



WARNGEN TROUBLESHOOTING GUIDE

AWIPS Site Support Team

November 10, 2003

VERSION OB2

Cover Photo: Tornado at Manhattan, Kansas, May 31, 1949. Photo courtesy of the NOAA Photo Library (National Oceanic and Atmospheric Administration/Department of Commerce)

WARNGEN TROUBLESHOOTING GUIDE

AWIPS Site Support Team

November 10, 2003

CONTENTS

DISCLAIMER	<u>Page 3</u>
1. INTRODUCTION	<u>Page 3</u>
2. NWS TEXT PRODUCT HEADINGS	<u>Page 4</u>
2.1 PRODUCT HEADING FORMATS	<u>Page 4</u>
2.1.1 WMO HEADING	<u>Page 4</u>
2.1.2 AWIPS PRODUCT ID	<u>Page 4</u>
2.1.3 UNIVERSAL GEOGRAPHIC CODE (UGC)	<u>Page 5</u>
2.1.4 VALID TIME EVENT CODE (VTEC)	<u>Page 6</u>
2.2 TEXT PRODUCT SEGMENTATION	<u>Page 8</u>
3. LOCALIZATION	<u>Page 10</u>
3.1 LOCALIZATION BASICS	<u>Page 10</u>
3.2 LOCALIZATION TASKS	<u>Page 10</u>
3.3 LOCALIZATION INPUT FILES AND UNIX	<u>Page 12</u>
3.4 LOCALIZATION INPUT FILES AND FILE OVERRIDE	<u>Page 13</u>
3.5 SITE SPECIFIC CONTROL FILES	<u>Page 15</u>
3.6 CUSTOMIZATION FILES	<u>Page 15</u>
3.7 LOCALIZATION DIRECTIVES	<u>Page 17</u>
3.8 PROCESS CONFIGURATION FILES AND WARNGEN	<u>Page 17</u>
3.9 LOCALIZATION OUTPUT FILES	<u>Page 17</u>
3.10 LOCALIZATION TROUBLESHOOTING	<u>Page 18</u>
4. GEOGRAPHY FILES USED BY D2D AND WARNGEN	<u>Page 20</u>
4.1 THE “-maps” LOCALIZATION TASK AND WARNGEN	<u>Page 20</u>
4.1.1 WARNGEN AND D2D SHAPE FILE SETS	<u>Page 20</u>
4.1.2 THE “-maps” LOCALIZATION TASK	<u>Page 21</u>
4.1.3 TROUBLESHOOTING SHAPE FILE PROBLEMS	<u>Page 22</u>
4.2 THE “-wwa” LOCALIZATION TASK AND WARNGEN	<u>Page 25</u>
4.2.1 GELTs (GEOGRAPHIC ENTITY LOOKUP TABLES)	<u>Page 26</u>
4.2.2 COUNTIES, ZONES AND GELTs	<u>Page 27</u>
4.2.3 INPUT FILES FOR newGELTmaker.csh	<u>Page 29</u>
4.2.4 GELT SCRIPT FILES	<u>Page 29</u>
4.2.5 GELT DIAGNOSTICS	<u>Page 29</u>
4.2.6 TEXT PRODUCT TEMPLATES	<u>Page 31</u>

4.3 THE “-station” LOCALIZATION TASK AND WARNGEN	<u>Page 31</u>
4.3.1 GOODNESS FILES	<u>Page 31</u>
4.3.2 STATION PLOT INFORMATION FILES	<u>Page 33</u>
4.3.3 LOCATION PLOT INFORMATION FILES	<u>Page 33</u>
4.3.4 D2D MENUS, DATA KEYS AND DEPICT KEYS	<u>Page 34</u>
4.3.5 D2D CITIES VS. WARNGEN CITIES	<u>Page 35</u>
4.3.6 LOCAL CITY NAMES	<u>Page 36</u>
5. WARNGEN TROUBLESHOOTING	<u>Page 36</u>
5.1 ITEMS DIRECTLY AFFECTING PRODUCT CREATION	<u>Page 36</u>
5.1.1 WARNGEN FULL BACKUP, PARTIAL BACKUP, PRIMARY AND SECONDARY SERVICE BACKUP	<u>Page 36</u>
5.1.2 MARINE WARNINGS IN WARNGEN PARTIAL BACKUP MODE	<u>Page 39</u>
5.1.3 CUSTOMIZING THE WARNGEN PRODUCT TYPE MENU	<u>Page 40</u>
5.1.4 LOCAL CITY NAMES	<u>Page 43</u>
5.1.5 TEXT PRODUCT TEMPLATES AND LOCALIZATION	<u>Page 45</u>
5.1.5.1 TEXT PRODUCT TEMPLATE SUMMARY	<u>Page 46</u>
5.1.5.2 PRODUCT SPECIFIC TEMPLATES	<u>Page 46</u>
5.1.5.3 LIST TEMPLATES	<u>Page 50</u>
5.1.5.4 WARNGEN LOCALIZATION DIRECTIVES	<u>Page 51</u>
5.1.5.5 TEMPLATE SUBSTITUTIONS	<u>Page 53</u>
5.1.5.6 SAMPLE WARNGEN TEMPLATE SYNTAX	<u>Page 53</u>
5.1.5.7 TEMPLATE TROUBLESHOOTING	<u>Page 62</u>
5.1.6 UGC TROUBLESHOOTING	<u>Page 64</u>
5.1.7 VTEC TROUBLESHOOTING	<u>Page 65</u>
5.1.8 OB2 ENHANCED QC TROUBLESHOOTING	<u>Page 67</u>
5.1.9 FOLLOWUP TROUBLESHOOTING	<u>Page 69</u>
5.1.10 WARNGEN PRODUCT SEGMENTATION	<u>Page 70</u>
5.1.11 WARNGEN AND THE TEXT PRODUCT SITE ID	<u>Page 70</u>
5.1.12 RECREATING A FIELD SITE’S D2D/WARNGEN ENVIRONMENT	<u>Page 74</u>
5.2 WARNGEN, UNIX AND OTHER AWIPS SOFTWARE	<u>Page 79</u>
5.2.1 WARNGEN UNIX PROCESSES	<u>Page 79</u>
5.2.2 WARNGEN AND THE WORKSTATION TEXT DISPLAY ..	<u>Page 80</u>
5.2.3 TEXT NOTIFICATION SERVER TROUBLESHOOTING	<u>Page 81</u>
5.2.4 AWIPS DISSEMINATION OF WARNGEN PRODUCTS	<u>Page 83</u>
5.2.5 TRACING WARNGEN PRODUCTS ON THE AWIPS WAN, SBN	<u>Page 86</u>
5.2.6 DUPLICATE STORAGE OF WARNGEN PRODUCTS IN THE TEXT DATABASE	<u>Page 88</u>

WARNGEN TROUBLESHOOTING GUIDE

AWIPS Site Support Team

November 10, 2003

DISCLAIMER

This guide is based on experience that the AWIPS Site Support Team (SST) has had with field sites in WarnGen troubleshooting. It does not cover all WarnGen problems, but only those which have been escalated to the SST. The guide was written as a concise WarnGen troubleshooting guide for centralized AWIPS software support personnel. In other words, it is not a complete reference, but gives enough information to diagnose and fix various common WarnGen problems. The guide should be useful for field personnel, but they may not have access to all procedures or software mentioned.

This guide was verified with the AWIPS 5.2.2, OB1 and OB2 software releases. As time allows, updates will be made to include AWIPS software changes and additional knowledge gained by the SST. Information on meteorological operations (e.g. service backup procedures, issuing warnings, use of VTEC) is not a statement of official NWS policy, but is presented only as general background for software support personnel. Any opinions expressed are solely those of the SST and are not official NWS policy. If you find an error or have suggested improvements for this guide, please contact Mike Rega (Michael.Reg@noaa.gov) at the AWIPS Site Support Team.

1. INTRODUCTION

This guide was written for WarnGen, but some of the items here also pertain to WWA. WarnGen is used for short fused warnings (e.g. tornado warnings, severe thunderstorm warnings) and WWA is used for long fused warnings (e.g. winter storm warnings, hurricane warnings). NWS operational directives specify the formats of NWS text products, and in some cases, specify what software application (WWA or WarnGen) should be used to create warning related products.

WarnGen is maintained by FSL in Boulder, CO. WWA is maintained by MDL at NWS headquarters. There is a plan (during about the OB6 era) to combine the features of WarnGen, WWA and XNOW (a local warning application) into a next generation warning tool.

Sections 2 to 4 cover some helpful background topics related to WarnGen:

- section 2 - product headings and VTEC
- section 3 - localization basics
- section 4 - geography files used by D2D and WarnGen

Section 5 discusses specific WarnGen troubleshooting techniques.

2. NWS TEXT PRODUCT HEADINGS

To understand WarnGen, it's helpful to have some basic knowledge about the communications headings of the NWS text products that WarnGen creates. This section has a brief description of the contents of these headings.

2.1 PRODUCT HEADING FORMATS

The communications heading of a typical WarnGen product looks something like this:

```
WFUS52 KMHX 061735
TORMHX
NCC047-061815-
/NEW.KMHX.TO.W.0003.0204061730Z-0204061815Z/
```

The above example is for a tornado warning issued by NWS office KMHX. The fourth line above (the VTEC) can be produced by OB2 WarnGen but won't be implemented until after the OB2 era. Many other NWS text products have a similar communications heading. The following sections contain some information about each line of the heading.

2.1.1 WMO HEADING

The first line is the WMO heading. The general format of the WMO communications header is:

```
TTAAii CCCC YYGGgg (BBB)
```

TTAAii - type of product and geographic area. This is defined by a fairly complex set of WMO tables.

CCCC - ID of the site that created the product

YYGGgg - day, hour, minute the product was prepared. For observations, this contains the official observation reporting time.

BBB - optional group specifying corrections, amendments or other special cases

The NWS page describing the WMO heading is <http://www.nws.noaa.gov/tg/headef.html>

2.1.2 AWIPS PRODUCT ID

The second line of the heading contains the AWIPS Product ID. The AWIPS product ID is derived from the old AFOS product ID (PIL). The PIL has the format CCCNNNXXX where:

CCC = AFOS node

NNN = product ID

XXX = local site ID

The AWIPS product ID is the NNNXXX part of the AFOS PIL. In our example, the AWIPS product ID is “TORMHX”. This is a tornado warning issued by NWS site MHX.

On AWIPS, the file `/awips/fxa/data/afos2awips.txt` specifies the WMO ID for each AFOS ID. If you do a Unix “grep” for the AWIPS product “TORMHX” you get this:

```
RDUTORMHX WFUS52 KMHX
```

This gives us the AFOS PIL and its associated WMO heading. We see that the AFOS node for site MHX is RDU. This information is useful in various types of AWIPS troubleshooting.

The file `ds1:/data/fxa/nationalData/afosMasterPIL.txt` lists all possible AFOS PILs. If we grep for “RDUTOR” the result is:

```
RDUTORILM
RDUTORMHX
RDUTORRAH
```

This tells us that AWIPS sites MHX, RAH and ILM all share the AFOS node RDU.

Localization produces an associated file on each AWIPS workstation. The full Unix path and file name is `/awips/fxa/data/localizationDataSets/LLL/afosMasterPIL.CCC` (where LLL is the local site ID).

The AWIPS processing of WMO headings, AFOS PILs, and AWIPS products IDs is quite complex and beyond the scope of this paper.

Some troubleshooting hints related to the XXX in the AWIPS product ID are in the section on troubleshooting the site ID beginning on page [70](#).

2.1.3 UNIVERSAL GEOGRAPHIC CODE (UGC)

The third line of the heading uses the UGC to describe the effective location and expiration time of the product. Here are some examples:

```
NCC055-290830-   NC county 055 (FIPS code for Dare county)
                  product expires on the 29th of the month at 0830Z

NCZ103-141905-   NC NWS forecast zone 103 (outer banks Dare county)
                  product expires on the 14th of the month at 1905Z
```

Some NWS products are based on counties (such as WarnGen warning products) and some are based on NWS forecast zones (such as the ZFP - zone forecast product). There can be various

combinations of counties/zones and states in the UGC line. A more complex example is:

DEZ002-003-MDZ015>018-032216-

This is from a ZFP and covers Delaware zones 002 and 003 and Maryland zones 015, 016, 017, 018 and expires on the 3rd of the month at 2216Z.

There is generally a one to one correspondence between NWS public forecast zones and county names, especially in the eastern U.S. Some counties contain more than one NWS forecast zone. A small number of NWS forecast zones span more than one county, especially in the western U.S. The forecast zones that are not co-incident with county boundaries generally are defined by a topographic or climate feature rather than by county boundaries. There are NWS web sites where you can see the zones and county maps. Go to <http://www.nws.noaa.gov/geodata/> and select either "AWIPS Counties" or "Zone Forecast Areas." To see a little more detail, go to <http://www.nws.noaa.gov/>. On the map of the U.S., click on an area of interest to go the web page for the local WFO. The web page for each WFO has a detailed map showing the NWS forecast zones.

For more details on the UGC see <http://iwin.nws.noaa.gov/emwin/winugc.htm> The UGC uses FIPS (Federal Information Processing Standard) geography codes. The full FIPS code has 6 characters. This 6 character code is also the SAME (Specific Area Message Encoding) code used by NOAA Weather Radio. For more info on FIPS see <http://www.itl.nist.gov/fipspubs/co-codes/states.htm> For more info on the NWR SAME codes see <http://205.156.54.206/nwr/indexnw.htm>

The following is an example of the full FIPS code:

- 037055 - for now, 1st digit is always 0
- 37 is the FIPS code for North Carolina
- 055 is the FIPS code for Dare County

Eventually, each county can be divided into up to 9 sections (sections 1 to 9) by using the first digit of the FIPS code. For example, FIPS code 137055 could denote the northwest part of Dare County, NC. There is a UGC troubleshooting section beginning on page [64](#).

2.1.4 VALID TIME EVENT CODE (VTEC)

In the future, watches/warnings/advisories will have the VTEC (Valid Time Event Code) code added just under the UGC line. The VTEC supplements the UGC and will provide for more sophisticated automated processing of NWS warnings. For more VTEC details, see the VTEC Implementation Page at <http://www.nws.noaa.gov/om/vtec> There is some VTEC requirements info on the NWS operational directives page <http://www.nws.noaa.gov/directives> (directive 10-1703). There is some VTEC training info from the Warning Decision Training Branch at:

http://wdtb.noaa.gov/modules/AWIPS_OB2/index.htm

There are two forms of VTEC. The primary VTEC (P-VTEC) will be included in all warnings. The hydrologic VTEC (H-VTEC) will be in flood or flash flood warnings and will follow on the line immediately under the P-VTEC.

Here is a quick and dirty summary of the P-VTEC format:

/AAA.CCCC.PP.S.####.YYMMDDHHNNZ-YYMMDDHHNNZ/

AAA - action (new, continued, extended in time/area, upgraded, cancelled, expired, routine, correction, test)

CCCC - NWS office modernized site ID that has normal responsibility for the affected area (eg KLWX)

PP - phenomena (blizzard, freeze, severe thunderstorm, tornado, etc.)

S - significance (warning, watch, advisory, statement, outlook, forecast, synopsis)

- event tracking number (unique ID for this product, each office has its own set of tracking numbers). Tracking numbers are incremented during each calendar year for each phenomena type. The event tracking number is referred to as ETN.

YYMMDDHHNNZ-YYMMDDHHNNZ - date/time when the product becomes effective (not when it is issued) followed by the date/time when the product expires.

Here is a quick and dirty summary of the H-VTEC format:

/S.IC.YYMMDDTHHMMZ.YYMMDDTHHMMZ.YYMMDDTHHMMZ.FR/

S - flood severity (none, minor, moderate, major, unknown)

IC - immediate cause (excessive rainfall, snow melt, dam or levee failure, ice jam, etc.)

YYMMDDTHHMMZ.YYMMDDTHHMMZ.YYMMDDTHHMMZ - date/time of the flood start, crest and end.

FR - flood record status (is this a record flood)

An example of an H-VTEC follows:

/1.ER.020426T2000Z.020430T1100Z.020503T1300Z.NR/

Here we repeat the MHX tornado warning heading with the P-VTEC:

WFUS52 KMHX 061735

TORMHX

NCC047-061815-

/NEW.KMHX.TO.W.0003.0204061730Z-0204061815Z/

This is a new tornado warning for NC county 47 (in the CWA of site KMHX), the 3rd tornado warning issued for the CWA during this calendar year. The warning is effective April 6, 2002 1730Z until April 6, 2002 1815Z. There is a VTEC troubleshooting section beginning on page 65.

2.2 TEXT PRODUCT SEGMENTATION

NWS text products are either segmented or non segmented. Non segmented products apply to only one geographic area, and the list of UGC codes are in the product heading. Generally warnings are non segmented. Here is a heading example of a non segmented product:

WUUS51 KLWX 111354
SVRLWX
MDC017-037-VAC099-111445-

This severe thunderstorm warning is for Maryland counties 017, 037 and Virginia county 099.

A segmented product applies to more than one geographic area and each segment of the product applies to one geographic area. A segmented product does not have a UGC list in the communications heading. An example of a segmented product (the beginning of a zone forecast) follows:

FPUS51 KLWX 050742
ZFPLWX

ZONE FORECASTS
NATIONAL WEATHER SERVICE BALTIMORE MD/WASHINGTON DC
300 AM EST WED NOV 5 2003

DCZ001-MDZ011-VAZ054-052100-
ARLINGTON-DISTRICT OF COLUMBIA-SOUTHERN BALTIMORE-
INCLUDING THE CITIES OF...ALEXANDRIA AND FALLS CHURCH...
BALTIMORE...WASHINGTON
300 AM EST WED NOV 5 2003

.TODAY...AREAS OF FOG AND DRIZZLE THIS MORNING WITH OCCASIONAL SHOWERS BY AFTERNOON. HIGHS IN THE UPPER 60S. SOUTHWEST WINDS AROUND 10 MPH.

.TONIGHT...OCCASIONAL SHOWERS. LOWS IN THE UPPER 50S. LIGHT WINDS.

.THURSDAY...SHOWERS LIKELY DECREASING IN THE AFTERNOON. HIGHS IN THE LOWER 60S. LIGHT WINDS BECOMING NORTH AROUND 10 MPH IN THE AFTERNOON. CHANCE OF SHOWERS 60 PERCENT.

.THURSDAY NIGHT...COOLER. MOSTLY CLOUDY. LOWS IN THE MID 40S.
.FRIDAY...PARTLY CLOUDY. HIGHS IN THE UPPER 50S.
.FRIDAY NIGHT...MOSTLY CLEAR. LOWS IN THE MID 30S.
.SATURDAY...MOSTLY CLEAR. HIGHS IN THE UPPER 40S.
.SUNDAY...MOSTLY CLEAR. LOWS IN THE LOWER 30S. HIGHS AROUND 50.
.MONDAY...PARTLY CLOUDY. LOWS IN THE UPPER 30S. HIGHS IN THE
UPPER 50S.
.VETERANS DAY...PARTLY CLOUDY. LOWS IN THE LOWER 40S. HIGHS
AROUND 60.

\$\$

MDZ002-WVZ048-049-052100-
ALLEGANY-GRANT-MINERAL-
INCLUDING THE CITIES OF...BAYARD AND PETERSBURG...CUMBERLAND...
KEYSER
300 AM EST WED NOV 5 2003

.TODAY...AREAS OF FOG THIS MORNING WITH OCCASIONAL SHOWERS BY
AFTERNOON. HIGHS 67 TO 72. SOUTHWEST WINDS AROUND 10 MPH.
.TONIGHT...OCCASIONAL SHOWERS...MAINLY BEFORE MIDNIGHT. LOWS IN
THE LOWER 50S. LIGHT WINDS.
.THURSDAY...SHOWERS LIKELY DECREASING IN THE AFTERNOON. HIGHS
IN THE MID 50S. LIGHT WINDS. CHANCE OF SHOWERS 60 PERCENT.
.THURSDAY NIGHT...MOSTLY CLOUDY. LOWS IN THE UPPER 30S.
.FRIDAY...PARTLY CLOUDY. HIGHS AROUND 50.
.FRIDAY NIGHT...MOSTLY CLEAR. LOWS AROUND 30.
.SATURDAY...MOSTLY CLEAR. HIGHS IN THE LOWER 40S.
.SUNDAY...MOSTLY CLEAR. LOWS IN THE MID 20S. HIGHS IN THE MID 40S.
.MONDAY...PARTLY CLOUDY. LOWS IN THE LOWER 30S. HIGHS IN THE MID
50S.
.VETERANS DAY...PARTLY CLOUDY. LOWS IN THE UPPER 30S. HIGHS IN THE
MID 50S.

\$\$

The first line is the WMO heading and the second line is the AWIPS product ID. The next three line section is the Mass News Disseminator (MND).

The segments follow the MND. Each “segment heading” begins with the relevant UGC list. The first segment above includes DC forecast zone 001, MD zone 011, and VA zone 054. The text of the product for this segment comes next. Each segment ends with the segment trailer “\$\$”.

The second segment pertains to MD zone 002 and WV zones 048 and 049. This pattern continues throughout the rest of the segmented product. Usually, after the last segment trailer “\$\$”, the forecaster’s initials appear.

3. LOCALIZATION

The files provided to a field site by an AWIPS software install or upgrade are not sufficient, in and of themselves, to support operations at the site. Localization is the process by which geographic specific software customization is completed. Localization also allows the site to modify the behavior of various aspects of AWIPS operation, according to their local needs. Localization is a complex subject and this guide focuses on localization topics important to WarnGen.

3.1 LOCALIZATION BASICS

To run localization, you must be user “fxa”. Localization relies on the user fxa environment variables to function correctly. Three of the important fxa environment variables are:

```
$FXA_HOME - set to “/awips/fxa”  
$FXA_CUSTOM_FILES - set to “/data/fxa/customFiles”  
$FXA_LOCAL_SITE - set to the three character local site ID, e.g. “LWX”
```

If localization is accidentally run by a user other than “fxa”, don’t abort the localization. Let the accidental localization complete, then run it as user “fxa”.

Run localization from directory /awips/fxa/data/localization/scripts using “./mainScript <options> <tasks>”. Some of the options are described in the “Localization TroubleShooting” section on page 18. The tasks refer to the various application areas of AWIPS that can be customized. All the options and tasks are described in the NWS “Localization Training and Reference Manual - Volume 1.” Also useful localization documentation from FSL is found in ds1:/awips/fxa/data/localization/documentation. If localizing workstations, the localizations should be run one at a time. That is, don’t try to save time by simultaneously running more than one workstation localization (per a 2002 awipsinfo posting).

The localization log is in /data/logs/fxa/localization.log. Each new localization is appended to the log file.

3.2 LOCALIZATION TASKS

The following summary shows what localization tasks are run for the various localization command line options. This section was derived from mainScript.csh, build OB2.

The following tasks are run in this order if the command “mainScript.csh” is used (no options).

This is the preferred way to set up full D2D workstation functionality for your local site:

```
dataSups scales clipSups tables text topo grids radar maps wwa station
```

The following tasks are run in this order using the -all option (i.e., “mainScript.csh -all”):

```
dataSups scales clipSups tables text laps msas topo grids radar maps wwa station dirs auxFiles
```

The tasks run using the -install option are:

```
dataSups scales clipSups tables text topo grids radar maps wwa station dirs auxFiles scan
```

The tasks run using the -syncGini option are:

```
dataSups scales clipSups tables syncGini
```

The tasks run using the -DS option are:

```
dataSups scales clipSups tables text topo grids radar dirs auxFiles
```

The tasks run using the -AS option are:

```
dataSups scales clipSups tables text laps msas topo grids radar
```

The -WS option runs the tasks needed for all applications running on a workstation. The -WS option should only be run at install time. These tasks are:

```
dataSups scales clipSups tables text topo grids radar maps wwa station scan
```

The -WWA option does the workstation localization tasks needed to support WarnGen full service backup (see page [36](#)). This is different than the “-wwa” task. The “-wwa” task only does things needed by the local version of WarnGen and is described in detail beginning on page [25](#). The “-WWA” tasks are:

```
dataSups scales clipSups tables text maps wwa station
```

The localization tasks most important to WarnGen are “maps”, “wwa” and “station.” The “maps” task uses ArcView shape files to create D2D vector map backgrounds. The “wwa” task creates geographic entity lookup tables (GELTs) used by WarnGen and WWA to find geographic features. The “wwa” task also creates the template files used by WarnGen and WWA to format the text of warning products. The “station” task constructs files needed by D2D to plot station and location information on map backgrounds. These three tasks are described in more detail in the section “Geography Files Used by D2D and WarnGen” beginning on page [20](#).

