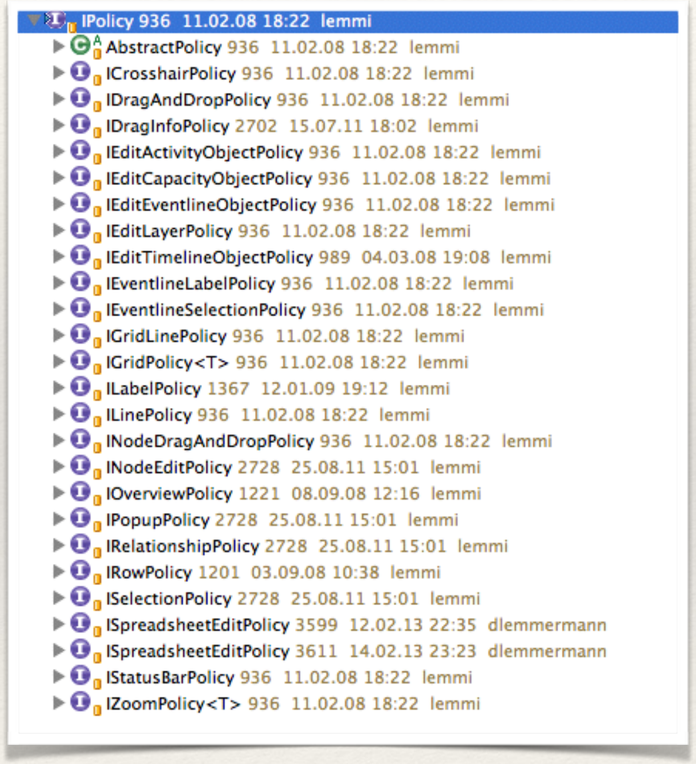
Introduction to FlexGantt

Topic: Policies

Dirk Lemmermann
Software & Consulting
Zurich, Switzerland

Policies


IPolicy 936 11.02.08 18:22 lemmi
AbstractPolicy 936 11.02.08 18:22 lemmi
ICrosshairPolicy 936 11.02.08 18:22 lemmi
IDragAndDropPolicy 936 11.02.08 18:22 lemmi
IDragInfoPolicy 2702 15.07.11 18:02 lemmi
IEditActivityObjectPolicy 936 11.02.08 18:22 lemmi
IEditCapacityObjectPolicy 936 11.02.08 18:22 lemmi
IEditEventlineObjectPolicy 936 11.02.08 18:22 lemmi
IEditLayerPolicy 936 11.02.08 18:22 lemmi
IEditTimelineObjectPolicy 989 04.03.08 19:08 lemmi
IEventlineLabelPolicy 936 11.02.08 18:22 lemmi
IEventlineSelectionPolicy 936 11.02.08 18:22 lemmi
IGridLinePolicy 936 11.02.08 18:22 lemmi
IGridPolicy<T> 936 11.02.08 18:22 lemmi
ILabelPolicy 1367 12.01.09 19:12 lemmi
ILinePolicy 936 11.02.08 18:22 lemmi
INodeDragAndDropPolicy 936 11.02.08 18:22 lemmi
INodeEditPolicy 2728 25.08.11 15:01 lemmi
IOverviewPolicy 1221 08.09.08 12:16 lemmi
IPopupPolicy 2728 25.08.11 15:01 lemmi
IRelationshipPolicy 2728 25.08.11 15:01 lemmi
IRowPolicy 1201 03.09.08 10:38 lemmi
ISelectionPolicy 2728 25.08.11 15:01 lemmi
ISpreadsheetEditPolicy 3599 12.02.13 22:35 dlemmermann
ISpreadsheetEditPolicy 3611 14.02.13 23:23 dlemmermann
IStatusBarPolicy 936 11.02.08 18:22 lemmi
IZoomPolicy<T> 936 11.02.08 18:22 lemmi

What is a Policy?

- A policy encapsulates a well defined (and limited in scope) piece of functionality. The type of functionality depends on the component that requires the policy.
- Example: the row policy covers anything related to row height, row resizability, and produces the command to alter row heights.
- Policies are pluggable. They can be registered with policy providers.

Why Policies?

