDecoder	as1 Y p	S2
wrOC fxa		
MetarDecoder	as1 Y p	S2
wrOC fxa		
MhsRequestServer	ds	Y m
S2R2 wrOC fxa /awips/fxa/bin/		
MhsServer	ds	Y m
S2R2 wrOC fxa /awips/fxa/bin/		
notificationServer	as1 Y S	
S2R2 wrOC fxa /awips/fxa/bin/		
profilerDecoder	as1 Y p	S2
wrOC fxa		
RadarMsgHandler	as1 Y r1	
S2R2 wrOC fxa /awips/fxa/bin/		
RadarServer	ds	Y r1
S1R1 wrOC fxa /awips/fxa/bin/		
RadarStorage	as1 Y r1	S2
wrOC fxa		
RadarTextDecoder	as2 Y t1	S2
wrOC fxa		
RaobBufRDecoder	as1 Y p	S2
wrOC fxa		

```

RedbookStorage                |as1|Y|t1   |S2
|wrOC|fxa|
RMR_Server                    |ds  |Y|r3
|S1R2|wrOC|fxa|/awips/fxa/bin/
Satdecoder                    |ds  |Y|s     |S2
|wrOC|fxa|
shefdecode -s stop_shefdecode  -r start_shefdecode |ds  |Y|t1   |
|wrOC|oper|/awips/hydroapps/shefdecode/bin/
StdDBDecoder                  |as2|Y|t1   |S2
|wrOC|fxa|
syncComms                     |ds  |N|r1   |S1
|wrOC|fxa|
TextDB_Server -Read           |ds  |Y|t1
|S2R2|wrOC|fxa|/awips/fxa/bin/
TextDB_Server -Write          |ds  |Y|t1
|S2R2|wrOC|fxa|/awips/fxa/bin/
textNotificationServer        |as1|Y|t1
|S1R1|wrOC|fxa|/awips/fxa/bin/
tStormDecoder                 |as1|Y|p     |S2
|wrOC|fxa|/awips/fxa/bin/
### SCANprocessor              |as1|Y|p     |
|wrOC|fxa|
### FFMPprocessor              |as1|Y|p     |
|wrOC|fxa|
WarnDBDecoder                 |as2|Y|t1   |S2
|wrOC|fxa|
wfoApi                        |ds  |Y|r1   |S2
|wrOC|fxa|

```

Figure 1. monitorProcesses.txt

Restart

The restart mechanism will selectively stop and restart processes running on the AS1, AS2, and DS machines. The restart procedure is initiated via the Data Monitor/Process Monitor GUI. This GUI is run through the Netscape Browser interface. The monitor keeps track of the state of the data and the ingest processes. The Process Monitor components updates themselves every minute while refreshing the page every 30 seconds.

Restart can be configured to define a process hierarchy. What this means is that it can be set up to stop and then restart processes in a certain order so that dependencies between processes within the system can be maintained. In this way restart will stop certain processes first before stopping the remaining processes. In the same manner, certain processes can be restarted first. For example, a communication processes can be up and running before the rest of the system is restarted. **Note:** Use of this hierarchy mechanism should be addressed with caution, as additions may affect existing priority assignments.

Near the bottom of the Process Monitor page one finds a [Restart](#) button which when clicked brings up a menu (Figure 2) offering the user several choices as to which subsystem to restart. (The screen on which the menu appears is determined by environment variables `FXA_RESTART_WORKSTATION` and `FXA_RESTART_SCREEN`, in `as1:/awips/fxa/environs`. By default, this is `ws1:0.0`.)

