Velocity Overview

Office of Operational Services (OOS)
04/24/2008

This document is only valid for TO-8 Release of AWIPSII.
It will have to be updated as new information comes available and with subsequent releases. It should be considered for informational purposes only.

Velocity Overview 1
Introduction: 1
Description: 1
Code Notes: 4
Task List 5
Comments: 5

Introduction:

The new template system for AWIPSII warngen is a complete rewrite of the software. The new software is implemented using the Velocity Template Language (VTL). VTL is a scripting language with syntax similar to javascript which contains variables, logic, and text. The text in the template remains static while the variables are inserted and the logic processed at runtime.

AWIPS 1 currently has 19 baseline templates and many custom, localized templates. The template language for AWIPS1 is not compatible with VTL. The architecture of the AWIPSII implementation of WarnGen is so different that a direct port of the existing templates in not practical.

Description:

The AWIPS II implementation of WarnGen consists of three major components. These components are:

- Configuration files (one for each template)
- Template files (one for each warning type)
- Java Code (WarngenActions.java)

And a few helper files:

- countyTypes.txt
- stateAbbreviations.txt
- VM_glogal_library.vm

File Descriptions:

tornado.cfg

Controls the display of the WarnGen gui. Sets the labels for the text boxes, select boxes, and selectable lists.

Code Snippet:

```
<!-- Tornado Warning configuration -->
<warngenConfig>
 <!-- durations: the list of possible durations of the warning -->
 <durations>
  <duration>10</duration>
  <duration>15</duration>
  <duration>20</duration>
…
  <duration>90</duration>
</durations>
 <!-- bullets: User specified text generation options
          bulletName: an id that is passed to the template when a bullet
                      is selected
          bulletText: the text presented to the user -->
 <bullets>
  <bullet bulletText="*********** BASIS FOR WARNING (CHOOSE 1) **********"/>
  <bullet bulletName="doppler" bulletText="Doppler radar indicated"/>
  <bullet bulletName="trainedSpoters" bulletText="Trained spotters reported"/>
  <bullet bulletName="public" bulletText="public reported"/>
  <bullet bulletName="lawEnforcement" bulletText="local law enforcement reported"/>
  <bullet bulletText="*********** THREAT (CHOOSE 1 each wind/hail) **********"/>
  <bullet bulletName="60mphWind" bulletText="60 mph wind"/>
  <bullet bulletName="70mphWind" bulletText="70 mph wind"/>
  <bullet bulletName="80mphWind" bulletText="80 mph wind"/>
  <bullet bulletName="pennyHail" bulletText="Penny size hail"/>
  <bullet bulletName="nickelHail" bulletText="Nickel size hail"/>
  <bullet bulletName="quarterHail" bulletText="Quarter size hail"/>
  <bullet bulletName="golfballHail" bulletText="Golf ball size hail"/>
  <bullet bulletName="baseballHail" bulletText="Baseball size hail"/>
  …
</bullets>
```

tornado.vm

The actual template that contains text for the products, variables to be inserted from the data, and logic to control the formatting of the text product generated by WarnGen's velocity engine.

Code Snippet:

```
ZCZC ${tornadoId} DEF
TTAA00 ${wmoValue} ${dateUtil.format(${now}, ${timeFormat.ddhhmm})}
/O.NEW.${vtecOffice}.TO.W.0001.${dateUtil.format(${start}, ${timeFormat.ymdthmz})}-
${dateUtil.format(${expire}, ${timeFormat.ymdthmz})}/

BULLETIN - EAS ACTIVATION REQUESTED
#if(${mode}=="test" || ${mode}=="practice")
TEST...TORNADO WARNING...TEST
#else
TORNADO WARNING
#end
<!-- NATIONAL WEATHER SERVICE ${officeShort} -->
${officeLong}
${dateUtil.formatLocal(${now}, ${timeFormat.header})}

#if(${mode}=="test" || ${mode}=="practice")
#set($testMessage = "THIS IS A TEST MESSAGE. ")
...THIS MESSAGE IS FOR TEST PURPOSES ONLY...
#else
#set($testMessage = "")
#end

${officeLong} HAS ISSUED A

* ${testMessage}TORNADO WARNING FOR...
#foreach (${area} in ${areas})
```

```
  ##
#if(${area.partOfArea})
#areaFormat(${area.partOfArea} 0) ##
#end
```

stateAbbreviations.txt

Translation of state name and two letter state abbreviation. Not need to change this

severethunderstorm.cfg
See tornado.cfg

severethunderstorm.vm
See tornado.vm

VM_glogal_library.vm

Contains some utilities for formatting dates, time, text etc. should need minimal modification.

