Apollo Common User Interface Controls

Progress Tree Control

# Disclaimer

This specification is only an initial specification. It is not completed and changes will be made over time.

# Goal

The goal is to create a progress control to hold several layers of progress controls. This control will be used in the Apollo User Interface to display the progress of the processing of data sets. Because data sets can have a parent-child relation it is necessary to have a progress control in which these relations become clear.

# Features

The Progress Tree Control has the following features:

* The individual elements of the tree can be selected. Upon selection an event will be fired indicating which element(s) have been selected. Programmatically selecting an element is also possible and also leads to the firing of the selection event.
* There can be many top-level nodes.
* It is possible to add a node above existing nodes, thereby forcing the node to become a child node.
* Each element of the Progress Tree Control contains:
  + A progressbar
  + An estimate for the remaining time
  + An indication of the current action that is being performed
  + Start, stop and pause buttons which action on the action which is being monitored
* The Progress Tree Control provides the following events:
  + SelectedItemChanged
* The Progress Tree Control allows data binding to elements that provide the right data.
  + How does this work?
* The progressbar in the elements will be displayed in different colors depending on the state of the data set.
  + Running data set: Green
  + Paused data set: Orange
  + Error in data set: Red
* The Progress Tree Control does not allow higher level controls to be directly data bound to the lower level controls. All progress should come from an outside source.
* The Progress Tree Elements allow the following states:
  + Running
  + Paused
  + Waiting
  + Error
* Progress Tree Elements action controls can be enabled or disabled to allow the user to change the action state (start, pause, stop) or to disallow the user to change the action state.

# Design

* Develop separate component for TreeNode.Header --> Has the progressbar etc.
* Tree actions:
  + Add new tree element
  + Remove tree element
  + Select tree element
* TreeNode actions:
  + Update progress
  + Update current action
  + Show details
  + Pause action
  + Stop action (--> leads to removal of the node)