

AWIPS-II Localization Companion

11. August 2010

TABLE OF CONTENTS	
Document Overview	Page 3
Outside Document References	Page 3
Document Variable Conventions	Page 3
Ingest Localization	Page 4
CPSBN Upstream Localization	Page 5
PX Downstream Localization	Page 5
Data Encryption	Page 6
EDEX Localization	Page 6
EDEX Data Filtering	Page 6
EDEX Databases	Page 7
FFMP shape file import	Page 8
afos2awips import	Page 9
EDEX Configuration Files	Page 9
common_static directory	Page 9
radarsInUse.txt	Page 14
edex_static directory	Page 17
GFE localConfig.py	Page 18
Text Triggers	Page 23
start.sh	Page 24
CAVE Localization	Page 25
CAVE Core Plugin Configuration File	Page 25
CAVE Map Customization	Page 28
CAVE Satellite Menu	Page 33
CAVE Upper Air Menu	Page 34
CAVE NCEP/Hydro Menu	Page 34
CAVE Warnings Display	Page 35
CAVE Warngen	Page 36
Warngen Templates	Page 36
Values In .cfg Files	Page 36
Values Passed To .vm Templates	Page 39
Editing Templates	Page 43

[CAVE D2D Procedures](#)

Page 45

[AVNFPS](#)

Page 46

OVERVIEW

Localization adapts (e.g., configures) the AWIPS national baseline software to the unique data and ConAps requirements of the site. AWIPS II performs localization dynamically at system startup using data from the localization data environment.

This document is designed as a guide to changing specific files in the localization environment to configure an EDEX installation to a specific site, as well as adding localization files needed for launching and running client applications such as CAVE and AlertViz. It has incorporated Raytheon document AWP.DOC.SWCTRTO11.LOC-01_Dft.

This document focuses on a SERVER-CLIENT localization paradigm, much like exists on AWIPS I. The steps taken should be those enough to guide a person familiar with AWIPS I systems to create a usable AWIPS II localization for an EDEX and CAVE instance.

It is a general note that you should **never edit files that are in a base directory tree**. Although as of the date of this document's creation, there are a few exceptions, anything beneath a directory named base in the configuration tree should never be edited. This is analogous to the nationalData directory in AWIPS I.

REFERENCES

Raytheon Document AWP.DOC.SWCTRTO11.LOC-01_Dft
Raytheon Document AWIPSII_Installation_Document

VARIABLE CONVENTIONS TABLE

<code>\${DATABASE_INSTALL_LOCATION}</code>	The location chosen for the database engine flat files and configuration files. Baseline: /awips2/db
<code>\${EDEX_INSTALL_LOCATION}</code>	The location chosen during install as the install root. Baseline: /awips2
<code>\${EDEX_HOME}</code>	The location chosen for the AWIPS-II software install, followed by the sub-directory edex. Baseline: /awips2/edex
<code>\${PATH_TO_SHARED_DRIVE}</code>	Any path to an NFS or other network shared drive which can be accessed on both the EDEX server, and the EDEX database server. Suggestion: /data/local

LOCALIZATION DATA TREE DESCRIPTION

CAVE plug-ins that are designed to be configurable have a `config.xml` file. The data in any `config.xml` can be overridden by the site. Likewise, each user can have overrides to the site `config.xml`. The utility service in EDEX copies the site and user overrides to the local CAVE instance into the `caveData` tree in the user's home directory during CAVE startup. The `caveData` tree is a local cache of localization data and improves the startup performance of CAVE after the initial startup. To force CAVE to reinitialize its localization, `caveData` can be deleted. Instructions on where and how to set this up are discussed below.

Specific CAVE plug-ins are able to be customized using the base/site/user hierarchical localization pattern. Each listed CAVE plug-in has a unique `config.xml` which is site/user localizable. The location of the site/user versions are on the EDEX utility data tree and need to be created for a new localization as follows:

```
${EDEX_HOME}/data/utility/cave_config/site/XXX/plugin_name/  
${EDEX_HOME}/data/utility/cave_config/user/UUU/plugin_name/
```

A list of configurable CAVE plugins can be found in the CAVE localization section.

CAVE general purpose data can be localized from the following location:

```
${EDEX_HOME}/data/utility/cave_static/site/XXX/{data item}  
${EDEX_HOME}/data/utility/cave_static/user/UUU/{data item}
```

General purpose data is defined as bundles/scales, colormaps, GFE text formatter combinations and override python files and menu entries.

Common data (data common to all areas of AWIPS II software) can be localized in the following locations:

```
${EDEX_HOME}/data/utility/common_static /site/XXX/  
${EDEX_HOME}/data/utility/common_static/user/UUU/
```

Common data is defined as GFE text files and edit areas, hydro Apps_defaults files, radar information for menu building and some text product files such as `afosMasterPil.txt`, `textCategoryClass.txt`, `textCCChelp.txt`, `textNNNhelp.txt` and `textOriginTable.txt`.

EDEX only data can be localized in the following location

```
${EDEX_HOME}/data/utility/edex_static/site/XXX/
```

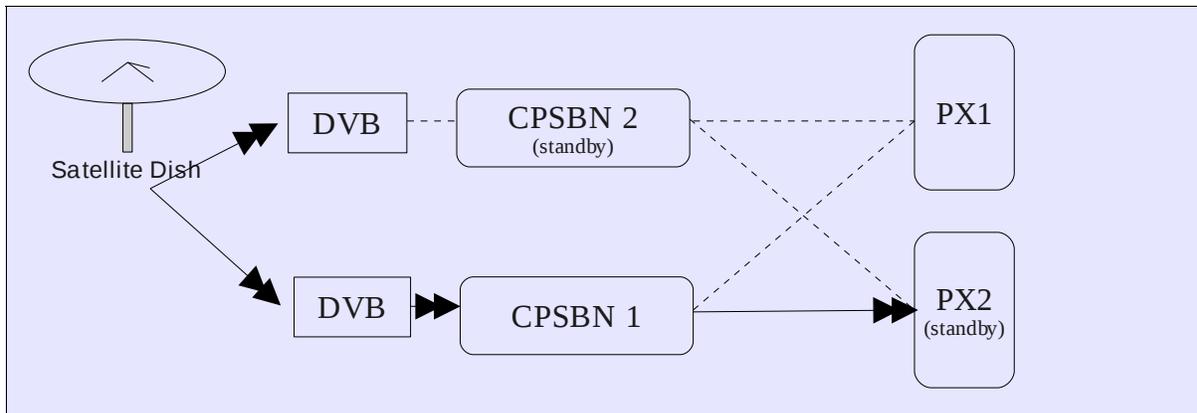
EDEX only data is centered around data ingest and GFE server duties. For example, data filters, distribution ingest patterns, GFE localConfig and siteConfig, and subgridding.

INGEST LOCALIZATION

Currently in AWIPS I, to localize ingest for a site, a user would have to consider filter patterns on each CPSBN machine, as well as the `acq_patterns.txt` file in `/awips/fxa/data` on DX3 and DX4. With AWIPS II, LDM is used as a primary ingest method. This eliminates filtering at the CPSBN level and adds filter files on the PX1 and PX2. For more information

and background on the LDM system than what is provided here, please visit <http://www.uidata.ucar.edu/software/ldm>

Overall the data flow will look similar to this diagram:



CPSBN Ingest (Upstream LDM)

The architecture runs an upstream LDM host on CPSBN1/2 directly connected to their respective DVB receivers. It allows an unfiltered full SBN stream into the system. This is analogous to the `acq_clients` on AWIPS I CPSBN machines. Currently, one CPSBN controls ingest of all data streams, unlike AWIPS I where the SBN and NWSSTG streams are spread between two physical machines.

To install and configure LDM on the CPSBN servers, reference Raytheon Document *AWIPSII_Installation_Document*.

PX Acquisition Ingest (Downstream LDM)

A downstream LDM client runs on PX2 in conjunction with the `a2px2apps` package and is used to filter the full SBN stream from the CPSBN and customize it to each site's needs. To localize which data is ingested, a site must edit the `/usr/local/ldm/etc/pqact.conf` file on px2 and px1. If unfamiliar with this file, it is highly recommended that the documentation on the LDM website be read before editing. This can be found as of the date on this document at the URL <http://www.unidata.ucar.edu/software/ldm/ldm-6.6.5/basics/pqact.conf.html>.

Once the `pqact.conf` file has been localized, it should be checked for syntax accuracy. After a successful check, it can be re-read with a restart of the LDM processes or by sending a HUP signal to the LDM processes which causes a re-read of the configuration files. As user root on the PX2, issue one of the following sets of commands:

```
# su - ldm
# ldmadmin pqactcheck
    NOTE: This should return the string "syntactically correct"
# ldmadmin stop && ldmadmin start
```

OR

```
# su - ldm
```

```
# ldmadmin pqactcheck
    NOTE: This should return the string "syntactically correct"
# ldmadmin pqactHUP
```

The second method is preferred, as it will not stop the processes.

Once verified, be sure to copy the pqact.conf file to the px1 in order to ensure changes take effect after a failover.

Encryption Key

It is important to note some data which is provided via the SBN is encrypted. In AWIPS-I, the decryption is handled by the software running on the CPSBN. This decryption, in AWIPS II, is handled through the downstream LDM client running on the PX2F server. Instructions are included in Raytheon Document AWIPSII_Installation_Document under the *Configuration of Downstream LDM Client* on installing the key on the PX1 and PX2 server. This will affect ECMWF and ACARS data.

NOTE: ECMWF Low-Res data is not encrypted

EDEX LOCALIZATION

While not all files in this section are specific to EDEX, the steps taken here will localize the server side EDEX system to a site ID other than OAX.

NOTE: These steps are not necessary to do for your backup site. You'll be able to provide service backup without localizing EDEX to your backup site. This is no different than AWIPS I, which does not require a localization to be run on your servers in order to perform service backup.

