

Kepler - Bug #2966

Add a 'Preview data' menu to data source actors

09/11/2007 06:01 PM - ben leinfelder

Status:	Resolved	Start date:	09/11/2007
Priority:	Normal	Due date:	
Assignee:	ben leinfelder	% Done:	0%
Category:	data access	Estimated time:	0.00 hour
Target version:	SanParksRelease	Spent time:	0.00 hour
Bugzilla-Id:	2966		
Description			
Add a 'Preview data' menu to data source actors to show the content of data table(s) a) Only shows the top 20 or so records of the table (easier) b) maybe only shows user-selected columns (more difficult from UI perspective)			
Related issues:			
Blocked by Kepler - Bug #2967: Add a 'View data' menu to data source actors			Resolved 09/11/2007

History

#1 - 12/31/2007 11:20 AM - ben leinfelder

Did an initial pass at this for EML200DataSource actor. Uses a jTable (with simple sorting). Have not tried it with *big* datasets - might encounter some problem there - all in memory.  
Preview is accessed from right-click context menu on actor. The org.kepler.actor.preview.Previewable interface should be implemented by any actor that wishes to expose a "Preview" of some sort. public void preview() method is the only requirement for implementing the interface (thus far).  
If there is an attempt to preview a non-Previewable actor, a warning message is displayed indicating that previews are not yet supported for that type of actor.

#2 - 01/02/2008 04:29 PM - ben leinfelder

Ported the PersistentTableModel from Morpho to make use of the [already] ported version of PersistentVector. This supports big big big datatables by using the disk with some records in memory.  
Note: this table model uses strings for all attributes. Since this is a "Preview" that may not be the biggest deal, but it does mean numbers may not be sorted as desired (i.e. 100 < 2 when you use string comparison)

#3 - 01/04/2008 12:12 PM - ben leinfelder

Added a recast() method to the PersistentTableModel implementation. This allows us to make an initial pass over the columns to divine the type correctly (which in turn allows the cells to be rendered with right-justified numbers, and left-justified strings and also to be sorted appropriately).  
Note: Since there is an initial pass over *every* cell in the table to divine the type of the column, there could be some overhead when dealing with large datasets.  
Note 2: "Missing values" are not really supported in that if a column of numbers includes something like "-" or "N/A" for a missing value then the entire column will be treated as a string (for display and sorting).

#4 - 02/05/2008 09:54 AM - ben leinfelder

this change has been included as part of 1.0.0RC1  
additional bugs have been reported in a different bugzilla entry.

#5 - 03/27/2013 02:21 PM - Redmine Admin

Original Bugzilla ID was 2966