# Warngen Template Customization

WarnGen? templates for the AWIPS II system are \*.vm files written in the **Velocity Template Language (VTL)** with supporting configurations in an xml file with \*.cfg suffixes. Velocity and the VTL is released under the Apache Jakarta Project and thus the current documentation can be found on the Apache Jakarta Project website. The following URLs are direct links to the web-based VTL reference.  http://svn.apache.org/repos/asf/velocity/engine/tags/V\_1\_0\_1/docs/vtl-referenceguide.html

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

**Configuration Values in WarnGen? \*.cfg files.**

*Values in italics are system level settings and should only be modified if necessary*

* polygonShape: Use “1” for the pathcast shape based on storm track or “2” for a square
* followups: Defines the options which can appear in the “UPDATE LIST” dropdown
* phenomena: Defines which Phenomena this template can create Followup products for.
* Significance: Defines which Significance this template can create Followup products for.
* DefaultDuration?
* Durations
* Bullets
  + bulletText: Defines the text displayed in the WarnGen? dialog.
  + bulletName: Defines a name to refer to the bullet as in the template code.
  + bulletGroup: Provides a mechanism for limiting selections in WarnGen?. Only one item per bulletGroup can be selected so related items should list the same bulletGroup.
  + bulletType: bulletType=”title” makes the bullet unselectable, bulletType=”basin” correlates the bullet to a geometry of the same name in the customlocations table.
* pathcastConfig enabled: “1” will enable the Storm Track, “2” will disable the Storm Track
  + defaultSpeedKt: Default Storm Track speed in knots
  + defaultDirection: Default Storm Track direction
  + overThreshold: Distance for a city or POI to be considered “over” the Storm Track
  + nearThreshold: Distance for a city or POI to be considered “near” the Storm Track
  + lineofStormsDistance: The distance in km between the 2 original storm points when Line of Storms is selected.
  + lineofStormsAzimuth: The azimuth between the 2 original storm points when Line of Storms is selected.
  + areaSource: Relation to a table in the maps database
  + pointField: Relation to a field name in a table in the maps database
  + pointFilter: A filter when making a query to the map database. This can be used to include or exclude points of interest with specific WarngenLev? values.
  + areaNotationField: Relation to a field name in a table in the maps database
  + areaField: Relation to a field name in a table in the maps database
* fipsField: Relation to a field name in a table in the maps database
* parentAreaField: Relation to a field name in a table in the maps database
* timezoneTable: Relation to a table in the maps database
* timezoneField: Relation to a field name in a table in the maps database
* areaNotationTranslationFile: File noting the difference between “county” and “parish”
* areaConfig
* inclusionPercent: The minimum percentage for a county or zone to be included in a warning
* inclusionAndOr: Determines which of the inclusion Area and Percentage are used for inclusion
* inclusionArea: The minimum area in square kilometers for a county or zone to be included in a warning
* pointField: Relation to a field name in a table in the maps database
* pointFilter: A filter when making a query to the map database. This can be used to include or exclude points of interest with specific WarngenLev? values.
* fipsField: Relation to a field name in a table in the maps database
* parentAreaField: Relation to a field name in a table in the maps database
* timezoneField: Relation to a field name in a table in the maps database
* closestPointsConfig
* numberOfPoints: This configuration sets the number of “Closest Points” which will be used in the first bullet. Any point included in the “Closest Points” variable will be excluded from the “Other Points” variable so that a location is not repeated.
* pointFilter: A filter when making a query to the map database. This can be used to include or exclude points of interest with specific WarngenLev? values.
* geospatialConfig
* pointSource: Relation to a table in the maps database
* areaSource: Relation to a table in the maps database
* parentAreaSource: Relation to a table in the maps database
* maskSource: Relation to a table in the maps database
* maskFilter: Filter used to exclude geometry outside of the local CWA
* basinConfig
* maskSource: Relation to a table in the maps database
* bulletColumn: Relation to a field name in a table in the maps database
* maskFilter: Filter used to exclude geometry outside of the local CWA

