Operational

Requirements

Specifications

Of the Eclipse inkstone Plugin

Document version : 2. 5

Document date : 25/09/2013

Author : Ofer Calvo

Version remarks : Requirements update at the “INK Properties View” requirements.

Content List:

1 Abstract 1

1.1 Abbreviations list 1

1.2 Definitions 1

1.3 Applicable Documents 1

2 Operational principles 1

3 Graphic interface design 1

3.1 Plugin views 1

3.1.1 INK Properties View 1

3.1.2 INK Kiosk View 1

3.1.3 INK Diagram View 1

3.2 Plugin dialogs 1

3.2.1 INK Diagram Deletion Warning 1

3.2.2 The inkstone general error message 1

3.2.3 DSL’s Selection Dialog 1

3.2.4 The inkstone Preference page 1

3.3 Plugin Action Commands 1

3.3.1 INK Toolbar group 1

3.4 Plugin perspective 1

4 Appendixes 1

4.1 Appendix A – INK Diagram 1

4.2 Appendix B – Operational scenarios 1

4.2.1 Scenario: Make project-based INK diagram 1

4.2.2 Scenario: Design INK-based with visual notations. 1

Figure List:

[Figure 1 – INK Properties View 1](#_Toc325923565)

[Figure 2 – INK Kiosk View 1](#_Toc325923566)

[Figure 3 – INK Diagram View 1](#_Toc325923567)

[Figure 4 - INK Diagram with compilation error 1](#_Toc325923568)

[Figure 5 – Confirm dialog of INK diagram deletion 1](#_Toc325923569)

[Figure 6 - Plugin general error message 1](#_Toc325923570)

[Figure 7 - INK Diagram Preference page 1](#_Toc325923571)

[Figure 8 - INK Kiosk Preference page 1](#_Toc325923572)

[Figure 9 - INK Toolbar group 1](#_Toc325923573)

[Figure 10 - Diagram filter options tree 1](#_Toc325923574)

[Figure 11 – INK Perspective 1](#_Toc325923575)

Table List:

[Table 1 - INK properties interactions 1](#_Toc325923576)

[Table 2 - INK kiosk interactions 1](#_Toc325923577)

[Table 3 - INK diagram interactions 1](#_Toc325923578)

[Table 4 - Actions commands 1](#_Toc325923579)

# Abstract

The inkstone project is about extending INK workbench project to enable graphical editing of INK metadata, using diagram drawing of notations (INK elements and relations between them).

The project goal is to simplify DSL design with INK, to domain specialists that are not programmers, and to enable modeling visual language between people that uses INK.

This paper is a specification document that lists the operational requirements of inkstone.

The Operational requirements will define:

* Operational principles (see chapter ‎2).
* GUI design (see chapter ‎3).
* INK notations (see Appendix B – INK Diagram, chapter ‎4.2).
* Operational scenarios (see Appendix B – Operational scenarios, chapter ‎4.3).

And, be a reference document to the functional requirements and the design of the plugin.

## Abbreviations list

|  |  |
| --- | --- |
| **DSL** | Domain Specific Language |
| **GUI** | Graphical User Interface |
| **SDL** | Simple Declarative Language |
| **SVG** | Scalable Vector Graphics |

## Definitions

|  |  |
| --- | --- |
| **INK Element** | A graphical shape used to note: INK meta-class, INK class or INK Object.  More in appendix A… |
| **INK Relations** | A graphical relation line used to note: Extension, Instancing or Referencing.  More in appendix A… |
| **INK File** | A text file with “\*.ink” suffix. Written in SDL format. |
| **Swing** | Part of Oracle's Java Foundation Classes (JFC), an API for providing a graphical user interface (GUI) for Java programs. |

## Applicable Documents

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Document Name** | **Version** | **Date** |
|  | inkstone FRS | 2.0 | 25/09/2012 |

# Operational principles

The graphical interface of inkstone should be based on the Eclipse development environment visual GUI components.

It shall be simple and fast to operate. Meaning, no more than 3 hits/clicks actions per each functionality usage.

