**serpISC**

**Version 1.8**

Les Colin - WFO Boise, ID

Author’s note: serpISC is strictly an editing tool—meteorology is not built into it. Meteorological coordination to within NDFD tolerance should have been completed beforehand. This tool’s purpose is to objectively clean up details to achieve final seamless borders.

Version 1.8 has a self-cancellation feature if average differences across any ISC border are too large. When that happens the tool reports which WFO(s) borders caused the problem.

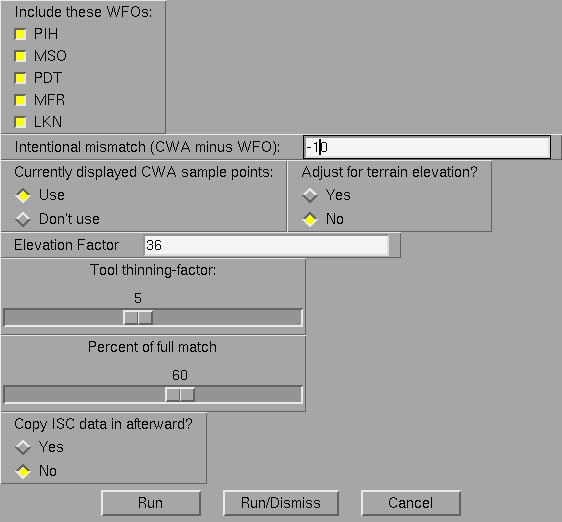
In response to several suggestions (National STSIT, GGW, BOI, among others) version 1.7 allows you to intentionally mismatch a neighbor’s grid. Suppose WFO A shares an ISC border with WFO B. A has 30 degrees on its side of the border while B has 40 degrees on its side, and A runs serpISC to match B. Later on, B decides A’s 30 was better after all. Now if B runs serpISC against A an artificial ridge of 40 will straddle the border, affecting both sides, and neither A nor B will be able to cleanly remove it.

This is where an intentional mismatch is useful. Either A or B can designate an edit area near the border and intentionally mismatch the other by 10 and save the result. Then the other can run a regular serpISC to match it.

(Version 1.6 corrected the return statement for wind, and repaired code when not adjusting for elevation. Version 1.6 has all the features of previous versions, including making the final ISC\_Copy optional in the tool GUI, which was introduced in version 1.5. With ISC\_Copy turned on, the copy is only done on the “lit” sites, and if an edit area has been specified, the copy is only done on the “lit” sites within the edit area.)

The serpISC smart tool produces a seamless match of your grids with any or all of the corresponding grids produced by your ISC neighbors. This tool is not intended as a substitute for ISC collaboration, but only as a supporting and finishing step. If other offices use it the national mosaic can become truly seamless.

serpISC presents a GUI like this:



The five choices in the upper left represent the five ISC sites surrounding BOI (the installation section will show you how to configure this for your site). Below that is the enhancement in version 1.7 that allows you to intentionally mismatch your neighbors. The next portion allows you to preserve sample values within your CWA. The middle right portion allows you to match by elevation, according to the elevation factor shown (see serp documentation for details). Below that is the “Tool-thinning factor”, a feature introduced in Version 1.4. serpISC can run very long (especially at 2.5 km resolution) if every border point is matched, so to speed it up you can now match to every second, or every third point, etc., according to your choice here. Next, the GUI allows you to make a partial match (i.e., nudge) toward your neighbor. The default is 100 percent, meaning a full match. If you set it to 50 (i.e., 50%) your grid will adjust only halfway toward your neighbor. Now if you send that grid via ISC, your neighbor will need to match full way (not half) to the newly received grid to make a seamless fit. The bottom of the GUI was the improvement for version 1.5—making the final ISC\_Copy step optional (the default is “No”).

