# Masking Requirements

## Preamble

Masking is considered a core area and it is vital that the final (re)design for masking is approved and accepted before work begins. Furthermore, the actual work will be done on a branch and thoroughly tested and reviewed before it is merged back into main.

## Requirements

1. There will be a pair of algorithms to save and load masks from disk. This will need to support loading from legacy formats, but saving to them may not be necessary. It is also likely that the format needs to either be binary or have a binary variant for instruments with a large number of pixels (i.e. SNAP).
2. A historic masking file format that refer just to spectrum numbers that need to be able to be applied to workspaces (direct inelastic). This will be done by loading the mask into a workspace and applying it.
3. Extract the mask from a workspace to another workspace (effectively a clone, but going to Boolean data) that can be applied to multiple workspaces.
4. Masks can be pixel and x-axis specific. In order to apply the same filter in MaskBins one currently uses a table workspace, it could be done by a new kind of mask workspace.
5. Algorithms need to operate on Masked data without special programming, unless grouping (in which case they should care about masking anyway). This is currently done using the value flag of 0 (not sure where, in MaskWorkspace LIVE\_VALUE=0).
6. You should be able to easily differentiate between 0 counts and masked.
7. You should be able to combine masks using addition and subtraction. BinaryOperateMasks should continue to do the logical operations of ‘not’ (inversion), ‘or’, ‘xor’, and ‘and’.
8. Masks should fundamentally operate at the detector level, this way they can always handle all grouping situations.
9. When applying masking it should be at the detector level, as long as both source and target workspace have instruments. Otherwise they can apply at a workspace indexes level with copious log messages explaining this.
10. Grouping detectors should work the same way as the DetectorGroup object currently does. A grouped detector is only masked when all of its constituent detectors are also masked.

## Questions that need to be answered:

1. Solid angle of a group when some detectors are masked. Do the masked detectors contribute to the solid angle?  
   [ND]Currently no, but a wider question is does this matter? How often do we have partially masked spectra, and can we actually decrease the frequency even further? See section Partially masked spectra.
2. Effective position of a group when some detectors are masked. Should the position move because some of the contributing detectors were masked?  
   [ND] The position of a group is currently not affected by the masked status of the spectra it contains. Again I think this is rare, and we can probably make it rarer. See section Partially masked spectra.
3. Masking data that does not have an associated instrument. Should this be allowed?  
   [ND] We need to be able to apply masks such as this to workspaces, but that does not mean that workspaces need to be able to store this information. So the MaskDetectors could take a file format (as it currently does) that masks on spectra number, but you can only extract Mask Workspaces from workspaces that have detectors (and therefore instruments.
4. Do we need to support the ability to unmask previously masked data after further processing algorithms?  
   [ND] No, the effort and complexity to support this far outweighs the expected usage, or the effort required to rerun the analysis with a different Mask applied.

# Possible solutions

## Detector based masking

1. Mask Workspaces should operate just like normal workspaces, they should not have any unique functions, in fact event having a derived class is probably a mistake as it would encourage the addition of functionality.
2. There were some areas where Mask workspaces made workflows faster, by eliminating the need to iterate the sp-det lookups to apply masking. Improvements and intelligent caching (in the WS param map object as a new cache) should be implemented for all matrix workspaces.
3. Bitwise mask add and subtract operation should be added, these should be separate from the add and subtract algorithms.
4. We should test the implications of using quiet NaNs as the masked value rather than 0.
5. The historic file format for inelastic data are a special case and should just be supported by MaskDetectors for application.
6. Masks should be applied at the detector level, as long as both source and target workspace have instruments. Otherwise they can apply at a workspace indexes level with copious log messages explaining this.

## Partially masked spectra

Currently spectra that originate from more than one detector could have one or more of those detectors masked. The rule currently is that the spectra is only considered masked if all of the contributing detectors are masked. Partially masked spectra are currently inconsistently supported in Mantid, and it may be better to not support them at all. Please note that this is referring to completely masked detectors, not masked time regions within a spectra.

As this stage it is worth understanding that the existence of any partially masked spectra is either an edge case, or the result of grouping a large region of the instrument and unintentionally including masked detectors. It is fair to say that any edge cases that occur during Mantid use are not well supported due to the inconsistent implementation at present, and indeed getting agreement as to the correct implementation across all stakeholders will be time consuming. The second case is where masked detectors exist within a group. Scientists mask detectors because they do not want them to contribute to their data, or data processing in any way, as such we have to wonder why we are allowing them to be part of a group at all.

We should seriously consider not supporting partially masked spectra, and altering the rule such that if any member of a group is masked then that member is removed from the group. The masking the last member of a group would be mask the spectra.

## Spectra Based Masking

Notes from the session this morning to add

We prefer this approach