# Graphic interface design

## Plugin views

### INK Properties View

Normal icon:  Error icon: 

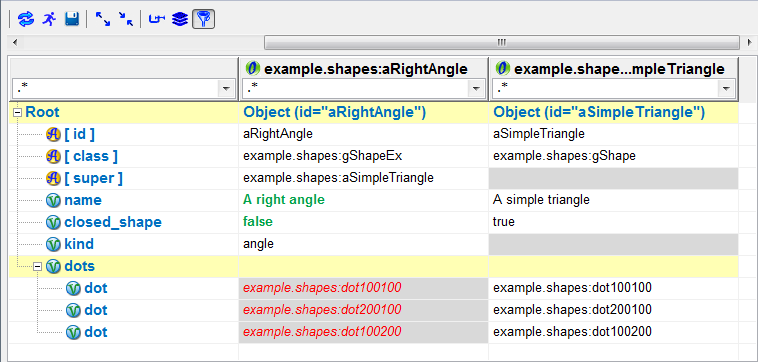


Figure 1 – INK Properties View

This view shall be used to edit selected INK elements properties. The view will be called “properties view” in this paper terminology.

The properties view will show the element hierarchy in the most left grid column, the selected element content in the second column beside the tree, and after that all the parents (“supers”) content in the continued columns.

The selected INK element inheritance data will be revealed if the user drags the upper slider to the right, and hidden when he drags it to the left (default first draw shell be the hidden inheritance, slider to the left).

On INK compilation error, the properties view icon will be replaced into the error icon (as shown above).

For more examples that helps illustrate the view requirements, please see Appendix A – INK Properties Diagram.

#### Popup menu options

This view shall support the next popup menu options, as shown in Figure 2:   
(Keyboard alternatives: mnemonics and accelerators are customary marked)

**A row popup menu**, to be available in each row:

* **Refresh (F5)** – Refresh all properties.
* **Go to Element (F3)** – Opens the INK file, which contains the INK element that is presented in the  
   properties view, on the elements declaration.
* **Collapse All (Alt "–")** – Close all properties rows.
* **Expand All (Alt "+")** – Open all properties rows.
* **Delete (Del)** – Deletes current selected node/cell row from the view.
* **Add (Alt A)** – Add new node/cell row to current selected position.  
   Contains a dynamic sub-menu with the current selected node available additions.

**A header popup menu**, to be available in each column header:

* **Go to Element (n/a)** – Opens the INK file, which contains the INK element that is presented in the  
   current properties tree column under the mouse.

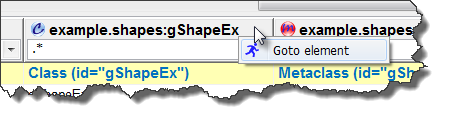
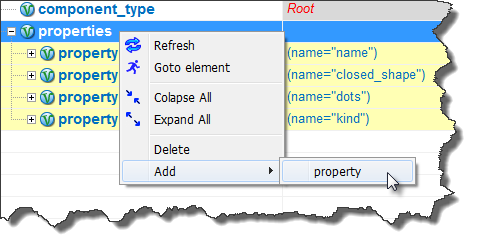


Figure 2 – Cell & Header popup menu

#### Mouse buttons options

This view shall support the next mouse commands:

* **Right mouse button** – One click opens the view popup menu/s.
* **Left mouse button** – One click selects one cell in the properties view.  
   Double click open a cell editor for manual edit (if cell not blocked for a change).

#### Views Interactions

|  |  |
| --- | --- |
| When … | Do … |
| INK element is selected in the “INK Kiosk view” | Show its properties from the INK core API. |
| Selected element properties changed by the user and saved. | Save the element to the INK model.  Update view icon if error reported from INK Plugin core. |