3.3 LOCALIZATION INPUT FILES AND UNIX

Localization inputs may come from either the /awips/fxa or /data/fxa file systems. It is helpful to understand how these Unix file systems are accessed by AWIPS. In general, files in the /awips/fxa file system are local to each AWIPS host. In general, files in the /data/fxa file system physically reside on DS1 and /data/fxa is NFS mounted by the workstations.

An older name for /data/fxa/nationalData is /awips/fxa/data/localization/nationalData (which is used in some FSL documentation). On the workstations, /awips/fxa/data/localization/nationalData is a Unix symbolic link to ds1:/data/fxa/nationalData. That is, if you refer to /awips/fxa/data/localization/nationalData, this is actually on ds1, not on the workstation, so this violates the general rule that /awips/fxa files are local to the workstation.

Before OB1, the /data/fxa file system physically resided only on ds1 and was NFS mounted by the other AWIPS hosts, i.e., /data/fxa was shared by all the workstations. The /awips/fxa file system was local to each AWIPS host, i.e., each workstation had its own copy of /awips/fxa. If you changed something in /data/fxa, it was changed for all workstations. If you changed something in /awips/fxa, it was been changed on only one host.

Beginning with OB1, ds1:/data/fxa can be referred to as “dsdata” when logged on to the workstations. Also, beginning with OB1, each workstation was given a local partition named /data/fxa. This can cause some confusion. In summary, you can still refer to the /data/fxa and /awips/fxa file systems, and things will work the same old way. If you need to create files or directories in these file systems, it’s helpful to know what’s happening Unix-wise.

Beginning with OB1, each workstation accesses ds1:/data/fxa via an NFS mount on the workstation in /dsdata. For example, lx1 has a mount point /dsdata, on which ds1:/data/fxa is mounted. Use the Unix command “mount” to see the currently mounted file systems. If logged onto ds1, this file system is accessed using “/data/fxa”. If logged on to a workstation, use either “/data/fxa” or “/dsdata”. For example, using “/data/fxa/nationalData” on ds1 and lx1 accesses the exact same directory as using “/dsdata/nationalData” on lx1.

Beginning with OB1, the workstations also have a local partition named “/data/fxa”. This is local to the workstation and contains lots of Unix symbolic links to directories on ds1:/data/fxa and px1:/data/fxa. If on lx1, you refer to “/data/fxa/nationalData”, this is a symbolic link to lx1:/dsdata/nationalData which is the lx1 location where ds1:/data/fxa/nationalData is NFS mounted. So a reference to lx1:/data/fxa/nationalData is actually a reference to ds1:/data/fxa/nationalData. If on lx1, you do something like “mkdir /data/fxa/flatText”, this is creating the flatText directory on lx1, not on ds1. If on lx1, you want to create directory ds1:/data/fxa/flatText (global to all workstations), you need to do “mkdir /dsdata/fxa/flatText”. If on ds1, you do “mkdir /data/fxa/flatText”, this creates the directory on ds1, and it is global to all workstations.

Similar things have happened with OB1 ds1:/data/fxa/customFiles. That is, each workstation has a /data/fxa/customFiles (local to the workstation) which is actually a symbolic link to ds1:/dsdata/customFiles, the workstation location where ds1:/data/fxa/customFiles is NFS mounted.

Here is info on manipulating Unix symbolic links. Below is the “ls -l” listing for the lx1 link to /dsdata/nationalData (that is, the link to ds1:/data/fxa/nationalData):

```
lrwxrwxrwx 1 root root 20 Sep 24 12:21 nationalData -> /dsdata/nationalData
```

If you need to remove this link, on lx1 as “root” use “rm /data/fxa/nationalData”. This will not affect ds1:/data/fxa/nationalData, you are just removing the lx1 reference to it. If you need to create this symbolic link, on lx1 as “root” use:

```
ln -s /dsdata/nationalData /data/fxa/nationalData
```

The first directory name is the physical directory (here /dsdata/nationalData - that is, ds1:/data/fxa is mounted on lx1:/dsdata) and the second directory is the name of the link (/data/fxa/nationalData).

3.4 LOCALIZATION INPUT FILES AND FILE OVERRIDE

There are various places where localization inputs may be located. If localization is looking for an input file named dataFile, it looks in the following places in the following order:

```
/awips/fxa/data/dataFile  
/data/fxa/nationalData/dataFile  
/awips/fxa/data/localization/realizations/RRR/RRR-dataFile  
/awips/fxa/data/localization/LLL/LLL-dataFile (“LLL-“ prefix must exist)  
/data/fxa/customFiles/dataFile  
/data/fxa/customFiles/LLL-dataFile  
/data/fxa/customFiles/$FXA_CUSTOM_VERSION/dataFile  
/data/fxa/customFiles/$FXA_CUSTOM_VERSION/LLL-dataFile
```

The directory /awips/fxa/data/localization/realizations is used only by RFC’s and National Centers. The /awips/fxa files are local to each workstation and the /data/fxa files are shared by all workstations.

Localization uses the dataFile version from the location found last in the list above (not the newest version of dataFile). This is called override. For example, if you have changed file /awips/fxa/data/dataFile but also have an older version of the same file in /data/fxa/nationalData/dataFile which has not been changed, then the old nationalData version will override the new /awips/fxa/data version and you will not see the change that you made

reflected in the localization output. For localizations run on the workstation, any localization input files in the /awips/fxa partition are physically on the workstation.

If a certain localization input file has been modified, be sure that the updated file has the current system date/time (use the Unix “touch” command if needed) so localization will see that the file has been updated. Localization will only process input files that it recognizes as being modified since the last localization. This is called “smart localization.”

There are six different kinds of override: functional, copy, replacement, append, include and mixed override. The file ds1:/awips/fxa/data/localization/documentation/fileChanges.html lists in detail all the files involved in override and what kinds of override occur by default.

Functional override means that the localization input file is used to create another localization file or control a localization process. No functional override file (localization input file) is permanently saved in /awips/fxa/data/localizationDataSets/LLL. For example, localization input file /awips/fxa/data/localization/LLL/LLL-wwaConfig.template creates localization output file /awips/fxa/data/localizationDataSets/LLL/wwa.config.

Copy override means that the override file (localization input file) is moved to /awips/fxa/data/localizationDataSets/LLL. Append override means that override files are appended to each other - that is the localizationDataSets file will contain first the nationalData items, then any items from localization/LLL/LLL*, then any items from customFiles. When a file is subject to copy override or append override, a version of that override file is always saved in localizationDataSets.

Replacement override means that the override file is moved to localizationDataSets/LLL only if the realizations/RRR/RRR-* file or localization/LLL/LLL-* file is present. We have not tested this, but replacement override implies that override files in /data/fxa/customFiles are ignored.

The different behaviors of the various types of override are subtle. See the “Localization Training and Reference Manual - Volume 1”, section 2.3 for more details. For example, you may want to have localization do a copy override (completely override the default configuration) instead of an append override (append local changes to the default configuration). A specific override troubleshooting example is in the site ID troubleshooting section beginning on page 70. An additional understanding of how file override works can be gained from the header documentation of the getPath.csh and fileGrab.csh scripts in /awips/fxa/data/localization/scripts.

In the case of replacement and append override only, you can set a switch to specify the override that you want. In the first line of the localization input file, add this to request replacement override:

```
#replace
```

Add the following to request append override:

```
#append
```

A local override file that can easily cause lots of problems is the D2D process configuration file `ws.config` which sets many D2D default behaviors. A local version of `ws.config` can be created which localization appends to the default `/data/fxa/nationalData/ws.config`. If the local `ws.config` is not created correctly, many items can be duplicated by localization when it does an append override. In other words, `/data/fxa/nationalData/ws.config` specifies all the national default parameters. If `/awips/fxa/data/localization/LLL/ws.config` is created by copying the default `ws.config` file and editing it, any items remaining from the default `ws.config` file will be appended by localization. This will result in duplicate `ws.config` entries in the localization output file `/awips/fxa/data/localizationDataSets/ws.config`. This can cause many types of strange D2D behavior and excessive log files being generated.

3.5 SITE SPECIFIC CONTROL FILES

Local changes to the default localization can be made via site specific control files or customization files.

Site specific control files are in directory `/awips/fxa/data/localization/LLL`. The site specific control files must begin with “LLL-“ where LLL is the site ID. Customization files are in directory `/data/fxa/customFiles`. The customization files can begin with “LLL-“ but do not have to begin with “LLL-“. Site specific control files are created by the site to control AWIPS software behavior on that particular workstation. The `/awips/fxa` partition is local to each workstation. Site specific control files are often modified versions of national default configuration files supplied by AWIPS software installs in directory `/data/fxa/nationalData`.

Two site specific control files of interest here are `LLL-wwaConfig.txt` and `LLL-mainConfig.txt`. `LLL-wwaConfig.txt` controls how WarnGen behaves (see the section on WarnGen localization directives beginning on page [51](#)). `LLL-mainConfig.txt` controls other localization features (see section the section on troubleshooting the site ID beginning on page [70](#)).

There are various other site specific control files in directory `/awips/fxa/data/localization/LLL`. These deal with many other things like data keys, depict keys, radar items, hydro items, etc.

3.6 CUSTOMIZATION FILES

Another way that a site can change the default localization is by using customization files. Customization files are also referred to as custom control files. Customization files are in directory `/data/fxa/customFiles`. Note that changes to customization files apply to all workstations, that is, the `/data/fxa` partition physically resides on `ds1` and is NFS mounted by all the workstations. Therefore, this is the easiest place to make the same custom change to all

workstations.

An important feature of directory `/data/fxa/customFiles` is that this directory is guaranteed not to be overwritten by AWIPS software upgrades. In other words, `/data/fxa/customFiles` is the **safest** place to put your site specific configuration changes. The directory `/awips/fxa/data/localization/LLL` is not as safe a location in AWIPS software upgrades.

Also be aware that in localization file override, customization items in `/data/fxa/customFiles` will override site specific control items in `/awips/fxa/data/localization/LLL`. See the list of localization input directories on page [13](#).

The customization files can begin with “LLL-“ but do not have to begin with “LLL-“ like the site specific control files in `/awips/fxa/data/localization/LLL`. Customization files that begin with “LLL-“ apply only to that specific site localization. For example, customization file `wwaConfig.txt` applies to all localizations run at the site. However, customization file `LWX-mainConfig.txt` applies only to localizations where LWX is specified as the site ID. This comes into play when D2D is configured in full service backup mode for a neighboring site (see page [36](#)). A detailed example of customization file use is explained below in the site ID troubleshooting section beginning on page [70](#).

You can create your own specialized custom localization directories in `/data/fxa/customFiles/$FXA_CUSTOM_VERSION`, for example, you could create a set of winter and summer customization files. If you created `/data/fxa/customFiles/winter`, you would set the environment variable `FXA_CUSTOM_VERSION` to “winter”, then run localization using the “n” option. For example, if you want to set up a workstation to use the winter localization files, do the following as user fxa:

```
cd /awips/fxa/data/localization/scripts
setenv FXA_CUSTOM_VERSION winter
./mainScript.csh n
```

If you want to be creative, you can specify a different custom directory for localization to use instead of using `/data/fxa/customFiles`. That is, on one workstation, you could create directory `/awips/fxa/awipsusr/ob2/customFiles` and put some extra special WarnGen customFiles items in this directory, then do the following as user fxa:

```
cd /awips/fxa/data/localization/scripts
setenv FXA_CUSTOM_FILES /awips/fxa/awipsusr/ob2/customFiles
./mainScript.csh n -wwa
```

To get really creative, you can use both `FXA_CUSTOM_FILES` and `FXA_CUSTOM_VERSION`. In this case, `FXA_CUSTOM_VERSION` must be a subdirectory of `FXA_CUSTOM_FILES`.

3.7 LOCALIZATION DIRECTIVES

Site specific control files and customization files can contain localization switches called directives. Directives control how localization output data sets are created. These determine characteristics of how various localization functions behave. For example, directives can control how large a section of a county must be within the warning box before WarnGen includes the county in a warning product. Directives can also be used in WarnGen configuration files in /data/fxa/nationalData and also in text product templates to produce direct text substitution.

The two site specific control files (in /awips/fxa/data/localization/LLL) that use directives are LLL-mainConfig.txt and LLL-wwaConfig.txt. LLL-mainConfig.txt contains directives that can define additional localizations. For more information, see the FSL documentation in file ds1:/awips/fxa/data/localization/documentation/localization.html. Customization versions of these files should be created in /data/fxa/customFiles. Beware that override also applies to directives, e.g. directives in /data/fxa/customFiles override directives in /awips/fxa/data/localization/LLL and /data/fxa/nationalData.

The file /data/fxa/nationalData/wwaDefaults.txt contains default national values for some of the directives in LLL-wwaConfig.txt. Any LLL-wwaConfig.txt values will override those in wwaDefaults.txt. More on how to use directives is in the WarnGen localization directives section beginning on page [51](#).

3.8 PROCESS CONFIGURATION FILES AND WARNGEN

Process configuration files control how a processes on the workstation or data ingest behave. The main process configuration file is /awips/fxa/data/fxa.config. Among other files this includes ws.config (which controls many D2D behaviors and is described briefly on page [15](#)) and wwa.config (which controls the WarnGen product menu).

The file /awips/fxa/data/localizationDataSets/LLL/wwa.config is generated from /data/fxa/nationalData/wwaConfig.template and contains the list of active WarnGen product titles and file names, as well as a list of the alternate site ID's which can restart WarnGen. Beware that wwaConfig.template is subject to override. File wwaConfig.template is described in more detail beginning on page [40](#). Process configuration files are explained in ds1:/awips/fxa/data/localization/documentation/localization.html.

3.9 LOCALIZATION OUTPUT FILES

Localization output files for site LLL are in directory /awips/fxa/data/localizationDataSets/LLL. These take up a lot of space on the workstation hard disk. To save space on the workstation, some sites link these workstation directories to directories on the DS, which contain the actual localization output files. If you want to remove a certain localization, you can just remove the appropriate /awips/fxa/data/localizationDataSets/LLL directory.

In diagnosing a localization problem it may be helpful to edit these output files to help understand how localization output files produce certain results in WarnGen. That is, make a change to the appropriate file in /awips/fxa/data/localizationDataSets/LLL, stop and restart D2D, then test WarnGen to see if you get the desired result.

It is much better to fix the localization input files that affect the WarnGen problem. Don't just change a localization output file to fix a localization problem. Doing this may temporarily produce the desired result in WarnGen, but the fix will be lost the next time localization is run.

3.10 LOCALIZATION TROUBLESHOOTING

If localization is hung, and you can't abort it, you'll need to use the Unix 'ps -efl' to find the associated localization programs that are running and then 'kill' them. The 'l' (el) option in 'ps' lets you see the often very long names of localization processes. As a result of localization ending abnormally, there will likely be a file /awips/fxa/data/localizationDataSets/LLL/.unSafe that must be removed before localization can be run again.

Look at the localization log file to see if there are errors. On each workstation, the log file is /data/logs/fxa/localization.log. In general, the localization options used will be listed at the top of the localization, but the "f" option will not be listed. Also, the "v" option (verbose) generates verbose output only to standard output, not to the log file. When using "v", you need to redirect the output to a file as described later in this section. The "t" option also does not generate information for the localization log file.

In the log, where you see "running makeMapFiles.csh" this is the localization task "-maps". The "running makeWWAtables.csh" section is the "-wwa" task. The "running makeStationFiles.csh" section is the "-station" task. There can be lots of other localization tasks in the log which are beyond the scope of this paper. For more details on the tasks related to WarnGen ("-maps", "-wwa", "-station") see the D2D/WarnGen geography files section beginning on page 20.

If localization produces an unexpected result, check for a rogue version of a localization input file in the list of localization input directories on page 13. An unwanted file override may be occurring. Past experience at the SST has shown that subdirectories should not be put under /data/fxa/nationalData. Doing this can have unpredictable results and will confuse localization.

Also, if localization isn't producing the result you expect, a good troubleshooting method is to use the "f" option which tells localization to process everything possibly needed for the particular task selected. This is a "brute force" method and may fix the problem. Using the "f" option will disable "smart" localization.

If localization produces an error message that you don't understand or generates a strange result, localize with the verbose option "v". This will echo the commands executed in each localization script and may help you to find errors. If the "v" option is used, logging is turned off and output

goes only to the standard output. To save the output from the “v” option (you’ll need to, since lots of stuff will be generated) use this:

```
./mainScript.csh v f -wwa |& tee loc.diag
```

This runs a verbose, forced wwa localization and puts the output into file loc.diag.

You can also add debug statements to localization scripts to print values of variables or other useful troubleshooting items. For example, to check the value of shell variable “cityId” add this statement:

```
echo “cityId = ‘$cityId’ “ (to do the substitution, use ‘apostrophe’, not `backquote`)
```

If you narrow a localization problem to a specific localization script or executable file, compare the file with the same file from another site. Sometimes these files get modified or corrupted.

If localization is failing only on certain workstations, one approach to fix the problem is to copy all the localization output files from a good workstation to the problem workstation. That is, rcp the entire /awips/fxa/data/localizationDataSets/LLL directory from the good to the bad workstation. Before the copy, remove all files (and subdirectories) in the localizationDataSets/LLL directory on the problem workstation. For example, if site DDC is having a problem on lx1, do the following. Log onto lx1 as root. Then enter:

```
cd /awips/fxa/data/localizationDataSets
rm -rf DDC (be absolutely certain that you are in the correct directory!)
```

This will also remove a volume browser subdirectory (named vb) under DDC. Now copy the good localizationDataSets directory from lx2:

```
rcp -rp lx2:/awips/fxa/data/localizationDataSets/DDC . (space then dot at the end)
```

This is a draconian approach which could fix the problem, but you won’t know what caused the problem. It is crucial that you copy only among the same types of workstations, that is, copy only from HP to HP workstations or from Linux to Linux workstations.

A drastic approach is to do a system trace for the offending program. For example:

1. Log onto lx1 as fxa
2. cd /awips/fxa/data/localization/scripts
3. strace -o /tmp/zap /awips/fxa/bin/localizeForFFMP keax

This runs the program “localizeForFFMP keax” and produces a listing of all system calls in file /tmp/zap. Lots of output will be generated, but it may be useful for the software developers in

diagnosing the problem.

If you can't figure out a problem at a field site, you can configure a local workstation to try to recreate the problem or even install the site's actual localization environment on your local workstation. These are described beginning on page 74. If you localize your test workstation as the remote site and don't see the problem, it is a site specific problem. To further isolate the problem, you can compare the localization input files or configuration files at the field site and on your test machine. You could try tweaking some of the appropriate localization input or configuration files. If the problem is consistent across several configurations, you may need to consult with FSL or open a DR.

Sometime in the AWIPS 5.x era, a time saving feature was started in AWIPS software installations which has caused some confusion and possible problems. Instead of running localization on each workstation, the localization is run on the DS and the localization output files are pushed to each workstation. In OB2, this was changed so that the localization is run on lx1 and then the localization output files are pushed to each workstation. There used to be some binary file differences between Linux and HP workstations, but this issue has been resolved.

4. GEOGRAPHY FILES USED BY D2D AND WARNGEN

There are many, many geography files used by D2D and WarnGen. They are configured by the localization tasks “-maps”, “-wwa”, and “-station”. This section only describes the geography files most likely to be involved in WarnGen troubleshooting.

This section groups the geography files according to their associated localization task. The “-maps” task uses ArcView shape files to create D2D vector map backgrounds. The “-wwa” task creates geographic entity lookup tables (GELTs) used by WarnGen to find geographic features. The “-wwa” task also takes the product template files and creates the product definition files used by WarnGen to issue products. The “-station” task constructs files needed by D2D to plot meteorological stations and geographic location names.

4.1 THE “-maps” LOCALIZATION TASK AND WARNGEN

The localization “-maps” task uses ArcView shape files to create D2D vector map backgrounds. This section describes shape files.

4.1.1 WARNGEN AND D2D SHAPE FILE SETS

The localization “-maps” task uses ArcView shape files to create D2D vector map backgrounds. ArcView is a widely used geographic information system (GIS) software package. Each vector map background theme in D2D (e.g. county boundaries, marine zones) is derived from an ArcView shape file set. Each shape file set consists of three files in a proprietary ArcView format. The three files created by ArcView have the extensions .dbf, .shp, and .shx.

In AWIPS, the shape file sets are in /data/fxa/nationalData. To save space, the .shp file is compressed, so on AWIPS, you see shape file sets with extensions .dbf, .shp.Z and .shx.

Here are the AWIPS shape file sets that are most important to WarnGen (there are other shape files, but they are beyond the scope of this discussion):

- usa_cwa - has cwa boundaries
- c11-zone - NWS public forecast zones, one closed polygon per zone
- uscounty - county boundaries, the huge majority have one closed polygon per county
- usa_cwa_total - county boundaries for Warngen partial backup
- marine_zones - marine zones
- marine_total - marine zones for Warngen partial marine backup - new in AWIPS 5.1.1
- usa_wsfo - boundaries for the SLS product (redefining area)

The AWIPS baseline shape file sets are maintained at NWS headquarters by OS&T, and are available for sites to download from the NOAA1 server. Instructions for downloading and installing shape file sets are in the AWIPS System Manager's Manual. Here are the shape file names with their associated NOAA1 file names (in parentheses):

- usa_cwa (w_ddmmyy) "ddmmyy" specify the file creation data on NOAA1
- c11-zone (z_ddmmyy)
- uscounty (c_ddmmyy)
- usa_cwa_total (wtddmmyy)
- marine_zones (mzddmmyy)
- marine_total (mtddmmyy)
- usa_wsfo (usa_wsforddmmyy)

Every now and then sites should download a fresh copy of the shape file sets. Sometimes they may be required to do so because of a local geography/zone change or a new AWIPS feature. OS&T creates shape file sets using ArcView. Some sites also have ArcView (normally a PC Windows version) and can create their own local versions of shapefiles. For more info on AWIPS shapefiles see <http://www.nws.noaa.gov/geodata>

A site can use ArcView to create their own local D2D background maps. In D2D the option to display the maps is under "maps" then "user shape file." There is some FSL information on displaying shape files in <ds1:/awips/fxa/data/localization/documentation/shapeFileDisplay.html>.

4.1.2 THE "-maps" LOCALIZATION TASK

After new shape files are downloaded, the localization `./mainScript.csh -maps` does most of the processing needed for the new shape files. Make sure that the new shape files have the current system date/time (so localization sees the new files) or use `./mainScript.csh f -maps` to have localization process all possible map files. Loading a new shape file may also affect items in the

“-wwa” and “-station” tasks also, so it’s best to run “./mainScript.csh f +maps” to make sure that all possible items associated with the shape files are processed.

The “-maps” task uses script makeMapFiles.csh to make the files that the workstation reads to draw D2D vector map backgrounds. Script makeMapFiles.csh uses program “shp2bcd” to convert ArcView shape files into binary cartographic data files (.bcd files) or extended binary cartographic data files (.bcx files). The .bcd and .bcx files are read by D2D to draw the vector map backgrounds. Also, makeMapFiles.csh uses program “bcdProc” to perform operations such as clipping and thinning .bcd and .bcx files.

.bcd files can be used only for drawing lines. The .bcx files can draw annotated lines (like the D2D interstate highways). Most other geographic entities, e.g. counties, have a separate label in the center of the county that comes from another file, so .bcd files are used for these shapes.

Remember that the .bcd files are localization output files. Some .bcd files are saved in ds1:/data/fxa/nationalData. Many are moved to the workstations in /awips/fxa/data/localizationDataSets/LLL. Larger .bcd files seem to be on ds1 and clipped or thinned versions of the larger ds1 .bcd files seem to be on the workstations. There are also .bcd files on the workstations that begin with “total.” These probably are used for WarnGen partial backup (since the “total” shape files are for WarnGen partial backup).

4.1.3 TROUBLESHOOTING SHAPE FILE PROBLEMS

If something looks strange in D2D map shapes (boundary lines, rivers, roads, etc), check in /data/fxa/nationalData to see how old the site’s shape files are. If they’re somewhat old, new shape file sets should be downloaded from the NOAA1 server. Make sure the new shape files have the current system date/time. If not, use the Unix “touch” command. This will assure that localization sees the new shape files. Localize using “mainScript.csh +maps”.