Earlier versions of serpISC treated every ISC-border point as a serp control point (see Tim Barker’s documentation for the serp tool), and compared your values with your neighbors’. Differences at each point were “serped” over your entire CWA domain (or edit region) and added onto your original grid. Starting with version 1.4 the comparison of values was made differently: Home-CWA values situated one grid box inside the border are compared with their nearest neighbors’ values inside their own borders. Averaging is done if two or more neighboring grid boxes are equally close to a given home-CWA grid box. The comparison decreases in proportion to elevation difference between neighboring grid boxes. For example, if A is a grid box just inside the home-CWA border and B is the nearest grid box in the neighboring ISC, and both A and B have the same elevation, then the full grid-value difference between A and B is used. But if A and B differ by 5000 feet or more, no adjustment is made there. Intermediate elevation differences are used proportionately. The reason for this coding change is to enable serpISC to run even after ISC\_Copy has been run. In fact, serpISC version 1.4 used to complete itself by running an equivalent ISC\_Copy on every selected border. The resulting grid had your data inside the home-CWA and ISC data in every selected ISC area. Version 1.5 made the final ISC\_Copy optional.

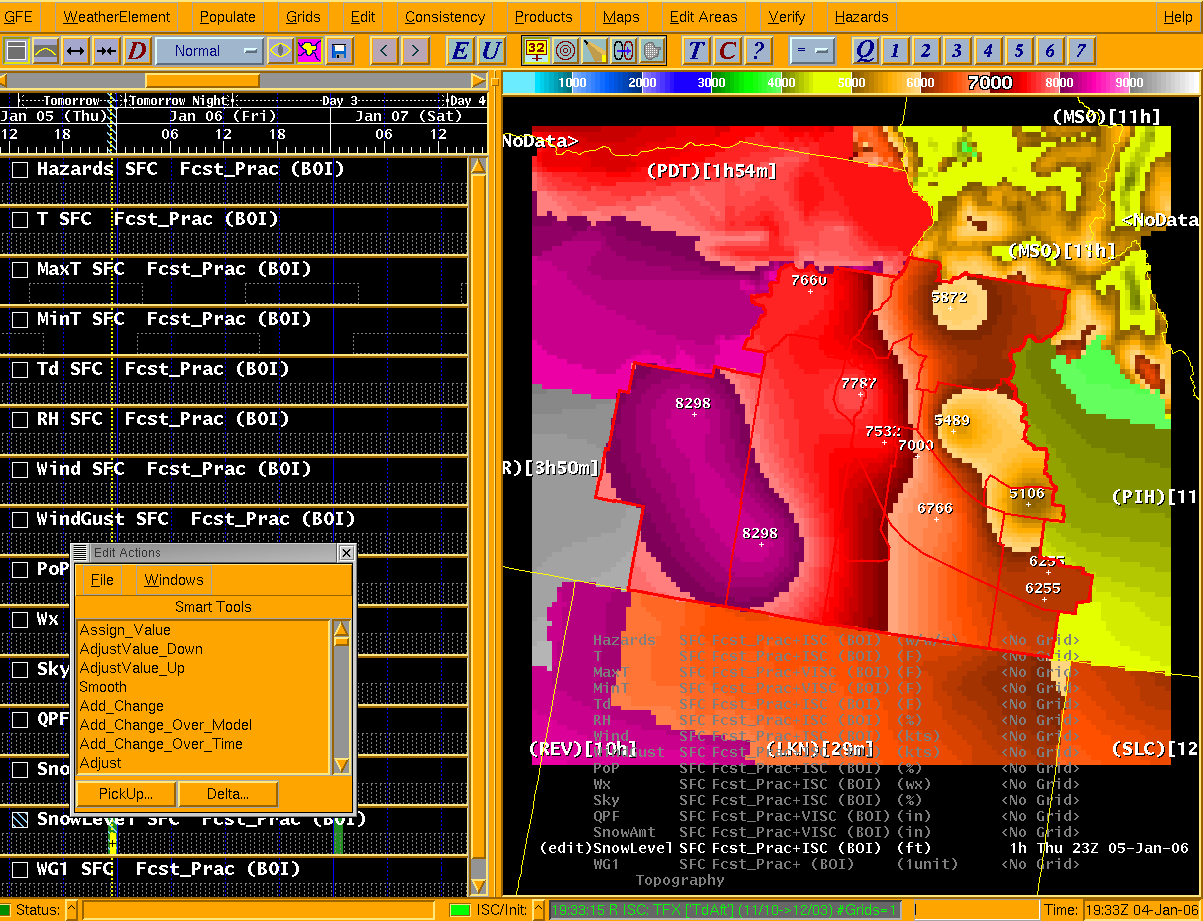
As in previous versions, no slider bars are needed to enter data differences — the tool already knows which adjustments to make at each border point. Only your grid changes — nothing happens to your neighbors’ grids.