Table 1 - INK properties interactions

#### View Tool Bar



Figure 3 - The properties view tool bar

|  |  |
| --- | --- |
| Icon | Role |
| C:\Users\OC\Pictures\paint_net\refresh.png | Refresh manually the properties tree (for instance, after manual editing the element in the INK textual editor).  Should be implement by a push-button. |
| C:\Users\OC\Pictures\paint_net\runman.png | Opens the INK file, which contains the selected element.  Should be implement by a push-button. |
| C:\Users\OC\Pictures\paint_net\save_icon.png | Save changed cell data to the data mode of the INK add-in (action will not include changes to the language base INK elements).  Should be implement by a push-button. |
| C:\Users\OC\Pictures\paint_net\expand.png | Expand all properties (open) rows in the properties tree.  Should be implement by a push-button. |
| C:\Users\OC\Pictures\paint_net\collaps.png | Collapse all properties (close) rows in the properties tree.  Should be implement by a push-button. |
| C:\Users\OC\Pictures\paint_net\clamps.png | Show/Hide empty variables rows of the selected element column (i.e. hide properties that the selected element is not using at all). This case is helpful for the declarative INK elements “Class” and “Meta Class”.  This function use should be to reduce the “Information overload” effect when the properties tree is large with none used properties by the selected elements for inspection.  Should be implement by a 2 state check-button (On/Off). |
| C:\Users\OC\Pictures\paint_net\Layers.png | Show/Hide inherited variables rows in the selected element column.  Should be implement by a 2 state check-button (On/Off). |
| C:\Users\OC\Pictures\paint_net\filter.png | Show/Hide The columns headers [regular expression](http://docs.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html" \l "sum) filter boxes (“Regex”).  Each box should have a Regex string that together (using the logical ***AND*** operator) should define a combined filter to the properties grid view.  Hiding the filter boxes cancels the filtered view (show all rows).  Showing the filter boxes resume last defined combine filter on the properties view (default Regex string should be: “**.\***” which means any character, zero or more times).  Removing the filter boxes will also cancel last filtered display, to a normal not filtered grid.  Should be implement by a 2 state check-button (On/Off). |

#### Display of Information

**Terminology for properties rows**:

* **Node** : A property row that show/hide child/s properties of a parent property.
  + GUI indication: A colored background row in the properties tree (can be expand or collapsed).
  + At the INK syntax, a word that followed by the scope marks by curly brackets ‘**{...}** ‘.
* **Attribute** : A property that describes an INK language characteristic value of node property.
  + GUI indication: marked by the C:\Users\OC\Pictures\paint_net\A Marble.png icon.
  + At the INK syntax, an equal clause ‘a=b’ between the node name and the curly brackets.
* **Variable** : A property that holds data child property of a node.
  + GUI indication: marked by the C:\Users\OC\Pictures\paint_net\V Marble.png icon.
  + At the INK syntax, an equal clause ‘a=b’ inside the node’s curly brackets ‘**{...}** ‘scope.

**Column data display rules**:

* Inheritance element columns should be disclosed by using the upper slider. Inheritance data columns display will be according to the “super” marking words in INK. Each column will represent an INK element that is a “super element” of the element in the column to its left. Missing “super” reference (or a cyclic reference) will bring this, sequence of inheritance display, to its end.
* Columns properties view rows should be display according to this cases:
  + On instance elements (regarding INK objects and INK enumerators) the rows will be a sum collection of all the INK class type properties, if used or not by the inheriting elements.  
    (The “class” attribute in the element is its type).
  + On declarative elements (regarding INK class & INK meta-class) the rows will be sum of all the used and the mandatory variables and sub properties scopes, from all the columns of elements property variables. Optional properties will be omitted if not used by the selected declarative element.
* Column variable data values will be formatted according to this default styling rules:  
   (For example see Figure 4.)
  + Original variable data will be shown in the next format style: some-original-data.  
    (A black on white normal text).
  + Inherited data from element A to B will be shown in the next format style: *some-inherit-data*.  
    (A red on gray italic text).
  + Override data from element A to B will be shown in the next format style: **some-override-data**.  
    (A green on white bold text).
* On node colored row, there should be on every element column (of the selected element and on inheritance parent elements) a variable clause ‘a=b’ to hint the user about the content of that node, without the need to expand it just to see what variables and attributes it contains.  
  For example see Figure 4.  
  The hint property should be selected according to this algorithm:
  + Use the “id” attribute.
  + Use the “name” variable (if “id” not available).
  + Use the “class” attribute (if “id” and “name” not available).
  + Use the first attribute (if “id”, “name” and “class” not available).
  + Leave empty if also no attribute exists for the node.

