Drag and Drop

AWT Data Transfer

Pre-reading

Please refer to http://java.sun.com/j2se/1.5.0/docs/api/java/awt/datatransfer/package-summary.html for general information on AWT data transfer. See the drag-and-drop specification and tutorial at http://java.sun.com/j2se/1.5.0/docs/guide/dragndrop/index.html.

Data Flavors

The mapping between the Java type and content type is the `SystemFlavorMap`. The mapping between content type and native format is hard-coded as long as each native format needs some code to perform the conversion. Harmony implementation does the data conversion is performed in two steps: from Java to intermediate format (which is platform-independent, and is one of content types listed above), then from intermediate format to the platform-dependent native format. The conversion from native format to Java is performed also in two steps: from native to intermediate, then from intermediate to Java.

The table below shows the mapping of data flavors to supported native Windows formats.

Java type	**Content type**	**Native format on Windows**		
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="462cc79c-16e5-4959-9cc5-b9dfff248a40"><ac:plain-text-body><![CDATA[	`String` `char[]` `[InputStream]`	text/plain	`CF_TEXT` `CF_UNICODETEXT`	]]></ac:plain-text-body></ac:structured-macro>
`Image`, `BufferedImage`	image/x-java-image	`CF_DIB` (device independent bitmap) `CF_ENHMETAFILE`		
`java.net.URL`	application/x-java-url	`CFSTR_INETURLW` `CFSTR_INETURLA`		
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="9fe6cf40-8e4f-4f4e-ba63-88d8a7311419"><ac:plain-text-body><![CDATA[	`String` `char[]` `[InputStream]`	text/html	"HTML Format"	]]></ac:plain-text-body></ac:structured-macro>
`java.util.List`	application/x-java-file-list	`CF_HDROP`		
`java.lang.Serializable`	application/x-java-serialized-object	`org.apache.harmony.awt.datatransfer:application/x-java-serialized-object;class=`		

Data structure

The picture below illustrates the structure of the intermediate data.