serpISC can be run against one ISC border at a time, or on all borders at once. You can adjust for elevation in the GUI (again see Tim Barker’s discussion of elevation in the serp documentation) if your neighbor’s grid has the grainy detail characteristic of an elevation-dependent grid, otherwise select “No” there.

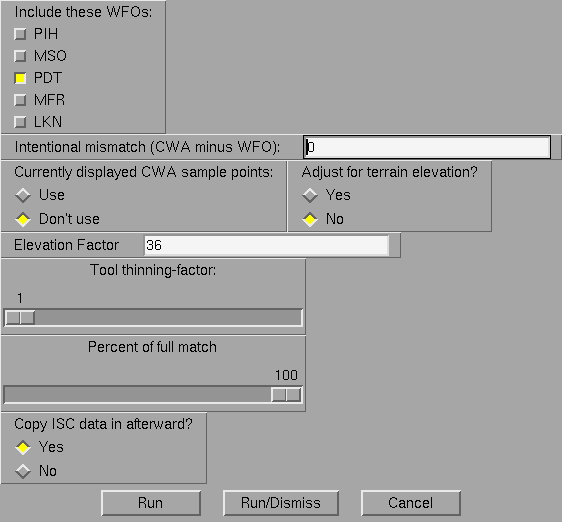
Some examples will illustrate the possibilities and implications of serpISC:

First of all, it’s important to start with a “good” forecast in your own CWA. This is because serpISC produces a “change”-grid which is added back onto your original. If changes are small, the structure and detail of your original grid will be essentially preserved, rather than replaced, by your neighbors’ data.

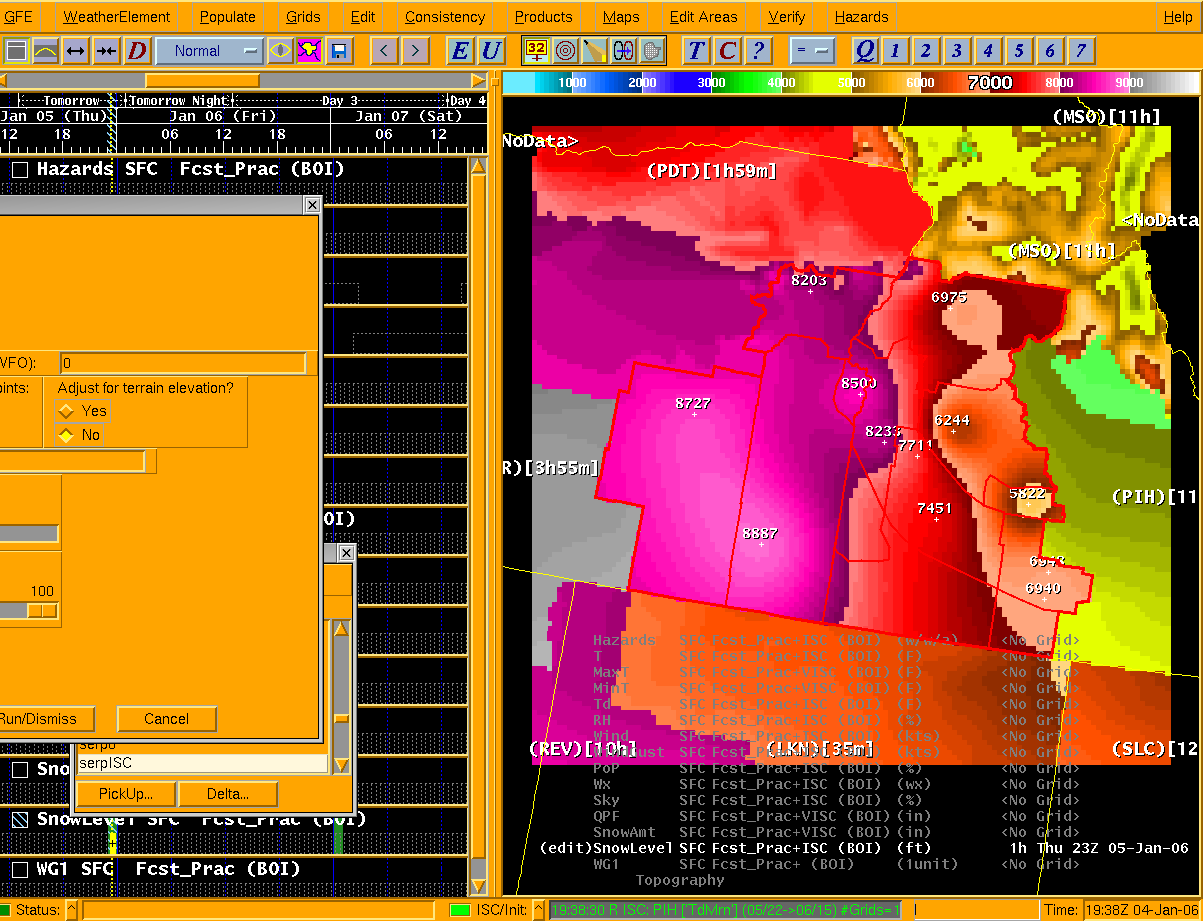
So starting with a good forecast, check the creation-times of all the surrounding ISC grids. Grids only a few minutes old can be matched right away. We’ll match a starting CWA SnowLevel forecast (shown next) with that of PDT (the neighbor to the northwest), with elevation not considered (we also let the sample points change by selecting “Don’t use” in the middle portion of the GUI):