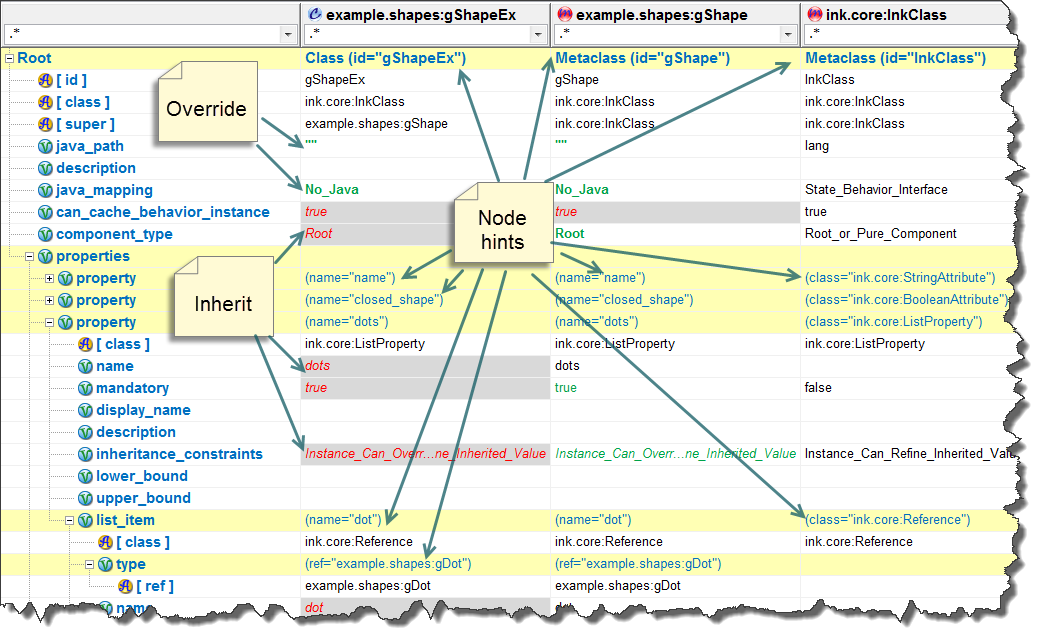


Figure 4 - Node hints example

**Editing and update data**:

* User Controls on the properties view:
  + Mouse Click (As describes above in ‎3.1.1.2)
  + Keyboard
    - Arrows to navigate the cells (selected cell changed cordially).
    - Enter Key to open a cell editor for manual edit (if cell not blocked for a change).
    - Esc key to cancel cell change.
* Cell editors:   
  An inner GUI controls that helps edit cell data in the properties view grid.
  + For text data, an in cell editor for text edit with an option for a small multiline expanded note.
  + For numerical data, an in cell editor for numbers.
  + For dates, an in cell combo to open a date picker dialog.
  + For Boolean data, an in cell combo box with (“true” or “false” values).
  + For Enumeration lists, an in cell combo with the defined ENUM items in it.
* Blocked cells  
  Cells which are protected against user edit actions.
  + Node caption text (most left column).
  + Node hints (auto calculated cells).
  + Columns names.
  + Columns with INK language base elements in them (like “ink.core.InkClass”).

### INK Kiosk View

Normal icon: 

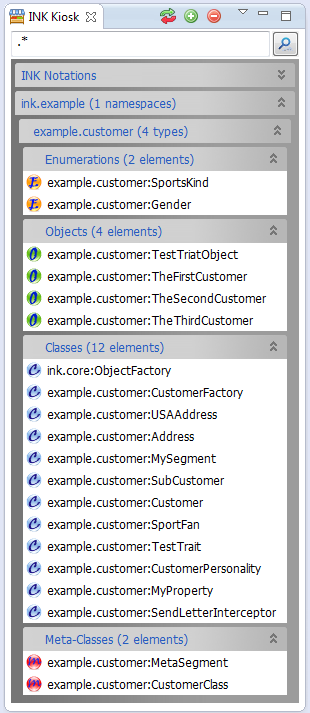
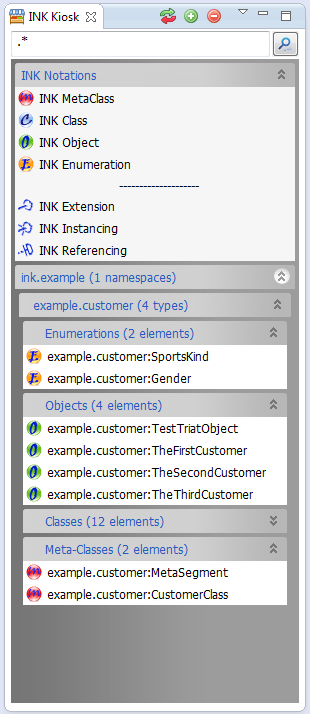
 

Figure 5 – INK Kiosk View examples

The kiosk view usage will be to store reusable notations of elements, to be dragged into the INK diagram view.