If changes are made to a localization input file and changes aren’t reflected in the output files or in WarnGen, try localizing with “mainScript.csh f +maps”. The “f” forces localization to process all possible items related to maps. The “maps” “wwa” and “station” localization tasks perform various functions related to geography. Instead of trying to figure out which localization task you need to run, it’s often easier to use “mainScript.csh f +maps.”

If the WarnGen shape files aren’t working, use “cksum” to be sure that each part of the shape file set is exactly the same as on the NOAA1 server. They must be ftp’d in binary mode. Also check for some fumble fingered typing when the shape files are renamed after they are downloaded. Be sure that the file extensions .dbf, .shp.Z, and .shx are correct.

There should be no files with extension .shp - this is the uncompressed version of the .shp.Z file. If both .shp and .shp.Z exist, that is bad news for localization - remove the .shp file. Also, be sure that the only shape files are in nationalData. They should not exist in any other location.

If the map shapes look wrong, you can download the offending shape file set and test it in ArcView. In ArcView you need the uncompressed version of the shape file set (use file *.shp instead of *.shp.Z). The SST can also check ArcView shape files.

Sometimes there is confusion about which sites perform service backup for a particular county (see the service backup section on page 36). You and/or the site may need to check the appropriate operational procedures that define service backup for the site. Sometimes there are errors in the shape file set, so OS&T would need to fix the shape files and put the corrected version on the NOAA1 server.

You can check the integrity of shape file sets (make sure the 3 pieces are in sync) by interactively running the AWIPS “shp2bcd “ command. For example “shp2bcd i usa_cwa” provides a summary of the usa_cwa shape file set. “shp2bcd i” should produce a listing with a “.shp FILE HEADER” section (this has a brief list of parameters) and a long “.dbf FILE HEADER” section with descriptions of the dbf records and fields. In the “.dbf FILE HEADER” section there must be fields and field names listed, or something is wrong. After this, the contents of each record and their fields are listed. In the “.dbf FILE HEADER” section, the “num dbf records” field must match the actual number of records that are listed. That is, if “num dbf records” is 121, the contents of record 0 through record 120 should be at the end of the listing. For details see the FSL documentation in ds1:/awips/fxa/localization/documentation/shp2bcd.doc.html.

The following is the beginning of the output from the command “shp2bcd i usa_cwa”:

```
.shp FILE HEADER:
-----
filecode 9994
filelength 4272398
version 1000
shape type 5
polygon
File bounding box:
(-2.68156e+154,6.85793e+42)
(2.68156e+154,2.68156e+154)
-----
.dbf FILE HEADER:
-----
Num dbf records 121
Length of dbf record 43
Fields:
-----
Field 0
-----
Field Name: WFO
```

Field type: character
field Length: 3
field decimal count: 0

Field 1

Field Name: CWA
Field type: character
field Length: 3
field decimal count: 0

Field 2

Field Name: LON
Field type: numeric
field Length: 18
field decimal count: 5

Field 3

Field Name: LAT
Field type: numeric
field Length: 18
field decimal count: 5

Field Names:

0: WFO Type: character
1: CWA Type: character
2: LON Type: numeric
3: LAT Type: numeric

---- Record 0 -----

WFO: EYW
CWA: EYW
LON: -81.5513
LAT: 24.6394

---- Record 1 -----

WFO: BMX
CWA: BMX
LON: -86.5298
LAT: 33.3195

```
-----  
---- Record 2 -----  
WFO: MOB  
CWA: MOB  
LON: -87.7369  
LAT: 31.2571  
-----
```

If you see a .bcd file with size zero, or the .bcd files have not been newly created by the localization, then something is wrong. D2D won't be able to draw the feature that has a size zero .bcd file. I don't know of any way to directly check the contents of .bcd files. If you get a non-zero size .bcd file after localizing, the way to check the .bcd file is to try to display the associated feature in D2D. That is, if localization has created a non-zero size c11-zone.bcd file, then in D2D, use the "maps" drop down menu and display the NWS forecast zones. Also, if localization does not create a new .bcd file, remove the old .bcd file to entice localization to build a new .bcd file.

Another source of errors with shape files are stray copies of .bcd files on the system. It seems like localization in some AWIPS software releases may put certain .bcd files in different locations. The result is sometimes an older .bcd file may not get overwritten which will confuse D2D when it tries to display map backgrounds. Poke around in ds1:/data/fxa/nationalData and on the workstation in /awips/fxa/data/localizationDataSets/LLL and see if there are any old or duplicate .bcd files. If so, remove all of them and localize using "./mainScript.csh f +maps".

It may be necessary to recreate the site's localization environment to help diagnose a problem. There are various ways to do this and they are covered beginning on page [74](#).

4.2 THE "-wwa" LOCALIZATION TASK AND WARNGEN

The "-wwa" task uses script makeWWAtables.csh to create geographic entity lookup tables (GELTs) used by WarnGen and WWA to determine what geographic features are at a given latitude/longitude coordinate. Script makeWWAtables.csh also takes the product template files and creates the product definition files used by WarnGen to issue products.

The script newGELTmaker.csh (this replaced program makeGEOtables) creates the GELTs. GELTs allow WarnGen to figure out which geographic entities exist at some latitude/longitude point or within some area (e.g. a warning box). Program newGELTmaker uses shape files, .bcd files, and ASCII files (e.g. CitiesInfo.txt) to create the GELT files used directly by WarnGen. Shape files are the main input used to create GELTs. Program newGELTmaker can be run interactively as a stand alone program. File ds1:/awips/fxa/data/localization/documentation/newGELTmaker.doc.html has the FSL documentation.

On the workstations, localization creates GELTs in /awips/fxa/data/localizationDataSets/LLL. Special GELTs are created for use on the D2D regional scale (these GELT file names begin with “reg”). Other GELTs are also created for use on the D2D state and WFO scale which include the CWA and service backup areas (these GELT file names begin with “wwa”).

4.2.1 GELTs (GEOGRAPHIC ENTITY LOOKUP TABLES)

Numerous GELTs are created by localization, but we’ll only discuss those which are used the most often by WarnGen. At the regional scale, there are these three GELTs:

- reg_counties - counties (include county FIPS codes)
- reg_zones - NWS forecast zones
- reg_marine - NWS marine zones

At the state and WFO scale, there are these GELTs:

- wwa_counties - counties (include county FIPS codes)
- wwa_zones - NWS forecast zones
- wwa_marine - NWS marine zones
- wwa_marine_sites - marine locations
- wwa_warn_city - cities used by WarnGen
- wwa_watch_counties - statewide counties for the SLS product
- wwa_watch_city - statewide cities for the SLS product

There are 6 files that comprise one GELT (in this case the wwa_zones GELT):

- wwa_zones.gelt - ASCII file describes the grid and list the other parts of the GELT
- wwa_zones.id - ASCII file lists each geographic entity with lat/lon
- wwa_zones.entity - ASCII file that lists each contiguous area in the table and maps each contiguous area to entities in the .id file
- wwa_zones.table - binary file with pointers to items in .entity file
- wwa_zones.NS - binary file describes how far N/S a point is within an entity
- wwa_zones.EW - binary file describes how far E/W a point is within an entity

The *.id files are useful for determining what items are in the GELT. Here are the first three lines of file wwa_zones.id:

```
93
2 35.862 -75.642 a 2 OUTER BANKS DARE | NCZ103 | MHX
3 35.769 -75.869 a 3 MAINLAND DARE | NCZ047 | MHX
```

The first line of the file always has the number of entries in the file. The first five fields on each subsequent line are space delimited. Fields one and five are sequence numbers that are internal

to the GELT software and can be ignored. Fields two and three are latitude/longitude in decimal degrees (not minutes, seconds). Field four is the entity type as follows:

p = point entity defined by one grid point
a = area centroid point is area weighted
A = area with arbitrary centroid provided by user
m = area centroid is point within the entity furthest from any boundary

The rest of each entry is arbitrary text associated with each entry. The vertical bars are field delimiters within the arbitrary text. Text product templates have translation control strings which control how the text is used in WarnGen products (see FSL doc TextTemplate.html). The text here contains the NWS forecast zone description, zone name and associated WFO.

When testing WarnGen related localizations, you can often look at the GELT .id files to see if the localization worked correctly. This is easier than exporting a site's D2D display to see what happens in WarnGen or having the site run WarnGen and report back.

Looking at the regional scale county GELT and the WFO scale county GELT, we observe the following. The reg_counties.id file lists all counties that D2D displays at regional scale. On D2D at regional scale, the county boundaries plot (using the .bcd file derived from the shape file), but county names cannot be displayed. At an east coast site, regional scale includes almost the entire eastern half of the U.S. It is unlikely that WarnGen would ever be used at this large scale, but at least the geography is available if needed. File wwa_counties.id lists counties that plot on D2D WFO scale. At this scale, the county names can also be displayed. Recall that the D2D graphics come from the shape files, while WarnGen uses the GELTs to determine names of places within a warning area.

There are other wwa_warn_city files that are related to the wwa_warn_city function. File wwa_warn_city.use is created by makeWWAtables.csh and is used in creating the wwa_warn_city and the wwa_watch_city GELTs. File wwa_warn_city.cities is also created and used by makeWWAtables.csh.

4.2.2 COUNTIES, ZONES AND GELTS

In most cases, there is a one-to-one relationship between NWS forecast zones and counties. Some counties contain more than one zone. A small number of zones span more than one county, especially in the western U.S. Zones that are not co-incident with county boundaries are based on local climatology to keep similar weather patterns in same zone.

Most WarnGen products are county based (such as warnings) and a small number zone based (e.g. short term forecast). In general, WarnGen uses the wwa_counties and wwa_zones GELTs to determine what counties and zones are in the warning box. These GELTs are also used to construct the lists of UGC codes and county/zone names that appear in the warning products.

There are some interesting situations in the way that GELTs map county, zone and UGC codes. In the shape files, each geographic entity (county or zone) is a closed polygon. There is not a separate shape file with UGCs, but the UGCs are assigned in the county shape file. Also, the WarnGen county GELTs contain the UGC, but the WarnGen zone GELT does not contain the UGC. Most WarnGen products are county based, so it doesn't seem too hard for WarnGen (using GELTs) to get the UGC, but it seems like it would be hard for WarnGen to get UGCs for zone based products, such as the NOW.

Some counties require more than one closed polygon to define it in the shape file. For example, the UGC for Dare County was assigned to the two names "Mainland Dare" and "Outer Banks Dare." This was also done for Monroe County, Florida, which also has a mainland part and a Florida Keys part. Localization defines a unique closed polygon using a combination of state names and county names. When the state/county combination does not result in only one unique item, FSL created a custom GELT script file for sites MHX and EYW named /data/fxa/customFiles/wwa_counties_gsf.txt (see page 30 for more on GELT script files) to use a different state/county area name combination to determine uniqueness. In effect this allows one UGC (one county) to have more than one name. In the case of NWS forecast zones, I don't know if there are any zones that are defined by more than one closed polygon, but I'd guess the GELT script file wwa_zones_gsf.txt may take care of this for WarnGen zone based products.

In the future, each county could be divided into as many as 9 subsections by making full use of the FIPS code (see the FIPS description on page 6). In other words, one county will have multiple UGCs. This will cause a complication for the GELTs. Another issue will be how NWS forecast zones will relate to the FIPS county subsections.

The following are examples of some entries from county and zone GELTs in the Dare County case. First the regional GELTs (in /awips/fxa/data/localizationDataSets/LLL), file reg_zones.id defines the NC zones 047 and 103:

```
819 35.824 -75.593 a 819 103 | NCZ103
820 35.774 -75.870 a 820 047 | NCZ047
```

Below, file reg_counties.id shows the two Dare County zones are assigned to the Dare County FIPS code of 055:

```
746 35.824 -75.593 a 746 Outer Banks Dare | NC| NCC055
747 35.774 -75.870 a 747 Mainland Dare | NC| NCC055
```

Now the local scale GELTs. File wwa_zones.id lists the Dare County zones 047 and 103 with their zone names:

```
2 35.862 -75.642 a 2 OUTER BANKS DARE | NCZ103 | MHX
3 35.769 -75.869 a 3 MAINLAND DARE | NCZ047 | MHX
```

File `wwa_counties.id` lists Dare County FIPS code 055 associated with the two zones:

```
2 35.862 -75.642 a 2 OUTER BANKS DARE |ee NC| NCC055 | MHX
3 35.769 -75.869 a 3 MAINLAND DARE |ee NC| NCC055 | MHX
```

4.2.3 INPUT FILES FOR `newGELTmaker.csh`

There are three types of files used as input for GELTs: shape files, ID files and cities files. The shape files are described above beginning on page [20](#). The FSL documentation says that the ID files have the same format as the `*.id` part of the GELT described above on page [26](#). I haven't been able to find any of these input ID files in AWIPS. Most of the county, zone and UGC info comes from the shape files, so it appears that the ID files may not be used currently. There is only one cities file: `/awips/fxa/data/CitiesInfo.txt`, with an optional override file `localCitiesInfo.txt`. `CitiesInfo.txt` and `localCitiesInfo.txt` are described beginning on page [43](#).

4.2.4 GELT SCRIPT FILES

GELT script files contain parameters that control how localization builds GELTs. GELT script files end in `_.gsf.txt` and in general, sites are not supposed to edit them. Most of them are in `/data/fxa/nationalData` and some are in `/awips/fxa/data/localization`. In rare cases, GELT script files can be in `/data/fxa/customFiles`. There are lots of keywords that can be used. See `ds1:/awips/fxa/data/localization/documentation/newGELTmaker.doc.html` for details.

4.2.5 GELT DIAGNOSTICS

If you suspect that a GELT file may be corrupted, you can use the command `GELTtest`. You must be user `fxa`. If you're testing the GELT for a site other than your local site ID, you need to be careful that you're using the GELT for the correct site. Use `"echo $FXA_LOCAL_SITE"` to get the default site ID. To change this to site `LLL`, use `"setenv FXA_LOCAL_SITE LLL"`. For example, you may need to test a GELT from site `CLE` and your AWIPS test machine has the default site ID of `LWX`. Make sure that you have enough room on the workstation, then localize using `"mainScript.csh -WWA CLE CLE"` so you can create a `CLE` GELT. Before running `GELTtest`, use `"setenv FXA_LOCAL_SITE CLE"` so that you access the `CLE` GELT and not the `LWX` GELT.

If WarnGen provides the wrong county name at a certain latitude/longitude area (say 41.25 N, 79.75 W), use the following `GELTtest` command (shown with two lines of output immediately below) to test the `wwa_counties` GELT:

```
/awips/fxa/bin/GELTtest t wwa_counties 41.25 -79.75
VENANGO
0 0 VENANGO
```

This says that WarnGen thinks Venango County is at 41.25 north latitude and 79.75 west longitude. The ‘t’ flag is needed for ‘test’. Specify north latitude in decimal degrees and hundredths (not degrees and minutes). Specify west longitude as negative degrees and hundredths. The GELT grid is accurate to about 0.01 degrees (about 1 km). More on GELTtest is at <http://www-sdd.fsl.noaa.gov/~fxa/doc/AWIPS5.0/GELTtest.doc.html>.

Here are some tricks to help determine how a GELT is being created.

Script newGELTmaker.csh can be used with the “v” option to determine the input configuration files used to create a given GELT. For example to see what is used to create the wwa_warn_city GELT, as “fxa” in /awips/fxa/data/localizationDataSets/LLL, do:

```
/awips/fxa/bin/newGELTmaker -v wwa_warn_city
```

The result follows:

```
wwa_warn_city.gelt
/awips/fxa/data/localizationDataSets/TOP/wwaTables.sup
/awips/fxa/data/localizationDataSets/TOP/wwa_warn_city.use
```

The first line is the file which describes the GELT being created. Following lines give the files that the GELT script will try to read.

The sup files define the geographic characteristics of the GELT. There are many sup files in /awips/fxa/data/localizationDataSets/LLL which describe various D2D geography items. The sup files are ASCII files but are not directly understandable. To see what parameters were used to create a sup file, that is, to see the useful geographic sup parameters, the mapsuparg program can be used. For example, to see the latitude, longitude boundaries of the wwa_warn_city GELT, we can analyze the wwaTables.sup file (from the “newGELTmaker -v” example above). To do this, as “fxa” in /awips/fxa/data/localizationDataSets/LLL, use:

```
/awips/fxa/bin/maksuparg wwaTables.sup
```

The output is:

```
maksuparg 1 39.022083 -96.486832 0.000000
1 36.875301 -99.089996 40.680466 -92.110909
1 40.729675 -99.235191 36.830593 -92.341698
x -0.018178 0.028962 -0.018482 0.015182
o 38.830315 -95.693665 300.276798 300.276798 214.438031 214.438031
o 39.022083 -96.486832 231.598222 368.991394 235.473003 193.428781
```

The second and third lines above (beginning with “1”) show the four corners of the GELT area.

In this case they are:

upper left (40.73 N, 99.24 W)
upper right (40.68 N, 92.11 W)
lower left (36.88 N, 99.09 W)
lower right (36.83 N, 92.34 W)

More details on mapsuparg are in
ds1:/awips/fxa/data/localization/documentation/maksuparg.html.

4.2.6 TEXT PRODUCT TEMPLATES

The “-wwa” task (script makeWWAtables.csh) also takes the product template files and creates the product definition files used by WarnGen to issue products. Templates are a complex topic and geography is only a part of the template story, so the template section is included in the general WarnGen troubleshooting section beginning on page [45](#).

4.3 THE “-station” LOCALIZATION TASK AND WARNGEN

The “-station” task uses script makeStationFiles.csh to construct station and location lists used by D2D to plot meteorological stations and geographic location names. The lists contain D2D progressive disclosure information.

Inputs to makeStationFiles.csh are goodness files, GELTs, and the cities files (CitiesInfo.txt or localCitiesInfo.txt). GELTs are discussed on page [26](#) and the cities files are discussed beginning on page [43](#). We discuss goodness files in this section. Outputs produced by makeStationFiles.csh are station plot files and location plot files.

4.3.1 GOODNESS FILES

Goodness files are lists of meteorological stations (e.g., METAR, RAOB stations) or locations (e.g. city names) that contain latitude/longitude coordinates, the station/location name and an arbitrary “goodness” value which controls progressive disclosure (i.e., what things are displayed at various D2D map scales). In practice, it seems that goodness files are mainly used in plotting station items. I don’t see evidence of goodness files being used to plot geographic location names such as cities. City names on D2D have progressive disclosure values, but the source of these are the goodness values in the CitiesInfo.txt file. For more details on plotting cities on D2D, see the section on local city names beginning on page [43](#).

Goodness files end with the file extension .goodness. Most goodness files are in directory ds1:/data/fxa/nationalData except the METAR goodness file which is in /awips/fxa/data/MTR.goodness. Information about the stations and locations are also saved in /awips/fxa/data in files ending with *Info.txt.

Many of the goodness files are maintained at NWS headquarters and are downloaded from the NOAA1 server. These are:

MTR.goodness
88D.goodness
raob.goodness
profiler.goodness

There is a BOUY.goodness file that is not used anymore. There are also files named airports.goodness, twebRoutes.goodness and twebStations.goodness. There is a spotters.goodness file, used by some sites for displaying SkyWarn items. The AWIPS National Datasets Maintenance web page (<http://www.nws.noaa.gov/ndm/>) has more information about goodness files. Here is a sample of contents from file MTR.goodness:

```
72403 KIAD 38.9600 -77.4500 98 26586
72405 KDCA 38.8600 -77.0300 20 27258
72417 KEKN 38.8900 -79.8500 609 27940
0 KJYO 39.0800 -77.5700 119 2975
```

The fields are the WMO block/station number (if one exists), station ID, latitude, longitude, elevation in meters, and an integer goodness value. The goodness value is an arbitrary integer that says how desirable it is to plot this station when it will overlap with another station. The larger the number, the more likely it will be displayed. If the goodness number is floating point, it is a progressive disclosure value which is the distance in kilometers to the nearest station that is at least as visible. The AWIPS System Manager's Manual provided these estimates of integer goodness values for each D2D map scale:

<u>Scale</u>	<u>Value</u>
NH	18100
NA	9000
CONUS	5900
REG	2000
STATE	900
WFO	460

A value of 2000 or so will make the item appear at the regional, state and WFO scales.

Script makeStationFiles.csh creates two kinds of station lists: station plot info files (these end with .spi) and location plot info files (these end with .lpi).

Script makeStationFiles.csh uses program va_driver to take goodness files and create the .spi and .lpi files (convert goodness factors into progressive disclosure parameters). See the FSL documentation in directory ds1:/awips/fxa/data/localization/documentation, files

localization.html (in the makeStationFiles.csh section), files masterToGoodness.doc.html and va_driver.doc.html.

Two related files in ds1:/awips/fxa/data are anchors.txt and selsAnchors.txt. These are used for labeling cross section baselines and decoding SAW products respectively.

4.3.2 STATION PLOT INFORMATION FILES

The station plot files control progressive disclosure for plotting meteorological station data on D2D such as MTR observations, buoy observations, RAOBs, etc. The .spi files are in /awips/fxa/data/localizationDataSets/LLL on the workstations and on the DS. Here are some entries from the MTR.spi file:

```
72403 KIAD 38.9600 -77.4500 98 37.997
72405 KDCA 38.8600 -77.0300 20 342.304
72417 KEKN 38.8900 -79.8500 609 173.754
0 KJYO 39.0800 -77.5700 119 16.895
```

The fields are the WMO block/station number (if one exists), station ID, latitude, longitude, elevation in meters, and a goodness value. The goodness value can be integer or floating point. If an integer, it indicates how desirable it is to plot this station when it will overlap with another station. The larger the number, the more likely it will plot. If the goodness number is floating point, it is a progressive disclosure value which is the distance in kilometers to the nearest station that is at least as visible. In the example above, station KDCA will plot at a larger scale than station KJYO.

4.3.3 LOCATION PLOT INFORMATION FILES

The location plot files are in /awips/fxa/data/localizationDataSets/LLL. Examples of entries in file cities.lpi:

```
39.3724 -77.4356 11.386 BALLENGER CREEK|MD
39.3008 -76.6106 126.097 BALTIMORE|MD
39.3300 -76.4200 1.662 BALTIMORE MARTIN ST|MD
```

These are the cities that are displayed for the D2D “Maps” menu, “Cities” button. Also under the “Maps” menu is a “Warning Locations” button, which displays items from file warn_cities.lpi. Here are the same cities from warn_cities.lpi:

```
39.3720 -77.4360 17.101 BALLENGER CREEK
39.3010 -76.6110 18.466 BALTIMORE
39.3300 -76.4200 1.686 BALTIMORE MARTIN ST
```

The cities listed in cities.lpi cover a larger geographic area than those in warn_cities.lpi. Just for fun, here are the same cities from wwa_warn_city.id (from the wwa_warn_city GELT):

```
158 39.372 -77.436 p 157 BALLENGER CREEK|2
159 39.301 -76.611 p 158 BALTIMORE|1
160 39.330 -76.420 p 159 BALTIMORE MARTIN ST|3
```

In this section, we focus on how D2D displays cities using the .lpi files. In the WarnGen troubleshooting section on page 43, we will discuss how WarnGen handles cities using GELTs and also the cities files.

4.3.4 D2D MENU, DATA KEYS AND DEPICT KEYS

This is a digression, but it's useful to know how D2D determines what file to look at when you push a given D2D menu button. The file that defines the D2D main pull down menus is /data/fxa/nationalData/dataMenus.txt. Various sub menus are defined in the files *Menu.txt. When a menu needs to display something (a depictable), file productButtonInfo.txt tells D2D how to do this. productButtonInfo.txt specifies a depict key which matches a product file with the desired D2D map scale. For each D2D map scale, only certain products can be displayed. The file /data/fxa/nationalData/scaleInfo.txt defines the various D2D map scales.

Manually maintained depict keys are defined in file depictInfo.manual. depictInfo.manual assigns a data key for each depict key. The data key is used to specify which file has the data that will be displayed. File dataInfo.manual assigns a source data file for each data key.