And here’s the tool GUI:

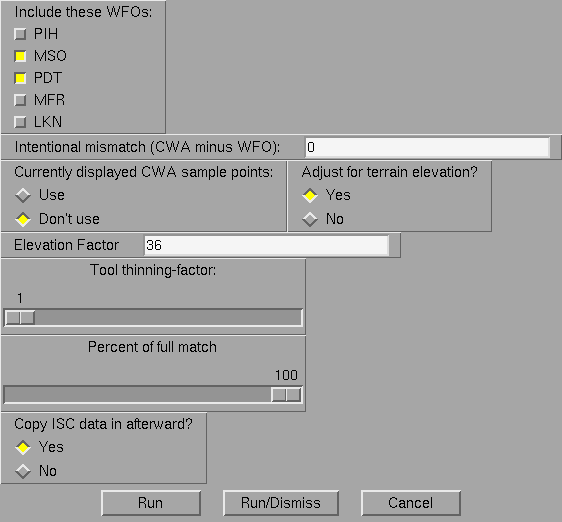


The result is:

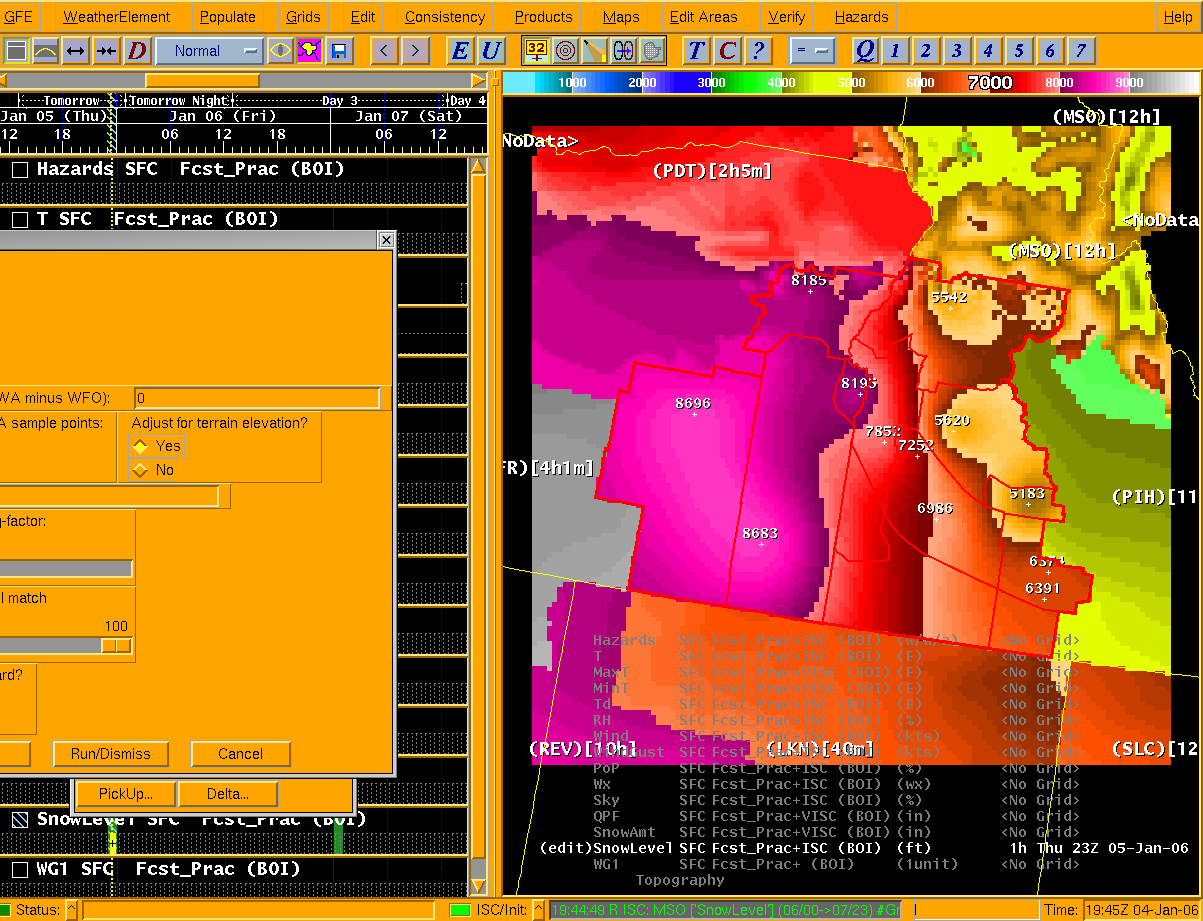


PDT’s data has blended into the CWA, but no other neighbor’s data has blended.

Now let’s run serpISC again but on MSO (the office to the north which supplied the grainy detail indicating elevation dependence). We answer “Yes” to the elevation question. We have to keep PDT lit so we don’t lose the previous result:



The result is:

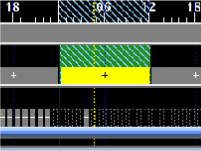


You can run serpISC again as each neighbor produces a new grid. In the latest example above it probably didn’t make much meteorological sense to transition from elevation “off” to elevation “on” across the CWA. This happened because different offices used different tools and methodologies to create their respective grids.

It’s probably better to either keep elevation “on” for both neighbors, or “off” for both neighbors, and sacrifice the perfect match for one neighbor.

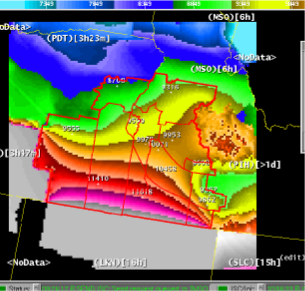
The order in which new offices are added to serpISC makes a slight difference. You must remember to keep previous ISC offices turned “on” as new grids come in, or else the old borders will be re-written. Eventually all the ISC offices will be turned “on” and you can run serpISC one last time.