- Swing models work with „Object.class“.
- Problem: FlexGantt can not ask „Object.class“ anything (e.g. „are you selectable?“).

```
public interface TreeModel {
    public Object getRoot();
    public Object getChild(Object parent, int index);
    public int getChildCount(Object parent);
    public boolean isLeaf(Object node);
    public void valueForPathChanged(TreePath path, Object newValue);
    public int getIndexOfChild(Object parent, Object child);
}
```

Good News

- Swing and FlexGantt both ship with default implementations of their models.
- The default models do not use „Object.class“
- DefaultGanttChartNode, DefaultTimelineObject
- Most policy decisions are directly delegated to the default model classes.
- Note: commands are ALWAYS retrieved via a policy.

Policy Provider

```
/**
 * A policy provider supplies the component (e.g. a tree table or a timeline)
 * that it is attached to with policy implementations. These implementations can
 * be registered with the provider. They can be looked up based on the policy
 * interface.
*/
public interface IPolicyProvider {

     /**
      * Registers a policy implementation for the given policy type (policy
      * interface).
      */
     <T extends IPolicy> void setPolicy(Class<T> policyType, T policyImpl);

     /**
      * Returns a policy implementation for the given policy type (policy
      * interface).
      */
     <T extends IPolicy> T getPolicy(Class<T> policyType);

     …
}
```

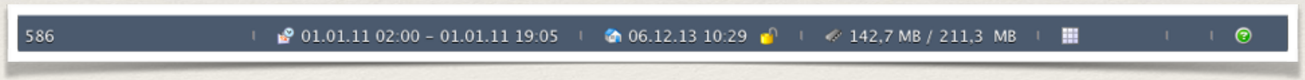
Policy Providers

- Used by:
  - AbstractGanttChart
  - LayerContainer
  - TreeTable
  - Dateline
  - Eventline

Where are policies used?

Policy User: AbstractGanttChart

- Uses a single policy: IStatusBarPolicy.
- Determines which fields are visible in the statusbar.
- Formats various strings for proper display in the statusbar.

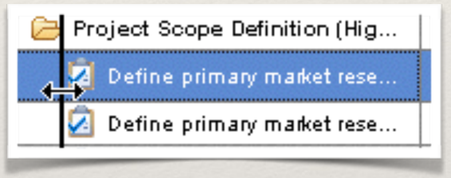


Policy User: LayerContainer

- IDragAndDropPolicy
- IEditTimeline / Activity / CapacityObjectPolicy
- ILabelPolicy
- IRelationshipPolicy
- IPopupPolicy
- ISelectionPolicy
- ICrosshairPolicy
- IOverviewPolicy
- ILinePolicy
- IEditLayerPolicy
- IGridPolicy
- IGridLinePolicy
- IDragInfoPolicy
- ISpreadsheetEditPolicy

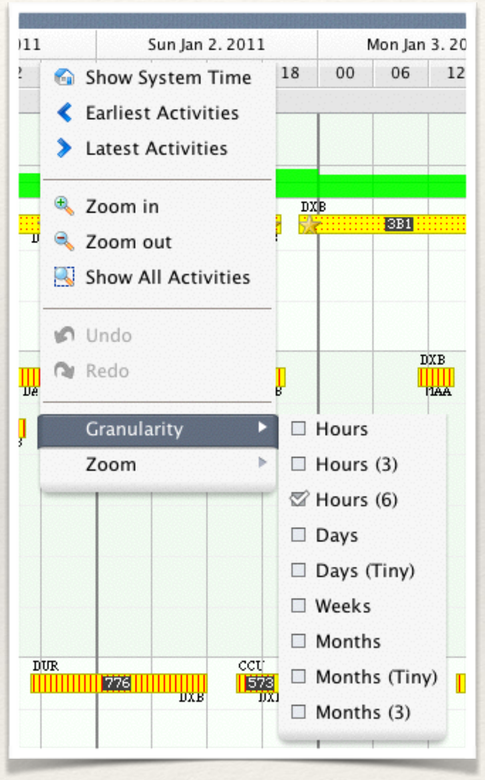
Policy User: TreeTable

- INodeDragAndDropPolicy
- INodeEditPolicy
- IRowPolicy

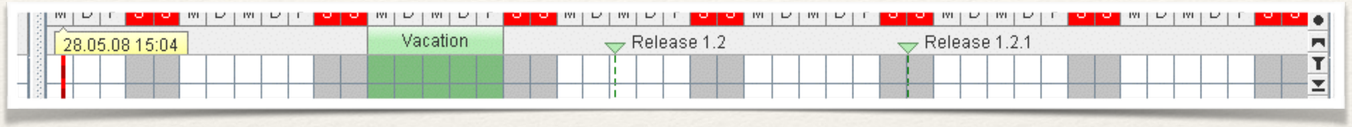


Policy User: Dateline

- IZoomPolicy
- Determines the available granularities.