EDEX Decoding Filters (distribution xml)

If changes were made to the pqact.conf file, these changes may need to be reflected in an EDEX configuration file in order for the data which is now ingesting to be recognized and processed by the respective EDEX plugins. The LDM software is configured to use a bridge to send over information about which products it has ingested, and into what directory it has flushed the product to a file. This information is sent over in a shared memory segment to the EDEX server via the AMQP message broker QPID. Based on the information provided by LDM, EDEX checks which plugin is interested in this product, and checks to see whether the product should be decoded against patterns inside of xml files in the following location:

`$EDEX_HOME/data/utility/edex_static/base/distribution/`

An example of a distribution file is listed below. This is the default text.xml, and any product which doesn't match one of these regular expressions will not be decoded by the text decoder plugin.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<requestPatterns xmlns:ns2="group">
  <regex>^S[AP].*</regex>
  <regex>^FXUS.*</regex>
  <regex>^FOUS.*</regex>
  <regex>^AS.*</regex>
  <regex>^FPUS.*</regex>
  <regex>^W[UFWG]US.*</regex>
  <regex>^F[TC][UX][SX].*</regex>
</requestPatterns>
```

After you have finished matching the distribution xml files to your pqact.conf file, a restart of EDEX will be necessary.

NOTE: The utility tree is shared between all servers running EDEX. You no longer have to copy changed files to all separate servers running EDEX. However, an EDEX restart will need to be performed on each individual server in order for changes to take effect.

Databases (PostgreSQL configuration)

As of TO11, there are databases specific to site OAX loaded on install. A site hoping to run an EDEX fully localized as any site other than OAX should load the last good backup of their AWIPS I databases for the following:

```
hd_ob92xxx (where xxx=site ID)
dc_ob7xxx (where xxx=site ID)
hmdb
lsrdata
```

To load these databases into the AWIPS II PostgreSQL engine, after installing the database server using the Raytheon Document AWIPS_Cluster_Deployment_FlowTag_RevTO11FIT.doc, issue the following commands on DX1-CCC where the AWIPS II database engine is installed and currently running:

```
# createdb -U awips hd_ob92xxx (where xxx=site ID)
# createdb -U awips dc_ob7xxx (where xxx=site ID)
# createdb -U awips lsrdata
# dropdb -U awips hmdb
# createdb -U awips hmdb
```

NOTE: The default password for the database user awips is awips.

Once the databases have been created, the `pg_hba.conf` must be edited to allow trusted connections to these particular databases. Add lines similar to the following (which are an example corresponding to the above examples) on your EDEX Database Server. The path for the `pg_hba.conf` file is the path chosen for the database location during the izPack

installer.

```
# su - awips
# cd ${DATABASE_INSTALL_LOCATION} [ e.g. cd /awips2/db ]
# vim pg_hba.conf
```

Add lines similar to these, but appropriate to the new site's databases, substituting the correct subnet for XXX below:

```
host hd_ob92xxx all 165.92.XXX.024 trust
host dc_ob7xxx all 165.92.XXX.0/24 trust
host lsrdata all 165.92.XXX.0/24 trust
```

```
# :wq! (saves and exits VIM session)
# pg_ctl reload -D ${DATABASE_INSTALL_LOCATION}
[ e.g. pg_ctl reload -D /awips2/db ]
# exit (returns to root shell)
```

At this point, the data from the AWIPS-I database dumps can be loaded into the newly created databases with the following commands:

```
# su - awips
# pg_restore -U awips -d <db_name> -Fc <path_to_backup_file>
```

The <path_to_backup_file> is the default location for AWIPS-I database backups when created by the `backup_pgdb` script. The location of that script's backup files is `/data/fxa/DAILY_BACKUP/postgres/<DAY_OF_WEEK>`

Repeat the `pg_restore` command substituting the names of the databases and their respective backup file locations (including the backup file name) in the appropriate spot.

Import Shape Files For FFMP

FFMP basin files are stored within the maps database in the AWIPS II PostgreSQL engine running on DX1. A tool is provided to import the shape files from current ESRI shape files, which are at least three files: .shp, .shx, and .dbf extensions. To import the new site's FFMP basins shape files, find the `FFMP_aggr_basins.shp`, `FFMP_aggr_basins.shx`, and `FFMP_aggr_basins.dbf` in `/data/fxa/nationalData` as well as the FFMP Streams from `FFMP_ref_sl.shp`, `FFMP_ref_sl.shx` and `FFMP_ref_sl.dbf`.

NOTE: If the shp files are gzipped, you'll need to run `gunzip` on them before running the `importShapeFile.sh` script

After locating the appropriate shape files, ensure they are in a location accessible by your EDEX Database Server. Then run the following tool, which will drop and reload the `ffmp_basins` and `ffmp_streams` map tables respectively:

```
# su - awips
# cd ${EDEX_INSTALL_LOCATION}/postgres/share/sql
# ./importShapeFile.sh ${FFMP_BASIN_SHAPE_FILE} mapdata ffmp_basins
0.064,0.016,0.004,0.001 awips 5432 ${EDEX_INSTALL_LOCATION}
```

```
# ./importShapeFile.sh ${FFMP_STREAM_SHAPE_FILE} mapdata fmp_streams  
0.064,0.016,0.004,0.001 awips 5432 ${EDEX_INSTALL_LOCATION}
```

NOTE: `$EDEX_INSTALL_LOCATION` above is the location chosen during the IzPack installer for the install location, and is normally the root for the `$EDEX_HOME`. For example, if `$EDEX_HOME` is `/awips2/edex`, then `$EDEX_INSTALL_LOCATION` would be `/awips2`

NOTE: You may be asked multiple times for the password for user `awips`, which is also `awips`

NOTE: The loading of the shape files may take as much as 30 minutes, or longer.

Import afos2awips.txt Into AWIPS-II Database

The information which is contained within the `/awips/fxa/data/afos2awips.txt` file in AWIPS-I now is contained in the metadata database for use by AWIPS-II processes. To import a custom version of `afos2awips.txt` into the metadata database, run the following commands:

Run from your primary EDEX server

(each `#` denotes a new command. Otherwise the commands continue across multiple lines)

```
# su - awips  
# cd ${EDEX_HOME}/bin  
# ./afos2awipstranslator.sh ${PATH_TO_AWIPS-I_AFOS2AWIPS}/afos2awips.txt > $  
{PATH_TO_SHARED_DRIVE}/afos2awips.sql  
# exit
```

NOTE: At this point if EDEX is running, it must be halted. Do so before running the database import in the following instructions.

Run from your database server

```
# su - awips  
# psql -U awips -d metadata -f ${PATH_TO_SHARED_DRIVE}/afos2awips.sql  
# exit
```

Restart EDEX after successful run of these commands.

EDEX Configuration Files

There are certain files which need to be localized in order for EDEX to function properly. Below is an attempt to identify which files need to be changed and in what manner.

`$EDEX_HOME/data/utility/common_static/site/XXX`

Create the directory listed above, replacing XXX with the site ID selected for localization (e.g. LWX) using a command similar to the following on the EDEX server:

```
# cd ${EDEX_HOME}/data/utility/common_static/site
# mkdir XXX (replace XXX with the site ID of the chosen localization site)
# chown awips:awips XXX
# chmod 755 XXX
```

Create, also, the list of sub-directories needed for subsequent steps:

```
# cd XXX
# mkdir ffmp [ new with R1G1 ]
# mkdir gfe
# mkdir goessndg
# mkdir hydro [ changed with R1G1 ]
# mkdir modelsndg
# mkdir poessndg
# mkdir radar [ changed with R1G1 ]
# mkdir shef [ new with R1G1 ]
# mkdir upperair [ new with R1G1 ]
# mkdir warngen
# chown -R awips:awips *
# chmod -R 755 *
```

Inside the site level XXX sub-directories, create the following files as specified below:

■ **ffmp**

This directory contains three files in the delivered OAX directory:

FFMPRunConfig.xml
FFMPSourceConfig.xml
FFMPTemplateConfig.xml

FFMPRunConfig.xml and FFMPTemplateConfig.xml are created by EDEX using the information found in FFMP section of the following file:

`\$EDEX_HOME/data/utility/common_static/site/XXX/hydro/hydroSitesInUse.txt`

Further details on the use of these configuration files will be provided in subsequent updates to this document.

■ **fog/monitoringArea/monitorAreaConfig.xml**

NOTE: As of AWIPS-II Build R1G1, this file and directory structure are not necessary to create. If the files exist, then EDEX will make use. If no configuration exists, it will default configure itself based upon the WFO location.

This holds true for Fog, Snow, and Safeseas configurations.

This file is created in the following format and lists county zones which are monitored by the fog monitor portion of EDEX.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MonitorAreaConfig xmlns:ns2="group">
  <AreaID type="REGULAR" id="IA043"/>
  <timeWindow>0</timeWindow>
  <shipDistance>0</shipDistance>
  <useAlgorithms>>false</useAlgorithms>
</MonitorAreaConfig>
```

This is similar to the AWIPS I file in `/data/fxa/workFiles/fog_monitor/setup` named `fm_ZoneCountyList.dat`. For each zone listed, create a new `<AreaID>` entry like entry in bold and red text above.

To easily parse out the zones and echo statements that can be cut and paste into the xml file, this is an example which can be used:

```
cat fm_zoneCountyList.dat | cut -f2 -d'|' | awk '{print $1}' | while read zone
do
  echo -e "\t<AreaID type=\"REGULAR\" id=\"$zone\"/>"
done
```

Ensure the proper ownership and permissions of the file:

```
# chown awips:awips monitorAreaConfig.xml
# chmod 755 monitorAreaConfig.xml
```

■ **gfe/**

This directory just needs to be created, not populated. It's contents are created on start of EDEX. Inside will reside GFE edit area groups, edit areas and sample sets.

The following sub-directories should be created on first start:

```
editAreaGroups
editAreas
sampleSets
```

- **goessndg/goesBufr.spi**

Take the `goesBufr.spi` file from the site's `/awips/fxa/data/localizationDataSets/XXX` directory and copy it to the goessndg sub-directory.