The view shall be built from a collapsible panel sections. The first mandatory section is dedicated to store basic INK notation for drawing of new diagram elements and relations. The other optional sections are dedicated to display a static display of currently selected INK elements. This while using hierarchy display of INK projects 🡪 name spaces (DSL’s) 🡪 INK type. INK types are: Meta-Class, Class, Object and Enumeration.

#### Popup menu options

This view shall support the next popup menu options:   
(Keyboard alternatives: mnemonics and accelerators are customary marked)

Expand-bars title menu:

* **Collapse All** **(Ctrl C)** – Open all the view sections.
* **Expand All** **(Ctrl E)** – Close all the view sections.

Expand-bars element list menu:

* **Open element** **(Ctrl O)** – Open the INK core text editor, at the definition of the selected element.

#### Mouse buttons options

This view shall support the next mouse commands:

* **Right mouse button** – Open the expand-bars title, or the expand-bars element list, popup menu.

#### Views Interactions

|  |  |
| --- | --- |
| When … | Do … |
| Dragging INK notation from the view into the INK diagram view. | Place new diagram notation in the diagram. |
| INK element is selected in the "INK Kiosk view" | Show the selected element properties at the "INK diagram view" and (if exist) mark the element in the current display of the “INK Diagram View”. |

Table 2 - INK kiosk interactions

#### View Tool Bar

|  |  |
| --- | --- |
| Icon | Role |
| C:\Users\OC\workspace\ink.inkstone\src\inkstone\utils\gallery\add_icon.png | Opens for selection of the INK projects & libraries tree, from the current active INK model. Selected libraries elements shall fill the Kiosk bars, after closing the dialog.  (See paragraph ‎3.2.3 DSL’s Selection Dialog) |
| C:\Users\OC\workspace\ink.inkstone\src\inkstone\utils\gallery\delete_icon.png | Clear Kiosk view. Leaving only the basic INK notation bar. |
| C:\Users\OC\workspace\ink.inkstone\src\inkstone\utils\gallery\refresh_icon.png | Refresh current Kiosk view data. Update Kiosk static display from the current saved INK model. |

#### Display of Information

The Kiosk view shall use two display methods to help dealing with “Information Overload”.

Firstly, with the view display of selected DSL’s elements (from INK model) in tree-like order of cascading bars.  
Each bar shall be titled with the project/library/INK kind name, plus the number of sub instance in brackets. And all project/name-spaces/elements names should be alphabetically ordered.

Secondly, with the help of a filter tool, at the top of the view, that should hide display elements according to a [regular expression](http://docs.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html#sum) string match of the element names.

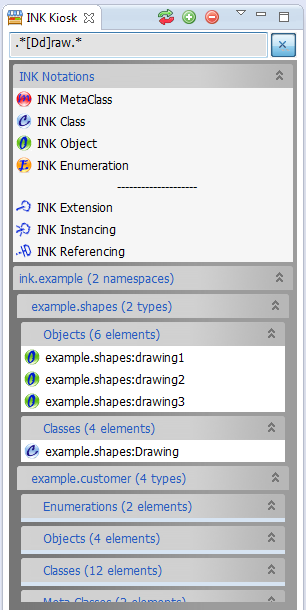


Figure – Kiosk view regex filtering example

### INK Diagram View

Normal icon:  Error icon: 

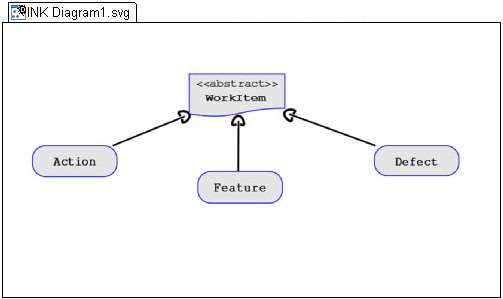


Figure 7 – INK Diagram View

The INK diagram should be built from a Swing component that will be used to display SVG documents graphics.