The logic that processes the templates is located in java code at.
com.raytheon.viz.warngen\src\com\raytheon\viz\warngen\gui\WarngenAction.java

countyTypes.txt
Used for County/Parish/Municipality for Louisiana and SanJuan
No need to modify.

stateAbbreviations.txt

Translation of state name and two letter state abbreviation. Not need to change this

WarngenAction.java

WarngenAction.java contains the java code and logic for processing the templates. Observing the code it is apparent that the application logic is not complete for TO-8 and considerable work needs to be done.

Code Snippet:

```
VelocityContext context = new VelocityContext();

        IPreferenceStore store = LocalizationManager.getInstance()
                .getLocalizationStore();
        String office = store.getString(PreferenceConstants.P_SITE_NAME);
        context.put("vtecOffice", office);
        context.put("tornadoId", "tornado123");
        context.put("wmoValue", "wmoid123");

        Hashtable<String, DateFormat> dateFormat = new Hashtable<String, DateFormat>();

        dateFormat.put("header", new SimpleDateFormat(HEADER_FORMAT));
```

```
        dateFormat.put("plain", new SimpleDateFormat(PLAIN_FORMAT));
        dateFormat.put("clock", new SimpleDateFormat(CLOCK_FORMAT));
        dateFormat.put("ymdthmz", new SimpleDateFormat(YMDTHMZ_FORMAT));
        dateFormat.put("ddhhmm", new SimpleDateFormat(DDHHMM_FORMAT));
        dateFormat.put("time", new SimpleDateFormat(TIME_FORMAT));

        /** Convenience tools for the template */
        context.put("timeFormat", dateFormat);
        context.put("list", new ListTool());
        context
                .put(
                        "officeShort",
                        Activator
                                .getDefault()
                                .getPreferenceStore()
                                .getString(

com.raytheon.viz.warngen.PreferenceConstants.P_OFFICE_SHORT));
        context
                .put(
                        "officeLong",
                        Activator
                                .getDefault()
                                .getPreferenceStore()
                                .getString(

com.raytheon.viz.warngen.PreferenceConstants.P_OFFICE_LONG));
        context.put("stormType", stormType);

        context.put("mathUtil", new MathTool());
        context.put("dateUtil", new DateUtil());

        AffectedAreas[] areas = null;
        try {

            areas = Area.findAffectedAreas(config.getAreaConfig(), config
                    .getGeospatialConfig(), geom);
            context.put("areas", areas);

            Wx wx = new Wx(stormLoc, stormSpeedMetersPerSec, stormAzimuth,
                    eventTime.getTime(), startTime.getTime(),
                    endTime.getTime(), 1000 * 60 * 5);

            context.put("now", new Date());
            context.put("start", wx.getStartTime());
            context.put("expire", wx.getEndTime());
            context.put("event", wx.getObsTime());

            context.put("areaPoly", GisUtil
                    .convertCoords(geom.getCoordinates()));
            ClosestPoint[] points = wx.findClosestPoints(config
                    .getGeospatialConfig(), config.getClosestPointsConfig());
            context.put("closestPoints", points);

…
```

Code Notes:

Context.put is the command to insert content into the location on the template where the matching variable exists. There must be a correlation of the variables in the code and the variables in the template. It is not known at this time what all of the variables are and where there values are derived.

It is apparent observing the code that it is not yet complete and there are values hardcoded in the code. It is also observed that the "default duration setting" is hardcoded and not modifiable with the templates or configuration files.

Task List

The following tasks will need to be completed in order to customize, and configure the new WarnGen templates implemented in velocity:

- Document the VTL syntax
- Document the configuration file syntax
- Provide modification examples
- Define the WarnGen Template requirements
    - Based on current WarnGen logic
        - Process flow
        - Rules
        - NWS requirements
- Document the available variables and their context
    - Read code in WarngenAction.java and document how the variables are populated
- Define best practices
    - Dos and don'ts of template modifications
- Receive complex templates
    - Current templates represent the most trivial use cases
- Receive completed code
    - Code contains hardcoded variables and missing blocks of logic

Comments:

An exercise to modify the current AWIPS II templates was undertaken with the following results.

1) Reset default duration setting
    a. Failed, default setting is hardcoded.
2) Add Call to Action Statement
    a. Required modifying template and configuration file
    b. First attempt failed – unable to recover
    c. Second attempt was successful and fairly straightforward if documentation is provided
3) Change WFO name
    a. First attempt was successful by changing variable name from $officeshort to $officeLong
    b. It is unknown how or where the code populates the officeLong and officeShort variables

4) Change default text
    a. It was trivial to change the order and content of the default text in the template.