Ensure the file has the following ownership:

```
user:      awips
group:    awips
permissions: 755
```

The following commands will ensure that it is set, when running from the directory containing the goessndg directory.

```
# chown -R awips:awips goessndg
# chmod -R 775 goessndg
```

- **hydro/Apps_defaults**

This file is analogous to the AWIPS I file `/awips/hydroapps/.Apps_defaults_site` file. It is the same in format and function for any re-engineered or wrapped hydro applications. The `/awips/hydroapps/.Apps_defaults_site` may be copied into the `hydro/Apps_defaults` file within the `common_static/site/XXX` directory tree with the following command (all on one line):

```
# cp /awips/hydroapps/.Apps_defaults_site
$EDEX_HOME/data/utility/common_static/site/XXX/hydro/Apps_defaults
```

Remember to substitute the EDEX install location for `$EDEX_HOME` and your site ID for `XXX`

The following other files are found in the OAX delivered directory and will be created on first EDEX start:

```
hydro/hydroapps/ffmp_templates
hydro/hydroapps/geo_data
hydro/hydroapps/whfs
```

- **hydro/hydroSitesInUse.txt**

This file is utilized to configure EDEX for FFMP and FOG and build appropriate CAVE menus for NCEP/Hydro. These entries used to be in the `radarsInUse.txt` file, but those entries are no longer utilized. This file should contain the following sections:

```
# DO NOT EDIT LINES BEGINNING WITH '#'
# FFMP - MUST HAVE THIS LINE
```

This entry is used to create the *RFCCFG* menus under NCEP/Hydro

```
# FOG - MUST HAVE THIS LINE  
  
top level menu in the D2D  
perspective inside of CAVE  
  
Format:  
radarid:CWA:RFCID  
  
Example:  
klwx:LWX:KRHA  
  
Output to:  
$EDEX_HOME/data/utility/cave_static/  
site/XXX/menus/hydro/index.xml  
Creates the Fog Monitor menus  
under the D2D perspective  
inside CAVE.  
  
Output to:  
$EDEX_HOME/data/utility/cave_static/  
site/XXX/menus/fog/index.xml
```

Ensure that both sub-directory and file have the following ownership:

```
user:      awips  
group:     awips  
permissions: 755
```

The following command can be used to achieve these permissions when currently in the common_static/site/XXX path:

```
# chown -R awips:awips hydro  
# chmod -R 755 hydro
```

■ **modelsndg/modelBufr.spi**

Take the `modelBufr.spi` file from an AWIPS-I localization store in the directory `/awips/fixa/data/localizationDataSets/XXX` and copy it into a modelsndg sub-directory of the common_static/site/XXX directory tree.

Ensure that both sub-directory and file have the following ownership:

```
user:      awips  
group:     awips  
permissions: 755
```

The following command can be used to achieve these permissions when currently in the common_static/site/XXX path:

```
# chown -R awips:awips modelsndg  
# chmod -R 755 modelsndg
```

■ **poessndg/poesBufFr.spi**

Take the *poesBufFr.spi* file from an AWIPS-I localization store inside of the */awips/fxa/data/localizationDataSets/XXX* directory and copy it into a poessndg sub-directory inside the common_static/site/XXX directory tree.

Ensure that both sub-directory and file have the following ownership:

```
user:      awips
group:    awips
permissions: 755
```

The following command can be used to achieve these permissions when currently in the common_static/site/XXX path:

```
# chown -R awips:awips poessndg
# chmod -R 755 poessndg
```

■ **radar/radarsInUse.txt and radar/eavConfigTable.txt**

Inside of the radar sub-directory, create a radarsInUse.txt file. This file is used for creating specific menus in the D2D perspective of CAVE dynamically. The following are a list of menu items are created using this file as of TO11DR10 by listing radar IDs below the following lines:

```
# LOCAL_RADARS (including
terminal) - MUST HAVE THIS LINE
                                Sets up stand-alone dedicated radar
                                menus in main menu of D2D perspective.
                                Also builds SCAN menus for dedicated
                                radars

                                Output to
                                $EDEX_HOME/data/utility/cave_static/si
                                te/XXX/menus/radar/index.xml

                                Output to
                                $EDEX_HOME/data/utility/cave_static/si
                                te/XXX/menus/scan/scan.xml
LINE CHANGED IN R1G1

# DIAL_RADARS - MUST HAVE THIS
LINE
                                Sets up dial radar menus for each
                                radar listed below this line. Also
                                build SCAN menus for dial radars

                                Output to
                                $EDEX_HOME/data/utility/cave_static/si
                                te/XXX/menus/radar/dialRadars.xml

                                Output to
                                $EDEX_HOME/data/utility/cave_static/si
                                te/XXX/menus/scan/scan.xml
LINE CHANGED IN R1G1
```

<i># Terminal Radars - MUST HAVE THIS LINE</i>	<i>Sets up a stand-alone radar in main menu of D2D perspective.</i>
	<i>Output to \$EDEX_HOME/data/utility/cave_static/site/XXX/menus/radar/airportRadars.xml</i>
<i># FFMP Radars - MUST HAVE THIS LINE</i>	<i>Sets up the FFMP launchers under the SCAN menu for each radar listed below this line.</i>
	<i>NOTE:</i> <i>any radar listed under the RadarServer and Dial Radar sections automatically have these menus created, so it is not necessary to list others here that are already in those sections.</i>
	<i>Output to \$EDEX_HOME/data/utility/cave_static/site/XXX/menus/radar/scan.xml</i>
<i># ASR_RADARS - MUST HAVE THIS LINE</i>	<i>Sets up ASR radar menu for each radar listed below this line.</i>
	<i>Output to \$EDEX_HOME/data/utility/cave_static/site/XXX/menus/scan/airportRadars.xml</i>
<i>LINE CHANGED IN R1G1</i>	
<i># ARSR_RADARS - MUST HAVE THIS LINE</i>	<i>Sets up ARSR radar menu for each radar listed below this line.</i>
	<i>Output to \$EDEX_HOME/data/utility/cave_static/site/XXX/menus/radar/airportRadars.xml</i>
<i>LINE CHANGED IN R1G1</i>	
<i># MOSAIC_RADARS - MUST HAVE THIS LINE</i>	<i>Sets up the radars included in the default mosaic menu selection.</i>
<i>LINE CHANGED IN R1G1</i>	
<i># CWA - MUST HAVE THIS LINE</i>	<i>Used within FFMP for determining which CWAs to monitor for input data.</i>
<i># FFMP_RUN - MUST HAVE THIS LINE</i>	<i>Different from FFMP Radars, these entries create the RFCFFG menus under NCEP/Hydro top level menu in the D2D perspective inside of CAVE</i>
	<i>Format: radarid:CWA:RFCID</i>
	<i>Example: klwx:LWX:KRHA</i>
	<i>Output to:</i>

```
LINE CHANGED IN R1G1, NO LONGER USED      $EDEX_HOME/data/utility/cave_static/site/XXX/menus/hydro/index.xml  
  
# FOG_RUN - MUST HAVE THIS LINE           Creates the Fog Monitor menus under the D2D perspective inside CAVE.  
  
Output to:  
$EDEX_HOME/data/utility/cave_static/site/XXX/menus/fog/index.xml  
LINE CHANGED IN R1G1, NO LONGER USED
```

The OAX version of this file can be used as a template.

The eavConfigTable.txt is exactly as it was in AWIPS-I. Find a copy in /data/fxa/nationalData and copy it into common_static/site/XXX/radar with the following command (all on one line).

```
# cp /data/fxa/nationalData/eavCofigTable.txt  
$EDEX_HOME/data/utility/common_static/site/XXX/radar
```

Ensure that both sub-directory and files have the following ownership:

```
user:      awips  
group:     awips  
permissions: 755
```

The following command can be used to achieve these permissions when currently in the common_static/site/XXX path:

```
# chown -R awips:awips radar  
# chmod -R 755 radar
```

■ **shef/metar.cfg**

This file is analogous to the following AWIPS I file:

```
/awips/hydroapps/whfs/local/data/log/metar2shef/metar.cfg
```

It can be used as a template in configuring the AWIPS II version. The OAX delivered metar.cfg should be used as a starting point, and the first three lines (in, out and err locations) should not be changed.

■ **upperair/raobSitesInUse.txt**

Used to configure upper air menus. The file consists of two main sections delineated by the following strings which always need to be present, and must appear exactly as shown. For this reason, it is suggested the delivered file with the OAX localization should be used as a template:

```
# UPPER_AIR menu  
# LOCAL_UPPER_AIR submenu
```

Under each would be entries for specific RAOB sites desired to show up under each specific menu. Each entry is in the format as follows:

4-Letter RaobID Raob WMO ID Raob Location

for example:

```
KOAX 72558 Omaha, NE
```

Each section is separated by a space. The RAOB WMO ID can be found at the following web address: <http://rucsoundings.noaa.gov/raob.short>

■ **warngen/**

See Warngen Templates in this document

```
$EDEX_HOME/data/utility/edex_static/site/XXX
```

If it does not already exist create the above directory, *replacing XXX with the new site ID (e.g. LWX)*. Create the following sub-directories underneath the first:

```
# cd ${EDEX_HOME}/data/utility/edex_static/site  
# mkdir XXX (replace XXX with the site ID of the chosen localization site)  
# chown awips:awips XXX  
# chmod 755 XXX
```

Create, also, this list of sub-directories needed for subsequent steps:

```
# cd XXX  
# mkdir -p config/gfe  
# mkdir -p grib/subgrids  
# mkdir plugin-filters  
# chown -R awips:awips *  
# chmod -R 755 *
```

Inside the site level XXX sub-directories, create the following files specified below:

■ **config/gfe/siteConfig.py**

File is re-usable from AWIPS-I, however as of TO11DR10 it is suggested that at a minimum for testing to copy the file from the OAX localization tree and change the bold

red sections in the following to mimic that for the chosen localization site:

```
#THIS FILE GENERATED AUTOMATICALLY DURING GFE INSTALL **
import os, socket

GFESUITE_HOME = os.environ['EDEX_HOME']+"/data/gfe"
GFESUITE_SERVER = "localhost"
GFESUITE_PORT = '98000000'
GFESUITE_SITEID = 'OAX'
GFESUITE_PRDDIR = GFESUITE_HOME+"/products"
GFESUITE_MHSID = 'OAX'
```

Ensure that both sub-directory and file have the following ownership:

```
user:      awips
group:     awips
permissions: 755
```

The following command can be used to achieve these permissions when currently in the edex_static/site/XXX path:

```
# chown -R awips:awips gfe
# chmod -R 755 gfe
```

■ **config/gfe/localConfig.py**

File is re-usable from AWIPS-I, however take careful note that any specific pointers to /data/fxa will be invalid in this configuration. The current AWIPS I location of the file is, from DX4, /awips/GFESuite/primary/etc/SITE. The following table is a list of substitution parameters known as of TO11DR10

AWIPS-I Token	AWIPS-II Token
serverConfig.D2DDIRS	serverConfig.D2DMODELS example: serverConfig.D2DMODELS.append(('SREF12', 'SREF')) You can retrieve the model name from the grib_models table in the metadata database with the following SQL statement: select distinct(modelname) from grib_models;

Ensure that both sub-directory and file have the following ownership:

```
user:      awips
group:     awips
permissions: 755
```

The following command can be used to achieve these permissions when currently in the `edex_static/site/XXX` path:

```
# chown -R awips:awips config
# chmod -R 755 config
```

■ ***grib/subgrids***

Add an xml file in this directory to “clip” the higher resolution models much as they are in AWIPS I. Sub-gridding is done by model name and by coordinates. Multiple xml files can be created for different sub-gridding areas, or one can be created for one area and multiple models.