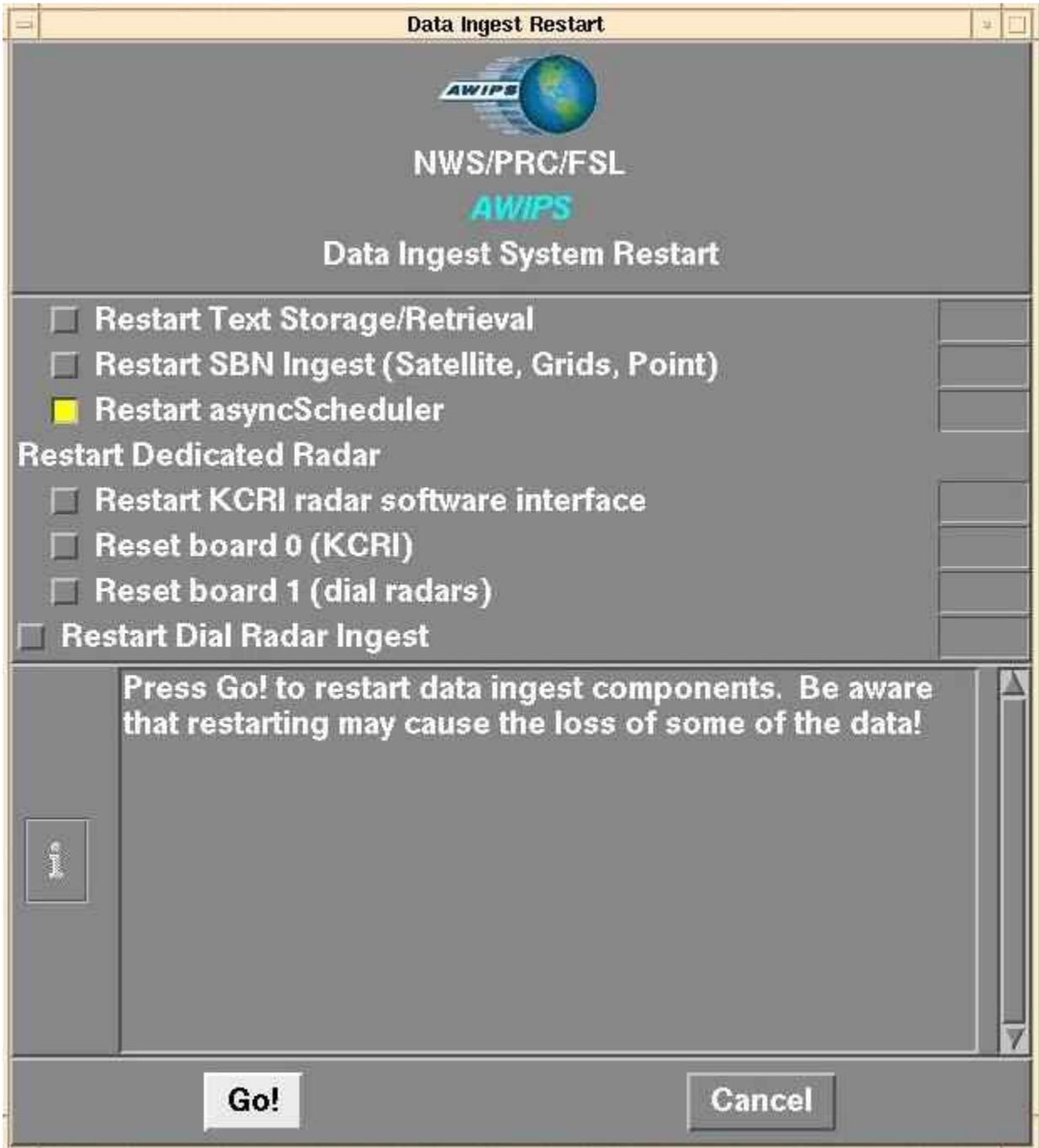


Figure 2. Restart menu

Since many processes interact with one another, executing the top three items will ensure that everything is working again. It also has the potential of causing the loss of the greatest amount of data, since all the processes are restarted. The more restrictive options are used when the user is confident that the problems exist within that particular subsystem and when it is obvious that restarting the entire data ingest system is unnecessary.

Radar - The choices in the Restart Dedicated Radar section are based on the local site's radar ingest configuration. A 'Restart *radar* software interface' option appears for each radar defined in file `/awips/fxa/data/localizationDataSets/$FXA_INGEST_SITE/portInfo.txt`, where *radar* is the actual ID of the radar. This option will restart the `wfoApi` process for that radar. Additionally, one or two more options will appear that state 'Reset board *n* (*radar1*, *radar2*)', etc. These choices reload the firmware to the Simpack board. Again, *radar1*, *radar2* will be the actual name(s) of the radar(s) listed in `portInfo.txt`. The first item will affect all radars on board 0 and the second will affect all radars on board 1. The Dial Radar option, as the name implies, restarts processes associated with dial-out radar.

Once the appropriate option is selected, another dialog box comes to the screen asking for input as to who is doing the restarting and the reason for the restart. This information is written to `$LOG_DIR/<yyyymmdd>/restart.log` for the purpose of documenting the event. Other restart* log files in the same location record the progress of the restart process.

Figure 3 gives an overview of the Restart Procedure on the AWIPS system.

