# Software Requirements Specification: Threads

## Requirements:

### File IO Need: Risk: Status:

User must be able to load a process from a static file Critical Low Alpha

User must be able to save a process to a static file Critical Low Alpha

### Display Need: Risk: Status

Display a process in steps or layers from right (end) to left (start) Critical Low Alpha

Use Case: As a scientist I would like to be able to view a manufacturing process from start to finish

When programmatically altering the display, should be animated Should High In Prog

UX: User expectation in modern software that movement of nodes should be animated in some way

### Controls: Need: Risk: Status:

User must be able to add a node to the display (vertex) Critical Low NS

User must be able to remove a node from the display Critical Low NS

User must be able to connect a node with another node (edge) Critical Med NS

User should be able to move a node up- or downstream NTH High NS

### Node Functions Need: Risk: Status:

Node must display name of node Critical Low Done

The node map will be nonsensical if some node identification isn’t displayed

Node should be able to display chemical structure NTH Med NS

Note: Not all threads will be associated with chemical structures (i.e. document structure)

Use Case: As a scientist, I would like to see the chemical structures of the process as I am displaying them in a manufacturing process

## Edge Functions Need: Risk: Status:

User should be able to add direction to a connection (edge) Should Low NS

Use Case: As a scientist, I would like to see the cause and effect of one node on another node

Note, it may be enough to have the software determine the direction, but it should be displayed, for example by the use of an arrow-head.

User should be able to add weight to an edge NTH Med NS

Note, this isn’t required for all types of thread diagram and should be prioritised only if cause-effect diagrams become a key part of the software.

As a scientist, I would like to visualise the size of an effect one node has on another node