Here is an example. File /data/fxa/nationalData/backgroundMenus.txt defines the "Maps" drop down menu in D2D. In backgroundMenus.txt, the "Cities" menu item uses product button key 1016. The entry in backgroundMenus.txt looks like this:

```
productButton: 1016 # cities
```

To find more information about the product button key 1016 use "grep Cities depictInfo.manual" which provides this output:

```
1016 |5 |1016| | |1 |Cities |Cities |1 |1 |1 |
```

The fields are "|" delimited. The first field is the depict key (1016), the third field is the data key (coincidentally also 1016). The seventh field is the label D2D uses when the product is displayed, in this case "Cities." To find the file containing the stuff to plot for data key 1016, enter command "grep 1016 dataInfo.manual":

```
1016 | | | | | |cities |.lpi | cities
```

The first field is the data key (1016), and D2D plots the data from file cities.lpi.

Everything displayed on D2D has similar data and depict keys. Another WarnGen depictable item is the “Local Warnings” menu which displays text products. The files involved in mapping D2D menu items to text products are in /awips/fxa/data/localizationDataSets/LLL. File textDataKeys.txt assigns a depict key for the AFOS PIL. File textDepictKeys assigns the D2D menu item for the depict key. The localization task “-text” updates textDataKeys.txt and textDepictKeys.txt.

For example, the WarnGen warning boxes can be displayed in D2D’s “Local Warnings” menu. To be able to display these, the files textDataKeys.txt and textDepictKeys.txt must be correct. File textDataKeys.txt assigns a depict key for the CCCNNNXXX, e.g., key 6250 is assigned to LAXSVRSGX:

```
6250 |||1 ||| LAXSVRSGX |||LAXSVRSGX
```

File textDepictKeys assigns the D2D menu item for the depict key. In this case, depict keys in addition to 6250 are assigned to the Local Warnings item:

```
6120 |43|6250,6251,6252,6253,6254,6255,6256,6257,6258,6259,6300,6301,6302,6303,6304,6305,6306,6307,6308,6309,6350,6351,6352,6353,6354| |0 |1 |Local Warnings |Local Warnings|1 |0 |1 |
```

4.3.5 D2D CITIES VS. WARNGEN CITIES

Different mechanisms are used to plot cities on D2D and by WarnGen to put cities into text products. The .lpi files are used to plot city names on D2D and the GELT tables are used within WarnGen to find cities that are in warning boxes (or near a weather feature) to place in the warning text. The .lpi files have D2D progressive disclosure information, while the GELT .id files have the WarnGen flag for major, intermediate, and minor cities which determines which city names are put in the warning text. If WarnGen sees a major city, it will include that first. If no major cities are found, then intermediate, then minor cities will be included.

Using our knowledge of depict keys and data keys, we can see how WarnGen gets city names. Examine the D2D “Maps” drop down menu. Under the “WarnGen Tables” option, the “Cities” button uses depict key 1480 which labels the display as “Warning Cities” (see file /data/fxa/nationalData/depictInfo.manual). This entry also uses data key 1480 which corresponds to data file wwa_warn_city.table (see file /data/fxa/nationalData/dataInfo.manual). The file wwa_warn_city.table is part of the wwa_warn_city GELT.

To see the “Warning Cities” items in the wwa_warn_city GELT, look on a workstation in file /awips/fxa/data/localizationDataSets/LLL/wwa_warn_city.id. Here are the corresponding entries in wwa_warn_city.id that we saw above in the .lpi files:

158 39.372 -77.436 p 157 BALLENGER CREEK|2
159 39.301 -76.611 p 158 BALTIMORE|1
160 39.330 -76.420 p 159 BALTIMORE MARTIN ST|3

Fields one and five are sequence numbers internal to the GELT and can be ignored. The number after the “|” is the WarnGen flag for major, intermediate and minor cities. These values can only be “1”, “2”, or “3”. The WarnGen flag may also have an urban area identifier appended which is used if cities are to be treated as areas rather than points. The urban area identifier can be “=”, “+” or “~”. To use the urban area identifier, the site must install the urban_bounds shape file set and the city needs to exist in the urban_bounds shape file. For more info, see ds1:/awips/fxa/data/localization/documentation/va_driver.doc.html.

4.3.6 LOCAL CITY NAMES

Localization treats local cities in both the “-wwa” task and the “-station” task, so I didn’t put this under one of the specific localization task items. Information on local cities is in the WarnGen section troubleshooting section on page [43](#).

5. WARNGEN TROUBLESHOOTING

This guide is written mainly from the software support perspective, so there are items of operational interest to forecasters that are not included. Also, this guide covers only topics that have been escalated to the AWIPS Site Support Team, so not all WarnGen problems are covered.

Here are some references to help become familiar with using WarnGen and operational aspects of WarnGen. Evan Bookbinder at site SGF has a nice WarnGen training guide and a set of sample WarnGen templates at <http://www.crh.noaa.gov/sgf/WarnGen> The AWIPS Users Guide, Chapter 5, has a helpful section on using WarnGen. A web site with some forecaster oriented tips is <http://www.nwstc.noaa.gov/AWIPS/WarnGen/WarnGen.htm> This web site has not been updated since 1999 so is somewhat out of date.

5.1 ITEMS DIRECTLY AFFECTING PRODUCT CREATION

This section covers topics directly related to creating or formatting WarnGen products such as service backup, the WarnGen menu, local city names, text templates, and WMO product headings.

5.1.1 WARNGEN FULL BACKUP, PARTIAL BACKUP, PRIMARY AND SECONDARY SERVICE BACKUP

Primary and secondary backup are WFO operational procedures to provide service backup for a neighboring WFO that is temporarily not operational. Each WFO has a primary backup site to provide service, and if the primary backup is not available, the secondary backup is used.

WarnGen full and partial backup are software configuration terms, not explicitly associated with primary and secondary service backup. In WarnGen full backup mode, warnings can be issued for the entire CWA of a neighboring WFO. In WarnGen partial backup mode, warnings can be issued only for part of the neighboring CWA, based on your local radar coverage.

In full backup mode, WarnGen is configured to run as if you were at another site (e.g. the D2D maps are centered on the CWA being backed up). A special localization needs to be run to do this. For example, if site PHI wanted to perform WarnGen full service backup for site LWX, site PHI would run the localization on a workstation:

```
./mainScript.csh -WWA LWX PHI (WWA is upper case)
```

Now, for PHI to perform full service backup for LWX, D2D needs to be restarted with the LWX localization. The D2D maps at PHI will now be automatically centered on the LWX CWA.

If PHI needed all the workstation functionality to backup site LWX, they would run:

```
./mainScript.csh LWX PHI
```

This will perform the tasks topo, grids and radar in addition to all the tasks performed above in the “-WWA” localization. See the localization tasks section on page [10](#) for details.

When you localize using a backup CWA, a new /awips/fxa/data/localizationDataSets subdirectory is created for the backup site. These take up a lot of disk space, so on the HP workstations, not many localizationDataSets/LLL (where LLL is a backup CWA) directories can be created. To allow more localizationDataSets/LLL directories to be created, some sites have physically put the directory on another host (e.g. ds1) and created a link from the workstation to the directory. Using the linked localizationDataSets directory may affect the local AWIPS network performance.

When D2D first starts, it asks which localization will be used. If there are localizationDataSets/LLL directories in addition to the one for your local site, you can select the other localizations from a D2D drop down menu. D2D gets the list of possible localizations from the list of localizationDataSets/LLL directories that exist on the workstation.

In WarnGen partial backup mode, you start D2D as your local site and the D2D maps are centered on your CWA, but you are allowed to issue products for some of the zones in the neighboring CWA based on your local radar coverage. The full and partial backup is mainly a WarnGen software configuration concept. When a site sends out a product in full or partial backup mode, the backup site ID information is included in the product heading, just as if the backup site itself had issued the product. That is, if PHL was backing up LWX, the LWX headings would be used.

If a site needs to set up WarnGen partial backup, they should download from NOAA1 the newest set of usa_cwa_total shape files and put them in ds1:/data/fxa/nationalData. If they need the marine zones, also download the marine_total shape files. Then run localization using:

```
./mainScript.csh f +maps    (optional 'f')
```

If this were done at site PHI, now PHI would be able to run WarnGen partial backup for all their neighboring sites and issue products for which they have adequate radar coverage. The AWIPS shape files define the zones for which each site can do WarnGen partial backup.

At the top of the WarnGen window, in the “backup” section, there are drop down lists of the available “partial” and “full” backup sites available. The “full” list comes from the SBID directive in the file LLL-wwaConfig.txt. See the WarnGen localization directives section (page 51) for more on file LLL-wwaConfig.txt. The SBID directive contains a list of potential service backup localization CWAs. It looks something like:

```
@@@SBID “TBW”, “SJU”, “EYW”
```

This example is from site MFL, saying that MFL can provide service backup for sites TBW, SJU and EYW. To actually implement the backup, the “-WWA” or default localization needs to be run with the backup site specified. After localization, to see the list of full backup sites, look in /awips/fxa/data/localizationDataSets/LLL/wwa.config. The “warngen.cwaIds” line will have the list of full backup sites plus the local site ID.

It’s possible for a CWA (e.g. site LLL) to be included in the SBID list (that is, the WarnGen full backup drop down list includes site LLL) even if the associated localizationDataSets/LLL directory does not exist. It’s not possible to directly use one of the CWA’s in the full backup list, D2D must first be restarted. In this case, when D2D first comes up, you’d see that starting D2D as site LLL is not possible even though site LLL was in the WarnGen full backup drop down list.

Now for the WarnGen “partial” drop down CWA list. The sites in this list come from the usa_cwa_total shape file (which specifies partial backup counties). If the site has marine responsibility, the marine_total shape file (which specifies partial backup marine zones) is involved. The localization output file that contains the list of partial backup sites is /awips/fxa/data/localizationDataSets/LLL/backupCWAs.txt.

A totally different idea is the operational concept of primary and secondary service backup. For example, if site LWX cannot issue products (possibly because of a software upgrade or serious hardware problems), neighboring sites will cover for LWX, based on the local radar coverage at each site. Sites PHI, CTP, PBZ, RLX, RNK and AKQ could provide service backup for various sections of the LWX CWA. Also, each of the backup sites could provide backup for one or more of the LWX service areas, such as marine, aviation, hydrology, etc. Each of these sites could use WarnGen in either partial backup or full backup mode while performing primary service backup

for LWX. If a primary backup site can't provide help, then a secondary backup site would cover. Secondary backup is rarely needed.

The various primary and secondary backup assignments are in the station operations procedures and are based on the local radar coverage. The site backup responsibilities are defined by each region in their ROML or regional supplements to NWS operational directives. OS&T uses these documents to define the AWIPS shape files. As an example, the Eastern Region ROML for backup responsibilities is at: <http://www.werh.noaa.gov/msd/romls/backup/erbatrans.htm>

In January 2003, Southern Region implemented a new "sister-sister" service backup plan in which each office backs up the entire CWA for neighboring offices. In this arrangement, each office has one neighboring office provide primary backup for the entire CWA and another neighboring office provides secondary backup for the entire CWA. They no longer provide service backup for only selected counties of another CWA; it's "all or nothing." To implement this change, sites downloaded new copies of the usa_cwa_total shape file set (and marine_total, if marine responsibility was included). Then workstation localization was run using "mainScript.csh f +maps". The usa_cwa_total shape file defines the backup counties for which WarnGen can issue warnings. The marine_total shape file defines backup marine zones.

5.1.2 MARINE WARNINGS IN WARNGEN PARTIAL BACKUP MODE

AWIPS build 5.1.1 introduced a new WarnGen feature which allows sites to issue Special Marine Warnings (SMW) in partial backup mode. To implement this feature, a new shape file was created, the marine_total shape file set. When sites implemented this, we recommended that they also download a fresh copy of the marine_zones shape file set to be sure that the local CWA marine zones and the backup marine zones were in sync. Sites then localized their workstations using "./mainScript.csh f +maps".

This feature worked in 5.1.1. Unfortunately, it did not work in AWIPS 5.1.2 because of a bad localization list template file: /data/fxa/nationalData/wwa_marine_ugc.preTemplate. The following AREA substitution was added at the beginning of the wwa_marine_ugc.preTemplate file to fix the problem:

```
<AREA |file=backupMarine |area=WWA |include_field=1  
      |include_text=$$CURRENT_CWA! |format=none>
```

The marine warnings in WarnGen partial backup mode works in 5.2.1 since the correct wwa_marine_ugc.preTemplate file has been restored.

During the 5.2.X era, we discovered that sites which normally do not have marine responsibility could not issue marine warnings for their partial backup sites. This problem was diagnosed at sites CAE and ILN during August 2002. The problem was caused by a logic error in the localization script makeWWAtables.csh. The error has been fixed in the AWIPS baseline and is

OK as of release 5.2.2.2.

5.1.3 CUSTOMIZING THE WARNGEN PRODUCT TYPE MENU

On the WarnGen menu, by default, three products appear in the “Product Types” box: Flash Flood, Severe Thunderstorm and Tornado. The “Other” option contains a long drop down list of products derived more or less from the list of WarnGen product pre-template files in /data/fxa/nationalData (files beginning with “wwa” and ending with “preWWA”).

The configuration file which controls these lists of products is wwaConfig.template. The default national version of this file is /data/fxa/nationalData/wwaConfig.template. The localization output version of this file on the workstations (which WarnGen uses) is /awips/fxa/data/localizationDataSets/LLL/wwa.config. The OB2 System Manager’s Manual has some information in Appendix J.1.

In pre AWIPS 5.2.2 releases, the local customized version of this file was in directory /data/fxa/customFiles. During the AWIPS 5.2.2 through OB1 era, this override did not work, and the local version of wwaConfig.template needed to be in /awips/fxa/data/localization/LLL/LLL-wwaConfig.template.

In OB2, the /data/fxa/customFiles override works again. The wwa.config override information is in the OB2 FSL documentation in ds1:/awips/fxa/data/localization/documentation, files localization.html and fileChanges.html. We have not tested it, but according to the OB2 FSL documentation, localization (task wwa on the workstation) looks for wwaConfig.template, then wwa.config in the following order, using these overrides to create file localizationDataSets/LLL/wwa.config:

- /awips/fxa/data/wwaConfig.template - not used
- /data/fxa/nationalData/wwaConfig.template - default input for wwa.config
- /awips/fxa/data/localization/LLL/LLL-wwaConfig.template - functional override (that means that /data/fxa/nationalData/wwaConfig.template is not used)
- /data/fxa/customFiles/wwaConfig.template and LLL-wwaConfig.template - no overrides
- /data/fxa/customFiles/\$FXA_CUSTOM_VERSION/wwaConfig.template and LLL-wwaConfig.template - no overrides
- /awips/fxa/data/wwa.config - no override
- /data/fxa/nationalData/wwa.config - no override
- /awips/fxa/data/localization/LLL/LLL-wwa.config - replace what has been found so far in *wwaConfig.template
- /data/fxa/customFiles/wwa.config - replacement override
- /data/fxa/customFiles/LLL-wwa.config - replacement override
- /data/fxa/customFiles/\$FXA_CUSTOM_VERSION/wwa.config - replacement override
- /data/fxa/customFiles/\$FXA_CUSTOM_VERSION/LLL-wwa.config - replacement override

To change one workstation, you can still edit /awips/fxa/data/localization/LLL/LLL-wwaConfig.template, run “mainScript.csh -wwa” and restart D2D. You can also edit file /awips/fxa/data/localization/LLL/LLL-wwa.config. I believe that LLL-wwaConfig.template and LLL-wwa.config would contain exactly the same things.

To make the custom change on all workstations, it is now easier to use the override files in /data/fxa/customFiles. These are files wwa.config or LLL-wwa.config. There are no override files wwaConfig.template nor LLL-wwaConfig.template in /data/fxa/customFiles.

To customize your list of WarnGen products, edit the wwa.config override file. The number of products which display on the main WarnGen “Product Types” menu is controlled by the following line in the override file:

```
warngen.numMajorProds: 3
```

In this case, three products display on the main menu (the default) and any additional products are accessed using the “other” option. This number can be modified, and according to the FSL documentation, nine is the maximum number allowed here.

Another section of the wwa.config override file provides the text that will display on the WarnGen menu. The default version looks like this:

```
#
# WarnGen Product Types
#
warngen.prodTypeNames: \
  "Flash Flood",\
  "Severe Thunderstorm",\
  "Tornado",\
  "XXX000",\
  "XXX001",\
  "XXX002",\
```

This specifies the three default products followed by placeholders for the products in the “other” drop down list. You can put whatever text you wish between the quotes to define site specific lists of products for the WarnGen “Product Types” menu. An example follows:

```
#
# WarnGen Product Types
#
warngen.prodTypeNames: \
  "SVR - Single Storm/Line of Storms",\
  "SVR - Line of Storms with High Wind Potential/Bow Echo",\
```

```

"SVR - Dangerous Storms/Marginal Tornadoic Potential/Extreme Wind/Hail",\
"TOR - Mesocyclone with Tornadoic Potential",\
"TOR - Strong Mesocyclone/TVS Detected - No Tornado Reported",\
"TOR - Strong Mesocyclone/TVS Detected - Tornado Reported",\
"TOR - Tornado Reported - No Radar Detection",\
"FFW - Flash Flood Warning",\
"SMW - Special Marine Warning"

```

In the example above, the placeholders were removed, so only the products explicitly listed will appear on the WarnGen menus. Note that the last entry has the “,\” removed. In this example, if the entry “warngen.numMajorProds: 3” remained, then the three SVR products would display on the main WarnGen menu and the others would display on the “other” drop down list.

The next section of the wwa.config override file tells WarnGen which product template to use (in directory /awips/fxa/data/localizationDataSets/LLL, files beginning with “wwa” and ending with “wwaProd”). Here is the default version:

```

#
# WarnGen Product Type File Names.
# A file names position in this list must match the position of its
# corresponding menu name in the prodTypeNames list.
#
warngen.prodTypeFileNames: \
  "wwa_ffw.wwaProd",\
  "wwa_svr.wwaProd",\
  "wwa_tor.wwaProd",\
  "YYY000",\
  "YYY001",\
  "YYY002",\

```

Here the warngen.prodTypeFileNames have been changed to match the warngen.prodTypeNames section:

```

#
# WarnGen Product Type File Names.
# A file names position in this list must match the position of its
# corresponding menu name in the prodTypeNames list.
#
warngen.prodTypeFileNames: \
  "wwa_svr.wwaProd",\
  "wwa_svrbow.wwaProd",\
  "wwa_svrdanger.wwaProd",\
  "wwa_tor.wwaProd",\

```

```
"wwa_tor2.wwaProd",\  
"wwa_tor3.wwaProd",\  
"wwa_tor4.wwaProd",\  
"wwa_ffw.wwaProd",\  
"wwa_specmarine.wwaProd"
```

Each template file name must be in the same order as the associated items above in the warngen.prodTypeNames section. Also, note that the last file listed should have no “,\” after the file name.

After making the changes to the wwa.config override file, localization needs to be run to implement the changes. Run “mainScript.csh f -wwa”, then restart D2D to see the modified WarnGen menu.

5.1.4 LOCAL CITY NAMES

This section describes how WarnGen and D2D know about city names. Localization treats the local cities in both the “-wwa” task and the “-station” task.

The file /awips/fxa/data/CitiesInfo.txt on the workstation is the source of almost all city information used by D2D on map backgrounds (via location plot information files) and used by WarnGen to insert into warnings (via GELTs). CitiesInfo.txt is not updated by the AWIPS software upgrades. CitiesInfo.txt is maintained at NWS headquarters and is downloaded from the NOAA1 server. On NOAA1, this file is in directory /awips/ftp/pub/maps (for ftp users, just cd to /pub/maps), file name ciddmmyy.txt (where ddmmyy is the date of file creation). Don't confuse this with a new shape file set, ciddmmyy, which is used with GFE. By default, CitiesInfo.txt contains mainly a list of U.S. Census cities, but local sites can request changes to the file if they feel that the changes are of interest beyond their CWA. Some information on CitiesInfo.txt is in the OB2 System Manager's Manual, Section 15.9.

A section of the CitiesInfo.txt file looks like:

```
39.3724 -77.4356 1900 MD BALLENGER CREEK|2  
39.3008 -76.6106 3330 MD BALTIMORE|1  
39.3300 -76.4200 80 MD BALTIMORE MARTIN ST|3
```

Each line contains the latitude and longitude in decimal degrees, negative west longitude. The next field is the ‘goodness’ value and is used for progressive disclosure. Normally the goodness value is not used in GELTs, but only in the D2D .lpi files. See the section on the “-station” localization task (page [31](#)) for more on goodness and progressive disclosure.

The next fields are the state and location name. The location can be any meaningful landmark - highway intersections, parks, etc. and can have any format. This section must begin with the

correct two letter state abbreviation. WarnGen treats the state/location combination as a unique item, so each state can only have one city with the same name.

If the ending “|#” is not present, normally this location won’t be included in the GELT. That is, if the ending “|#” is present, then this location is available to WarnGen. The “#” is a WarnGen flag set to “1” for major cities, “2” for places of average importance, “3” for minor places. If WarnGen sees no “1” or “2” cities in the warning area, it will put “3” cities in the product. No other values are allowed in this field.

There can be an optional “urban area” identifier after the “|#” which indicates how to treat the city if it is too large in area to be considered a point. The urban area identifier can be “=” (city name is the centroid of the city), “+” (city centroid determined using area weighted averaging) or “~” (city centroid determined using a point furthest from any city border). The urban area identifier can only be used if the urban boundaries shape file is being used. The urban boundaries shape file has a huge number of cities and towns mapped with their boundaries.

There are various other configuration options dealing with formatting the output GELT city text and an override file dealing with centroids, duplicate city names, etc. which are beyond the scope of this discussion. See directory ds1:/awips/fxa/data/localization/documentation, files newGELTmaker.doc.html and va_driver.doc.html for details.

It is interesting to note that in the state of Virginia, independent cities are treated as counties in the uscounty shape file set. This is the only state where this happens. In the c11-zone shape file set (which defines NWS forecast zones), the Virginia independent cities are not part of the zone definitions.

Cities not in the national version of CitiesInfo.txt can be placed in file LocalCitiesInfo.txt, then run workstation localization using “mainScript.csh +wwa”. Items from LocalCitiesInfo are appended to CitiesInfo.txt. The resulting “-wwa” localization task will add the local cities to the WarnGen GELTs. The resulting “-station” task will add the cities to the location plot information files for displaying the cities on the D2D background. If a site wants some local cities to be available to their service backup sites, they can provide their LocalCitiesInfo.txt entries to their backup sites.

The OB2 documentation (ds1:/awips/fxa/data/localization/documentation/fileChanges.html) from FSL states how localization override works with CitiesInfo.txt and LocalCitiesInfo.txt, but we haven’t tested it. Localization (tasks “wwa” and “station” on the workstation) looks for CitiesInfo.txt, then LocalCitiesInfo.txt in the following order, using these overrides:

- /awips/fxa/data/CitiesInfo.txt - default input for cities files
- /data/fxa/nationalData/CitiesInfo.txt - no override
- /awips/fxa/data/localization/LLL/LLL-CitiesInfo.txt - functional override (that means that /awips/fxa/data/CitiesInfo.txt is not used)

/data/fxa/customFiles/CitiesInfo.txt and LLL-CitiesInfo.txt - no overrides
/data/fxa/customFiles/\$FXA_CUSTOM_VERSION/CitiesInfo.txt and LLL-CitiesInfo.txt
- no overrides
/awips/fxa/data/LocalCitiesInfo.txt - no override
/data/fxa/nationalData/LocalCitiesInfo.txt - no override
/awips/fxa/data/localization/LLL/LLL-LocalCitiesInfo.txt - appended to what has been
found so far in *CitiesInfo.txt
/data/fxa/customFiles/LocalCitiesInfo.txt - append override
/data/fxa/customFiles/LLL-LocalCitiesInfo.txt - append override
/data/fxa/customFiles/\$FXA_CUSTOM_VERSION/LocalCitiesInfo.txt - append
override
/data/fxa/customFiles/\$FXA_CUSTOM_VERSION/LLL-LocalCitiesInfo.txt - append
override

In summary, localization starts with /awips/fxa/data/CitiesInfo.txt on the workstation, then if file /awips/fxa/data/localization/LLL/LLL-CitiesInfo.txt exists, it is used instead. Next, localization looks for LocalCitiesInfo.txt files. If /awips/fxa/data/localization/LLL/LLL-LocalCitiesInfo.txt exists on the workstation, it will be appended to CitiesInfo.txt. Lastly, if ds1:/data/fxa/customFiles/LocalCitiesInfo.txt exists, it will be appended. If all workstations are to be configured the same, it's best to use /data/fxa/customFiles/LocalCitiesInfo.txt to add your local cities.