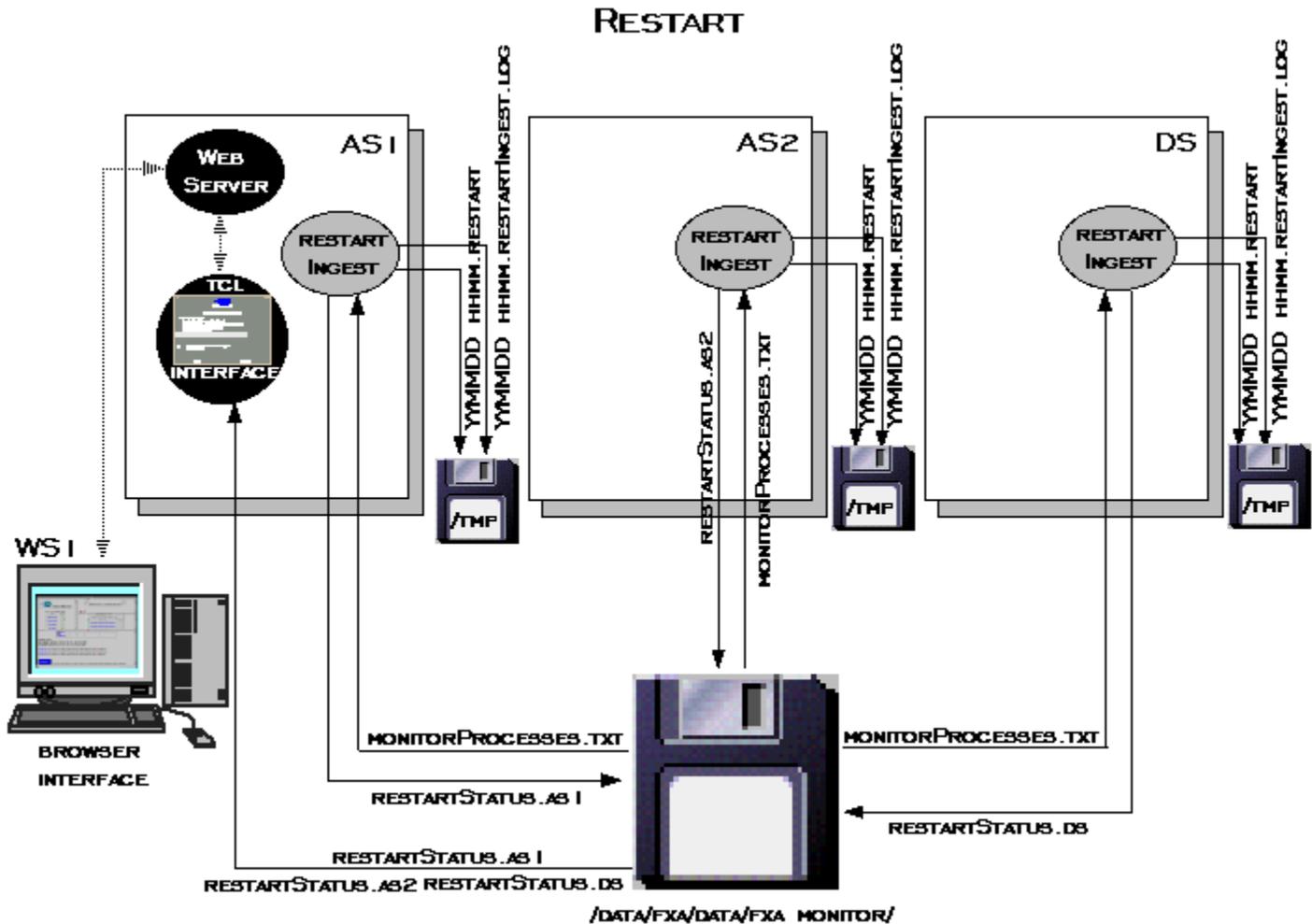


Figure 3. Restart Overview

After the documentation information is entered, restart begins on the three servers. The processes on the AS1 server are stopped and restarted almost immediately; however, the restart process needs to remotely log onto the AS2 and DS machines to stop and restart processes locally on those two machines. This remote logging-on requires some additional time and a message appears on the restart menu window indicating this situation. Once everything is on its way, the menu window updates itself indicating that the restart is underway ("busy" indicator next to the option) and a message appears indicating approximately how long the process should last. When completed, the status of the procedure is written to the browser window stating if the restart procedure was successful or whether it failed. If the restart runs, and the restart process is able to send termination signals to the processes, the message would indicate success. Were the restart process unable to send a termination signal (due to some underlying UNIX system problem), the restart process would terminate itself, and indicate that restart was aborted. In this case, the user is instructed to try again. If this second try is unsuccessful, then there is some inherent problem with the machines requiring system administrator intervention. At this point the NCF should be contacted if possible to address the problem.

A "successful" status message may not necessarily indicate that processes were stopped or restarted correctly. It just means that the interrupt signals were sent successfully and that the operating system will deliver them to the intended processes. There are times when, depending upon the state of a given process, it may ignore the interrupt signal that was sent to it. The true test of whether or not a process is up and running again is watching the Ingest Processes display and making sure that only green check marks appear and that there are no red Xs. Keep in mind that since the Process Monitor updates itself only every 60 seconds, one needs to wait for two refresh cycles to occur in order to be sure that the monitor is showing the true state of the system. If the red Xs remain after several refresh cycles, try restarting again and if things still don't return to normal after a second restart, notify the NCF.

Authors: Ron Kahn and Joe Wakefield
Last update: 17 Nov 00