The following is the syntax needed for the xml file. It is recommended that the delivered file with the OAX localization be used as a starting point:

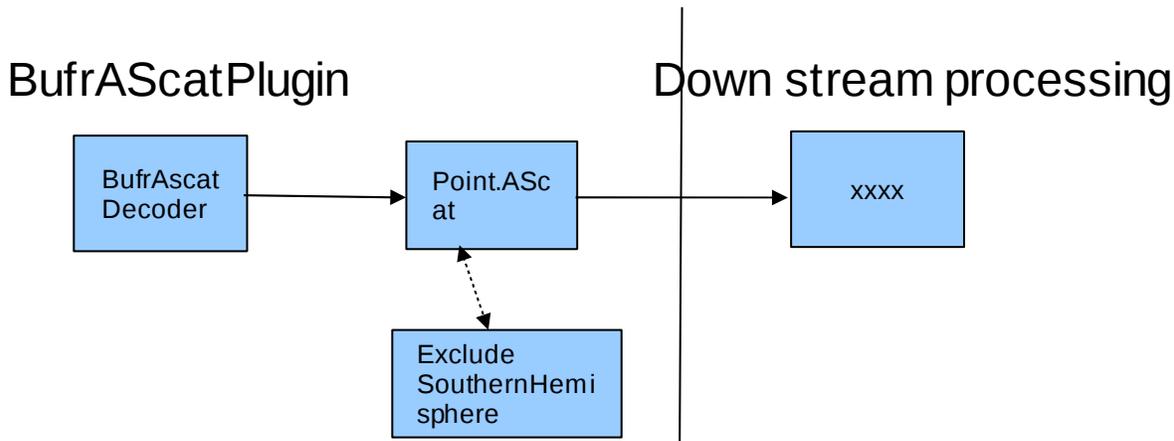
```
<subGridDef>
  <modelName></modelName>
  <lowerLeftLat></lowerLeftLat>
  <lowerLeftLon></lowerLeftLon>
  <upperRightLat></upperRightLat>
  <upperRightLon></upperRightLon>
</subGridDef>
```

The `modelName` can be extracted from the metadata database, inside the `gribModels.xml` file which resides in `$EDEX_HOME/data/utility/common_static/base/grid`

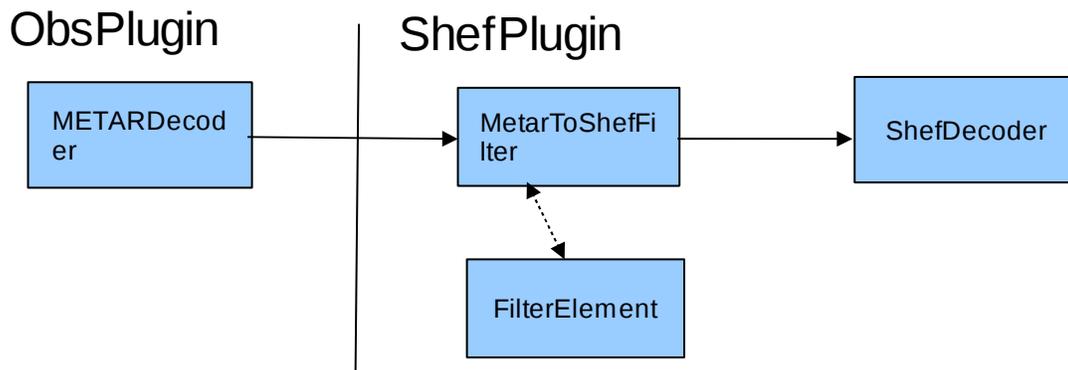
■ ***plugin-filters***

The purpose of plugin filters is to provide a data filtering capability that is base or site configurable without recompiling baseline code.

Plugin filters may be used to filter output from a process:



Or used to filter the input to a process.



Filtering can be done by EXCLUDING data, INCLUDING data or a mix of EXCLUDES and INCLUDES.

When using INCLUSIVE filtering, the output of the filtering operation returns those records which match the filtering criteria. For example:

```
<pluginDataObjectFilter>
  <filterElements xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="rectFilterElement">
    <filterElementName>OAX.WFO</filterElementName>
    <!-- Filter that includes all observations within the Omaha Valley WFO -->
    <filterType>INCLUDE</filterType>
    <upperLeftLat>44</upperLeftLat>
    <upperLeftLon>-100</upperLeftLon>
    <lowerRightLat>40.0</lowerRightLat>
    <lowerRightLon>-94.90</lowerRightLon>
  </filterElements>
  <filterName>OAX.MetarToShef</filterName>
</pluginDataObjectFilter>
```

When using EXCLUSIVE filtering, the output of the filtering operation returns those records which are not matched by the filtering criteria. For example:

```
<pluginDataObjectFilter>
  <filterElements xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="rectFilterElement">
    <!-- Exclude all southern hemisphere data. -->
    <filterElementName>Exclude SouthernHemisphere</filterElementName>
    <filterType>EXCLUDE</filterType>
    <upperLeftLat>-0.1</upperLeftLat>
    <upperLeftLon>-180.0</upperLeftLon>
    <lowerRightLat>-90.0</lowerRightLat>
    <lowerRightLon>180.0</lowerRightLon>
  </filterElements>
  <filterName>Point.QuikScat</filterName>
</pluginDataObjectFilter>
```

Filtering can be done on the following elements as of the date on this document:

RectFilterElement	Select based upon reports contained within a rectangular bounding
RadiusFilterElement	Select based upon reports within a specified distance from a given latitude/longitude
StationIdFilterElement	Select based upon reports matching a specified station identifier. Regular expressions may be used.

Within the **base** plugin-filters directory the following files reside to filter out all Southern Hemisphere data. Finer grain filtering may be done within a site-specific plugin-filters directory.

ascat_filters.xml	Allows only northern hemisphere ASCAT data to be stored
modelsounding_filters.xml	Allows only northern hemisphere model sounding data to be stored
pointssmi_filters.xml	Allows only northern hemisphere SSMI data to be stored
quikscat_filters.xml	Allows only northern hemisphere QUIKSCAT data to be stored

The following plugins support filtering currently:

mesowest_filters.xml	As delivered in the OAX localization, this filters out all data except that within the site's CWA for the mesowest plugin by latitude / longitude.
metarToShefFilter.xml	As delivered in the OAX localization, filters out all data except that within the site's CWA by latitude/longitude.

Fields within the plugin-filter xml are described below:

filterElementName	Global	For information purposes only. Appears in EDEX log for tracing loading and functionality of filters.
filterName	Global	For information purposes only. Appears in EDEX log for tracing loading and functionality of filters.
filterType	Global	INCLUDE or EXCLUDE
upperLeftLat	RectFilterElement	Latitude measure for upper left corner point for filter area.
upperLeftLon	RectFilterElement	Longitude measure for upper left corner point for filter area.
lowerRightLat	RectFilterElement	Latitude measure for lower right corner point for filter area.
lowerRightLon	RectFilterElement	Longitude measure for lower right corner point for filter area.
pointLat	RadiusFilterElement	Latitude of center point for radius filtering

pointLon	RadiusFilterElement	Longitude of center point for radius filtering
radius	RadiusFilterElement	Radius around center point to either include or exclude data.

NOTE: Latitude values may range from -90 degrees (south) to 90 degrees (north); Longitude values may range from -180 degrees (west) to 180 degrees (east); If a box must cross the date line (180) two filter boxes must be created to handle the filtering.

An example of a plugin filter is listed below

```
<pluginDataObjectFilter>
  <filterElements xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="rectFilterElement">
    <filterElementName>OAX.WF0</filterElementName>
    <!-- Filter that includes all observations within the Omaha Valley WF0 -->
    <filterType>INCLUDE</filterType>
    <upperLeftLat>44</upperLeftLat>
    <upperLeftLon>-100</upperLeftLon>
    <lowerRightLat>40.0</lowerRightLat>
    <lowerRightLon>-94.90</lowerRightLon>
  </filterElements>
  <filterName>OAX.MetarToShef</filterName>
</pluginDataObjectFilter>
```

The fields highlighted in red indicate how to define the type of filter (rectangular, radius, or station) and the type of filter (INCLUDE or EXCLUDE). Multiple <filterElements> are allowed in each xml file, and are read in the order they are listed in the file.

Triggers

Triggers, as of TO11, are configured manually using the textdb CLI executable. For more information on the commands which can be used to load triggers, see section 1.3.9 of Raytheon Document AWP.DOC.SWCTRTO11.LOC-01_Dft.

A quick way to get current triggers from an AWIPS-I site loaded into the database, is to copy the /awips/fxa/postgres/fxatextTriggerActions.txt into a temporary location on the EDEX server and run the following script.

NOTE: Proceeding should be done with caution as some triggers automatically send products to the WAN.

```
cat fxatextTriggerActions.txt | while read line
do
    PIL=$(echo $line | cut -f1 -d'|')
    SCRIPT=$(echo $line|cut -c13-)
    $PATH_TO_CLI/textdb -ldad -a $PIL "$SCRIPT"
done
```

set_hydro_env

The AWIPS II instance of set_hydro_env is delivered with the site path of OAX hard coded in the line for Apps_defaults. It is located in the following directory:

`${EDEX_HOME}/data/hdf5/hydroapps`

Change the file so the site which is currently being localized is reflected properly:

```
export APPS_DEFAULTS_SITE=${LOCALIZATION_ROOT}/site/OAX/hydro/Apps_defaults
```

start.sh

The final step to switching EDEX to a new site is to modify the start.sh if EDEX was not originally installed as the site which the system is currently being localized. This file resides in the following location on each server on which EDEX is running.