According to the OB2 FSL documentation, you can create .lpi files in /data/fxa/customFiles which will append extra items to display on D2D when you localize using “-station”. Files cities.lpi, warn_cities.lpi and marine_sites.lpi can be placed in customFiles. This will add items to the D2D display only, not make them available to WarnGen. We have never tried this.

5.1.5 TEXT PRODUCT TEMPLATES AND LOCALIZATION

Templates are boilerplate text files used by WarnGen, WWA and NWR to generate products. When a warning is created, WarnGen automatically substitutes text in the template, based on complex template syntax rules. Sites customize templates quite a bit to suit their local preferences, so it's hard to give a lot of detailed help on site specific template formats.

Localization task “-wwa” (script makeWWAtables.csh) does the template processing needed by WarnGen to create products. If all the localization items seem OK and the text in a product is messed up, there may be a template syntax problem. Generally sites know how to set up templates to their liking. Also there are various template postings on awipsinfo. We can offer some rudimentary template help. Most of the WarnGen problems escalated to the SST relate to various geographic topics, so this paper focuses on the use of geography in templates.

5.1.5.1 TEXT PRODUCT TEMPLATE SUMMARY

The text product boilerplates which are input to localization are called pre-templates. Localization uses these to create site specific product templates which WarnGen uses directly to substitute text and create a specific product. Another name which FSL uses for the templates created by localization are “product definition files.”

There are two types of templates. Product specific templates define the text for specific products. List templates describe how to make generic lists of geographic entities (lists of counties, cities, zones or UGC code headings) that can be used by different product templates.

An issue bearing on product templates are NWS operational directives (not to be confused with localization directives). The NWS directives spell out certain operational procedures. Directives define some of the formats for NWS text products (for example directives 10-1701, 10-1702). As a result of new directives, sometimes new national versions of product templates need to be distributed. Sites may be affected because they may need to add their customizations to the new templates. Information about NWS operational directives is at <http://www.nws.noaa.gov/directives/>

An FSL summary of the template formats and syntax is on <ds1:/awips/fxa/data/localization/documentation/TextTemplate.html>. Some helpful notes on WarnGen use and sample templates are at Evan Bookbinder’s page at <http://www.crh.noaa.gov/sgf/WarnGen> The OB2 System Manager’s Manual, beginning with Appendix J.2, has some information on template formats.

5.1.5.2 PRODUCT SPECIFIC TEMPLATES

The product pre-template file names end in .preWWA and are inputs to the localization “-wwa” task. The national default pre-templates are in /data/fxa/nationalData. In general, it is better for customized pre-templates to be in /data/fxa/customFiles. This way they will never be overwritten by an AWIPS software upgrade. The pre-templates in /data/fxa/customFiles are used by all workstations since the /data/fxa partition is NFS mounted. If a site only wants a specific workstation to use a custom template, the pre-template is put in /awips/fxa/data/localization/LLL. However, field site experience has shown that AWIPS software upgrades may overwrite files in /awips/fxa/data/localization/LLL.

Generally pre-template file names beginning with “WWA” are used by the WWA software, those with “nwr” are used by NOAA Weather Radio, and those with “wwa” are used by WarnGen. For example, file “/data/fxa/nationalData/wwa_svr.preWWA” is a default national WarnGen SVR pre-template file.

Site specific versions of the pre-templates are maintained in /data/fxa/customFiles. These can be prefixed with the site ID. For example, in /data/fxa/customFiles, files “wwa_svr.preWWA” and

“LLL-wwa_svr.preWWA” are site specific WarnGen SVR pre-templates. The prefix “LLL-“ means that this pre-template is used only for that particular site localization. That way, a site can do full service backup and use the customized pre-templates from the site they are backing up.

There are some pre-template files ending with .preWWAB. These are not used anymore. In the past, the .preWWA and .preWWAB had different bullet formats. The SRWARN directive in file /awips/fxa/data/localization/LLL/LLL-wwaConfig.txt apparently was involved in this transition to different product bullet formats.

The template files (or product definition files) are outputs from the “-wwa” localization task. WarnGen uses the template files directly to substitute text and create a specific product. Template files are in directory /awips/fxa/data/localizationDataSets/LLL and end with .wwaProd. They also begin with “WWA” “nwr” or “wwa” like the pre-template files: “WWA” templates are used by WWA, “nwr” are used by NOAA Weather Radio and “wwa” are used by WarnGen. For example, file /awips/fxa/data/localizationDataSets/LLL/wwa_svr.wwaProd is the SVR template which WarnGen uses to substitute text and create a specific SVR product. The pre-template files seem to be almost the same as the template files, except for a few substitutions that localization makes.

Also in /data/fxa/nationalData are *.abbrev files which WarnGen uses to translate various geography abbreviations into plain language in the templates.

Below are lists of national default WarnGen pre-template files from OB2 and their associated product name. The NNN is the shorthand name generally used to refer to the product. This is the NNN from the AWIPS product ID (see page 4). These files are located in /data/fxa/nationalData. The following short fused products are normally created by WarnGen:

FILE NAME	PRODUCT NAME	NNN
wwa_tor.preWWA	Tornado Warning	TOR
wwa_svr.preWWA	Severe Thunderstorm Warning	SVR
wwa_svrwx_sta.preWWA	Severe Weather Statement	SVS
wwa_svrwx_sta_county.preWWA	Severe Weather Statement (by county)	SVS
wwa_specmarine.preWWA	Special Marine Warning	SMW
wwa_mar_wx_sta.preWWA	Marine Weather Statement (follows an SMW)	MWS
wwa_mws_nosmw.preWWA	Marine Weather Statement (no SMW issued)	MWS
wwa_ffw.preWWA	Flash Flood Warning	FFW
wwa_ffw_svr.preWWA	FFW with Severe Thunderstorm	FFW
wwa_dam_break.preWWA	Dam Break (non convective FFW)	FFW
wwa_fflood_sta.preWWA	Flash Flood Statement	FFS

wwa_fflood_sta_county.preWWA	Flash Flood Statement (by county)	FFS
wwa_urbssflood_adv.preWWA	Urban/Small Stream Flood Advisory	FFS

These long fused warning and other products have the following national default WarnGen pre-templates but they are normally created by WWA:

<u>FILE NAME</u>	<u>PRODUCT NAME</u>	<u>NNN</u>
wwa_blizzard_wrn.preWWA	Blizzard Warning	WSW
wwa_bloodust_adv.preWWA	Blowing Dust Advisory	NPW
wwa_blosnow_adv.preWWA	Blowing Snow Advisory	WSW
wwa_cst_fld_sta.preWWA	Coastal Flood Statement	CFW
wwa_cst_fld_wat.preWWA	Coastal Flood Watch	CFW
wwa_cst_fld_wrn.preWWA	Coastal Flood Warning	CFW
wwa_fflood_wat.preWWA	Flash Flood Watch	FFA
wwa_flood_wat.preWWA	Flood Watch	FFA
wwa_flood_wrn.preWWA	Flood Warning	FLW
wwa_fog_adv.preWWA	Dense Fog Advisory	NPW
wwa_freez_wrn.preWWA	Freeze Warning	NPW
wwa_frost_adv.preWWA	Frost Advisory	NPW
wwa_frzdrzl_adv.preWWA	Freezing Drizzle Advisory	WSW
wwa_frzrain_adv.preWWA	Freezing Rain Advisory	WSW
wwa_heat_wrn.preWWA	Excessive Heat Warning	NPW
wwa_hiwind_wat.preWWA	High Wind Watch	NPW
wwa_hiwind_wrn.preWWA	High Wind Warning	NPW
wwa_hls.preWWA	Hurricane Local Statement	HLS
wwa_hvysnow_wrn.preWWA	Heavy Snow Warning	WSW
wwa_hvysurf_adv.preWWA	Heavy/High Surf Advisory	CFW
wwa_lakesh_fld_sta.preWWA	Lakeshore Flood Statement	LSH
wwa_lakesh_fld_wat.preWWA	Lakeshore Flood Watch	LSH
wwa_lakesh_fld_wrn.preWWA	Lakeshore Flood Warning	LSH
wwa_shortterm_fcst.preWWA	Short Term Forecast	NOW
wwa_snoblsno_adv.preWWA	Snow/Blowing Snow Advisory	WSW
wwa_snow_adv.preWWA	Snow Advisory	WSW
wwa_svrhiwind_wrn.preWWA	Severe High Wind Warning	NPW
wwa_sws.preWWA	Special Weather Statement (by zones)	SPS
wwa_sws_county.preWWA	Special Weather Statement (by county)	SPS
wwa_wintstrm_wat.preWWA	Winter Storm Watch	WSW
wwa_wintstrm_wrn.preWWA	Winter Storm Warning	WSW
wwa_wndchil_adv.preWWA	Wind Chill Advisory	NPW

Sites often create their own pre-templates in /data/fxa/customFiles. As a result, you are likely to see other wwa*preWWA files not listed here. Locally named pre-templates must be added to the WarnGen product list to be used (see page 40).

NWS operational directive 10-511 (dated 10/1/03) specifies whether WarnGen or WWA is to be used to create certain severe weather products. For more details, the directive is available at <http://www.nws.noaa.gov/directives> For other products, it is not cast in concrete whether WarnGen or WWA is used to create them. Also some of the hydrology related products (such as Dam Breaks, Flood Warnings) may be created by software from the Office of Hydrological Development. That is why there are many default WarnGen templates, but few of them are normally used at an individual site. If there is a doubt about which software is used to create a particular product, you should contact the site that issued the product.

To summarize the templates for one particular product, here is a list of possible product template files that could be associated with a WarnGen tornado warning, listed in the order that localization processes them:

- /data/fxa/nationalData/wwa_tor.preWWA (default pre-template file)
- /awips/fxa/data/localization/LLL/LLL-wwa_tor.preWWA (custom pre-template for site LLL at this workstation)
- /data/fxa/customFiles/wwa_tor.preWWA (custom pre-template for local site at all workstations)
- /data/fxa/customFiles/LLL-wwa_tor.preWWA (custom pre-template for site LLL at all workstations)
- /awips/fxa/data/localizationDataSets/LLL/wwa_tor.wwaProd (localization output template for site LLL at this workstation - this is used directly by WarnGen to substitute text and create the warning)

The WarnGen warning products are supposed to have a follow up. That is, if a warning is issued, another product is issued soon after, providing an update about this same weather event. The follow up could be an updated storm location or intensity, or a warning expiration or cancellation. Here is a list of WarnGen products and their normal follow up products:

- TOR is followed up with SVS
- SVR is followed up with SVS
- FFW is followed up with FFS (Western Region follows up FFW with FLS)
- SMW is followed up with MWS

In /data/fxa/customFiles, sites can create their own special products which may not be in the default list above. If you're not sure what custom products a site has created, you can get a handy list of their custom templates. To see a summary of the pre-templates, use the following "grep" command (in data/fxa/customFiles):

```
grep "/" *wwa*preWWA*
```

Templates have a comment line at the top with the product name. The comment line in templates begins with “/”, so you’ll get the product name, plus any other comments (normally there are very few other comments in templates).

For example, here are some customized Severe Weather Statements (SVS) pre-template files in /data/fxa/customFiles that came from the above grep command:

```
wwa_svs_svrcont.preWWA:/"Severe Weather Statement - SVR Continues"  
wwa_svs_svexpired.preWWA:/"Severe Weather Statement - SVR exp/cncl"  
wwa_svs_torcont.preWWA:/"Severe Weather Statement - TOR Continues"  
wwa_svs_toreexpired.preWWA:/"Severe Weather Statement - TOR exp/cncl"
```

To see what WarnGen templates are actually used (that is, to see the templates that are invoked from the Product List on the WarnGen GUI), you can look in the WarnGen product type configuration file: /awips/fxa/data/localizationDataSets/LLL/wwa.config. This file is described beginning on page 40.

5.1.5.3 LIST TEMPLATES

List template localization input files are named /data/fxa/nationalData/*.preTemplate. The preTemplate files specify how to make generic blocks of text such as lists of counties, cities, zones or UGC code headers. For example, the file nationalData/wwa_county_ugc.preTemplate is processed by localization to produce /awips/fxa/data/localizationDataSet/LLL/wwa_county_ugc.template. The wwa_county_ugc.template is #included into whatever product templates in /awips/fxa/data/localizationDataSets/LLL that need lists of county UGC codes. Here are the list templates from AWIPS release OB2:

sls.preTemplate	wwa_county_block.preTemplate
sls_county_all.preTemplate	wwa_county_ugc.preTemplate
sls_county_block.preTemplate	wwa_county_zone_ugc.preTemplate
sls_state_header.preTemplate	wwa_marine_block.preTemplate
wcn_city.preTemplate	wwa_marine_ugc.preTemplate
wcn_county_all.preTemplate	wwa_wsfo_cnty_ugc.preTemplate
wcn_county_block.preTemplate	wwa_wsfo_zone_ugc.preTemplate
wcn_state_header.preTemplate	wwa_zones_all.preTemplate
wwa_city_list.preTemplate	wwa_zones_block.preTemplate
wwa_county_all.preTemplate	wwa_zones_ugc.preTemplate

Hacked up versions of .preTemplate and .template can cause WarnGen problems.

5.1.5.4 WARNGEN LOCALIZATION DIRECTIVES

Directives are introduced in the section on localization directives (page 17). Most WarnGen directives result in direct text substitution in the WarnGen product definition files (templates). A small number of directives have a functional impact on how localization works.

The file `ds1:/data/fxa/nationalData/wwaDefaults.txt` contains centrally provided national default values for some directives used in controlling WarnGen's behavior. Here are examples:

```
@@@CIFA |min_fraction=0.030 |min_area=25
@@@ZOFA |min_fraction=0.050 |min_area=50
@@@COFA |min_fraction=0.030 |min_area=25
```

The directive tags CIFA, ZOFA and COFA control how sensitive WarnGen is in selecting cities, zones and counties (using GELTs) to place in a text product when part of the city, zone or county falls within the warning box.

There are other WarnGen site specific directives on the workstations in directory `/awips/fxa/data/localization/LLL`. These can be customized by each site. File `LLL-wwaConfig.txt` contain more directives that pertain to WarnGen. An example follows:

```
@@@SRWARN FALSE
@@@ORABOUT TRUE
@@@SBID "TBW", "SJU", "EYW"
@@@OFFH MIAMI FL
@@@OFFT IN MIAMI
@@@COPE |portions |extreme |central
@@@CIPE |portions |extreme |central
@@@PTPE |portions |extreme |central
```

The directives in `LLL-wwaConfig.txt` generally apply to WarnGen text substitution in product templates. If a site wants to set their own values for WarnGen directives, they should do this in `ds1:/data/fxa/customFiles/LLL-wwaConfig.txt`.

Here is a list of the directives that relate to various WarnGen geography items:

- OFFH - name of the office to use in product header (e.g. "SPRINGFIELD MO")
- OFFT - name of the office to use with the text of products (e.g. "IN SPRINGFIELD")
- AREAS - FSL definition: "a term used to generically describe what a list of geographic entities is."
- COPE - how to describe portions of counties (values are "portions", "extreme", "central")
- ZOPE - how to describe portions of zones (values are "portions", "extreme", "central")
- CIPE - how to describe cities listed as locations within the warned area (values are

- “portions”, “extreme”, “central”)
- PTPE - how to describe cities used to describe the location of a storm (values are “portions”, “extreme”, “central”)
- COFA - set thresholds for excluding small fragments of counties in the warning box (qualifier values are “min_area=aaa”, “min_fraction=fff” or “test_both”)
- ZOFA - set thresholds for excluding small fragments of zones in the warning box (qualifier values are “min_area=aaa”, “min_fraction=fff” or “test_both”)
- CIFA - set thresholds for excluding small fragments of cities in the warning box (qualifier values are “min_area=aaa”, “min_fraction=fff” or “test_both”)

For COFA, ZOFA and CIFA, the “aaa” used with “min_area” qualifier is an area in square km. For example, when you draw the WarnGen warning box, any county that has less than aaa square km within the box will be excluded from the warning text. The “fff” used with the “min_fraction” qualifier is a decimal fraction. When you draw the WarnGen warning box, any county with less than “fff” of its area within the box will be excluded from the warning. By default, the county, zone or city can pass only one test (“min_area” or “min_fraction”) to be included in the warning. When “test_both” is used, the county, zone or city must pass both the “min_area” and “min_fraction” tests to be included in the warning.

To use directives in site specific control files or customization files, the line with the directive must begin with the literal “@@@”, for example:

```
@@@COFA |min_fraction=0.030 |min_area=25| test_both
```

This specifies directive tag “COFA” with values for the qualifiers “min_fraction”, “min_area” and using the qualifier “test_both.”

Directives can also be used in text product template files to produce direct text substitution in WarnGen products. If there is a conflict between a directive that is in LLL-wwaConfig.txt and also in a template, I imagine that the normal localization override applies (e.g. a directive value in /data/fxa/nationalData will be overridden by a directive value in /data/fxa/customFiles).

To use directives in pre-template files, place the literal “@@@” with the directive tag name where the text substitution is desired. For example, the following uses the OFFT tag to substitute the office name in the text product:

```
THE NATIONAL WEATHER SERVICE @@@OFFT HAS ISSUED A...
```

Directive qualifiers can be used directly in the pre-template files to help control text substitution. For example, the “min_area” and “min_fraction” qualifiers can be used in the pre-template to define the areal or fractional thresholds for including counties in the warning.

In practice, COFA, ZOFA, CIFA don’t seem to be used much in product templates. They can be

used in file LLL-wwaConfig.txt to control WarnGen behavior for all product templates. COFA applies to county based products and ZOFA applies to zone based products. CIFA only applies if you treat your cities as areas instead of points; in this case you must use the urban bounds shape file to define areas for your cities.

For more details, see the FSL documentation in directory ds1:/awips/fxa/data/localization/documentation. File directives.html discusses directives and file TextTemplates.html discusses the many, many qualifiers that can be used in templates.

5.1.5.5 TEMPLATE SUBSTITUTIONS

An important concept in templates is text substitutions. Details are in ds1:/awips/fxa/data/localization/documentation/TextTemplates.html. In templates, the format of text substitutions is:

```
“<substitution_type | qualifier_type=qualifier_value | ... >”
```

Two important substitution types of interest here are “AREA” and “WX.” The AREA substitution causes text to be generated which describes the area of the warning. The WX substitution causes text to be generated which describes the location of the weather event. Both of these substitutions make use of a GELT.

These substitutions have many, many qualifiers that can be used to specify how the substitution is done. Some of the directive qualifiers also seem to serve as qualifiers used in substitutions. Here is an example of an “AREA” substitution used in a text template:

```
<AREA |file=wwa_warn_city |area=wwa_counties  
|exclude_field=2 |exclude_text=3 @@@CIFA  
|file=wwa_warn_city |area=wwa_counties  
|exclude_field=0 |var=dummyDummy>
```

We will explain some of these items in the next section.

5.1.5.6 SAMPLE WARGEN TEMPLATE SYNTAX

This section has a quick summary of how WarnGen uses a product template to generate a warning. The SST doesn't work much with template syntax, so only a cursory summary is given here. We'll look through an OB2 tornado warning template and see how WarnGen uses the template to fill in various kinds of text to produce the tornado warning. The template fragments show here are just representative samples. At any site, the template likely is somewhat different. The items below in bold are the actual text of a sample tornado warning.

The following is a sample tornado warning that we will examine:

**WFUS52 KTBW 061110
TORTBW
FLC101-061130-**

**BULLETIN - EAS ACTIVATION REQUESTED
TORNADO WARNING
NATIONAL WEATHER SERVICE TAMPA BAY AREA - RUSKIN FL
710 AM EDT SAT SEP 6 2003**

THE NATIONAL WEATHER SERVICE IN RUSKIN HAS ISSUED A

*** TORNADO WARNING FOR...
EASTERN PASCO COUNTY IN FLORIDA.**

*** UNTIL 745 AM EDT**

*** AT 710 AM EDT...NATIONAL WEATHER SERVICE DOPPLER RADAR
INDICATED A TORNADO 6 MILES WEST OF ZEPHYRHILLS...MOVING
EAST AT 15 MPH.**

*** THE TORNADO WILL AFFECT...
WESLEY CHAPEL BY 705 AM EDT.
DADE CITY BY 715 AM EDT.
ZEPHYRHILLS BY 720 AM EDT.**

**THIS IS A DANGEROUS STORM! MOVE INTO THE INTERIOR ROOM ON
THE LOWEST FLOOR OF A STURDY BUILDING...AWAY FROM WINDOWS.
COVER YOUR HEAD AND BODY WITH PILLOWS OR BLANKETS.**

LAT...LON 2836 8236 2817 8236 2819 8212 2839 8210

\$\$

Now we'll examine each section of the template to show how the warning text is created. At the top of the template is a comment line (beginning with "//") which gives the name of the product. Also there are definitions of warning time durations and a depict key for displaying the warning on D2D. Here is a the associated template text:

```
//"Tornado"  
  
<DURATIONS | 15 | 30=default | 45 | 1:00>  
<DEPICT_KEYS|1083>
```

The first line of the product has the WMO heading followed by the AWIPS product ID:

**WFUS52 KTBW 061110
TORTBW**

The following template text creates the WMO heading. The file /awips/fxa/data/localizationDataSets/LLL-offtIncl.txt is “#included” to make use of the variables cccValue and xxxValue.

```
#include "$ {CURRENT_CWA}-offtIncl.txt"  
ZCZC $$cccValue!TOR$$xxxValue! ALL&  
TTAA00 KTBW <NOW | ddhhmm | gmt>&&
```

Next, in non segmented products, comes the UGC line. This has the list of all counties (or zones) included in the product plus the day and time of product expiration:

FLC101-061130-

The county list of UGC codes is created by including the wwa_county_ugc list template as shown below from the template:

```
#include "wwa_county_ugc.template"
```

The county names come from the wwa_counties GELT. There is a UGC troubleshooting section on page 64. If this were a segmented product, the UGC would not be here, but later after the Mass News Disseminator. Also, segmented products could have multiple UGC lines, each indicated in the template by the “#include” above.

The VTEC (when implemented) will come next in the warning:

/NEW.KTBW.TO.W.0043.0309061100Z-0309061130Z/

The VTEC format is described beginning on page 6. The following line in the tornado template specifies the VTEC:

```
&/NEW.$$wmoValue!.TO.W.$$ETN_VAL!.<START|ymdthmz|gmt>-<EXPIRE|ymdthmz|gmt>/
```

The wmoValue is used to get the WMO site ID (here KTBW). There is a VTEC troubleshooting section on page 65.

Next, the templates have an “AUX_INFO” substitution. The “AUX_INFO” is a special substitution that generates no text and looks something like:

```
<AUX_INFO |geo_descriptor=2 |wwa_type=2
|wx_hazard=Tornado
|issue_prod=$$cccValue!TOR$$xxxValue! >
```

The FSL documentation says that in AUX_INFO “each qualifier is a key, and each value is some text that can be passed back to the client based on that key.” AUX_INFO allows the WarnGen created products to be tracked in the WWA software.

The next item in the warning is the Mass News Disseminator (MND):

```
BULLETIN - EAS ACTIVATION REQUESTED
TORNADO WARNING
NATIONAL WEATHER SERVICE TAMPA BAY AREA - RUSKIN FL
710 AM EDT SAT SEP 6 2003
```

The following template text produces the MND:

```
BULLETIN - EAS ACTIVATION REQUESTED&
TORNADO WARNING&
#include "${CURRENT_CWA}-headerIncl.txt"
<NOW | header | local >
```

In the MND, the issuing NWS office name is obtained by including the file /awips/fxa/data/localizationDataSets/LLL/LLL-headerIncl.txt. It’s also possible to use the OFFH directive (from file /awips/fxa/data/localization/LLL/LLL-wwaConfig.txt). To use the OFFH directive, use “@@@OFFH” in the template.

Next comes the warning headline:

```
THE NATIONAL WEATHER SERVICE IN RUSKIN HAS ISSUED A
```

The associated template text has:

```
THE NATIONAL WEATHER SERVICE $$offtText! HAS ISSUED A
```

In the template, the warning headline is hardcoded except that variable offtText is used (defined in /awips/fxa/data/localizationDataSets/LLL/LLL-headerIncl.txt). The directive “@@@OFFT” could also be used, and is defined in /awips/fxa/data/localization/LLL/LLL-wwaConfig.txt.