Policy User: Eventline

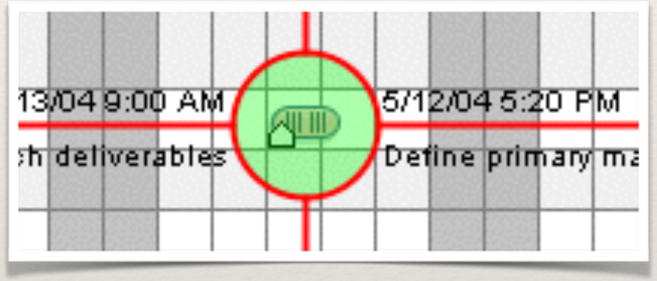


- IEventlineLabelPolicy
- IEditEventlineObjectPolicy
- IEventlineSelectionPolicy
- IGridPolicy

Policies

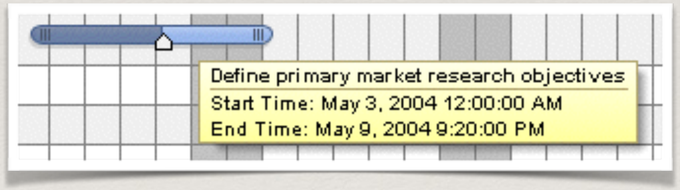
ICrosshairPolicy

- Very simple: returns the text for the four corners of the crosshair.



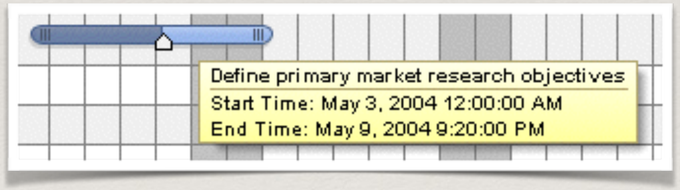
IDragAndDropPolicy

- Can a timeline object be dragged? If yes, how? (Move, Copy, Move & Copy, None).
- Which command has to be executed for a given timeline object when it gets dropped?



IDragInfoPolicy

- Returns the model object used for displaying a popup while moving a timeline object, setting the „percentage complete“ value, or changing the „capacity used“ value.

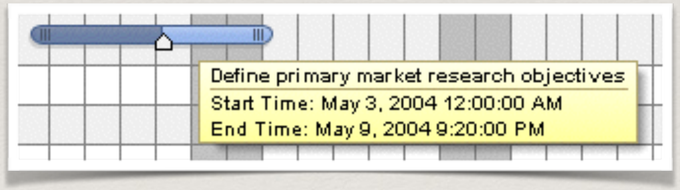


IEditTimelineObjectPolicy

- Determines which editing operations are allowed for a given timeline object.
  - Create, Delete, Start Time, Duration, In-Place Editing
- Returns the necessary commands to perform the changes.

IEditActivityObjectPolicy

- Determines if the „percentage complete“ value of an activity is editable.
- Returns the command to set a new value for „percentage complete“.

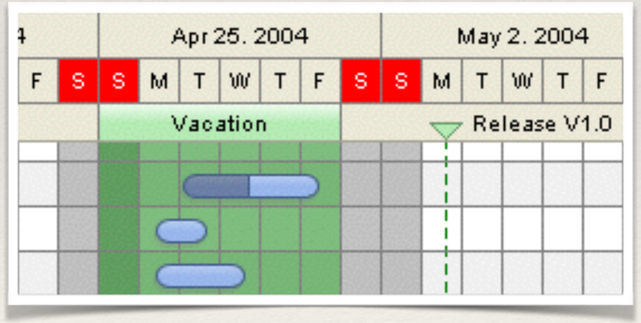


IEditCapacityObjectPolicy

- Determines if the „capacity used“ value of a capacity object is editable.
- Returns the command to set a new value for „capacity used“.

