**Common Changes when Converting Smart Tools and Procedures to AWIPS-2**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/CommonSmartToolChanges#CommonChangeswhenConvertingSmartToolsandProcedurestoAWIPS-2)

The following document lists common changes focal points should reference when converting smart tools and procedures from AWIPS-1 to AWIPS-2. This list is based on information from Tom Lefebvre.

**Note**: This list likely does not include all potential changes to smart tools or procedures needed at your site!

**WeatherElementEdited[¶](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/CommonSmartToolChanges" \l "WeatherElementEdited" \o "Link to this section)**

You cannot set WeatherElementEdited to None (WeatherElementEdited = “None”)! Doing so will cause your tool to crash with an error that gives no clue as to what is wrong. You must set WeatherElementEdited to a valid element, or to “variableElement.”

**numpy Arrays**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/CommonSmartToolChanges#numpyArrays)

When declaring a numpy array using one of the following methods:

* array()
* ones()
* zeros()

If a data type such as “Float32” is included in the array, the data type must be changed to one of the following (where the letter represents the type and the number is the size in bytes):

* 'i2'
* 'i4'
* 'f4' = Float32
* 'f8'

For example:

myArray = zeros(grid.shape, Float32) # AWIPS 1

myArray = zeros(grid.shape, 'f4') # AWIPS 2

**Scale Types in VariableList Definitions**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/CommonSmartToolChanges#ScaleTypesinVariableListDefinitions)

VariableList definitions of scale type GUIs cannot define values as string as with AWIPS 1. All values must be integers. For example:

VariableList = [

("Elevation to subtract from Wet Bulb Zero", **"750"**, "scale", [[0,1500]](https://collaborate.nws.noaa.gov/trac/ncladt/log/ncladt/?revs=0%2C1500)), ]

VariableList = [

("Elevation to subtract from Wet Bulb Zero", **750**, "scale", [[0,1500]](https://collaborate.nws.noaa.gov/trac/ncladt/log/ncladt/?revs=0%2C1500)), ]

**Data Type Inquiry**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/CommonSmartToolChanges#DataTypeInquiry)

When tools needed to know about the data type on which they were operating in AWIPS-1, the GridInfo object was used to inquire about the type. The gridded data type was represented in AWIPS 1 as an integer. In AWIPS-2, they are represented as the strings: "SCALAR", "VECTOR", "DISCRETE", "WEATHER". To access this string, use the following code in your tool:

gridType = variableElement\_GridInfo.getGridType().toString()

**GFE Locks**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/CommonSmartToolChanges#GFELocks)

GFE grid locks are different in AWIPS-2. Use the following code to access GFE grid locks:

parm = self.getParm(self.mutableID(), "T", "SFC") # get the parm

if parm is None:

print "parm is None"

lt = parm.getLockTable() # get the lockTable

locksByMe = lt.lockedByMe() # gets locks by me

locksByOthers = lt.lockedByOther() # gets locks by other GFE users

for lock in locksByMe.toArray(): # .toArray() converts from PyJObject to iterable

print lock

# These items (lock) are a JPyObject representation of a TimeRange

# so here's how to get the thing into a real GFE TimeRange

# must import AbsTime, TimeRange of course

start = l.getStart().getTime() / 1000

end = l.getEnd().getTime() / 1000

tr = TimeRange.TimeRange(AbsTime.AbsTime(start), AbsTime.AbsTime(end))