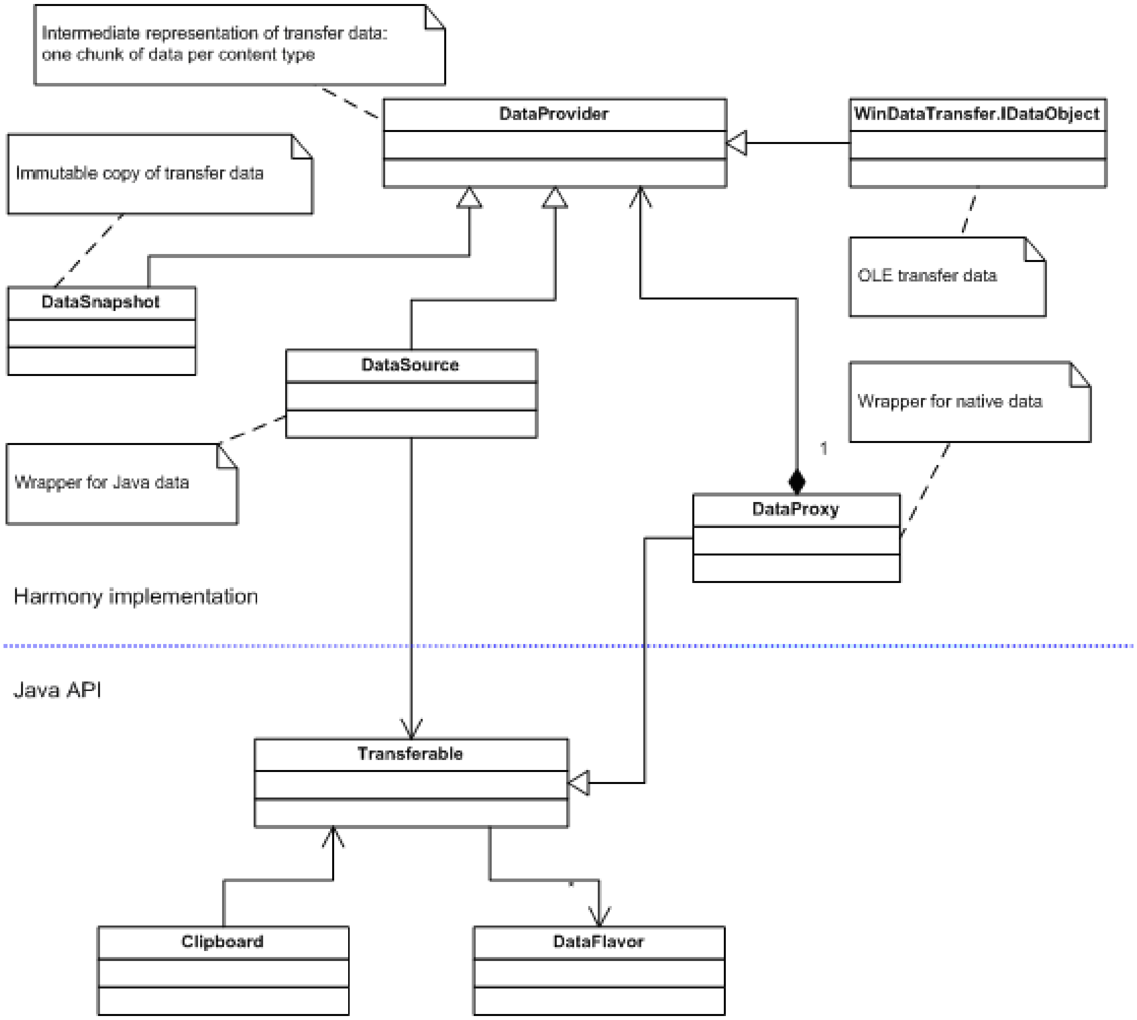