`${EDEX_HOME}/bin`

At the top of the file there are localization environment variables. Change the appropriate variables, which are highlighted in red below, to correspond to the desired localization.

```
### EDEX localization related variables ###
export AW_SITE_IDENTIFIER=OAX

# database names
export DC_DB_NAME=dc_ob7OAX
export FXA_DB_NAME=fxatext
export HM_DB_NAME=hmdb
export IH_DB_NAME=hd_ob83OAX

### end of localization variables ###
```

NOTE: As of TO11 there is no longer a need to edit the environment.xml or hibernate.cfg.xml files.

CAVE LOCALIZATION

CAVE Plugin Configuration Files

The CAVE plug-ins can be customized within the following data tree:

```
${EDEX_HOME}/data/utility/cave_config/site/XXX/plugin_name/  
${EDEX_HOME}/data/utility/cave_config/user/UUU/plugin_name/
```

The following is a table of plug-ins which can be customized

CAVE Plug-In Name	Configuration File	Comment
com.raytheon.viz.core	config.xml	CAVE general configuration
com.raytheon.viz.warngen	config.xml	Warngen Office
com.raytheon.viz.gfe	config.xml	GFE defaults
com.raytheon.uf.viz.alertviz	config.xml	AlertViz defaults
com.raytheon.viz.aviation	config.xml	AvnFPS setup config
com.raytheon.viz.avnconfig	config.xml	AvnFPS defaults
com.raytheon.viz.radar	config.xml	Radar general
com.raytheon.viz.skewt	config.xml	SkewT defaults

The CAVE core plugin contains information about the core CAVE visual experience. Here, the color selector for data displayed in the pane can be customized.

The file used to localize this feature is located in

`${EDEX_HOME}/data/utility/cave_config/site/XXX/com.raytheon.viz.core/` and is named `config.xml`.

Below is an example adding white to the list of colors shown when selecting *Change Color* from the right click menu.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<configuration>
  <defaultGraphicColor>white</defaultGraphicColor>
  <defaultGraphicColor>green</defaultGraphicColor>
  <defaultGraphicColor>coral</defaultGraphicColor>
  <defaultGraphicColor>cyan</defaultGraphicColor>
  <defaultGraphicColor>burlywood</defaultGraphicColor>
  <defaultGraphicColor>yellow</defaultGraphicColor>
  <defaultGraphicColor>violet</defaultGraphicColor>
  <defaultGraphicColor>darkkhaki</defaultGraphicColor>
  <defaultGraphicColor>OrangeRed</defaultGraphicColor>
  <defaultGraphicColor>dodgerblue</defaultGraphicColor>
</configuration>
```

Warngen

General Warngen configuration takes place in the following location:

`$EDEX_HOME/data/utility/cave_config/site/XXX/com.raytheon.viz.warngen/config.xml`

Below is an example of the file, localized for OAX. Edit the appropriate fields for the new site's localization.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<configuration>
  <warngenOfficeShort>OMAHA/VALLEY NE</warngenOfficeShort>
  <warngenOfficeLoc>OMAHA</warngenOfficeLoc>
  <backupCWAs>EAX/KANSAS CITY,DMX/DES MOINES,BOX/BOSTON,LBF/NORTH
  PLATTE,PQR/PORTLAND</backupCWAs>
  <siteNode>OMA</siteNode>
  <defaultTemplate>severethunderstorm</defaultTemplate>
  <mainWarngenProducts>Flash Flood/ffw,Severe
  Thunderstorm/severethunderstorm,Tornado/tornado</mainWarngenProducts>
  <otherWarngenProducts>Severe Weather Statement/SVS,Flash Flood
  Statement/ffs,Extreme Wind Warning/eww,Extreme Wind SVS/ewws,non-convective FFW
  (Dam Break)/dambreak,non-convective Flash Flood Statement/dambreakffs,Areal Flood
  Warning/flw,Areal Flood Warning Followup/fls,Areal Flood Advisory/fla,Areal Flood
  Advisory Followup/flas,Special Marine Warning/smw,Marine Weather Statement (SMW
  Follow)/smws,Marine Weather Statement standalone/marinestatement,Short Term
  Forecast/shortterm</otherWarngenProducts>
  <followupListRefreshDelay>5000</followupListRefreshDelay>
</configuration>
```

Information on the fields and their purpose can be found the table below:

XML TAG	DESCRIPTION
warngenOfficeShort	The office description included in the heading of the warning. AWIPS-I File: <i>/awips/fxa/data/localizationDataSets/XXX/headerInc.txt</i>
warngenOfficeLoc	The short office description included in the heading of the warning. AWIPS-I File: <i>/awips/fxa/data/localizationDataSets/XXX/wwaConfig.txt</i> Look at the line beginning @@@OFFT
backupCWAs	A comma-separated list of sites which appear in the Backup pull-down menu in the Warngen GUI, including a description of the site to be included in the template. Format: XXX/DESCRIPTION Each entry for a site must include both the XXX and DESCRIPTION, where XXX is the AWIPS Site ID. AWIPS-I File:

XML TAG	DESCRIPTION
	<i>/awips/fxa/data/localizationDataSets/XXX/backupCWA.id</i>
defaultTemplate	The Warngen template identifier of the template whose radio button is selected by default when the Warngen GUI is launched. Format:
mainWarngenProducts	A comma-separated list of templates which appear under "Product Type" on the Warngen GUI and are selectable via radio buttons. Format: <warngen template id>/<template description>
otherWarngenProducts	A comma-separated list of templates which appear under the Other drop down selector in the Warngen GUI within the "Product Type" section. Format: <warngen template id>/<template description>
followupRefreshDelay	Timer delay in milliseconds to update the start and stop time in the Warngen GUI.
siteNode	Provides a default site node if one cannot be found in the afos2awips table within the metadata database. This field is unused if an entry is found in the table.

Map Customization Including Menu Drop-Down

The OAX localized D2D perspective in CAVE consists of 6 maps, which are loaded as bundles. When the default five panes are loaded initially during a fresh CAVE startup, this is called a procedure (a combination of bundles saved in a specific state). To localize a D2D perspective to a different site, you must at least create new maps for the Regional, State(s), and WFO scale. The Northern Hemisphere, North American and CONUS can be re-used for CONUS sites. You can also create and add new maps at this time.

To create a replacement map, or new map, based on current D2D maps, use the tool *maksuparg* in conjunction with AWIPS-I sup files. As a user which belongs to the fxalpha group on any workstation, change directories into the */awips/fxa/data/localizationDataSets/XXX* directory. Run the following commands, and note the needed information:

SCALE REFERENCE FOR maksuparg

- | | |
|-----------------------------------------------------|-----------------------------|
| 1 = Stereographic | 7 = Satellite View |
| 2 = Orthographic | 8 = Cylindrical Equidistant |
| 3 = Lambert Conformal (rotation is second parallel) | 9 = Mercator |
| 4 = Azimuthal Equal Area | 10 = Mollweide |
| 5 = Gnomonic | 16 = Azimuth Range |
| 6 = Azimuthal Equidistant | 17 = Radar AZ Ran |

NOTE: As of R1G1, Lambert Conformation around second parallel is no longer supported. Use rotation around first parallel to create maps which return a scale reference of 3 from the table above. This will not affect the projection of the map being created.

maksuparg regionalScale.sup

```
maksuparg 3 25.000000 -95.000000 25.000000
l 34.343422 -86.801041 49.222343 -58.918396
l 52.075909 -85.166916 31.691475 -64.714157
x 0.046138 0.173107 -0.761975 -0.635005
o 42.471703 -73.894989 998.277192 998.277192 998.276958 998.276958
o 25.000000 -95.000000 -762.554509 2861.088761 -1066.558964 3165.092723
```

For this example:

Scale	=	Lambert Conformal 1SP
Central Meridian	=	-95.000000
Latitude Of Origin	=	25.000000
Standard Parallel 1	=	25.000000
Lower left	=	34.343422 -86.801041
Upper Right	=	49.222343 -58.918396

maksuparg stateScale.sup

```
maksuparg 3 25.000000 -95.000000 25.000000
  l 38.864445 -78.779984 45.511536 -66.190941
  l 46.854691 -77.411484 37.564293 -68.394897
  x 0.087403 0.144539 -0.727058 -0.669922
  o 42.328495 -72.694107 449.618694 449.618694 449.619045 449.619045
  o 25.000000 -95.000000 -1444.578129 2388.918468 -1643.655304 2587.996382
```

For this example:

Scale	=	Lambert Conformal 1SP
Central Meridian	=	-95.000000
Latitude Of Origin	=	25.000000
Standard Parallel 1	=	25.000000
Lower left	=	38.864445 -78.779984
Upper Right	=	45.511536 -66.190941

maksuparg wfoScale.sup

```
maksuparg 1 42.177322 -71.496742 0.000000
  l 40.076515 -74.199295 44.210030 -68.610909
  l 44.210045 -74.381912 40.076500 -68.793571
  x -0.018051 0.018056 -0.018055 0.018052
  o 42.177151 -71.496422 230.005066 230.005066 230.005066 230.005066
  o 42.177322 -71.496742 229.976610 230.029216 230.022298 229.983528
```

For this example:

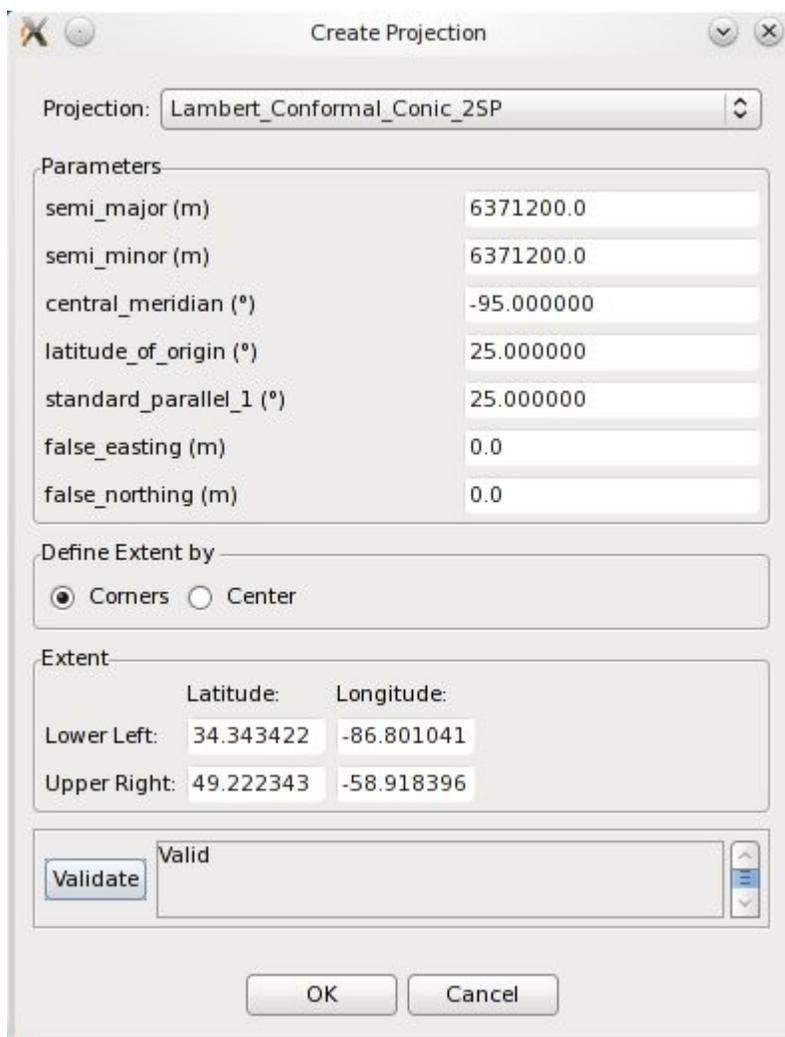
Scale	=	Stereographic
Latitude Of Origin	=	42.177322
Central Meridian	=	-71.496742
Lower left	=	40.076515 -74.199295
Upper Right	=	44.210030 -68.610909

Repeat these steps for any additional scales needed to be reused from AWIPS I. You can determine which scales are needed by looking at the file `scaleInfo.txt` in the localizationDataSets directory. To see which five scales load on default D2D launch, look in the file `scales.config` in `/data/fxa/nationalData` or `/data/fxa/customFiles`

To create the maps, use a CAVE session and follow the menu selections to the CAVE --> New --> Map Projection GUI. Fill in the fields as recorded above. Once the map is displayed as desired, use the CAVE --> Procedures --> Save Bundle menu selection and save the file to a temporary location.

NOTE: After the creation of each map, to ensure that the proper scale is displayed on the drop-down menu selector you should vi the xml file created during the save process. Change the scale=" " value at the top of the file in the first <displays tag to the scale you just created (e.g. Regional).

Below is a screen shot for an example in creating the Regional Scale for site LWX.



Once all needed bundles have been saved they must be copied to the following location on the EDEX servers:

```
$EDEX_HOME/data/utility/cave_static/site/XXX/bundles/scales
```

To copy a replacement of the State(s).xml the following command can be used to copy the file capturing the special characters in the file name

```
# scp State\s\).xml $EDEX_HOME/data/utility/cave_static/site/XXX/bundles/scales
```

Be sure to check that each file name is copied with the file extension of .xml. Below is an example of the directory listing for a baseline LWX localization

```
$EDEX_HOME/data/utility/cave_static/site/LWX/bundles/scales/Regional.xml  
$EDEX_HOME/data/utility/cave_static/site/LWX/bundles/scales/State(s).xml  
$EDEX_HOME/data/utility/cave_static/site/LWX/bundles/scales/WFO.xml
```

NOTE: As of release R1G1 you do not need to copy the CONUS, N. Hemisphere or N. American xml files into place. They are delivered as default and can be called from default-procedure without residing in the site-specific localization tree.

Ensure proper ownership and permissions (USER: awips; GROUP: awips; Permissions: 755) with the following commands:

```
# cd $EDEX_HOME/data/utility/cave_static/site/XXX  
# chown -R awips:awips *  
# chmod -R 755 *
```

Once all desired maps (bundles) have been created, ensure the five you wish to load by default when a D2D perspective is launched are currently loaded in the correct panes, and use the CAVE --> Procedures --> Save Procedure... menu selection to save this as the *default-procedure*.

NOTE: You may overlay maps, and give maps colors (for example, put the Lat/Lon on the Regional scale and color it yellow.) Do not load any data, or zoom before saving the procedure as the default-procedure outlined in section 1.3.2.1 of Raytheon Document AWP.DOC.SWCTRTO11.LOC-01_Dft.

Be sure to correct any inconsistencies left in the scale= values inside of the default-procedure.xml created, mimicing the changes you made in each individual map bundle before.

Once the default-procedure has been saved to any location, copy it into the localization store on the EDEX servers ensuring it has an .xml extension:

```
$EDEX_HOME/data/utility/cave_static/site/XXX/default-procedure.xml
```

At this point, to configure the scale selector drop-down in the CAVE D2D perspectives, copy the *plugin.xml* file from the OAX localization into the new site's cave_plugin directory tree. Start by copying the directory structure if it does not already exist :

Commands run from the \$EDEX_HOME/data/utility/cave_plugin/site directory:

```
# mkdir XXX (Substitute the new site ID for XXX)
# cp -a OAX/com.raytheon.viz.site_1.0.0 XXX/
```

If new maps have been created outside of the normal CONUS scales, they should now be added to the plugin.xml which resides in the following directory:

`$EDEX_HOME/data/utility/cave_plugin/site/XXX/com.raytheon.viz.site_1.0.0/`

Add a section using the rest of the file as a guide, or changing existing menu entries if they are not needed for the selected localization. The tags which must be created/edited are the `<command />` tags like shown below:

```
<command
  commandId="com.raytheon.viz.ui.gMap.setScale"
  id="TRACON"
  label="TRACON">
  <parameter
    name="scale"
    value="LWX-tracon">
  </parameter>
</command>
```

The *id* should be unique but it not used or displayed (used internally for identification)

The *label* is the text seen in the menu drop-down

The *value* is the name of the bundle file, without .xml. The file itself needs to end in .xml

GENERAL NOTE: In order for the map selector pull-down and the clear button to work correctly, the `scale=` entries in each bundle/procedure must match the `value=` entry in the plugin.xml file. This, in turn, must match the file name (without the .xml extension). It is highly recommended to use the same default names for all scales which are being replaced (e.g. Regional, State(s) and WFO).

Satellite Menu

In order to have a full satellite menu, the following sub-directory and index file must be created:

`$EDEX_HOME/data/utility/cave_static/site/XXX/menus/sat/index.xml`

It is suggested to copy in the file from the delivered OAX localization.

Commands run from the `$EDEX_HOME/data/utility/cave_static` directory:

First, make the site ID directory if it doesn't already exist, otherwise skip the following step

```
# mkdir -p XXX/menus/sat (Substitute the new site ID for XXX)
# chown -R awips:awips XXX
```

```
# chmod -R 755 XXX
```

Copy the menus/sat directory and contents to the new site ID directory

```
# cp -a OAX/menus/sat/index.xml XXX/menus/sat/
```

Edit the `index.xml` using vi and change the sector values to their appropriate values. A list of valid sectors can be found in the metadata database, inside the `satellite_sector_ids` table. Issue the following command to retrieve the list:

```
# psql -h ${DB_HOST} -U awips -d metadata -c "select * from satellite_sector_ids;"
```

As of the date on this document, the following table is a listing of all entries from the above command:

sectorid	sectorname
0	Northern Hemisphere Composite
1	East CONUS
2	West CONUS
3	Alaska Regional
4	Alaska National
5	Hawaii Regional
6	Hawaii National
7	Puerto Rico Regional
8	Puerto Rico National
9	Supernational
10	NH Composite - Meteosat-GOES E-GOES W-GMS

For most sites, either "East CONUS" or "West CONUS" depending on the value in the AWIPS-I file `/awips/fxa/data/localizationDataSets/XXX/whichSat.txt` will be used.

Change the value of sector1 to whichever one was not chosen for sector0.

Upper Air Menu

As of the date on this document, the upperair directory and structure are created when EDEX starts from the following configuration file discussed in another section of the document:

```
$EDEX_HOME/data/utility/common_static/site/XXX/upperair/raobSitesInUse.txt
```

NCEP / Hydro Menu

In order to pull up the RFC whose Flash Flood Guidance (FFG) that is normally displayed

through the RFC Flash Flood Guidance --> 1hr/3hr/6hr FFG Mosaic images the hydro index.xml must be localized. This file resides in the following directory

```
$EDEX_HOME/data/utility/cave_static/site/XXX/menus/hydro/index.xml
```

This file should be automatically created when EDEX starts based on the entries in the following file:

```
$EDEX_HOME/data/utility/common_static/site/XXX/hydro/hydroSitesInUse.txt
```

Note: For each RFC listed, a new trio of menus are added to the configuration file.