**Values passed to Warngen \*.vm templates**

* vtecOffice – is the 4 letter id set by the logic in “getSite4LetterId” shown above
* siteId – String: The 3 letter WFO set in Localization
* officeShort – String set in site localization config.xml
* officeLoc – String set in site localization config.xml
* EX: officeShort: “OMAHA/VALLEY NE” officeLong: “OMAHA”
* backupSite – The officeLoc value for a site that WarnGen? is backing up via backup mode.
* localtimezone – Time Zone where the warning starts in single letter format
* secondtimezone – If the Warning spans multiple time zones, the second Time Zone is stored here
* stormType – “line” if it is a line of storms, “single” otherwise
* now – Date value representing now
* start – Date value representing the start time
* expire – Date value representing the expire time
* event – Date value representing the event time
* fipsline – String representing the UGC Header
* fipslinecan – String representing a UGC Header
* areaPoly – String representing the polygon
* movementInMph – double representing movement speed
* movementDirectionRounded – double representing movement direction
* movementDirection – double representing movement direction
* movementInKnots – double representing movement speed
* action – String representing the VTEC Action
* oldvtec – String representing the old VTEC tracking number for followup products phenomena – String representing the VTEC pp field
* mode – String representing test mode if applicable
* bullets – Array of Strings representing the bullets selected in Warngen
* eventLocation – array of Point2D values (java.awt.geom.Point2D) – Do not edit
* otherPoints – array of Strings representing other points impacted by the storm – Do not edit
* pathCast – PathCast? value (com.raytheon.viz.warngen.gis.PathCast?) – Do not edit
* closestPoints – array of ClosestPoints? (com.raytheon.viz.warngen.gis.ClosestPoint?) – Do not edit
* areas – array of AffectedAreas? (com.raytheon.viz.warngen.gis.AffectedAreas?) – Do not edit
* cancelareas – array of AffectedAreas? (com.raytheon.viz.warngen.gis.AffectedAreas?) – Do not edit

timeFormat – Hashtable containing 6 SimpleDateFormats?

|  |  |
| --- | --- |
| Hashtable key | Hashtable Value |
| header | hhmm a z EEE MMM d yyyy |
| plain | hhmm a z EEEE |
| clock | hmm a z |
| ymdthmz | yyMMdd'T'HHmm'Z' |
| ddhhmm | ddHHmm |
| Time | HHmm |

• list – ListTool? (org.apache.velocity.tools.generic.ListTool?) Method use examples: $primes is an array of integers containing {2, 3, 5, 7} $lists.size($primes) -> 4 $lists.get($primes, 2) -> 5 $lists.set($primes, 2, 1) -> (primes[2] becomes 1) $lists.get($primes, 2) -> 1 $lists.isEmpty($primes) -> false $lists.contains($primes, 7) -> true

• mathUtil – WarnGenMathTool? (extending

org.apache.velocity.tools.generic.MathTool?) Methods: ( http://velocity.apache.org/tools/devel/javadoc/org/apache/velocity/tools/generic/MathTo ol.html) $mathUtil.roundTo5(num) $mathUtil.roundToInt(num, multiple) $mathUtil.abs(num) $mathUtil.add(num1, num2) $mathUtil.ceil(num1) $mathUtil.div(num1, num2) $mathUtil.floor(num1) $mathUtil.getAverage(nums) $mathUtil.getRandom() $mathUtil.Total(nums) $mathUtil.matchType(num1, num2) $mathUtil.max(num1, num2) $mathUtil.min(num1, num2) $mathUtil.mod(num1, num2) $mathUtil.mul(num1, num2) $mathUtil.pow(num1, num2) $mathUtil.random(num1, num2) $mathUtil.round(num1) $mathUtil.roundTo(decimals, num2) $mathUtil.sub(num1, num2) $mathUtil.toDouble(num1) $mathUtil.toInteger(num1) $mathUtil.toNumber(num1)

• dateUtil – DateUtil? (com.raytheon.viz.warngen.util.DateUtil?) Methods $dateUtil.format(Date, DateFormat?) $dateUtil.format(Date, DateFormat?, Interval) $dateUtil.format(Date, DateFormat?, Interval, TimeZone?) $dateUtil.format(Date, DateFormat?, TimeZone?)