Each INK diagram will be connected to a single SVG file, which will store the vector graphics data of the diagram.

It will let the user manipulate the displayed SVG document and its elements, with zooming and panning commands.

Selected INK element in the diagram should be visually marked with yellow highlight border and its INK properties should be shown in the INK properties view.

INK compilation errors in the diagram should be marked with a red highlight border around elements with error. When one (or more) of the diagram elements has INK error, the view icon should change to an error icon. Also, the error tooltip should popup when clicking on such an element.

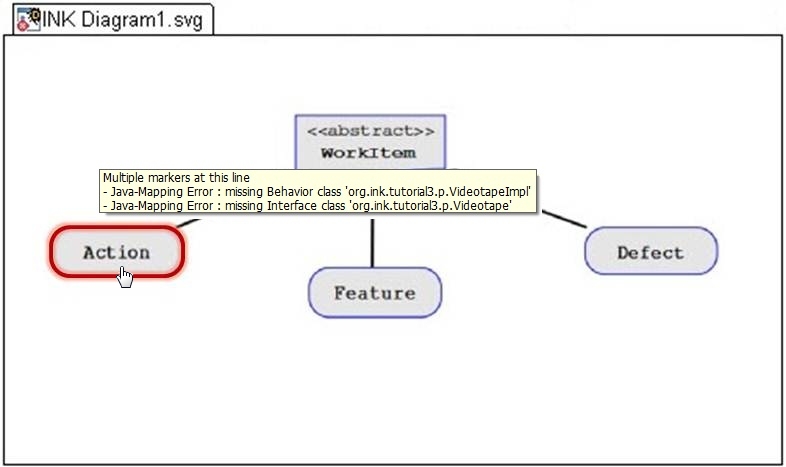


Figure - INK Diagram with compilation error

#### Popup menu options

This view shall support the next popup menu options:   
(Keyboard alternatives: mnemonics and accelerators are customary marked)

* **Add INK Element** **(Alt E)** – Add a new empty elements from the this options:
  + INK Meta-Class
  + INK Class
  + INK Object
* **Add INK Relation(Alt R)** – Add a new empty elements from the this options:
  + INK Extension
  + INK Instancing
  + INK Referencing
* **Delete Notation (Del)** – Deletes current selected notation from the diagram.
* **Zoom** **(Alt Z)** – Zoom diagram options:
  + 25% – Zoom out to one quarter of original size.
  + 50% – Zoom out to half of original size.
  + 75% – Zoom out to three quarters of original size.
  + 100% – Set diagram to original size.
  + 150% – Zoom in to one and half of original size.
  + 200% – Zoom in to double original size.
* **Copy Image** **(Alt C)**  – Copy image to the system clipboard (to paste/export image).
* **Default INK file (Alt H)** – Open project file dialog tree, to select a new default INK file to the diagram.
* **Go to Element (F3)** – Opens the INK file, which contains the currently selected INK element.

#### Mouse buttons options

This view shall support the next mouse commands:

* **Right mouse button** – Open the view popup menu.
* **Left mouse button** – Holds the diagram (while down) for panning with mouse movement.
* **Mouse wheel** – Zoom in/out in the range of 10% to 500% of original size.

#### Views Interactions

|  |  |
| --- | --- |
| When … | Do … |
| Dragging INK element around in the diagram. | Change element location and relation lines path to/from it when the all object is selected.  Resizing element size when one corner is selected.  Show element properties in the INK Properties view. |
| Drop INK **element** from the kiosk view. | Add new element to the diagram and show its properties in the INK Properties view. |
| Drop INK **Relation**, from the kiosk view, on INK element. | Change mouse icon to a cross and wait to a second left mouse click on the targeted element in the diagram.  Place the relation line if element selected, or cancel operation if "Esc" key is pressed. |
| Dropping **INK file** from the Package Explorer view, into the INK Diagram view. | Add all none already displayed elements to the diagram. |

Table 3 - INK diagram interactions

#### View Tool Bar

<TBD>

#### Display of Information

<TBD>

## Plugin dialogs

### INK Diagram Deletion Warning

Usage: confirm an INK diagram deletion. See FRS scenario: “Remove INK Diagram”.