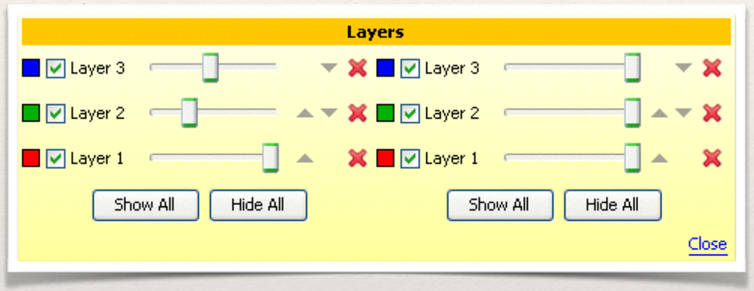
IEditEventlineObjectPolicy

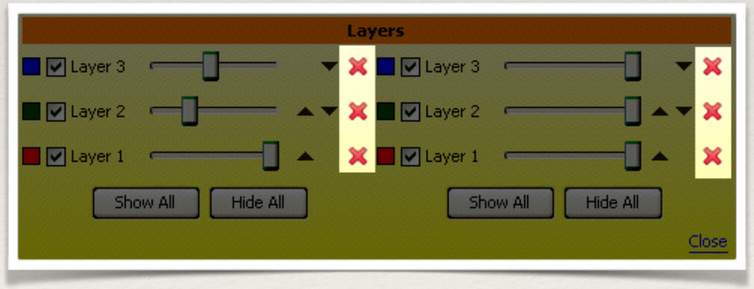
- Determines which editing operations are allowed for a given eventline object.
  - Create, Delete, Start Time, Duration
- Returns the necessary commands to perform the changes.



IEditLayerObjectPolicy

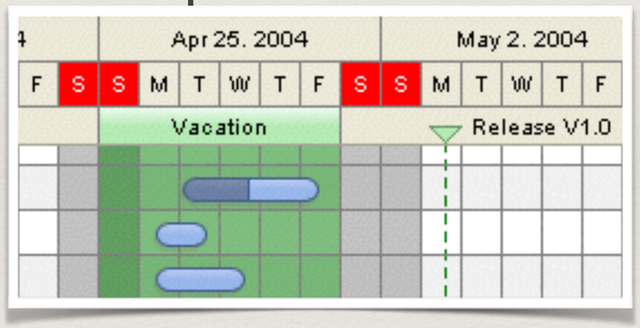
- Returns commands for adding and removing





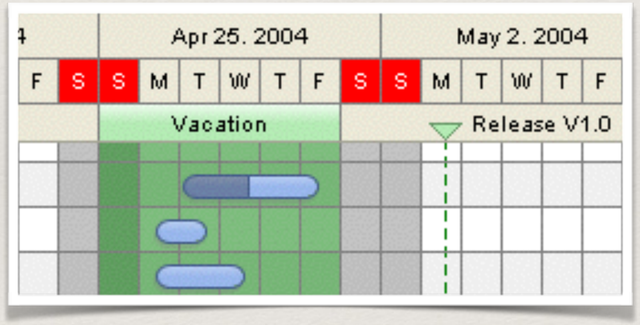
IEventlineLabelPolicy

- Returns the various labels used by eventline objects
  - Name, Description, Tooltip