Warnings Display

In order to localize the warning display for the chosen site localization, the following directory and index.xml should be created:

```
$EDEX_HOME/data/utility/cave_static/site/XXX/menus/warnings/index.xml
```

It is recommended using the index.xml delivered with the OAX localization as a guide to this file, and replacing the site specific information for the chosen localization in the following locations:

First, make the site ID directory if it doesn't already exist, otherwise skip the following step

```
# mkdir -p XXX/menus/warnings      (Substitute the new site ID for XXX)
# chown -R awips:awips XXX
# chmod -R 755 XXX
```

Copy the menus/warnings directory and contents to the new site ID directory

```
# cp -a OAX/menus/warnings/index.xml XXX/menus/warnings/
```

Using VI or another editor, make the local changes needed for the new site's localization. Attention should be paid to the bold red text below.

```
<menuContributionFile>
  <include installTo="menu:obs?after=HAZARDS"
    fileName="menus/warnings/baseWarnings.xml">
    <substitute key="site" value="KOAX"/>
    <substitute key="sites" value="KABR, KFSD, KUNR, KBIS, KFGF, KBOU, KGJT,
KPUB, KCYS, KRIW, KGID, KLBF, KOAX, KDDC, KGLD, KICT, KTOP, KEAX, KLSX,
KSGF, KDMX, KDVN, KDLH, KMPX, KARX, KGRB, KMKX, KILX, KLOT, KJKL, KLMK,
KPAH, KIND, KIWX, KAPX, KDTX, KGRR, KMQT">
  </include>
</menuContributionFile>
```

Warngen Templates

Warngen templates can be used on a user, site or base level. Base templates should never be edited, and are stored on the EDEX localization server in the following directory:

`$EDEX_HOME/data/utility/common_static/base/warngen`

Warngen templates are created using velocity template language (VTL) with supporting configurations in an xml formatted file with a .cfg extension. More information on this can be found:

http://svn.apache.org/repos/asf/velocity/engine/tags/V_1_0_1/docs/vtl-reference-guide.html

<http://click.sourceforge.net/docs/velocity/vtl-reference-guide.html>

Site level templates and product configuration files can be stored in the following location on the EDEX localization server:

`$EDEX_HOME/data/utility/common_static/site/XXX/warngen`

Where XXX is a site's localization ID. Site specific templates should also be named in the following manner, substituting the site ID for XXX:

`<templateName>_XXX`

For example: dambreak_OAX

NOTE: If custom templates are not loaded from the EDEX server, try creating `~/caveData/common/site/XXX/warngen` and restarting CAVE.

Configuration Values in Template .cfg Files	
polygonShape	Use "1" for the pathcast shape based on storm track or "2" for a square.
followups	Defines the options which can appear in the "UPDATE LIST" dropdown Note: This is a system-level setting and should only be modified if necessary.
phenomena	Defines which Phenomena this template can create Followup products for. Note: This is a system-level setting and should only be modified if necessary.
Significance	Defines for which Significance this template can create Followup products. Note: This is a system-level setting and should only be modified if necessary.

Configuration Values in Template .cfg Files	
DefaultDuration	
Durations	
Bullets	
bulletText	Defines the text displayed in the WarnGen dialog
bulletGroup	Provides a mechanism for limiting selections in WarnGen. Only one item per bulletGroup can be selected so related items should list the same
bulletType	bulletType="title" makes the bullet unable to be selected bulletType="basin" correlates the bullet to a geometry of the same name in the custom locations table
pathcastConfig	
enabled	"1" will enable the Storm Track "2" will disable the Storm Track.
defaultSpeedKt	Default Storm Track speed in knots
defaultDirection	Default Storm Track direction
overThreshold	Distance for a city or POI to be considered "over" the Storm Track
nearThreshold	Distance for a city or POI to be considered "near" the Storm Track
lineofStormsDistance	The distance, in km, between the 2 original storm points when Line of Storms is selected.
lineofStormsAzimuth	The azimuth between the 2 original storm points when Line of Storms is selected.
areaSource	Relation to a table in the maps database Note: This is a system-level setting and should only be modified if necessary.
pointField	Relation to a field name in a table in the maps database Note: This is a system-level setting and should only be modified if necessary.
areaNotationField	Relation to a field name in a table in the maps database Note: This is a system-level setting and should only be modified if necessary.
areaField	Relation to a field name in a table in the maps database

Configuration Values in Template .cfg Files	
	Note: This is a system-level setting and should only be modified if necessary.
fipsField	Relation to a field name in a table in the maps database Note: This is a system-level setting and should only be modified if necessary.
parentAreaField	Relation to a field name in a table in the maps database Note: This is a system-level setting and should only be modified if necessary.
timezoneField	Relation to a field name in a table in the maps database. Note: This is a system-level setting and should only be modified if necessary.
closestPointsConfig	
numberOfPoints	This configuration sets the number of “Closest Points” that will be used in the first bullet. Any point included in the “Closest Points” variable will be excluded from the “Other Points” variable so that a location is not repeated.
pointFilter	A filter used when making a query to the map database. This can be used to include or exclude points of interest with specific WarngenLev values.
geospatialConfig	Note: This is a system-level setting and should only be modified if necessary.
pointSource	Relation to a table in the maps database Note: This is a system-level setting and should only be modified if necessary.
areaSource	Relation to a table in the maps database. Note: This is a system-level setting and should only be modified if necessary.
parentAreaSource	Relation to a table in the maps database Note: This is a system-level setting and should only be modified if necessary.
maskSource	Relation to a table in the maps database. Note: This is a system-level setting and should only be modified if necessary.
maskFilter	Filter used to exclude geometry outside of the local CWA.

Configuration Values in Template .cfg Files	
	Note: This is a system-level setting and should only be modified if necessary.
basinConfig	Note: This is a system-level setting and should only be modified if necessary.
maskSource	Relation to a table in the maps database Note: This is a system-level setting and should only be modified if necessary.
bulletColumn	Relation to a field name in a table in the maps database. Note: This is a system-level setting and should only be modified if necessary.
maskFilter	Filter used to exlude geometry outside the local CWA Note: This is a system-level setting and should only be modified if necessary.

Values Passed to WarnGen *.vm Templates	
The following values from the Java WarnGen plug-in code are passed to the templates for WarnGen products.	
vtecOffice	The 4-letter id set by the logic in “getSite4LetterId” shown below.
siteId	A String: The 3-letter WFO set in Localization
officeShort	String set in site localization config.xml
officeLoc	String set in site localization config.xml <u>EXAMPLE:</u> officeShort: “OMAHA/VALLEY NE” officeLong: “OMAHA”
backupSite	The officeLoc value for a site that WarnGen is backing up via backup mode.
localtimezone	Time Zone where the warning starts in single letter format
secondtimezone	If the Warning spans multiple time zones, the second Time Zone is stored here.
stormType	“line” if it is a line of storms “single” otherwise.
now	Date value representing now.

<u>Values Passed to WarnGen *.vm Templates</u>	
The following values from the Java WarnGen plug-in code are passed to the templates for WarnGen products.	
start	Date value representing the start time.
expire	Date value representing the expiration time.
event	Date value representing the event time
fipsline	String representing the UGC Header
fipslinecan	String representing a UGC Header
areaPoly	String representing the polygon
movementInMph	Double representing movement speed
movementDirectionRounded	Double representing movement direction
movementDirection	Double representing movement direction
movementInKnots	Double representing movement speed
action	String representing the VTEC Action
oldvtec	String representing the old VTEC tracking number for followup products
phenomena	String representing the VTEC pp field
mode	String representing test mode if applicable
bullets	Array of Strings representing the bullets selected in WarnGen
eventLocation	Array of Point2D values (java.awt.geom.Point2D): NOTE: DO NOT EDIT
otherPoints	Array of Strings representing other points impacted by the storm NOTE: DO NOT EDIT
pathCast	PathCast value (com.raytheon.viz.warngen.gis.PathCast) NOTE: DO NOT EDIT
closestPoints	Array of ClosestPoints (com.raytheon.viz.warngen.gis.ClosestPoint) NOTE: DO NOT EDIT
areas	Array of AffectedAreas (com.raytheon.viz.warngen.gis.AffectedAreas): NOTE: DO NOT EDIT
cancelareas	Array of AffectedAreas

<u>Values Passed to WarnGen *.vm Templates</u>															
The following values from the Java WarnGen plug-in code are passed to the templates for WarnGen products.															
	(com.raytheon.viz.warngen.gis.AffectedAreas): NOTE: DO NOT EDIT														
TimeFormat	Hashtable containing 6 SimpleDateFormats <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #444; color: white;">Hashtable Key</th> <th style="background-color: #444; color: white;">Hashtable Value</th> </tr> </thead> <tbody> <tr> <td>header</td> <td>hhmm a z EEE MMM d yyyy</td> </tr> <tr> <td>plain</td> <td>hhmm a z EEEE</td> </tr> <tr> <td>clock</td> <td>Hmm a z</td> </tr> <tr> <td>ymdthmz</td> <td>yyMMdd'T'HHmm'Z'</td> </tr> <tr> <td>ddhhmm</td> <td>HHmm</td> </tr> <tr> <td>Time</td> <td>HHmm</td> </tr> </tbody> </table>	Hashtable Key	Hashtable Value	header	hhmm a z EEE MMM d yyyy	plain	hhmm a z EEEE	clock	Hmm a z	ymdthmz	yyMMdd'T'HHmm'Z'	ddhhmm	HHmm	Time	HHmm
Hashtable Key	Hashtable Value														
header	hhmm a z EEE MMM d yyyy														
plain	hhmm a z EEEE														
clock	Hmm a z														
ymdthmz	yyMMdd'T'HHmm'Z'														
ddhhmm	HHmm														
Time	HHmm														
list	ListTool (org.apache.velocity.tools.generic.ListTool). Method use examples: <pre> \$primes is an array of integers containing {2, 3, 5, 7} \$lists.size(\$primes) -> 4 \$lists.get(\$primes, 2) -> 5 \$lists.set(\$primes, 2, 1) -> (primes[2] becomes 1) \$lists.get(\$primes, 2) -> 1 \$lists.isEmpty(\$primes) -> false \$lists.contains(\$primes, 7) -> true </pre>														
mathUtil	WarnGenMathTool (extends org.apache.velocity.tools.generic.MathTool). Methods: (http://velocity.apache.org/tools/devel/javadoc/org/apache/velocity/tools/generic/MathTool.html) <pre> \$mathUtil.roundTo5(num) \$mathUtil.roundToInt(num, multiple) \$mathUtil.abs(num) \$mathUtil.add(num1, num2) \$mathUtil.ceil(num1) \$mathUtil.div(num1, num2) \$mathUtil.floor(num1) \$mathUtil.getAverage(nums) \$mathUtil.getRandom() \$mathUtil.Total(nums) </pre>														

<u>Values Passed to WarnGen *.vm Templates</u>	
The following values from the Java WarnGen plug-in code are passed to the templates for WarnGen products.	
	<pre> \$mathUtil.matchType(num1, num2) \$mathUtil.max(num1, num2) \$mathUtil.min(num1, num2) \$mathUtil.mod(num1, num2) \$mathUtil.mul(num1, num2) \$mathUtil.pow(num1, num2) \$mathUtil.random(num1, num2) \$mathUtil.round(num1) \$mathUtil.roundTo(decimals, num2) \$mathUtil.sub(num1, num2) \$mathUtil.toDouble(num1) \$mathUtil.toInteger(num1) \$mathUtil.toNumber(num1) </pre>
dateUtil	FateUtil (com.raytheon.viz.warngen.util.DateUtil). Methods: <pre> \$dateUtil.format(Date, \$dateUtil.format(Date, \$dateUtil.format(Date, TimeZone) \$dateUtil.format(Date, DateFormat) DateFormat, Interval) DateFormat, Interval, DateFormat, TimeZone) </pre>

How to Access the Data Contained in a Value Inside A Template

In general, any String, primitive type method or attribute can be accessed directly within the WarnGen template using the following syntax:

String or primitive type: `#{myStringValue}`

In the above context, myStringValue was a String passed directly into the template

String method: `#{dateUtil.format(#{pc.time}, #{timeFormat.clock}, #{localtimezone})}`

In the above context:

dateUtil is a com.raytheon.viz.warngen.util.DateUtil Object that contains the method format, which accepts 3 arguments.

