AWIPS II Satellite Expandability Whitepaper

A review of the code is underway to assure approximately 60 WFOs will continue to receive additional satellite imagery and products after they have transitioned in AWIPS II. Software changes are requested to allow for the expandability of non-operational satellite imagery and products (from GINI and netCDF3 format) within AWIPS II. The functionality sought was previously available in AWIPS. These changes will make it easier to add satellite data sets without recompiling the code for unique legends and color maps. Furthermore, this document assumes that the netCDF3 EDEX plug-in will use the same satellite viz code as the current SatelliteDecoder for GINI data.

This document has been prepared by Jordan Gerth (Jordan.Gerth@noaa.gov) at the request of National Weather Service Alaska Region Headquarters to assure all added satellite imagery is configurable.

The proposed changes are three new elements within the current satelliteImageryStyleRules.xml. An example is below, with additions in yellow:

```
<styleRule>
  <paramLevelMatches>
     <parameter>Imager 11 micron IR</parameter>
  </paramLevelMatches>
  <imageStyle>
     <displayUnits>C</displayUnits>
     <displayLegend>GOES 11 um IR Window ( C )</displayLegend>
     <range scale="LINEAR">
        <minPixel>0</minPixel>
        <maxPixel>255</maxPixel>
        <minValue>55</minValue>
        <maxValue>-109</maxValue>
     </range>
     <defaultColormap>Sat/IR/CIRA (IR Default)</defaultColormap>
      <colorbarLabeling>
            <pixels>0 30 60 90 120 160 200</pixels>
            <values>40 20 0 -20 -40 -60 -80</values>
         </colorbarLabeling>
  </imageStyle>
</styleRule>
```

The *displayLegend* element would override any value returned from SatelliteConstants class as a result of the getLegend action in SatResource.java:

```
private String getLegend(PluginDataObject record)
```

The *pixels* element would override the pixel-to-value mapping returned from one of the many classes extending UnitConverter in com.raytheon.uf.common.dataplugin.satellite.units (for example, IRPixelToTempConvert). Where *pixels* is a sequential, ascending one-dimensional array (list) of integers between *minPixel* and *maxPixel* inclusively the same length as *values*, *aPixel* is the argument to be converted to a value, and *result* is the value, the approximate logic to extend the IRPixelToTempConvert class would be:

```
public double convert(double aPixel) throws ConversionException {
  result = ((aPixel-pixels[i])/(pixels[i+1]-pixels[i]))*(values[i+1]-values[i]);
  return result;
}
```

If the *pixels* element is excluded but the *minPixel* and *maxPixel* are included, then there would only be two elements in the pixels array. If the *minPixel* and *maxPixel* elements are excluded, the current behavior would suffice.

In addition, the *minPixel* and *maxPixel* elements would replace the DataMin and DataMax parameters set within SatResource.java (currently hard-coded as 0 and 255, respectively):

```
initializeFirstFrame(SatelliteRecord record)
...
        colorMapParameters.setDataMin(0.0f);
        colorMapParameters.setDataMax(255.0f);
```

Lastly, com.raytheon.edex.plugin.satellilte/res/scripts/sources.sql should be expanded with these additional entries (values arbitrary):

```
insert into awips.satellite_source values (2, "ARH")
insert into awips.satellite_source values (3, "PRH")
insert into awips.satellite_source values (4, "WRH")
insert into awips.satellite_source values (5, "CRH")
insert into awips.satellite_source values (7, "SRH")
insert into awips.satellite_source values (8, "ERH")
insert into awips.satellite_source values (9, "CIMSS")
```

This document was based on a code review conducted of OB11.4. Examples are advisory only and should not suggest a complete code assessment has been done to assure dependencies have been satisfied if changes are applied in the areas referenced.

Last modification: *May 6, 2011*