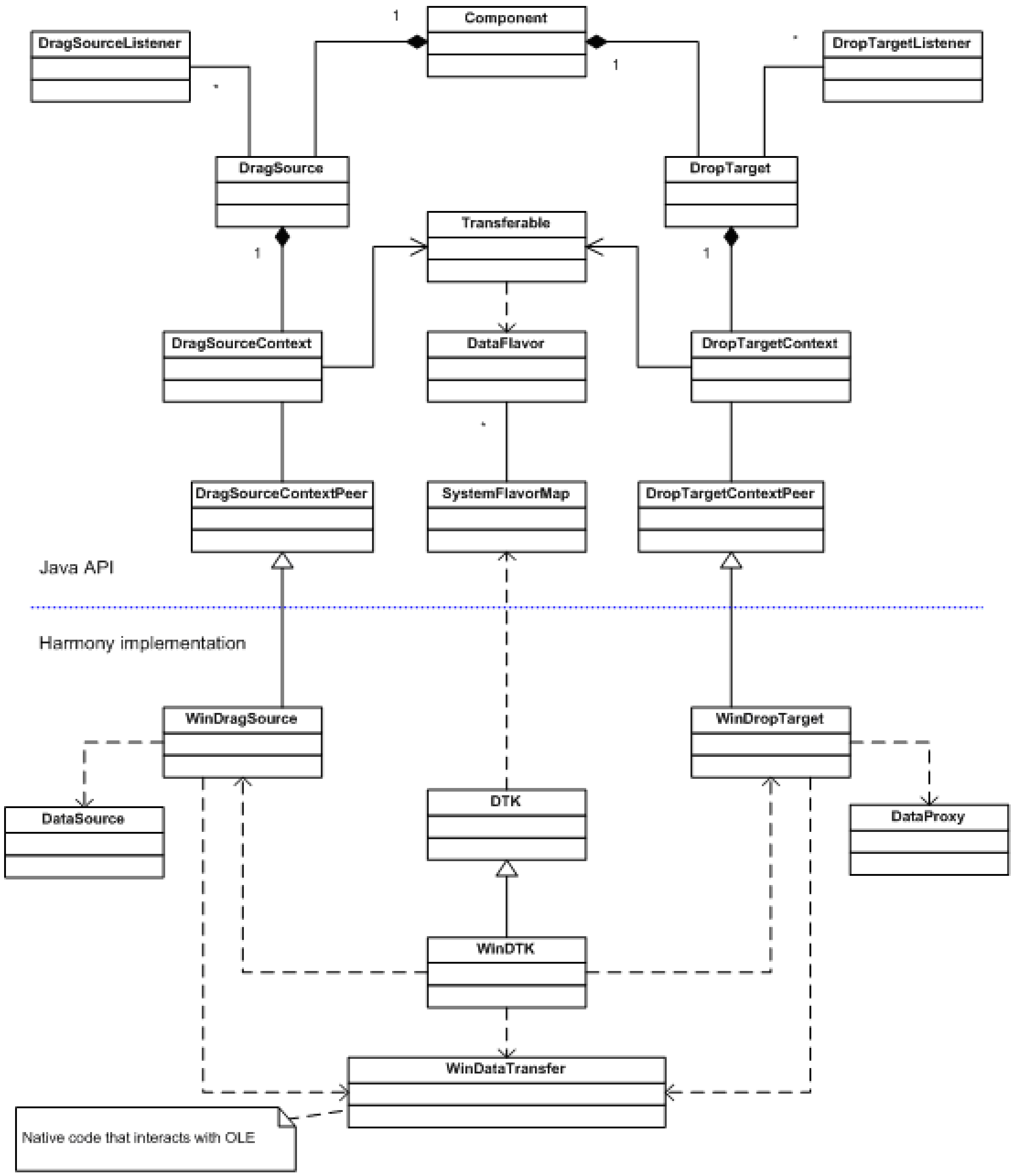