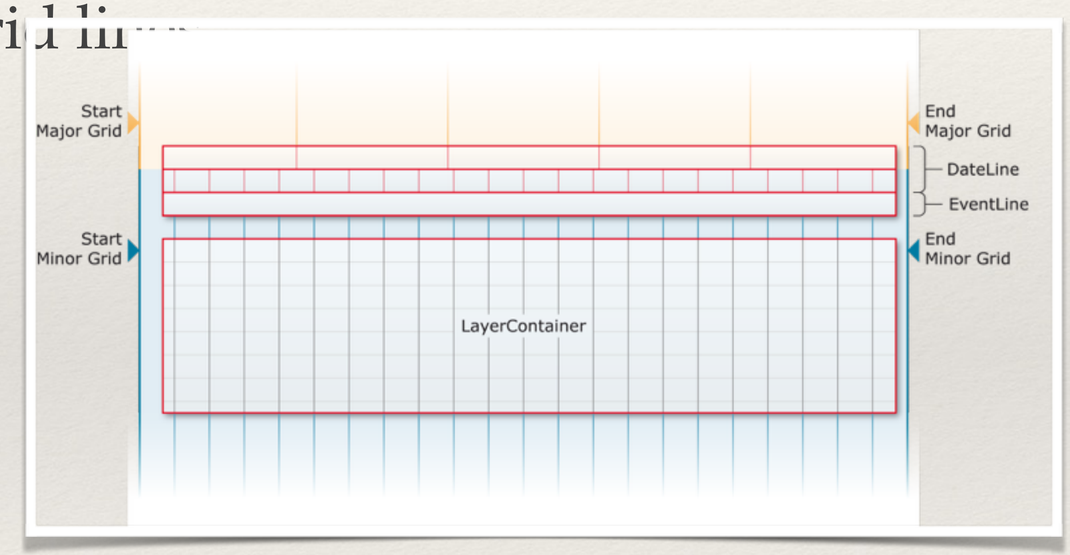
IEventlineSelectionPolicy

- Used only for the purpose of determining whether an eventline object is selectable or not.



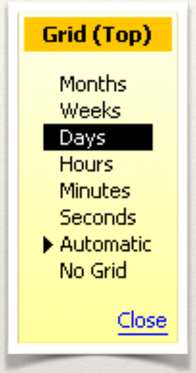
IGridLinePolicy

- Used to check visibility of major and minor grid li



IGridPolicy

- Controls a virtual (non-visible) grid that is used to make timeline objects „snap“ to grid locations after editing operations.




```
T[] getGridGranularities(); // hours, minutes, seconds

long getGridAdjustedStartTime(T granularity,
     long unadjustedStartTime,
     IDatelineModel<T> datelineModel, boolean autoGrid);

long getGridAdjustedEndTime(T granularity, long unadjustedEndTime,
     IDatelineModel<T> datelineModel, boolean autoGrid);

ITimeSpan getGridAdjustedTimeSpan(T granularity,
     ITimeSpan unadjustedTimeSpan, IDatelineModel<T> datelineModel,
     boolean autoGrid);
```

ILabelPolicy

- Determines the visibility and the text for various timeline object labels.

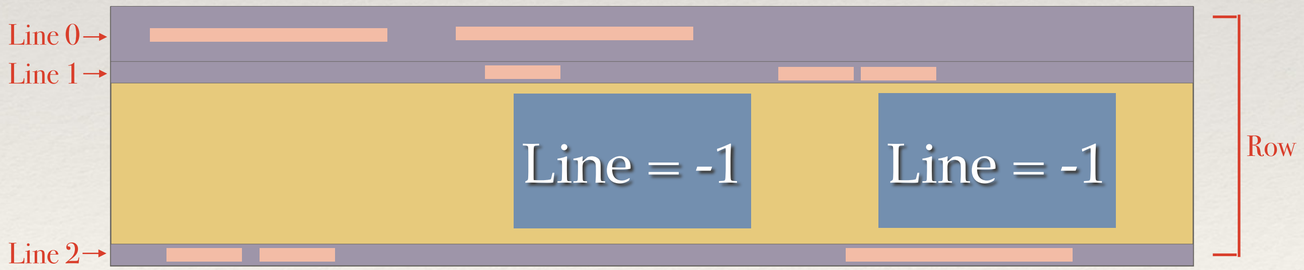
```
public enum LabelType {

    NAME, // shown in various places
    TOOLTIP, // when popups are off, regular tooltips can be used
    DESCRIPTION, // shown to the right of the timeline object
    STATUS, // shown in the statusbar
    DRAG_INFO_TITLE // shown in the title of the drag info popup

}
```

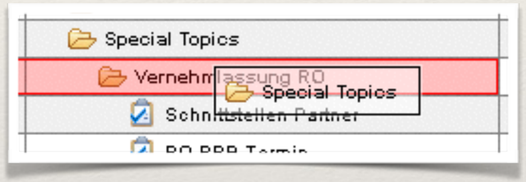
ILinePolicy

- Used for controlling „inner lines / rows“.
  - Line count
  - Height of each line
  - Location of each line (lines can overlap)
  - Visibility of each line
  - Line index for a given timeline object / placement