Even after doing this you may find that some grids still don’t match. This happens when the duration of your grid is longer than that of your neighbor:



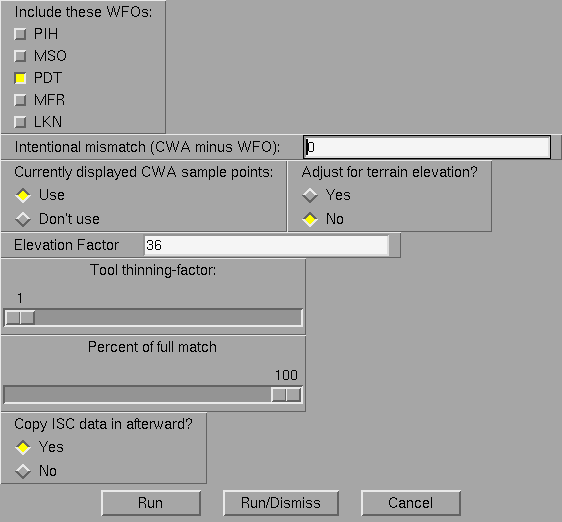
Here, your yellow duration spans at least two grids from your neighbor, so your neighbor’s grids are averaged first (except with PoP which uses the maximum instead), before serpISC runs. To deal with this you can split out one hour in your grid, run serpISC on that one hour, then widen the resulting hourly grid to the original time interval.

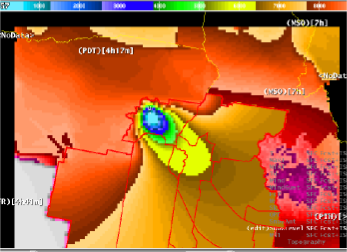
When two adjacent ISC neighbors don’t match each other serpISC will still fit both of them but it won’t look good, because all contours between the mismatched values will have to converge to the point of the mismatch. Here, PDT’s grid does not match MFR’s grid (on the western edge of this image). So all intermediate values have to squeeze into the point of discontinuity.



The problem here is that serpISC was run before the two neighbors coordinated with each other.

As stated before, you can preserve sample values inside your CWA (samples outside the CWA belong to the ISC offices and are not used. Those samples are walled off by the ISC borders anyway). This means that the change-input at those sample points is to be zero. Now consider what happens when one of those sample points is near an ISC border. At the border, large changes may be required, while at your sample point no change is allowed. Here, we show a single sample point with value “0” in the northwest part of your CWA, and we’re going to match PDT (the neighbor to the northwest, closest to the sample point):