Diagram legend:

- Triangle: ancestor
- Black diamond: owner
- Solid line arrow: reference
- Line without arrow: bidirectional reference
- Dashed line arrow: dependency/usage
- Asterisk: one-to-many relationship

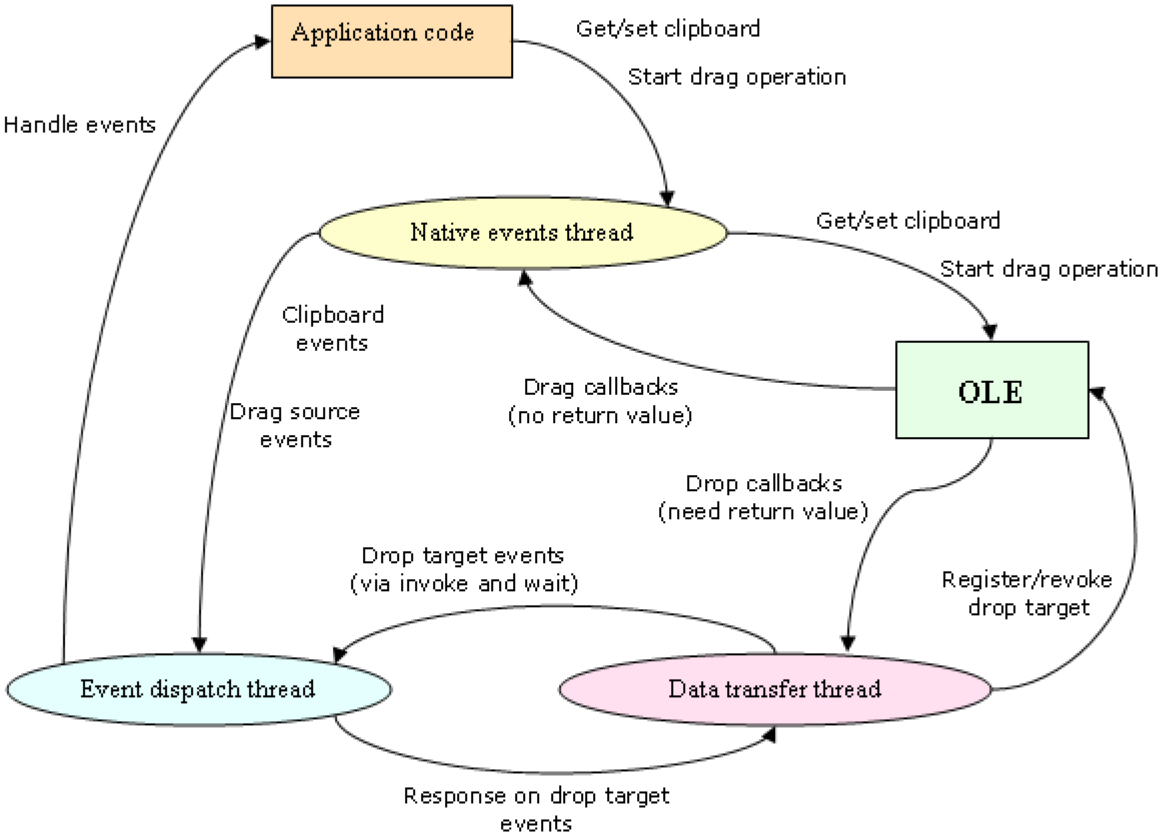
Drag-and-drop subsystem

The diagram below shows the relationship between public API and implementation classes. The classes that have “Win” prefix are for Windows platform, on other platforms there should be similar platform-specific classes.



Event handling and threads

The following figure gives an overview of the message routing and thread interaction in Harmony implementation on Windows.



The things which are missing and/or wanted

Cross-platform

- Clipboard events: flavors changed, ownership lost
- Drag source: setCursor() – use custom cursors for copy, link or cut operations.
- Drop target: acceptDrag() and rejectDrag()
- Drop target for lightweight components: track for mouse enter/exit, post appropriate events
- Drag-and-drop: detect local (in-process) transfer and do it in simplified way
- System flavor map: add system locale's default charset to the list of available charsets

Windows

- Handle clipboard messages
- Use cached copy of transfer data if the clipboard contents wasn't changed

Linux (X11)

Support system selection

- Get data from selection
- Take/revoke the ownership on selection
- Provide selection data for other applications
- Multi-format data handling
- Time-out detection and handling

Selections are

- PRIMARY – for simple applications
- CLIPBOARD – for fully functional copy & paste
- Xdnd – for drag-and-drop operations

Support drag-and-drop (Xdnd)

- State machine for Xdnd protocol

- Implement AWT drag-and-drop peers, post the events to event queue
- Time-out detection and handling

Implement data conversion from/to native formats

Please refer to http://www.freedesktop.org/wiki/Standards for X11-related standards and specifications.

DnD Development Plan

AWT/Windows (6w)

- Implement, support with tests and debug unimplemented methods of the following classes:
  - `java.awt.dnd.DragSource`
  - `java.awt.dnd.DragSourceDragEvent`
  - `org.apache.harmony.awt.datatransfer.DataSnapshot`
  - `org.apache.harmony.awt.datatransfer.windows.WinClipboard`
  - `org.apache.harmony.awt.datatransfer.windows.WinDragSource`
  - `org.apache.harmony.awt.datatransfer.windows.WinDropTarget`
- Debug classes:
  - `java.awt.dnd.DropTarget`
  - `java.awt.dnd.DragGestureRecognizer`
  - `java.awt.dnd.Autoscroll`
- Debug dragging and dropping of text, a UI element, a palette element, an image, a custom object in order.

Swing/Windows (6w)

- Implement, support with tests and debug unimplemented methods of `javax.swing.TransferHandler`.
- Implement swing GestureRecognizer.
- Implement DnD support and write corresponding tests for 46 components.
- Implement and debug mouse tracking.