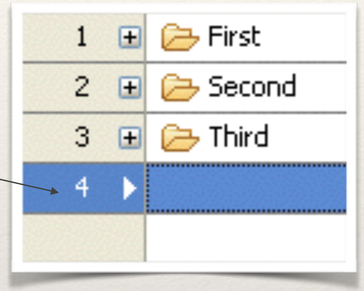
INodeDragAndDropPolicy

- Covers drag and drop operations within the tree table.
- Returns possible drag actions for a given node.
- Returns possible drop actions for a given target node.
- Returns the command to actually perform the necessary model changes.



INodeEditPolicy

- Covers various node editing aspects:
  - Can new nodes be created?
  - Is a node deletable?
  - Is the key of a node editable?
  - Is a certain column value editable?
  - Is a node reassignable to a new parent node (e.g. for indentation purposes)?
  - Is a node selectable?
- Returns commands for changing a key / column value, for creating nodes, for deleting nodes, for inserting nodes.



IOverviewPolicy

- Returns a status object for a given timeline object.
- Status object is used by overview display to highlight problems

IPopupPolicy

- Returns the two input objects for popup renderers.
  - Title and Content (standard and extended).

IRelationshipPolicy

- Defines „linking“ capabilities.
  - Is a link deletable?
  - Is a timeline object a possible link „source“?
  - Is a timeline object a possible link „target“?

IRowPolicy

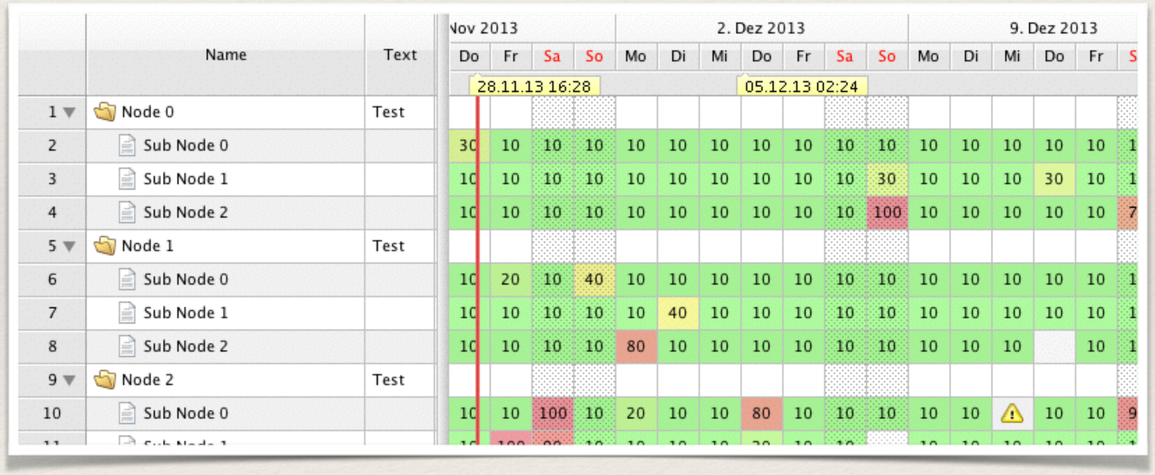
- Used for retrieving information about individual rows.
  - Current / Minimum / Maximum height.
  - Visibility of row divider line (good for grouping rows).
  - Is row resizable?
  - Returns the „row resize“ command (so it can be undone / redone).

ISelectionPolicy

- Controls which objects can be selected by the user.
  - Timeline objects
  - Tree nodes
  - Relationships / Links

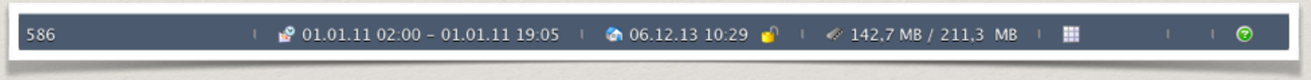
ISpreadsheetEditPolicy

- Only used to lookup the command for modifying a spreadsheet value.



IStatusBarPolicy

- Determines which fields are visible in the statusbar.
- Formats various strings for proper display in the statusbar.



IZoomPolicy

- IZoomPolicy
- Determines the available granularities.