Next in the template is an AREA substitution that looks something like:

```
// establish set of cities we will use (use 3's?)
<AREA |file=wwa_warn_city |area=wwa_counties
|exclude_field=2 |exclude_text=3 |min_fraction=0.030 |min_area=25
|file=wwa_warn_city |area=wwa_counties
|exclude_field=0 |var=dummyDummy>
```

This seems to generate a “global” list of cities for the warning. Other places in the template that generate lists of cities use cities from this global list. The text “file=wwa_warn_city” means to use the wwa_warn_city GELT. The text “area=wwa_counties” means to restrict the cities listed to the area covered by the wwa_counties GELT.

The following is a line from the wwa_warn_city.id file to help illustrate the “exclude” items:

```
3 36.476 -93.773 p 3 BEAVER|3
```

The “exclude_field” and “exclude_text” items say to exclude WarnGen priority 3 cities from the product. In the file wwa_warn_city.id file, the fields are the items at the end of each line (the field delimiter is “|”). The template example above says to exclude items in field 2 (“exclude_field=2”) whose value is 3 (“exclude_text=3”). In this case, BEAVER would not show up in the warning, since it is a priority “3” location. The “min_fraction” and “min_area” qualifiers are described in the WarnGen localization directives section beginning on page [51](#).

Many items in the following sections in the warning can be selected or excluded by using the WarnGen “optional bullets” window. This example shows only some of the possible options.

Next in the warning come four mandatory “bullets.” Each bullet begins with “*”. The first mandatory bullet contains the warning name and locations included in the warning:

```
* TORNADO WARNING FOR...
EASTERN PASCO COUNTY IN FLORIDA.
```

An AREA substitution is used in the template to produce the above list. The list could contain several counties plus optional lists of cities in the warning area. The following is the template section that produced the above warning text:

```
* TORNADO WARNING FOR...&
<AREA |file=wwa_counties |format=simple |group_by=[table]
|used=begin |unique_by=[1] [51] [52] [0,table]
|item_format=
~~[county_type]~IN FLORIDA.&[599]
|portions
|accumulate |cross=wwa_counties |sort_by=[52] [51] [50,table]>
```

The “file” qualifier tells which GELT to use. In the list of counties, we use the `wwa_counties` GELT. The qualifiers “portions”, “extreme”, “central” can be used to specify which sections of the county are included. If cities were also desired in this bullet, “file=`wwa_warn_city`” would also be used to access the `wwa_warn_city` GELT. Most of the other items are beyond the scope of this discussion.

There have been various questions on controlling how WarnGen takes the warning box from D2D and determines which geographic items to include in the warning. The COFA, ZOFA and CIFA directives control how WarnGen does this for counties, forecast zones and cities.

There are several different ways that the COFA, ZOFA and CIFA directives and their associated qualifiers (“`min_area`”, “`min_fraction`”, “`test_both`”) can be specified to localization to effect the desired WarnGen behavior. I have not determined the subtleties of the different ways that these directives and qualifiers can be used. The COFA, ZOFA and CIFA directives seem to be used mainly in the `LLL-wwaConfig.txt` file. I’ve seen a case of COFA being used directly in the first bullet of the template, though I don’t see COFA, ZOFA and CIFA directly used much in templates.

The qualifiers for these directives (“`min_area`”, “`min_fraction`” and “`test_both`”) can be used directly in templates in the “AREA” and “WX” substitutions without specifying the directive. That is, the “`min_area`” qualifier can be used in a template without specifying the COFA, ZOFA or CIFA directive. You can use the qualifiers in the template if you want to change how a specific product behaves instead of using `LLL-wwaConfig.txt` to have a standard behavior for all templates. That is how you can change the D2D sensitivity to be different for different types of warnings.

The second mandatory bullet tells the expiration time of the warning:

*** UNTIL 745 AM EDT**

The following template line produces the “until” bullet:

*** UNTIL <EXPIRE | clock | local |interval=15>**

The third mandatory bullet is the basis for warning bullet. This tells how the weather event was detected, details what kind of event is occurring and the current location and movement of the event:

*** AT 710 AM EDT...NATIONAL WEATHER SERVICE DOPPLER RADAR INDICATED A TORNADO 6 MILES WEST OF ZEPHYRHILLS...MOVING EAST AT 15 MPH.**

There is lots of complex template syntax that generated the above text:

```

<AREA |file=wwa_counties |multiple=no |format=simple
  |item_format=[799,TORNADO]
  |file=wwa_counties |multiple=yes |format=simple
  |item_format=[799,LINE OF TORNADO PRODUCING STORMS]
  |var=reportType1>

<VAR |lead=A $$reportType1! WAS LOCATED |var=reportType>
{ | }
{ *** RADAR *** |
<VAR |lead=NULL
  |var=SEP>
}
{ ^Doppler radar indicated |
<VAR |lead=NATIONAL WEATHER SERVICE DOPPLER RADAR INDICATED A
  $$reportType1!
  |var=reportType>
}

<AREA |file=wwa_counties |multiple=no |format=simple
  |item_format=[799,STORM]
  |file=wwa_counties |multiple=yes |format=simple
  |item_format=[799,LINE]
  |var=reportType2>

%%* AT < NOW | clock | local>...$$reportType!
<WX | file = wwa_warn_city |used=accumulate
  |portions |include_field=2 |include_text=1 |filter
  |format=simple |item_format=[2199,FROM~][2799,~TO~][~-401]
  |proximal=8
  | file = wwa_counties
  |item_format=[2199,FROM~][2799,~TO~][~-400,county_type]
  |portions >.\
  <MOVEMENT | move_lead=..MOVING~ |interval=5
  |stationary=~THE $$reportType2! WAS NEARLY STATIONARY>.

```

When the warning is issued, WarnGen menu items can be selected to indicate how the severe weather was detected or observed. The “VAR” substitution specifies these items in the template. The geography is generated by “AREA” and “WX” substitutions.

The “file” qualifiers here specify the wwa_counties and wwa_warn_city GELTs. The PTPE and COPE directive tags can be used to get city names to describe the location of the weather event within the county. Optional lists of other cities in the warning area can be included also.

The fourth and last mandatory bullet is the pathcast:

*** THE TORNADO WILL AFFECT...
WESLEY CHAPEL BY 705 AM EDT.
DADE CITY BY 715 AM EDT.
ZEPHYRHILLS BY 720 AM EDT.**

The following complex template syntax created the above pathcast:

```
<AREA |file=wwa_counties |multiple=no |format=simple
  |item_format=[799,THE TORNADO WILL AFFECT...]
  |file=wwa_counties |multiple=yes |format=simple
  |item_format=[799,TORNADO PRODUCING STORMS WILL AFFECT...]
  |var=pathcastLead>
{ | }
{***** PATHCAST FEATURE . . . (Choose One) ***** | }
{^ Cities - arrival times |
<AREA |file=wwa_warn_city |format=simple |interval=5 |delta=0 |used=accumulate
  |lead=* $$pathcastLead!&
  |proximal=99 |proximal=
  |item_format=
  [199,~~][101,azrn1][-101,azrn2][-799,...][799,~BY~][700,clock][799,.&]
  |sort_by=[itime] [dist] |unique_by=[itime] [1] |group_by=[itime]
  |max_count=-12 |max_groups=-6 |in_group=-1 |max_dist=25
  |file=wwa_warn_city |max_dist=20
  |file=wwa_warn_city |max_dist=15
  |file=wwa_warn_city |max_dist=10
  |file=wwa_warn_city |max_dist=5
  |max_count=-12 |max_groups=-6 |in_group=-1
  |file = wwa_counties |portions
  |item_format=~~RURAL [0,county_type]~BY~[0,clock].&
  |extreme=no |central=no |interval=5 |delta=0 |unique_by=[1]
  |max_groups=-6 |max_dist=999999>
}
{ Cities - no times |
<AREA |file=wwa_warn_city |format=simple |interval=5 |delta=0 |used=accumulate
  |lead=* $$pathcastLead!&
  |proximal=99 |proximal=
  |item_format=
  [199,~~][101,azrn1][-101,azrn2][-799,...][799,.&]
  |sort_by=[itime] [dist] |unique_by=[itime] [1] |group_by=[itime]
  |max_count=-12 |max_groups=-6 |in_group=-1 |max_dist=25
  |file=wwa_warn_city |max_dist=20
```

```

|file=wwa_warn_city|max_dist=15
|file=wwa_warn_city|max_dist=10
|file=wwa_warn_city|max_dist=5
|max_count=-12|max_groups=-6|in_group=-1
|file = wwa_counties|portions
|item_format=~~RURAL [0,county_type].&
|extreme=no|central=no|interval=5|delta=0|unique_by=[1]
|max_groups=-6|max_dist=999999>
}{|}

```

The wwa_warn_city and wwa_counties GELTS are specified. Cities can be listed with or without storm arrival times. Sometimes extra lists of affected towns are included after the last bullet, especially if they are not included in the first bullet or in the pathcast.

Call to action statements come next in the warning:

THIS IS A DANGEROUS STORM! MOVE INTO THE INTERIOR ROOM ON THE LOWEST FLOOR OF A STURDY BUILDING...AWAY FROM WINDOWS. COVER YOUR HEAD AND BODY WITH PILLOWS OR BLANKETS.

The various call to action statements are options on the WarnGen GUI. Here are some possible call to action options that could be in the template (normally there are many more):

```

{*** CALLS TO ACTION (Choose up to 2) *** |
<VAR |lead=NULL
|var=SEP>
}
{ Dangerous storm! |
THIS IS A DANGEROUS STORM! MOVE INTO THE INTERIOR ROOM ON THE
LOWEST FLOOR OF A STURDY BUILDING...AWAY FROM WINDOWS. COVER
YOUR HEAD AND BODY WITH PILLOWS OR BLANKETS.
}
{ If outside... |
IF CAUGHT OUTSIDE...MOVE INTO A STURDY BUILDING. AS A LAST
RESORT...SEEK SHELTER IN A CULVERT...DITCH...
OR LOW SPOT...AND COVER YOUR HEAD WITH YOUR HANDS.
}
{Leave vehicles |
DO NOT USE YOUR VEHICLE TO OUTRUN A TORNADO. VEHICLES MAY BE
MOVED BY THE STRONG WINDS. IF YOU ARE IN THE PATH OF A TORNADO
...LEAVE THE VEHICLE AND GET IN A DITCH OR LOW SPOT...BUT WATCH
FOR RISING WATERS.
}

```

Next in the warning comes the list of latitude/longitude coordinates defining the warning box:

LAT...LON 2836 8236 2817 8236 2819 8212 2839 8210

To generate the list above, the following template item is used:

<COORDS>

The warning must end with “\$\$”. This is hardcoded in the template and would look like “\\$\$”. Each character following a “\” is taken literally with no special meaning and inserted into the warning.

After the end of warnings, the name or initials of the person issuing the warning usually is included. This can be done using the “must modify” markers in the template:

!**NAME/INITIALS**!

When the AWIPS text editor sees the **!** **!** markers, editing is required within the markers before the product can be transmitted. Alternatively, template variables can be set up so the person’s name can be selected from the WarnGen GUI.

5.1.5.7 TEMPLATE TROUBLESHOOTING

All of the pre-template files are subject to override. If you’re not getting the expected result from a template, check the list of places where localization checks for input files (page [13](#)). You may be accidentally overriding a change that you want.

There are numerous template files and sometimes it’s hard to tell exactly which template file is being used at a particular site to create a specific WarnGen product. To see what WarnGen templates are actually used (for the products listed on the WarnGen GUI), look in the WarnGen product type configuration file: `/awips/fxa/data/localizationDataSets/LLL/wwa.config`. This file is described beginning on page [40](#).

If a site uses different templates for themselves and their backup sites, here is a diagnostic test that can be useful. For example, if the site’s own TOR template doesn’t work, have them restart D2D as one of their backup sites (this is full service backup) and try creating the TOR product using the backup site’s TOR template. If the backup site TOR is OK, then there is likely some syntax difference in the local template and the backup site template that the site can determine. If the backup site ID is LLL, the backup site TOR pre-template could be in `/awips/fxa/data/localization/LLL/LLL-wwa_tor.preWWA` or `/data/fxa/customFiles/LLL-wwa_tor.preWWA`.

If a particular template isn’t working, another possible trick is to try installing a template from

another site. For example, at site VEF, an SVS template was causing WarnGen to hang when the “create text” button was pushed. However, when D2D was started as site FGZ (one of the VEF backup sites), the SVS template for FGZ worked fine. As a test, I moved the SVS template file /awips/fxa/data/localizationDataSets/FGZ/wwa_svs_cont_svr.wwaProd from site FGZ to /awips/fxa/data/localizationDataSets/VEF/wwa_svs_svr.wwaProd at site VEF. This was the follow up to the TOR product and the two sites had different names for this template. To test at VEF, I edited the WMO heading in file wwa_svs_svr.wwaProd to change site FGZ to VEF and changed the AFOS node from PHX to RNO. After D2D was restarted as VEF, the bogus SVS template worked. This trick allowed us to test the template without having to localize the workstation. The site was then able to compare the VEF and FGZ templates to figure out what was wrong in the template syntax.

You can also tinker with the template files (the .wwaProd files) to see what effect that syntax changes have when WarnGen is run. This way localization won’t need to be run again for every change that you make. There is very little difference between the pre-template (localization input file) and template files (localization output in /awips/fxa/data/localizationDataSets/LLL). Once you figure out what you want in the template, be sure to make the appropriate change in your pre-template file in /data/fxa/customFiles and localize so that your change won’t be overwritten by the next “-wwa” localization.

If a particular template isn’t working, compare it with a similar template that does work and maybe you’ll see a difference in syntax. There are many combinations and permutations of template syntax that sites use to customize their products.

If you can’t figure out a template problem at the field site, you can configure a local workstation to try to recreate the problem. Make sure you have enough disk space on the workstation. Each localizationDataSets/LLL directory takes up quite a bit of disk space. To remove an unwanted localization, simply remove the appropriate directory /awips/fxa/data/localizationDataSets/LLL.

For example, if site GYX has a template problem, you can localize your test workstation to run WarnGen as GYX (even if the local ingest is not site GYX):

```
./mainScript.csh -WWA GYX GYX
```

“-WWA” will perform the minimum localization needed to run WarnGen as if it were at site GYX, i.e. perform WarnGen full service backup for site GYX. Try generating a test warning (be sure not to transmit it to the AWIPS WAN) using only the default national template to see if the problem may be with the site specific version of the template. You can also try installing the site specific template, localizing and see if the problem persists on your test system. Trying these should help determine if the problem is site specific or a more widespread AWIPS problem.

5.1.6 UGC TROUBLESHOOTING

This section describes how WarnGen gets the UGC codes that are placed in warnings. UGC codes are introduced beginning on page 5. All the AWIPS UGC codes are defined in the “uscounty” shape file. See page 20 for more on shape files. To see the UGC code in the shape file, do the following as user “fxa” in ds1:/data/fxa/nationalData:

```
shp2bcd i uscounty | more
```

As an example, to see the FIPS code for Montgomery County, MD, scroll down to the MD section. The Montgomery County entry is:

```
----- Record 999 -----  
DONE: 0  
STATE: MD  
CWA: LWX  
COUNTYNAME: Montgomery  
FIPS: 24031  
TIME_ZONE: E  
FE_AREA: cc  
LON: -77.2043  
LAT: 39.1397  
-----
```

The FIPS code for Montgomery County is 24031. The “24” stands for Maryland, “031” is Montgomery County.

When the “-wwa” localization is run, the UGC info from the shape file is included in the WarnGen GELTs (Geographic Entity Lookup Tables). For more on GELTs, see page 26. GELTs are used by WarnGen to put all geography items into warning products. To verify that the Montgomery County UGC, is in the “wwa_counties” GELT, do the following:

```
cd /awips/fxa/data/localizationDataSets/LLL      (where LLL is the site ID)  
ls -l wwa_counties.*
```

The output will look something like:

```
-rwxrwxr-x 1 fxa    fxalpha 218400 Feb 13 18:33 wwa_counties.EW  
-rwxrwxr-x 1 fxa    fxalpha 218400 Feb 13 18:33 wwa_counties.NS  
-rwxrwxr-x 1 fxa    fxalpha 17824 Feb 13 18:33 wwa_counties.entity  
-rwxrwxr-x 1 fxa    fxalpha 105 Feb 13 18:33 wwa_counties.gelt  
-rwxrwxr-x 1 fxa    fxalpha 30297 Feb 13 18:33 wwa_counties.id  
-rwxrwxr-x 1 fxa    fxalpha 436800 Feb 13 18:38 wwa_counties.table
```

Check that all 6 pieces of the GELT were created when the localization was run. You can see the UGC code in the GELT by doing this:

```
grep MD wwa_counties.id | grep MONTGOMERY
```

The output has the UGC in the next to last field. Here we see Maryland county 031 (MDC031):

```
49 39.148 -77.192 a 49 MONTGOMERY |cc MD| MDC031 |LWX
```

The “-wwa” localization also builds the product templates used by WarnGen to create an individual warning. In this example, we’ll look at a tornado warning. The localization output tornado template is in /awips/fxa/data/localizationDataSets/LLL/wwa_tor.wwaProd. Near the top of the template, just after the WMO heading, is a “#include” line:

```
#include "wwa_county_ugc.template"
```

This tells WarnGen to use the list template “wwa_county_ugc.template” to get the county list of UGCs. For more on list templates, see page [50](#). The localization output version of the list template (which is used by WarnGen) is in /awips/fxa/data/localizationDataSets/LLL/wwa_county_ugc.template. Sites rarely edit the list templates, so if you suspect a list template problem, the list template can be compared at another site to see if there are differences.

5.1.7 VTEC TROUBLESHOOTING

OB2 introduced three new WarnGen features: VTEC, enhanced product QC and a more automated followup capability. The following two web sites have useful technical information on the new OB2 WarnGen features:

<http://www-sdd.fsl.noaa.gov/~ramer/noaa/ob2-wgn/ob2-setup.html> - FSL summary of migration, testing, and training issues for OB2 WarnGen

http://www.ops1.nws.noaa.gov/awips_install.htm - the AWIPS System Administration page has a link to the document “OB2 WarnGen WFO Implementation Instructions.” This is based on the FSL OB2 WarnGen migration, testing and training paper and has detailed step by step instructions.

VTEC is introduced on page [6](#). The WarnGen VTEC is implemented in AWIPS release OB2 using WarnGen product templates and Unix configuration files. VTEC will become operational nationwide at a later date.

For WarnGen to produce VTEC, the configuration file /data/fxa/nationalData/warnGenVTEC.mode must be set correctly. The file

warnGenVTEC.mode can contain one of the following three options:

ON VTEC is turned ON
OFF VTEC is turned OFF
TEST VTEC is turned on, but in the AWIPS text editor, the VTEC will be within the “must modify” markers (!** **!) and the product can’t be transmitted until the “must modify” markers are removed.

If the file warnGenVTEC.mode does not exist, VTEC is assumed to be ON. Also, if the warnGenVTEC.mode file contains anything but “OFF” or “TEST”, then VTEC will be turned on for operational use.

Both files warnGenVTEC.mode and textQC.config (the QC configuration file) can be overridden. That is, to change the behavior of one workstation, different versions of warnGenVTEC.mode and textQC.config can be put in directory /awips/fxa/data/localizationDataSets/LLL. Doing this will override the associated VTEC and QC settings in /data/fxa/nationalData. Note that no localization is needed to implement this change since the change is made in the localization output directory /awips/fxa/data/localizationDataSets/LLL.

To change the VTEC operation of one workstation, do the following as user “fxa”:

```
cd /awips/fxa/data/localizationDataSets/LLL      (where LLL is your site ID)
echo CCC > warnGenVTEC.mode                      (CCC is “ON”, “OFF”, or “TEST”)
```

Beware that the VTEC setting and the OB2 QC setting must be compatible. The OB2 enhanced QC cannot be turned on and VTEC turned off. The QC check will quit immediately when it finds the VTEC line missing. QC and VTEC should both be on or both be turned off.

For WarnGen to generate VTEC, the P-VTEC and/or H-VTEC lines need to exist in the WarnGen template file. The VTEC line comes right after the WMO heading. Here is the P-VTEC line from a TOR:

```
ZCZC $$cccValue!TOR$$xxxValue! DEF&
TTAA00 $$wmoValue! <NOW | ddhhmm | gmt>&&
#include "wwa_county_ugc.template"
&/NEW.$$wmoValue!.TO.W.$$ETN_VAL!.<START|ymdthmz|gmt>-<EXPIRE|ymdthmz|gmt>/
```

Here are the P-VTEC and H-VTEC lines from the FFW template:

```
ZCZC $$cccValue!FFW$$xxxValue! DEF&
TTAA00 $$wmoValue! <NOW | ddhhmm | gmt>&&
#include "wwa_county_ugc.template"
&/NEW.$$wmoValue!.FF.W.$$ETN_VAL!.<START|ymdthmz|gmt>-<EXPIRE|ymdthmz|gmt>/
&/0.$$imCause!.000000T0000Z.000000T0000Z.000000T0000Z.0/
```

Some of the templates have slightly different combinations of the P-VTEC and H-VTEC lines. For example, the Flash Flood with Severe Thunderstorm template has a P-VTEC for the flash flood, another P-VTEC for the severe thunderstorm, plus an H-VTEC for the flash flood.

In OB2, the following default WarnGen templates in /data/fxa/nationalData have VTEC:

FILE NAME	PRODUCT NAME	NNN
wwa_tor.preWWA	Tornado Warning	TOR
wwa_svr.preWWA	Severe Thunderstorm Warning	SVR
wwa_svrwx_sta_county.preWWA	Severe Weather Statement	SVS
wwa_specmarine.preWWA	Special Marine Warning	SMW
wwa_mar_wx_sta.preWWA	MarineWeather Statement	MWS
wwa_ffw.preWWA	Flash Flood Warning	FFW
wwa_ffw_svr.preWWA	Flash Flood w/ Svr Thunderstorm	FFW
wwa_dam_break.preWWA	Non convective FFW (dam break)	FFW
wwa_fflood_sta_county.preWWA	Flash Flood Statement	FFS
wwa_fflood_sta.preWWA	Non Convective Flash Flood Stmt	FFS

In build 5.2.1, on Linux workstations, you got a VTEC ETN alert because the file system /awips/adapt was physically located on ds1 but Linux did not NFS mount this file system in 5.1.2. In 5.2.1, the Linux boxes NFS mounted the /awips/adapt file system, correcting the problem.

There was a WarnGen VTEC configuration problem in build 5.2.2.2 and OB1. Beginning in 5.2.2.2, the “VTEC|0” switch in /awips/adapt/ifps/data/wwaSiteConfig.txt no longer worked for WarnGen (it did still control VTEC for WWA). In wwaSiteConfig.txt, “VTEC|1” meant VTEC was turned on for WWA, “VTEC|0” meant VTEC was turned off for WWA. As a result, the only way to disable VTEC for WarnGen products was to comment out the VTEC line in WarnGen templates. WarnGen should not send out products with VTEC until VTEC is officially implemented. The workaround was to add a “//” at the beginning of the VTEC line (to comment it out) like so:

```
//&/NEW.$$wmoValue!.TO.W.ETN#.<START|ymdthmz|gmt>-<EXPIRE|ymdthmz|gmt>/
```

This change was made in the site maintained WarnGen pre-template files in /data/fxa/customFiles/wwa*preWWA. Then each workstation was localized using “./mainScript.csh -wwa” and D2D restarted. For more information on this issue, see the AWIPS Admin listserver posting from 2/28/03.

5.1.8 OB2 ENHANCED QC TROUBLESHOOTING

In OB2, enhanced WarnGen product QC was added. The QC is done by the AWIPS text editor after the WarnGen “create text” button is used. Here is a quick summary of the QC items that are checked (from the OB2 Release Notes):

1. proper UGC and VTEC codes
2. UGC line matching text counties, zones, etc.
3. internally consistent times
4. product type matching the Mass News Disseminator section and text in the first bullet
5. correct contents and format of the Mass News Disseminator and the first three bullets
6. “\$\$” at the end of the product

The QC is done for these products: TOR, SVR, FFW, SMW. There is no QC done in WarnGen followup products (SVS, FFS, MWS).