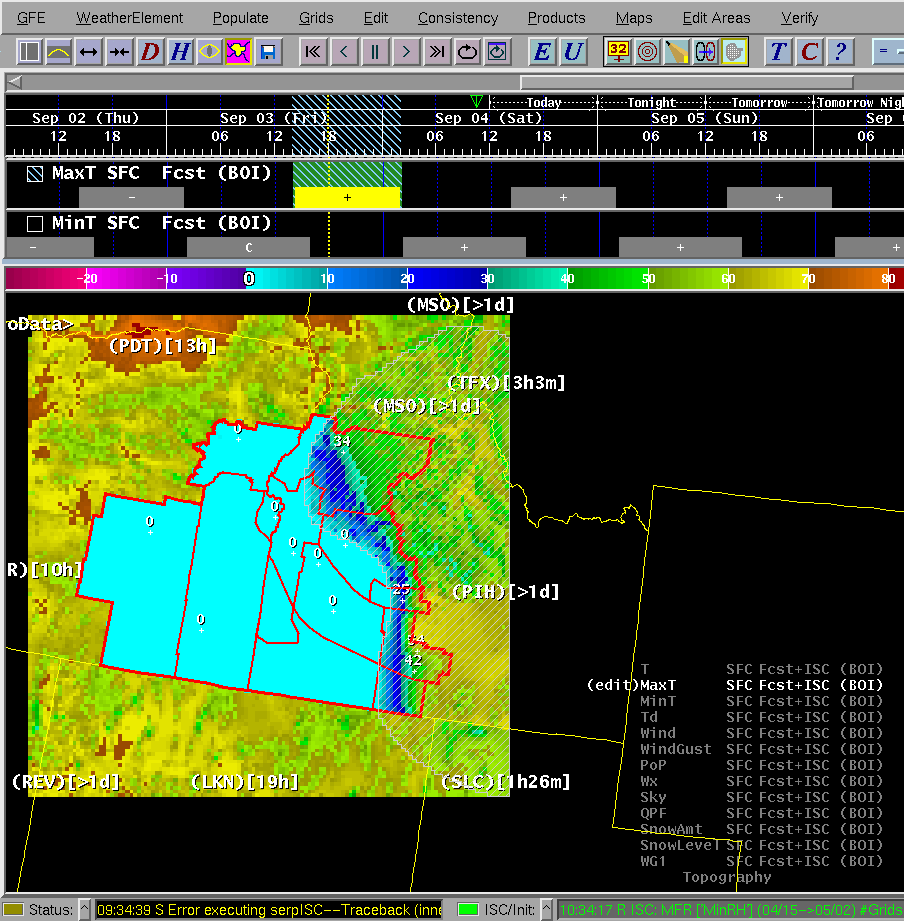


You can see how the army of PDT’s border-values blows past a sample point that must hold its value. It should be noted that isolated sample points are weighted more heavily by serp and have more influence on the resulting change-field than clustered samples or border points (which always have lots of nearby neighbors). This behavior is part of serp.

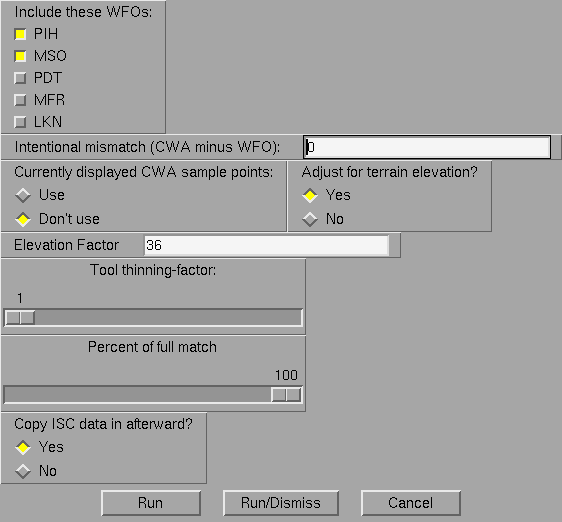
Sample-values near an ISC border can develop a large and possibly unrealistic gradient between the sample point and the border. To prevent this, either let the sample point change by turning it off (sample points can be turned off one at a time by middle-mousing just below them), or better still, coordinating the border values better with your neighbor first.

Version 1.3 introduced a feature, suggested by Wade Earle, that confines adjustments to an edit area rather than spanning the entire home CWA. A taper effect is created, starting from a full (or partial) adjustment at designated ISCs, to zero change inside the home CWA where the edit area stops.