/\*\*

Converts a 3 letter site ID into a 4 letter ID, e.g. OAX to KOAX

@param site3LetterId the 3 letter site id

@return

\*\*/

public static String getSite4LetterId(String site3LetterId) {

// this code was ported from legacy GFE

if (site3LetterId.equals("SJU")) {

return "TJSJ";

} else if (site3LetterId.equals("AFG") || site3LetterId.equals("AJK") || site3LetterId.equals("HFO") || site3LetterId.equals("GUM")) {

return "P" + site3LetterId;

} else if (site3LetterId.equals("AER") || site3LetterId.equals("ALU")) {

return "PAFC";

} else {

return "K" + site3LetterId;}}

}

How to access the data contained in a value in a template.

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

String or primitive type: ${myStringValue}

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

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

In the above context, dateUtil is a com.raytheon.viz.warngen.util.DateUtil? Object which contains the method format, which accepts 3 arguments. Additionally, ${pc.time} (A date value from the pathCast) is replaced by the time attribute of the pc object, ${timeFormat.clock} is replaced by the SimpleDateFormat? addressed in the timeFormat Hashtable by “clock,” and ${localtimezone} is replaced by the localtimezone object. Template Modifications

Plain text in Warngen templates can be freely edited as needed. When modifying the Velocity Template Language careful attention should be paid to the syntax so that errors are not introduced. Any line which begins with a pound (#) symbol or any variable surrounded by ${} could potentially be dangerous to alter.

Modifying Bullets in Warngen Bullets can be added to warngen by editing cave/etc/warngen/severethunderstorm.cfg. The <bullets> tag contains several existing bullets and additional bullets can be added by following the basic pattern <bullet bulletName=”x” bulletText=”y”> in the proper order. After the bullet has been successfully added to severethunderstorm.cfg, the cave/etc/warngen/severethunderstorm.vm file must be modified to take that bullet into account.

When adding a Call To Action a statement similar to Example #4 in the following section should be followed. First check if the list contains the bulletName using an if statement. Insert the text for the Call to Action on the following line, followed by an empty line. Finally, finish the if statement with a #end.

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

Example #1:

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

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

Example #2:

${officeLong} HAS ISSUED A

The above example will output “THE NATIONAL WEATHER SERVICE IN OMAHA HAS ISSUED A” if officeLong is set correctly in caveData/configuration/site/OAX/com.raytheon.viz.warngen/config.xml Example #3:

#if(${list.contains($bullets, "doppler")})

#if(${stormType} == "line")#set ($report = "NATIONAL WEATHER SERVICE DOPPLERRADAR INDICATED A LINE OF SEVERE THUNDERSTORMS")

#else

#set ($report = "NATIONAL WEATHER SERVICE DOPPLERRADAR INDICATED A SEVERETHUNDERSTORM")

#end

#end

T he above example shows a compound control structure. If the bullets list sent from warngen contains the “doppler” bullet (as indicated by the bulletName in cave/etc/warngen/severethunderstorm.cfg) the inner if statement is reached. At that point if the stormType is line then the report text is set to a message indicating a line of storms. Otherwise the report text is set to a message indicating a single storm. Finally, the first #end ends the stormType if statement and the second #end ends the list.contains if statement.

Example #4:

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

#end

This example is a typical call to action line and should be replicated for any additional Call to Action lines. The line beginning with #if determines if the torWatchRemainsInEffect bullet was highlighted in Warngen. If it is, the contents of the testMessage variable are printed followed by the next three lines of text. The #if statement ends at the #end line.

Example #5:

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

#if(${city.roundedDistance} < 3) ## close enough to not need azran, considered OVER the area${city.name}

#else ## needs azran information

#${city.roundedDistance} MILES

#direction(${city.roundedAzimuth}) OF ${city.name}

#end

# end

This example shows a foreachloop. pc.pointsis an array of cities affected by the pathcast of the storm. These objects should not be modified, but they can be used and displayed.

The loop covers the first line of the example until the final line for each item in the array pc.points. Each item in this array is temporarily renamed to citywhen it goes through the rest of the template. The #if statement checks if the roundedDistanceof the city (which determines how far the city is from the pathcast) is less than 3 miles. If this is true the cities name is printed. Otherwise, the roundedDistanceis printed to indicate in the warning that the storm is roundedDistancemiles in roundedAzimuthdirection of the city. Notice that the line beginning with two pound symbols (##) is a comment. These can be used anywhere within a template.

Top of Form

Bottom of Form