The configuration file which controls the OB2 enhanced QC is /data/fxa/nationalData/textQC.config. This file controls which products are QC'd and controls how the text editor responds to errors that are found. OB2 is installed with a default version of textQC.config named /data/fxa/nationalData/textQC.config.VTEC in which all OB2 QC is turned off. The following are the settings in the textQC.config file when VTEC and OB2 QC are turned on:

NNN	Code_Name (proc/file)	Code_Type (EXE/INT)	Send_Warn Y/N	UGC3 Y/N	B_GoAhead Y/N	B_Abort Y/N	B_WGN Y/N	B_WRK Y/N	WarnType string
TOR	{localWarningInfoTest TOR}	EXE	Y	N	Y	Y	N	N	{Tornado Warning}
SVR	{localWarningInfoTest SVR}	EXE	Y	N	Y	Y	N	N	{Severe Thunderstorm Warning}
SVS	NONE	INT	Y	Y	Y	Y	N	N	{Severe Weather Statement}
FFW	{localWarningInfoTest FFW}	EXE	Y	N	Y	Y	N	N	{Flash Flood Warning}
FFS	NONE	INT	Y	Y	Y	Y	N	N	{Flash Flood Statement}
SMW	{localWarningInfoTest SMW}	EXE	Y	N	Y	Y	N	N	{Special Marine Warning}
MWS	NONE	INT	Y	Y	Y	Y	N	N	{Marine Weather Statement}
SLS	NONE	INT	Y	N	Y	Y	N	N	{Watch Box Areal Outline}

The fields “Code_Type” and “UGC3” control the QC. The other fields control how the AWIPS text editor responds to QC errors that are found. See the documentation in file textQC.config for details.

Executable program /awips/fxa/bin/localWarningInfoTest (with one argument) is invoked for the enhanced OB2 QC check on TOR, SVR, FFW and SMW products. For products SVS, FFS, MWS and SLS, only the simpler pre OB2 QC is done - that is done by program /awips/fxa/bin/textQC.tcl.

Both files warnGenVTEC.mode (the VTEC configuration file) and textQC.config can be overridden. That is, to change the behavior of one workstation, different versions of warnGenVTEC.mode and textQC.config can be put in directory /awips/fxa/data/localizationDataSets/LLL. Doing this will override the associated VTEC and QC settings in /data/fxa/nationalData. Note that no localization is needed to implement this change since the change is made in the localization output directory

/awips/fxa/data/localizationDataSets/LLL.

Beware that the VTEC setting and the OB2 QC setting must be compatible. The OB2 enhanced QC cannot be turned on and VTEC turned off. The QC check will quit immediately when it finds the VTEC line missing. QC and VTEC must both be on or both turned off.

5.1.9 FOLLOWUP TROUBLESHOOTING

OB2 introduced more automated WarnGen followup capabilities. The “followup” pull down menu was added in the WarnGen “Product Type” section. The “followup” lists relevant recent warnings and has the following options for each warning:

1. Update List - update the followup list to show recent warnings. There is no automatic notification of new products and the update must be requested manually.
2. CAN - cancel a warning
3. CON - continue a warning
4. CA2 - in the case of a combined FFW/SVR, cancel the second part (SVR) and continue the first part (FFW).
5. EXP - this warning has expired, but can be used as the basis for a reissuance
6. Reissue - issue a new product for an ongoing quasi-steady severe event. The other followup options do not allow the box or time to be edited. This option allows the box and time to be edited because a new warning is being issued.

The followup list is selective in the products that are listed. If an SVS is selected in the WarnGen “Product Type” list, recent TORs and SVRs will be shown, since either of these can be followed up with the SVS. Similarly, if a MWS is chosen, only SMW’s will display in the followup list. If a warning is selected, recent warnings of the same type are displayed, with the actions that can be performed, e.g. cancel or continue the warning. When a followup product is created, neither the warning times nor the box defining the warning area can be edited.

The following web page has the FSL summary of the WarnGen followup capability:
<http://www-sdd.fsl.noaa.gov/~ramer/ob2-wgn/followUp.html>

In OB2, WarnGen needs to find all recent warnings and followup products to populate the followup list. WarnGen does this by reading all versions of the following local products from the text database: TOR, SVR, FFW, SMW, SVS, FFS, MWS. If there are lots of old versions in the text database, WarnGen will take a long time to launch, or it may take a long time for “create text” to finish. Brief inquiries by the SST at field sites showed that the delay may become unacceptable when more than about 50 old versions of these products are stored. When the problem occurred, the following types of message appeared numerous times in the warnGenWish display log under /data/logs/fxa/display (the example concerns a TOPSVRGLD product:

```
warnGenWish 31786 1063997047.548471 18:44:07.548 PROBLEM: Mismatch between
vtecType of individual product and product vtecType
warnGenWish 31786 1063997047.548565 18:44:07.548 PROBLEM: `KTOP.SV.W'
`SV.W'
```

If the WarnGen delay occurs in OB2, and there are lots of old local versions of these products in the text database, purge the extra products from the text database. To do this, log onto ds1 as fxa. To see a summary of all versions of product WBCSVSLWX, use the following command:

```
textdb -A WBCSVSLWX
```

If there are 100 versions of WBCSVSLWX stored, this command will reduce to 20 versions:

```
textdb -v WBCSVSLWX 20
```

FSL plans to add a check to assure that WarnGen does not read an excessive number of products from the text database (OB2 DR 13210, OB3 DR 13209).

Western Region has a requirement that the FFW be followed with the FLS (Flood Statement) instead of the FFS (Flash Flood Statement). This situation is described in DRs 13168 and 13175. This capability is expected to be implemented in the OB2.2 maintenance release.

5.1.10 WARNGEN PRODUCT SEGMENTATION

Most WarnGen products are non segmented and OB2 WarnGen can't create segmented products. For background on product segmentation, see page 8. OB2 WarnGen requires all product pre-templates to have a UGC in the third line (just under the AWIPS product ID), that is, requiring them to be non segmented. If there is no UGC in the third line, when "create text" is used, WarnGen gives the red banner message "NO UGCs in product when at least one is expected" and won't create the product. There are plans in the VTEC era to enable WarnGen to create multi-segmented SVS products, because of the need to remove counties from active warnings while leaving other counties in the warning.

5.1.11 WARNGEN AND THE TEXT PRODUCT SITE ID

This section describes how WarnGen gets the site ID to use in product headings. It also illustrates how WarnGen determines the AWIPS product ID and also some nitty gritty details on localization override and product templates.

At site SGX, WarnGen coded the wrong site ID in the AWIPS ID of certain products. The site ID had changed from SAN to SGX. WarnGen still coded SVRSAN instead of SVRSGX, even though various configuration files had been changed from SAN to SGX and localization completed. The problem existed on all workstations but not for all products.

For the sake of completeness, we'll mention where the entire CCCNNNXXX comes from, even though the problem only relates to the XXX. Recall that the AFOS product ID (PIL) has format CCCNNNXXX (eg LAXSVRSGX):

CCC = AFOS node
NNN = product ID
XXX = local site ID

The AWIPS product ID consists of the NNNXXX part of the AFOS PIL. The source of all the CCCNNNXXX items is file /data/fxa/nationalData/afosMasterPIL.txt. This file is maintained on the NOAA1 server and it's good for sites to have the nationally maintained version so everyone has the valid PILs. If we do "grep LAXSVR afosMasterPIL.txt" we see the desired PIL LAXSVRSGX, plus the PILs for other sites with AFOS node LAX:

LAXSVRBIH
LAXSVRLAX
LAXSVRLOX
LAXSVRSAN
LAXSVRSGX

Another important file for product headings is /awips/fxa/data/afos2awips.txt. If we do "grep LAXSVR afos2awips.txt" we see the associated WMO headings for the LAXSVR products:

LAXSVRBIH WUUS56 KHNX
LAXSVRLAX WUUS56 KLOX
LAXSVRLOX WUUS56 KLOX
LAXSVRSAN WUUS56 KSGX
LAXSVRSGX WUUS56 KSGX
LAXSVRSMX WUUS56 KLOX

File afosMasterPIL.txt is used by the localization task "-text" to update the file /awips/fxa/data/localizationDataSets/LLL/afosMasterPIL.CCC. When localization runs, it looks in the following places for inputs (in the following order):

/awips/fxa/data/dataFile
/data/fxa/nationalData/dataFile
/data/fxa/data/localization/realizations/RRR/RRR-dataFile
/awips/fxa/data/localization/LLL/LLL-dataFile
/data/fxa/customFiles/dataFile
/data/fxa/customFiles/LLL-dataFile
/data/fxa/customFiles/\$FXA_CUSTOM_VERSION/dataFile
/data/fxa/customFiles/\$FXA_CUSTOM_VERSION/LLL-dataFile

If localization sees similar things in different places above, it will use the LAST thing it sees in the list above (not the NEWEST thing). This is called override and is introduced on page [13](#).

Basic localization directives for a specific workstation are in /awips/fxa/data/localization/LLL/LLL-mainConfig.txt. The site had changed SGX-mainConfig.txt in ds1:/data/fxa/customFiles and on the workstation in /awips/fxa/data/localization/SGX. The contents of ws1-sgx:/awips/fxa/data/localization/SGX/SGX-mainConfig.txt were:

```
@@@WFO SGX
@@@CCC LAX
@@@XXXL CA LAX SGX LOX PHX PSR LAS VEF SFO MTR FAT
@@@XXXR CA LAX SGX LOX PHX PSR LAS VEF SFO MTR FAT
@@@PUP_TABLES TRUE
@@@XXX SGX
```

The contents of ds1-sgx:/data/fxa/customFiles/SGX-mainConfig.txt (this overrides file ws1:/awips/fxa/data/localization/SGX/SGX-mainConfig.txt) were:

```
@@@WFO SGX
@@@CCC LAX
@@@XXXL CA LAX SGX LOX PHX PSR LAS VEF SFO MTR FAT
@@@XXXR CA LAX SGX LOX PHX PSR LAS VEF SFO MTR FAT
@@@PUP_TABLES TRUE
@@@XXX SGX
```

So SGX-mainConfig.txt was OK. SGX was declared as the value for the XXX localization directive in all places. This specified the AFOS XXX value in locally created products. The WFO directive specifies the CWA for the localization geography. The XXXL directive specifies the XXX values that are important to the CWA local area. This should include both 3 character site IDs and 2 character state abbreviations. The XXXR directive specifies the XXX values that are important to the CWA regional area.

Now we look at WarnGen specific directives (see page [51](#) for more on these). LLL-wwaConfig.txt contain directives that pertain to WarnGen. Most WarnGen directives result in direct text substitution in the WarnGen product definition files (templates). Some directives have a functional impact on how localization works. The file ds1:/awips/fxa/data/localization/documentation/directive.html has details on directives.

The contents of file /awips/fxa/data/localization/SGX/SGX-wwaConfig.txt follow:

```
@@@SRWARN FALSE
@@@ORABOUT TRUE
@@@SBID
@@@OFFH SAN DIEGO CA
@@@OFFT IN SAN DIEGO
@@@COPE |portions |extreme |central
@@@CIPE |portions |extreme |central
@@@PTPE |portions |extreme |central
```

There was no override file like ds1:/data/fxa/customFiles/*wwaConfig.txt. After checking that the basic WarnGen configuration files are OK, we looked at the WarnGen product templates at site SGX.

On ds1-sgx:/data/fxa/customFiles, SGX had two versions of the SVR pre-template: wwa_svr.preWWA and SGX-wwa_svr.preWWA. Localization looks first at wwa_svr.preWWA, then sees SGX-wwa_svr.preWWA and uses it instead (override), ignoring the dates that the pre-templates were last modified.

In the wwa_svr.preWWA pre-template file the ZCZC line had SVRSGX hard coded:

```
ZCZC $$cccValue!SVRSGX DEF&
```

Next localization saw the SGX-wwa_svr.preWWA pre-template file, which overrides the wwa_svr.preWWA file. In this file the ZCZC line still had SVRSAN hard coded:

```
ZCZC $$cccValue!SVRSAN DEF&
```

The error was caused when SGX had "SVRSAN" hardcoded in the SGX-wwa_svr.preWWA pre-template, so WarnGen produced the wrong AWIPS product ID (SVRSAN) in the product heading. That is, "mainScript.csh -wwa" produced the template file on the WS: /awips/fxa/data/localizationDataSets/SGX/wwa_svr.wwaProd with "LAXSVRSAN" as the AFOS PIL in the heading. The TOR pre-templates had the same problem, but the FFW pre-templates were OK.

To fix the problem, the SGX-wwa_svr.preWWA pre-template was edited (also the TOR pre-template) so it used the XXX variable to get the correct XXX:

```
ZCZC $$cccValue!SVR$$xxxValue! DEF&
```

We ran localization on one workstation using "./mainScript.csh -wwa", restarted D2D and verified that the problem was fixed. Finally, the other workstations were localized.

5.1.12 RECREATING A FIELD SITE'S D2D/WARNING ENVIRONMENT

Sometimes in troubleshooting a problem, it helps to actually see the problem in person. Also, it may be necessary to recreate the field site's D2D/WarnGen environment at your test site in order to diagnose the problem. This section gives instructions on how to see a remote site's D2D display, see their WarnGen product that is sent to the text editor, and install their D2D/WarnGen environment on your local workstation.

The following are instructions on how to export a remote site's D2D display to your local workstation. This works best if the remote workstation is a Linux workstation. D2D needs to be launched on the remote workstation and this takes a while on HP workstations and may also cause performance problems on HP workstations when running another D2D session. The remote workstation and your local workstation can be the same type or different (e.g. HP-UX vs. Linux workstations).

On your local test workstation, open a telnet window as user "awipsusr." Get your workstation IP address from file /etc/hosts. For example, if you are using lx2, do "grep lx2 /etc/hosts." For this example, pretend that your lx2 IP address is 123.123.123.123. If you are using the left hand graphic display, this is denoted as ":0.0", the right hand display is ":0.1." In the telnet window, enter the following command if you are using the left hand display on your local lx2:

```
setenv DISPLAY 123.123.123.123:0.0
```

If using the right hand display, end the command with ":0.1" instead. Next enter this:

```
xhost +
```

You'll get a message like "access control disabled, clients can connect from any host." Next, rlogin to the workstation at the remote site, then use:

```
su - awipsusr
```

to become "awipsusr." Next enter:

```
cd /awips/fxa/bin
setenv DISPLAY 123.123.123.123:0.0      (repeat your local display address)
./start-d2d
```

You'll see a message like "shutting down any previous D2D." This refers only to any D2D instances on the current Unix terminal (telnet) session. Multiple instances of D2D can run on one workstation. You may get a "keypad error" message: use "cancel" and ignore this error.

Now D2D will start up on the remote workstation and the D2D display will be exported to your

local display. The response will be somewhat slow (especially on remote HP workstations), but you can see exactly what the site sees on the D2D and WarnGen display. When done, be sure to exit D2D, then exit the telnet window on your local workstation.

You can do WarnGen map related testing like this, but if you need to “create text”, the text editor will pop up at the remote site. For more on the text workstation, see page [80](#).

If you need to see the text that WarnGen produces, you can redirect the text to a remote Unix flat file instead of sending it to the text editor at the remote site. To do this, some preparation is needed first. Open a second telnet window on your local workstation, then rlogin to the remote workstation and use:

```
su - fxa
```

to become user “fxa.” You need to create a directory on the remote workstation, where WarnGen will deposit the warning text. Enter this:

```
mkdir -p /data/fxa/flatText/CCCNNNXXX (use literal “CCCNNNXXX”)
```

If you also want to test the WarnGen followup product text, you need to create a subdirectory for each of the product types you will test. The products are TOR, SVR, FFW and SMW. If you are testing a followup for a TOR at site LWX, create a directory with the corresponding AFOS PIL for TOR:

```
mkdir /data/fxa/flatText/WBCTORLWX
```

In the same way, create a directory for each of the products that will have a followup. Don’t create a subdirectory for the followup products (SVS, FFS and MWS).

Recall that beginning in OB2, /data/fxa on the workstation is local to that workstation. From the workstation, to access /data/fxa on the ds, use “dsdata” instead, which is a link to ds1:/data/fxa. If you’re logged onto ds1, /data/fxa is local to the ds1. In this case, we want to create the directories in /data/fxa which is local to the workstation. More details on these items are on the localization input file section on page [12](#).

Now go back to the first telnet window where you are logged on to the remote workstation as awipsusr. You should already have done the following:

```
cd /awips/fxa/bin
setenv DISPLAY 123.123.123.123:0.0 (repeat your local display address)
```

and exited D2D, if needed. Now enter the following to redirect the WarnGen text:

```
setenv FXA_FLAT_FILE_TEXT TRUE
setenv FXA_WARNGEN_PRODUCT_ID CCCNNNXXX
```

Start D2D using:

```
./start-d2d -nokeypad
```

Now, when you use “create text” in WarnGen, the file will be sent to the remote directory /data/fxa/flatText/CCNXXX with a file name based on the current date and time. For example, if you created the product on 7/22/03 at 12:34Z, the file created will be “20030722_1234.” To make sure this is set up right, you may want to first create a more innocuous product like a NOW to be sure it goes to the flat file instead of the text editor. In the second telnet window, you can use the Unix commands “cat” or “more” to view the file and check if WarnGen created the correct warning text. When done, be sure to exit D2D, then exit both telnet windows on your local workstation.

If you want to test the WarnGen followup function, you first need to make the workstation think that the original warning (TOR, SVR, FFW or SMW) has been stored in the text database and transmitted. If a WBCTORLWX has been created on 7/22/03 at 12:34Z, do the following on the second telnet window (this can be done as user awipsusr or as fxa):

```
cd /data/fxa/flatText/CCNXXX
cat 20030722_1234          (to see the TOR)
mv 20030722_1234 ../WBCTORLWX
```

The “mv” will make the workstation think that the product has been “stored” in the text database and transmitted. Now go back to WarnGen and you should see the original warning listed in the WarnGen “followup” pull down menu with the various followup options that are available. Select one of the followup options to create the SVS followup product for the TOR. If the followup SVS was created at 12:40Z, you will see it in file /data/fxa/flatText/CCNXXX/20030722_1240. This followup product must be “moved” to the subdirectory of the product that it follows, in this case, directory WBCTORLWX. If the SVS follows a SVR, the SVS would be moved to subdirectory WBCSVRLWX. In the same way, an FFS would be moved to subdirectory WBCFFWLWX and a MWS would be moved to subdirectory WBCSMWLWX.

When done, be sure to exit D2D, then exit both telnet windows on your local workstation.

You can configure a local workstation to try to recreate the site’s D2D environment, that is localize your local workstation to run D2D as the remote site. Make sure you have enough disk space on the workstation. Each localizationDataSets/LLL directory takes up quite a bit of disk space. To remove an unwanted localization, simply remove the appropriate directory /awips/fxa/data/localizationDataSets/LLL.

For example, if site GYX has a WarnGen problem, you can localize a workstation to run D2D/WarnGen as GYX (even if the local ingest is not site GYX):

```
./mainScript.csh -WWA GYX GYX
```

“-WWA” will perform the minimum localization needed to run WarnGen as if it were at site GYX, i.e. perform WarnGen full service backup for site GYX. Now you can simulate the WarnGen environment at the field site.

If you need to do more testing in the actual remote site’s environment, you can install their entire WarnGen environment on your local workstation. In most cases, you can install the entire /awips/fxa/data/localizationDataSets/LLL directory from the remote workstation onto your local workstation, and then WarnGen will run just like at the remote site. The remote workstation and your local workstation need to be the same type (HP-UX or Linux). You need quite a bit of room on the /awips/fxa partition on your local workstation, so Linux workstations are better.

On the remote workstation (in this example, at site GRR), as user “fxa” do the following:

```
cd /awips/fxa/data/localizationDataSets
tar cvfp GRR.tar GRR
```

This creates a tar file with all the contents of directory GRR in file GRR.tar. The “p” option preserves the original file ownership and permissions. If you get an error on the tar command, try it as user “root” instead. The tar file will be very large, so it needs to be compressed. Use:

```
gzip GRR.tar
```

This will remove file GRR.tar and create compressed file GRR.tar.gz. Another option is to use:

```
compress GRR.tar
```

which produces file GRR.tar.Z. Gzip does a better compression than compress. On the remote system, save the compressed tar file in ds1:/data/local (a partition with lots of room) and ftp it to your local ds1:/data/local partition. On your local ds1, do the following (best as user “root”):

```
cd /data/local
ftp n.n.n.n          (where n.n.n.n is the IP address of the remote ds1)
log on to the remote host
cd /data/local
hash
binary
get GRR.tar.gz      (or GRR.tar.Z if you used “compress”)
```

Now do something else for 15 minutes or so. When the file transfer completes, make sure you have enough room on your test workstation to add the new localizationDataSets directory. Copy the compressed tar file to your workstation directory /awips/fxa/data/localizationDataSets. Then do the following to uncompress the tar file and create file GRR.tar:

```
gunzip GRR.tar.gz    (or uncompress GRR.tar.Z)
```

Now install the tar file using:

```
tar xvpf GRR.tar
```

This creates the directory /awips/fxa/data/localizationDataSets/GRR. Now you can restart D2D and run D2D as site GRR. You will have all the same WarnGen configurations as used by GRR.

In extreme cases, you may need to install other items from the remote AWIPS on your local AWIPS. These are all the areas from which D2D and WarnGen access. In these extreme cases, some files from the following directories may be needed:

```
/awips/fxa/data  
/data/fxa/nationalData  
/awips/fxa/data/localization/LLL  
/data/fxa/customFiles
```

Getting these files can be dangerous since other AWIPS configuration files are stored in these locations. The same procedures can be used as above, except that you need a tar file with only certain files in a directory, not the entire directory contents. For example, for WarnGen testing, you need to tar up only the files in /awips/fxa/data and exclude the subdirectories. Use this to get a list of the file names:

```
cd /awips/fxa/data  
ls -al | grep -v ^d | awk '{print $9}' > file.names
```

The “grep” excludes directories (ls entries beginning with “d”). The “print” extracts field number 9 from the ls entry (the file name). Now use this “tar” command to save the files listed in file.names:

```
tar cvfp file.tar -I file.names
```

In Linux, use “-T” instead of “-I”. Beware that you need to be within the /awips/fxa/data directory to use the “tar cv” and subsequent “tar xv”.

5.2 WARNGEN, UNIX AND OTHER AWIPS SOFTWARE

This section describes how WarnGen relates to Unix, the text workstation, the AWIPS notification server, and AWIPS product dissemination services.

5.2.1 WARNGEN UNIX PROCESSES

Each pane in D2D has an IGC_process, so a workstation with two monitors should have ten active IGC processes (assuming one large pane and four small panes per monitor). The IGC_process starts the daemon /awips/fxa/bin/warnGenWish, which checks to see if the WarnGen button has been pushed on D2D. Normally there is one warnGenWish processes active for each D2D that is running. When WarnGen is invoked, warnGenWish runs the tcl script /awips/fxa/bin/warnGen. The warnGen script comments say that it displays the WarnGen dialogue box and invokes warnGenWish to start a WarnGen session. I don't quite understand this because, after WarnGen is invoked, you don't see another warnGenWish process starting, you just see the same warnGenWish process (same Unix process ID).

The warnGenWish process is associated with the big pane in D2D. If the big pane is swapped out with WarnGen active, another warnGenWish process starts up for the new big pane, and the warnGenWish associated with the old WarnGen session (now in a small pane) keeps running. If the small pane (with WarnGen active) is swapped back in as the big pane, all the warnGenWish processes remain active.

In rare cases, in WarnGen troubleshooting it is helpful to look at the workstation display logs. The display logs are in directory /data/logs/fxa/display. There are subdirectories for each graphics display and any text displays driven by this workstation. The subdirectory name is defined by the "DISPLAY" environment variable. The left display directory name ends with :0.0, the right display ends with :0.1. For three headed workstations, the names will end with :0.0, :0.1 and :0.2. The directories may also contain the name of the workstation. For example on two headed ws1 at site LWX, the display directories may be named ws1-lwx:0.0 and ws1-lwx:0.1, or they may be named just :0.0 and :0.1. If the "DISPLAY" environment variable is not set, the directory name is just ":0". If a display was exported to another host, you'll see a directory name with the remote IP address. For example, if the left hand display was exported to host 123.123.123.123, the directory name would be 123.123.123.123:0.0.