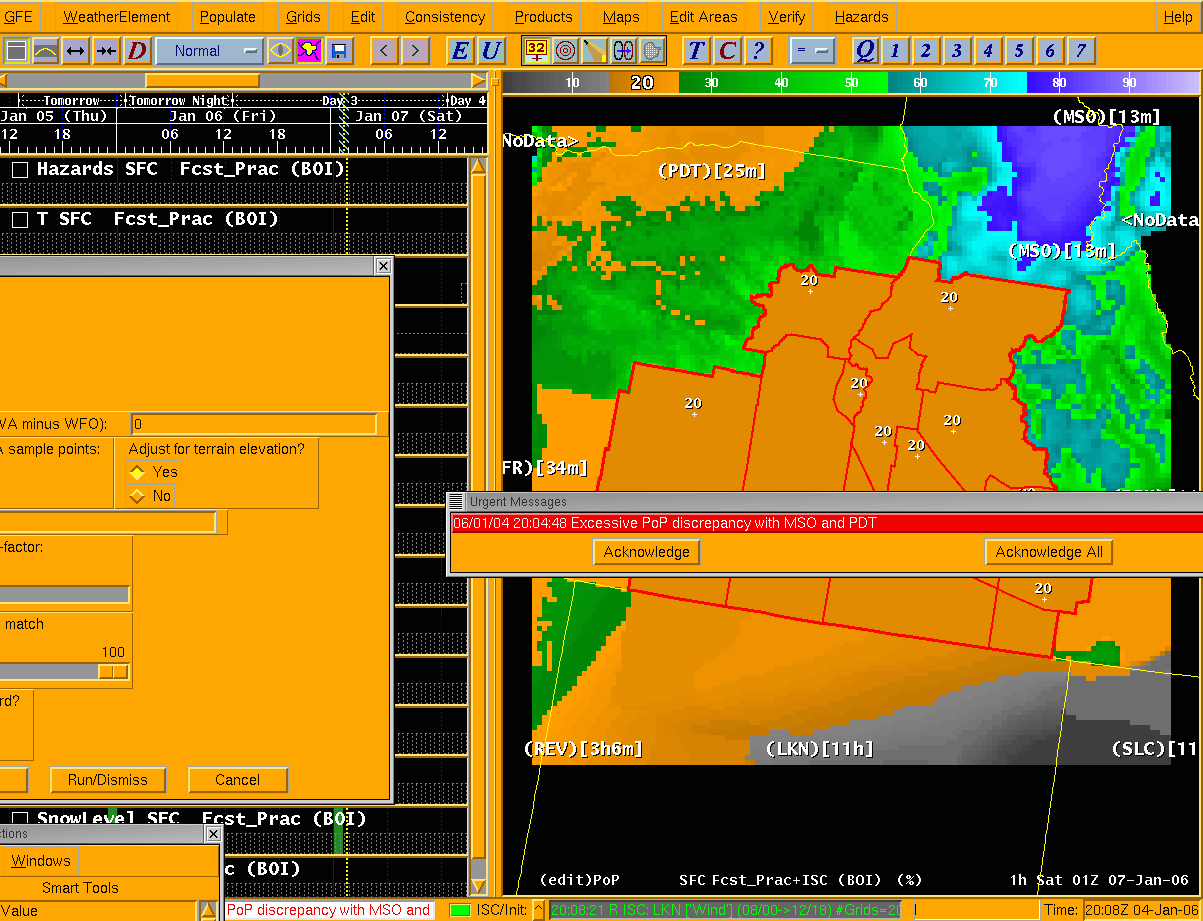
In the image below we started with a zero-valued grid in the home CWA, and designated an edit area on its eastern side. (In practice, you could not start with such large ISC discrepancies -- the tool would self-abort.) We set the GUI to use the two ISC neighbors within the edit area (MSO and PIH). We also let sample points be overwritten. The taper-effect can be seen in this artificial example. Note that if we had lit PDT instead (which does NOT touch the edit area), and PDT was the only site lit, no adjustments would have been made. On the other hand, you must be careful to keep your edit region from intersecting a lit neighbor’s border. In that case an unrealistic situation would occur in which zero values on the edit-boundary inside the home CWA meet the (potentially large) non-zero values at that neighbor’s border.



And here is the GUI setting for the above:



As stated earlier, the tool will self-cancel, i.e., abort, if initial ISC discrepancies exceed NDFD consistency limits. In the example below, we attempt to match all neighbors but we find that the northern sites exceed NDFD PoP limits. The tool outputs a red banner identifying those neighbors (in this case MSO and PDT) with whom further coordination is necessary before the tool will work.



Some final points: The tool seems to bog down and won’t post all results if too many grids are highlighted, say, all the Sky grids for seven days. Better to highlight only one or two grids at a time.

This version of serpISC will work even if there are duplicate sample points on the CWA. Early versions would stall when they encountered duplicate samples.

**Tool Installation:**

You can use the ifpServerText program to place this tool into user SITE by entering the following commands:

ifpServerText –u SITE –s –n serpISC181 –f serpISC181.tool –c SmartTool

ifpServerText –u SITE –s –n ObjAnal –f ObjAnal.utility –c Utility

In the serpISC tool itself note the section near the top where it says “Part to modify for local configuration”. Change the defaultCWA to your own CWA and change the VariableList entry “Include these WFOs:” to contain the names of your ISC neighbors. (At BOI we already have edit areas for all of them, as well as for BOI itself.) Your office will need to do something similar.

**Acknowledgement**

The author thanks Tim Barker (SOO at BOI) for assisting in the creation and various modifications of this tool and for re-writing serp as a utility.