#{pc.time} (a date value from the pathCast) is replaced by the time attribute of the pc object

`{timeFormat.clock}` is replaced by the SimpleDateFormat addressed in the timeFormat Hashtable by "clock," `{localtimezone}` is replaced by the localtimezone object.

Modifying / Editing Templates

Plain text in WarnGen templates can be freely edited as needed. When modifying the VTL, careful attention should be paid to the syntax so that errors are not introduced.

NOTE: Any line that begins with a pound (#) symbol or any variable surrounded by `{ }` could potentially be dangerous to alter.

To Add Bullets:

Bullets can be added to WarnGen by editing or creating a site or user level .cfg file.

Look at the `<bullets>` tag in an existing template and note it contains several existing bullets and additional bullets can be added by following the basic pattern `<bullet bulletName="x" bulletText="y">` in the proper order.

After the bullet has been successfully added to .cfg file, the associated .vm file must be modified to take that bullet into account.

Call To Actions

When adding a Call To Action a statement, Example #4 below should be followed.

First, check to see if the list contains the bulletName is using an if statement. Insert the text for the Call To Action on the following line, followed by an empty line. Finally, finish the if statement with a `#end`.

Examples

The following VTL examples are taken directly from the severethunderstorm.vm template:

Example #1

```
#if({mode}=="test" || {mode}=="practice")
TEST...SEVERE THUNDERSTORM WARNING...TEST
#else
SEVERE THUNDERSTORM WARNING
#end
```

The above example is a typical if statement. If the mode is set to test or practice then "TEST...SEVERE THUNDERSTORM WARNING...TEST" is printed to the SVR product, otherwise "SEVERE THUNDERSTORM WARNING" is printed.

Example #2

```
${officeLong} HAS ISSUED A
```

The above example will output “THE NATIONAL WEATHER SERVICE IN OMAHA HAS ISSUED A” if officeLong is set correctly in the `com.raytheon.viz.warngen/config.xml`.

Example #3

```
#if(${list.contains($bullets, "doppler")})  
    #if(${stormType} == "line")  
        #set ($report = "NATIONAL WEATHER SERVICE DOPPLER RADAR INDICATED A LINE OF  
SEVERE THUNDERSTORMS")  
    #else  
        #set ($report = "NATIONAL WEATHER SERVICE DOPPLER RADAR INDICATED A  
SEVERETHUNDERSTORM")  
    #end  
#end
```

The above example shows a compound control structure.

If the bullets list sent from WarnGen contains the “doppler” bullet (as indicated by the bulletName in severethunderstorm.cfg) the inner if statement is reached.

At that point if the stormType is equal to line then the report text is set to a message indicating a line of storms. Otherwise the report text is set to a message indicating a single storm. Finally, the first #end completes the stormType if statement, and the second #end completes the list.contains if statement.

Example #4

```
#if(${list.contains($bullets, "torWatchRemainsInEffect")})  
  
${testMessage}A TORNADO WATCH REMAINS IN EFFECT FOR THE WARNED AREA. IF A TORNADO IS  
SPOTTED... ACT QUICKLY AND MOVE TO A PLACE OF SAFETY IN A STURDY STRUCTURE...SUCH AS  
A BASEMENT OR SMALL INTERIOR ROOM.  
  
#end
```

This example is a typical Call To Action line and should be replicated for any additional Call To Action lines.

The line beginning with #if determines if the torWatchRemainsInEffect bullet is highlighted in WarnGen. If it is, the contents of the testMessage variable are printed followed by the next three lines of text. The #if statement ends at the #end line.

Example #5

```
#foreach (${city} in ${pc.points})
```

```
#if({city.roundedDistance} < 3)
## close enough to not need azran, considered OVER the area {city.name}##
#else
## needs azran information
{city.roundedDistance} MILES #direction({city.roundedAzimuth}) OF {city.name}##
#end
#end
```

This example shows a foreach loop.

pc.points is an array of cities affected by the pathcast of the storm. These objects should not be modified, but they can be used and displayed.

The loop covers the first line of the example until the final line for each item in the array pc.points. Each item in this array is temporarily renamed to city when it goes through the rest of the template. The #if statement checks if the roundedDistance of the city (which determines how far the city is from the pathcast) is less than 3 miles. If this is true, the city's name is printed. Otherwise, the roundedDistance is printed to indicate in the warning that the storm is roundedDistance miles in roundedAzimuth direction of the city.

Note: Lines beginning with two pound symbols (##) are comments. These can be used anywhere within a template.

Warngen - Adding Local Points

Currently, DR 2284 has been opened against this feature. This section will be updated upon the completion of work on this DR.

D2D Procedures

Procedures in AWIPS-II are created in the same way procedures in AWIPS-I are created. Using the top menu after launching CAVE, select the following options:

File --> Procedures --> Open / Load /

The GUI to create procedures is the same as AWIPS-I. Once a procedure is saved, it is stored to the user's caveData directory in the following path:

~/caveData/etc/common/procedures/

At this point, it is also synced to the EDEX localization server into the following path:

\$EDEX_HOME/data/utility/cave_static/user/xxxxxx/procedures

If, at any point, a user's ~/caveData directory is removed, the procedures will automatically

be transferred back into the caveData directory tree when that user first attempts to open each procedure.

Procedures can be shared from user to user by copying the file from one user's directory into a new user's directory tree on the EDEX server.

AVNFPS Localization

Further information on localizing AVNFPS can be found in section 1.3.7 of Raytheon Document AWP.DOC.SWCTRTO11.LOC-01_Dft

AVNFPS localization files reside in the following directory tree:

```
$EDEX_HOME/data/utility/cave_static/site/XXX/aviation.
```

It is recommended that localization be completed through using the GUI setup, as in AWIPS-I. However, certain files can be re-used from an AWIPS-I location.

The directories for the specific TAFs configured in `/awips/adapt/avnfps/etc/tafs/XXXX` can be copied to `$EDEX_HOME/data/utility/cave_static/site/XXX/aviation/config/tafs` with a command similar to the following example:

```
# cd /awips/adapt/avnfps/etc/tafs/  
# cp -a KDCA $EDEX_HOME/data/utility/cave_static/site/XXX/aviation/config/tafs/
```

Note: Remember to substitute your EDEX install directory for `$EDEX_HOME`. You will also need to remove the following files after copying the directory:

```
*.template
```

TAF configuration resides in the following directory and file:

```
$EDEX_HOME/data/utility/cave_static/site/XXX/aviation/avnsetup/TafConfig.xml
```

It is, again, recommended to use the GUI setup utility, but if configuring manually the current OAX version can be copied from the OAX localization tree and used as a guide for all the TAFs which are needed for the new localization.

```
# cd $EDEX_HOME/data/utility/cave_static/site/  
# mkdir -p XXX/aviation/avnsetup/  
# cp -a OAX/aviation/avnsetup/TafConfig.xml XXX/aviation/avnsetup
```

A listing of everything currently in the AWIPS I template directory (which goes in the `<template></template>` section of the `TafConfig.xml`) by running the following script from the `/awips/adapt/avnfps/etc/tafs` directory. Be sure to substitute the name of the default configuration file for `$DEFAULT_CFG`.

```
# tafList=( $(grep idents $DEFAULT_CFG | cut -f2 -d= | sed 's/,/ /g') )
# for taf in ${tafList[@]}
# do
# echo -e "<$taf>"
# cd $taf
# for file in $( ls *.template )
# do
# hr=$( echo $file | cut -f1 -d. )
# echo -e "<Z$hr>$( cat $file )</Z$hr>"
# done
# cd ..
# echo -e "</$taf>"
# done
```

Under the <products></products> portion of the TafConfig.xml file, be sure to list the new site in place of KOAX as the <default></default> and be sure to create a list for the site. A list of sites can be extracted from current configuration by issuing the following script in the /awips/adapt/avnfps/etc/taf directory (be sure to substitute the new AWIPS Site ID for XXX:

```
# tafList=( $(grep idents $DEFAULT_CFG | cut -f2 -d= | sed 's/,/ /g') )
# echo -e "<KXXX>" && for taf in ${tafList[@]}
# do
# echo -e "<list>$taf</list>"
# done && echo -e "</KXXX>"
```

This output can then be pasted into the <products></products> portion of the TafConfig.xml file.

The AVNFPS user list is configured in the the following file:

`$EDEX_HOME/data/utility/cave_static/site/XXX/aviation/avnwatch/aviationForecasterConfig.xml`

This is best localized through the GUI, but the OAX version of the file may also be used as a guide for the new localization:

```
# cd $EDEX_HOME/data/utility/cave_static/site/
# mkdir -p XXX/aviation/avnwatch/
# cp -a OAX/aviation/avnwatch/aviationForecasterConfig.xml XXX/aviation/avnwatch
```

Appendix A

List Of Configurable CAVE Plugins

CAVE Plug-In Name	Configuration File	Comment
com.raytheon.viz.core	config.xml	CAVE general configuration
com.raytheon.viz.warngen	config.xml	WarnGen office
com.raytheon.viz.gfe	config.xml	GFE defaults
com.raytheon.uf.viz.alertviz	config.xml	AlertViz defaults
com.raytheon.viz.aviation	config.xml	AvnFPS setup config
com.raytheon.viz.avnconfig	config.xml	AvnFPS defaults
com.raytheon.viz.radar	config.xml	Radar general configuration
com.raytheon.viz.skewt	config.xml	SkewT defaults