In each display directory are subdirectories for each day; their names have a format of yyyyymmdd. In each daily directory are the D2D logs. The main D2D log file is named fxaWish<PID> where <PID> is the Unix process ID for the main D2D process. There are subdirectories named fxaWish<PID>.children which contain logs for the various processes spawned by D2D. In the "children" directory, an IGC log exists for each D2D pane. There normally should be 5 of these log files named IGC_Process<PID>. Also in the "children" directory is the WarnGen log, which is named warnGenWish<PID> where <PID> is the Unix process ID for the warnGenWish process.

For example, the WarnGen log file for the left hand display for ws1 at site LWX, on March 11, 2003, is /data/logs/fxa/display/ws1-lwx:0.0/20030311/fxaWish4436.children/warnGenWish4511 (where 4436 is the main D2D Unix process ID and 4511 is the warnGenWish process ID).

The logs for the text display are only on the HP workstations. For example, on ws1 at site LWX, they would be in /data/logs/fxa/display/xt1-lwx:0. This contains text display log data from the local HP workstation and also from any associated Linux client workstation. To determine which HP workstation serves a given Linux workstation, do the following. If we are using lx1 at site LWX, look at file /awips/fxa/data/.environs.lx1-lwx. Then use:

```
grep FXA_WARNGEN_PRODUCT_ID .environs.lx1-lwx
```

You'll get output like:

```
FXA_WARNGEN_PRODUCT_ID WRKWG3
```

The "WG3" indicates that ws3 is the client for lx1.

If the normal logs don't reveal enough information, you can enable verbose and debug logging for WarnGen. As user fxa, edit the file /awips/fxa/data/displayLogPref. Normally, the environment variable LOG_PREF contains this path name, so use "vi \$LOG_PREF". To turn on debug and verbose logging for WarnGen, add the following two lines at the end of the file:

```
warnGen          all    file    all = on
warnGenWish      all    file    all = on
```

Now restart D2D and WarnGen to see the verbose logs. When done, be sure to turn off the extra logging by commenting out the above entries:

```
#warnGen          all    file    all = on
#warnGenWish      all    file    all = on
```

5.2.2 WARNGEN AND THE WORKSTATION TEXT DISPLAY

In WarnGen, a warning is initiated using the graphics display. When the 'create text' button is pushed, a text window is supposed to automatically pop up on the text display so the forecaster can edit the product and disseminate it. If the text window does not pop up quickly, this is very bad news. To help the text window come up faster, several suggestions are available in the SST handout "AWIPS check list for improved system performance during severe weather." This document is available at the following web site http://www.ops1.nws.noaa.gov/awips_install.htm. We suggest that sites perform these actions before anticipated severe weather. Here are suggestions to speed up the text and graphics workstations:

1. Close all windows and programs running on the text workstation. Log out of the text workstation and power it off. Power it back on and log back on.
2. Exit out of the Monitoring and Controller window. This should not be running on your warning workstation.
3. Make sure that the Update Obs feature is not turned on in any of the text windows.
4. Do not run any other local applications on your text workstations.
5. Limit the number of looping frames in your D2D windows. If possible, do not loop products in the small panes.
6. Limit the number of alerts you send to that text workstation.

If the text window does not pop up at all, the forecaster can manually call up the new warning and use text editor to edit and disseminate the product. The PIL would normally be a WRK product such as WRKWG# where # is the workstation number.

If the text window does not pop up at all, here are some troubleshooting tips. If this happens on all workstations and other workstation applications seem OK, there likely is a problem with the textNotificationServer on as1. If this happens only on one workstation, that workstation may not be registered with the textNotificationServer, or there may some kind of workstation localization problem. See the “Text Notification Server Troubleshooting Section” on page [82](#) for more details.

If all looks OK with the textNotificationServer, and the text window still won't pop up, then there could be some obscure localization problem. In this case, try to localize the workstation using “mainScript.csh f” (no tasks included, just the “f” option). This will do a simple workstation localization. If this still doesn't help, you may need to pursue the draconian approach of copying the entire localization output directory (/awips/fxa/data/localizationDataSets/LLL) from a good workstation (the “Localization Troubleshooting” section on page [18](#) has details).

As far as we know, there are no plans to include text editing capability within WarnGen, so it looks like the separate text editor window outside of WarnGen will continue.

5.2.3 TEXT NOTIFICATION SERVER TROUBLESHOOTING

To check the textNotificationServer do the following. Log onto as1 as fxa. Use “ps -ef | grep -i noti” and verify that process textNotificationServer is running. This checks for new text products and automatically does certain things when a new text product is available. Beware that there is also a plain notificationServer which alerts the workstation when new graphics products become

available. When WarnGen creates a new local product, textNotificationServer automatically opens a text edit window on the text display for the new product. The NCF alarms the textNotificationServer process and automatically restarts it if it is not running, so normally this is not a problem.

If textNotificationServer is not running, stop and restart the textDB processes on as1. To do this, log onto as1 as user fxa:

```
cd /awips/fxa/bin
./stopTextDB.as1 (ignore error saying wrong # of parameters)
./startTextDB.as1
```

This also stops and starts the asyncScheduler on as1. Now process textNotificationServer should be running on as1.

Now each workstation likely needs to be re-registered with the textNotificationServer. The site should log off the text display and log back on as user "textdemo". Normally this will re-register the workstation and a new WarnGen text product should automatically pop up on the text display. The site should not need to exit and restart D2D.

If there are still problems with the textNotificationServer, here are some troubleshooting tips. You can check the textNotificationServer log on as1 to see if things are OK. For example, if no text window pops up when the WarnGen "create text" button is pushed on ws4 at site ABR, check to see when the last text product was created on ws4-abr and check if this product appears in the textNotificationServer log.

First log onto ws4 as user fxa. We need to get the AFOS PIL to know what product to look for in the log. In /awips/fxa check the dot file ".environs.ws4-abr" Look at field "FXA_WARNGEN_PRODUCT_ID" In this case, the AWIPS ID for products created on this workstation is WRKWG4. Now we can look in the AWIPS text database to see when the last WRKWG4 product was stored.

Use "textdb -r WRKWG4" to display the latest product generated by ws4. In this example, at the top of the product, note the PIL "FSDSVRABR". The AFOS node for site ABR is FSD. It is also handy to see all recent text products generated by ws4. Use "textdb -A FSDWRKWG4" to get this list. In this example, the latest FSDWRKWG4 was a FSDSVRABR at Jan 30 2002 at 10:20:33 GMT.

Now we can check the textNotificationServer log on as1. Log onto as1 as user fxa and enter "logs" to go to the log directory (in this case /data/logs/fxa/20020130). Look at log file "textNotificationServer*" Use "grep FSDWRKWG4 textNoti*" to look for the ws4 product.

The following log entry shows that product FSDWRKWG4 was last received at 10:20:33. The

textNotificationServer seems to be recognizing ws4 products OK.

```
10:20:33.119 TextNotificationServer.C EVENT: Received text notification for:
FSDWRKWG4
```

Now check if ws4 is registered properly with the textNotificationServer. Look at the as1 configuration file /data/fxa/workFiles/textNotificationServerClientList.txt. Each workstation must be listed here. In this case there were no workstations listed so the textNotificationServer was finding the ws4 products, but couldn't send them to the text display. To fix this, have the site log off the ws4 text display and log back on as user "textdemo". After this, ws4 should show up in file textNotificationServerClientList.txt with an entry like this:

```
ws4-abr/2674/22166
```

The last number should be the process ID of the process "textAlarmQueue.tcl" on ws4. If the ws4 entry doesn't appear, you may need to have the site reboot the entire workstation or maybe restart the text notification server on as1.

The process textAlarmQueue.tcl should be active whenever user textdemo is logged onto the text workstation. textAlarmQueue.tcl handles new product notifications, including handling and display of alarm and alerts on the workstation. textAlarmQueue.tcl is invoked by textWS.tcl.

A problem occurred in which the WarnGen text window would not display on text workstation xt5-pah and also no alarms/alerts were received on the workstation. The textAlarmQueue.tcl started up when user textdemo logged on, but quickly died. The problem was with the configuration file /data/fxa/textWSwork/xt5-pah:0/textAlarmAlertProducts.txt. To fix the problem, we copied the xt4 version of the textAlarmAlertProducts.txt file to the xt5 directory. In the same directory, the file textPAProducts.txt may also be involved in the textAlarmQueue configuration.

5.2.4 AWIPS DISSEMINATION OF WARNGEN PRODUCTS

After the forecaster presses the "send" button in the text editor to disseminate the WarnGen product, lots of things happen in AWIPS to process the warning. The text editor sends the product to the AWIPS WAN for dissemination and saves the product in the local text database. Also some processing is initiated by Informix triggers when the product is stored in the local Informix text database.

One trigger likely will send the product to the Console Replacement System (CRS), i.e, send the product for transmission on the NOAA Weather Radio. Another trigger may send the warning to the LDAD fax package for fax dissemination. Other triggers may send the warning to the LDAD for other possible dissemination or processing, to local applications or to the AWIPS async scheduler.

The AWIPS Message Handling System (MHS) controls the AWIPS WAN. A useful log file is `/awips/ops/logs/ds1-LLL/x400sc.log` where “LLL” is the site ID in lower case. This shows all products that are sent from the site to the AWIPS WAN. These go to the NCF (in the log it says it is sent to “DEFAULTNCF”) for dissemination on the AWIPS SBN to all AWIPS sites. Another useful MHS log file, `x400rd.log`, shows all products received at the site from the AWIPS WAN.

Also, the MHS sends products to the NOAA Weather Wire System (NWWS). In the `x400sc.log` file these are sent to the NWWS primary and secondary uplink sites. The NWWS uplink sites for each local site are defined on `ds1` in file `/awips/ops/data/mhs/nwwsup_dlist.data`. At each site, products are by default all sent to the NWWS. If a product is not to be sent to the NWWS, it must be listed in the file `/data/fxa/workFiles/wanMsgHandling/NWWS_exclude_LLL.txt` (where LLL is the site ID).

At the site creating the warning, in the log file `/awips/ops/logs/ds1-LLL/x400sc.log`, you should see the warning sent to the Weather Wire primary and secondary uplink sites. At the NWWS uplink sites, the `ds1` executable `NWWSProduct` and the `as1` executables `NWWSchedule` and `NWWSTransmit` perform the uplink. At the NWWS uplink sites, the log files `as1:/data/logs/fxa/YYYYMMDD/NWWSchedule*` indicate when products are uplinked (that is, sent to the Calypso box).

To see products received from the SBN, you need to look at the `acqserver` log files in the files `/data/logs/fxa/YYYYMMDD/acq*`. The file `/awips/fxa/data/acq_patterns.txt` controls the local processing of files received from the SBN.

Most products received at a site come from the SBN. As a backup, warnings also are disseminated to sites via the AWIPS WAN. So sites should see the same warning listed both in the `x400rd.log` MHS log file and in the SBN `acqserver` logs. An exception are locally issued warnings. Locally issued warnings should be seen in the `acqserver` logs and the `x400sc.log` (MHS send log). The warning products, however, should be stored only once in the text database. This is because warning products received via the SBN and the WAN are processed by the `WarnDBDecoder` which checks for duplicate warnings. The `WarnDBDecoder` checks all SBN and WAN warnings byte for byte against warnings already stored in the text database. For more on this topic see the duplicate storage section beginning on page [88](#).

Also, the text editor archives the warning as a flat file on `ds1` in directory `/data/fxa/archive/OUP/archive`. Today’s products reside in individual files. Products from past days are put into one tar file for each day. The products are archived here for 30 days then purged.

`WarnGen` products can be sent to various applications by means of triggers in the Informix text database. The triggers are defined in the file `ds1:/awips/fxa/informix/fxatextTriggerActions.txt`. To see if the triggers are actually active in the Informix database, do the following as user “`fxa`”

on ds1:

```
cd /awips/fxa/awipsusr
ksh
dbaccess fxatext << !
> unload to 'dummy.txt'      (use any temporary file name except test)
> select * from watchWarn
>!
```

CRS items often are sent to the CAFÉ formatter for translation into a file used by the CRS automated voice system. A trigger entry with an item like “/home/CRS/NNN/nwrnnn.csh” (where NNN and nnn are the upper and lower case product code - from the CCCNNNXXX in the WMO heading) may indicate a product being sent to the CAFÉ formatter. The products that are sent to the NOAA Weather Radio are in directory ds1:/data/fxa/workFiles/nwr/sent. There are also MDL developed NOAA Weather Radio formatters which are considered the baseline AWIPS formatters.

WarnGen products can also be sent directly to the NWR browser. When products are sent to the NWR browser, then can be automatically transmitted or they can be put in a NWR browser pending area. Products in the pending area can be edited for NWR voice pronunciation problems, then transmitted.

Informix triggers are used to send products to LDAD for external dissemination. They can be sent to the textdbNotify process on ds1. The log file is ds1:/data/logs/fxa/YYYYMMDD/textdbNotify*. All products sent to LDAD and received from LDAD will be logged in file ds1:/data/logs/ldad/YYYYMMDD/listener*. Triggers can also send products to local application programs for processing. Products may be sent to the AWIPS async scheduler on as1, which sends products to local hosts via a serial line at the AWIPS site. The async scheduler log is in as1:/data/logs/fxa/YYYYMMDD/asyncScheduler*.

It is theoretically possible to specify that certain products be disseminated only to certain sites via the AWIPS WAN. The SST has not verified this. The file /data/fxa/workFiles/wanMsgHandling/siteDistList.txt file allows you to define which sites should receive certain products (see the OB2 System Manager’s Manual, section 14.2.1). For example, the following entry in siteDistList.txt:

```
AMBR  LOT,LSX,DVN,PAH
```

says that the AMBR product should only be disseminated to site LOT, LSX, DVN and PAH. In this case, the AMBR product is sent directly to the other sites via the AWIPS WAN, the NCF is bypassed, and the product is not broadcast on the SBN.

Text products that are not created by the text editor (such as the HWR and CLI) are disseminated

by programs handleOUP and distributeProduct. “OUP” means “Official User Product.” handleOUP stores the official version of the product in the local text database using the “textdb -w” command. Then handleOUP calls the distributeProduct executable to disseminate the product on the AWIPS WAN. If the product is transmitted on the NWS, handleOUP will call distributeProduct again to send the product to the MHS for dissemination to the NOAA Weather Wire uplink sites.

There are some useful handleOUP and distributeProduct log files. If these processes were invoked from the ds or as, the logs are in directory /data/logs/fxa/<YYYYMMDD> where YYYYMMDD is the year, month, day. The log file names are distributeProduct and handleOUP.log. If these processes were invoked from a workstation, the logs are in /data/logs/fxa/display/<DISPLAY>/<YYYYMMDD> where DISPLAY is defined by the DISPLAY environment variable.

5.2.5 TRACING WARNGEN PRODUCTS ON THE AWIPS WAN, SBN

This section describes how to trace AWIPS products from the AWIPS text editor, to the text database, MHS, AWIPS WAN and SBN. These can help diagnose various product dissemination problems and duplicate text storage problems. The logs need to be examined soon after the error occurs, since most logs are purged after one or two days.

It’s good to start out by verifying what products are actually stored in the text database. If we’re checking a TORGSP, use “textdb A CAETORGSP” to get a summary of all the TORGSP products that are stored in the text database. Use “textdb -r TORGSP” (if you are logged on to site GSP) to see the latest TORGSP. To see the previous TORGSP, use “textdb -r -1:TORGSP”.

If a product of interest has been purged from the text database, they can be found in the OUP archive in ds1:/data/fxa/archive/OUP/archive. Today’s products are in individual files. The previous 30 days are in tar files. To see a tar file contents (for example file arch_27.tar), copy the tar file to an empty scratch directory, then do “tar xvpf arch_27.tar.” The products from the 27th of the month will be extracted.

To verify that a product was sent to the AWIPS WAN and stored in the text database, look at the workstation text display log. For example, to look for product TORGSP that was created on text workstation one (on April 21, 2003), look in log files /data/logs/fxa/display/xt1-gsp:0/20030421/textWish*. The following entry shows that the TOR was sent to the WAN and stored in the text database at 21:45 UTC:

```
textWish 5553 1050961537.224858 21:45:37.224 EVENT: WFUS52 KGSP 212145  
[CAETORGSP] sent to DEFAULTNCF,NWWSUP via MhsServer  
textWish 5553 1050961544.215366 21:45:44.215 EVENT: WFUS52 KGSP 212145  
[CAETORGSP] written to db
```

We can see the local version of the product stored in the text database using the TextDB_Server logs on ds1 in files /data/logs/fxa/20030421/TextDB_Server* files. There are two TextDB_Server log files, the larger is the write log, the smaller is the read log. Look in the write log for the CAETORGSP and you should see a socket connection closed message from the workstation where the product was created. The product should be stored immediately before the display log says it was “written to db” with a log entry something like this:

```
21:45:44.163 TextDB_Server.C EVENT: CAETORGSP WFUS52 250227 8018 0.252
21:45:44.166 DataSocket.C DEBUG: Connection to ws1-gsp/2778/10640 has been closed by the peer.
21:45:44.168 DataSocket.C DEBUG: Closing the socket to ws1-gsp/2778/10640
```

We can see the TOR sent out to the WAN in the outgoing MHS logs /awips/ops/logs/ds1-LLL/msgreq_svr.log and most importantly in /awips/ops/logs/ds1-LLL/x400sc.log where “LLL” is the site ID in lower case. The x400sc.log file is for today, the x400sc.old file is for yesterday. The x400rd* logs show products received via the WAN. The outgoing product should show in the x400sc* file immediately after the display log says it was “sent to DEFAULTNCF” with a log entry something like this:

```
Mon Apr 21 21 21:45:37 2003 <x400sc:12245> MTA submit OK
GSP-949284: Routine to=DEFAULTNCF, from=GSP, pri=2 code=4 enc=1 len=915
GSP-949284: WFUS52 KGSP 240735 TORGSP (/data/x400/msgtbl/GSP-949284.msg)
```

The TORGSP should not appear in the x400rd log at GSP, since the NCF does not send it back to the site via the WAN. The TORGSP, however, should appear in the x400rd log at all other sites than GSP, since the NCF sends warnings to all sites via both the WAN and SBN.

The TORGSP is received back at GSP from the SBN. The file ds1:/awips/fxa/data/acq_patterns.txt is used to do Unix pattern matching on the WMO heading (WFUS52) to determine what to do with the product. The TORGSP appears in the acqserver logs on ds1 in files /data/logs/fxa/20030421/acqserver*. The acqserver logs only list the product by WMO header. The TORGSP should appear in the acqserver log twenty seconds or less after it was transmitted from the site. The log entry looks like this:

```
acqserver 22290 21:45:49.173 EVENT: NCF_ENTRY: WFUS52 KGSP 250709 Cat: TEXT 2045293
#44605024
acqserver 22290 21:45:49.174 EVENT: NCF_STORE: WFUS52 KGSP matched NO_STORE
```

The following WarnDBDecoder log entry shows that the TOR was processed after being received from the acqserver. The log file is as2:/data/logs/fxa/20030421/WarnDBDecoder*:

```
21:45:49.663 StdTextDecoder.C EVENT: NCF_ENTRY WFUS52|KGSP|212145
21:45:49.665 StdTextDecoder.C EVENT: Duplicate check on CAETORGSP
21:45:50.721 StdTextDecoder.C EVENT: Not duplicate (data differs)
21:45:50.723 TextDB_Decoder.C EVENT: NCF_STORE CAETORGSP
```

We see that about 13 seconds after the TOR was sent to the WAN, it had come back to GSP via the SBN and the WarnDBDecoder duplicate check failed. The example shows that the product received via the SBN is different than the product that was first stored in the text database. This is bad, because the two copies of the product are supposed to be identical. The WarnDBDecoder log contains StdTextDecoder messages because the WarnDBDecoder and the StdDBDecoder are two separate executables, but they basically share the same source code. There are separate logs for the WarnDBDecoder and the StdDBDecoder. The StdDBDecoder processes non warning text products.

An item of interest is a test message that is sent every minute from the NCF to test throughput on the WAN and SBN. This is product NCFTSTNCF NTUS99 KNCF. It can be used in the StdDBDecoder log to see how long it takes to store products in the text database. The log file is `as2:/data/logs/fxa/YYYYMMDD/StdDBDecoder*`. If you use “`grep NTUS99 StdDBDecoder*|more`” there will be entries like:

```
00:20:33.195 StdTextDecoder.C EVENT: NCF_ENTRY NTUS99|KNCF|260020
```

The end of the line shows when the NTUS99 was sent from the NCF (here 00:20Z). The beginning of the line shows when it was stored in the text database (here 00:20). In this example there is less than a one minute delay, so that is good. Another similar test message is NCFWTSNCF NTUS96 KNCF.

5.2.6 DUPLICATE STORAGE OF WARNGEN PRODUCTS IN THE TEXT DATABASE

When a site uses WarnGen to issue any product (for example a TOR), the text editor (script `textWindow.tcl` is the driver which controls the text editor) sends the TOR (using procedures `wan_send_msg_wmo` or `wan_send_msg` in file `textWanWmo.tcl`) to the MHS for distribution on the AWIPS WAN to the NCF. `textWindow.tcl` then stores the TOR in the text database (using procedures `wan_write_DB_wmo` and `wan_write_DB_afos` in file `textWanWmo.tcl` and also uses `writeTextDB.tcl`).

The NCF then disseminates the TOR to all sites via the SBN. Because the TOR is a warning, the NCF also disseminates it to all sites via the WAN (except that the NCF does not send the TOR back to the originating site). Therefore, the original site gets another copy of their TOR from the SBN. The WarnDBDecoder does the duplicate product check, and the extra copy of the TOR should not be stored in the text database. All other sites (sites that did not issue the TOR) receive two copies of this same TOR - one from the SBN and one from the WAN. The WarnDBDecoder performs the duplicate check and only one copy of the TOR should be stored in the text database. It doesn't matter whether the SBN copy of the TOR or the WAN copy arrives first, the duplicate check should work regardless.

However, there are cases in which the duplicate product check fails which causes the same

warning to be stored twice in the text database. Sites often notice this when the CRS tone alerts the same product twice (since it is stored twice in the text database). What causes this problem is that somehow the SBN version of the product is different than the local version stored by the text editor into the text database. This difference causes the as2 WarnDBDecoder duplicate check to fail. A number of things could cause the local version and the SBN version of the products to be different as the product traverses the complex path from the AWIPS text editor, through the MHS, WAN, NCF, NWSTG, SBN and back to the AWIPS site.

DR12548 covers the problem where the local version of text products is stored before the SBN version and for cases of non local warnings (copies of the warning are received on both the WAN and SBN). The fix (included in OB2) is an improved dup check algorithm in the WarnDBDecoder and StdDBDecoder. The new dup check algorithm works only for products ending with "\$\$". Warnings are required to end with "\$\$" and non warnings are not too big a problem.

DR12745 covers the problem where the storage of locally produced text products in the textDB is delayed until after the SBN version of the product arrives back at the site. In this case, the SBN version is stored first, then the text editor uses script writeTextDB to invoke "textdb -w" command to store the product in the local textDB. The problem is that sometimes the "textdb -w" dup check algorithm fails. To fix this, OB3 is scheduled to have an improved "textdb -w" duplicate checker.

Duplicate storage of handleOUP products (non text editor products) has been caused when the product is issued just before a new minute rolls over on the system clock. The products are time stamped to the nearest minute. It is possible for handleOUP to store a product in the text database at 10:50 and 59 seconds (with a time stamp of 10:50 UTC). It is then possible for distributeProduct to send the product to the WAN at 10:51 and 00 seconds (with a time stamp of 10:51 UTC). As a result, the same product has two different time stamps and the duplicate check fails when the product is comes back via the SBN. This problem does not exist if AWIPS software applications use the option "-d ddHHMM" to invoke handleOUP and distributeProduct. This allows the time stamp to be specified so that all versions of the same product have the same time stamp.

A similar duplicate storage problem was noted for SWR products in 5.2.2 (DR 11736) where distributeProduct required every product to end with an end of line character (carriage return or line feed). If the last line of the product did not have the end of line character, the last line of the product was dropped before being sent to the WAN, so the subsequent duplicate check failed. In 5.2.2 there was a duplicate check problem when the AWIPS product ID was less than 6 characters long. The AWIPS product ID was supposed to have extra blanks appended so that it was 6 characters long, but the extra blanks were not appended. For example, the second line of the heading for products SFPOH and SWRUT are supposed to have a blank appended. This was fixed in 5.2.2.1.