

Parameter Panel Collection Level

This is the outermost container of the UI and is described by the abstract class **ParameterPanelCollection**. This usually represents a collection of tabs. Implementing classes must override a few methods and are described in the javadoc. The important members in this collection are listed below:

Parameter Panel Collection Members	Description
public List<ParameterPanel> getPanels()	Override this method to return a list of panels (tabs) within the ParameterPanelCollection .
public String getParameterPanelCollectionClassName()	This method returns a CSS class name to be applied to the Panel Collection.
protected Map<String, Map<String, ?>> data	This member variable is meant for holding data from the Panels within the Panel Collection. The toJSON() method populates it. Implementations may change what's in this object by overriding getData() .
public Map<String, ?> getData()	The default implementation returns the member variable 'data'. A subclass of ParameterPanelCollection may override to modify it. toJSON() converts whatever getData() returns to JSON.
public JSONObject toJSON()	This method converts everything in the Panel Collection to an org.json.JSONObject (Jackson) object. The object attributes are name, description, panels, data and parameterPanelCollectionClassName . <ul style="list-style-type: none">The panel attribute holds a JSONArray of JSONObject's returned by each ParameterPanel.
protected byte[] getFile(Integer fileId) protected String getText(Integer textId)	If the ID is known, these methods allow a file or a CLOB to be read from the Yellowfin configuration database. The ID may be stored in the component's configuration. In case of Data Transformations, this would be a Step Option.
public List<ParameterDisplayRule> getDisplayRules()	This method need not be implemented at this level. Yellowfin currently does not use these rules, but may do so in a future release.

Implementation

Yellowfin includes an implementation of **ParameterPanelCollection** for use with the Data Transformation steps. The **ETLStepPanels** concrete class is easy to use. It contains the **addPanels()** for adding **ParameterPanel** objects.

```
ETLStepPanels panels = new ETLStepPanels();  
panels.addPanel(panel);
```