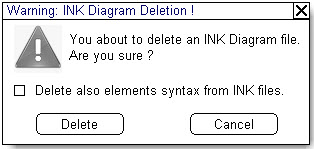


Figure 9 – Confirm dialog of INK diagram deletion

### The inkstone general error message

Usage: Display general simple error messages. The label will be the calling object (general the view) and the message will be an input from that calling object.



Figure 10 - Plugin general error message

### DSL’s Selection Dialog

Usage: Opens for selection of the INK projects & libraries tree, from the current active INK model.  
Selected libraries elements shall fill the Kiosk bars, after closing the dialog (Done button).

INK libraries are also called “DSL’s” in INK.

The tree should have check-boxes for each project/library. Select all, and Deselect all buttons should be placed in order to make mass selection more easy.

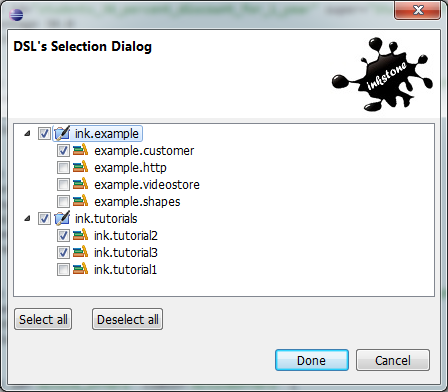


Figure – DSL’s Selection Dialog example

### The inkstone Preference page

A preference is data that is persisted between work sessions. The plugin preference will be integrated with Eclipse preference dialog and will be on a per project basis.

#### INK Diagram Preference

Set the project folder for INK Diagrams and the default INK file for every INK Diagram.

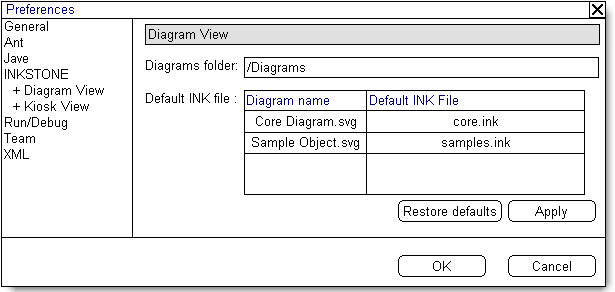


Figure - INK Diagram Preference page

The Apply button is to submit changes.

The Restore defaults button set the diagram folder back to “Diagrams”.

#### INK Kiosk Preference

Set the linked INK files of the current project.

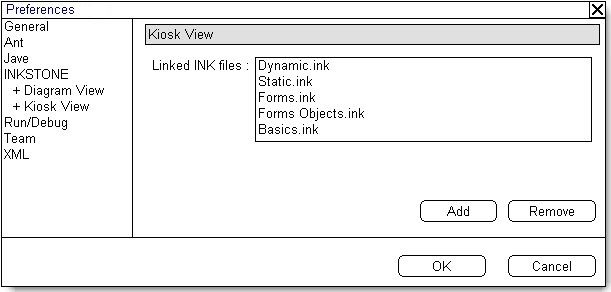


Figure - INK Kiosk Preference page

The Add button opens tree dialog of projects INK files, and add them to the list.

The Remove button deletes current selected rows from the list.

## Plugin Action Commands

The plugin action commands will use the Eclipse environment GUI menus and toolbar icons (an inkstone toolbar group adding to the Eclipse main toolbar).

|  |  |  |  |
| --- | --- | --- | --- |
| Command | Workbench Menu/s | Active when… | INK Toolbar |
| Add INK Diagram view | Ink menu.  File/New menu. | Always |  |
| Show INK Kiosk view | Ink menu.  File/New menu. | Kiosk view closed |  |
| Show INK Properties view | Ink menu.  File/New menu. | Properties view closed |  |
| Diagram Filter tool | Ink menu. | Diagram view in focus | C:\Users\OC\Documents\אופ תואר שני\פרויקט מתקדם במדעי המחשב - 22997\הצעה\ORS\graphics\Edit Filter icon.gif |
| Show inkstone perspective | Window/Open Perspective menu. | inkstone perspective not selected |  |

Table 4 - Actions commands

### INK Toolbar group

The toolbar group added to the workbench toolbar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | ▼ |

Figure - INK Toolbar group

#### Diagram Filter Tool

A drop down toolbar menu button with a tree of check-box option to mark the visible /none visible INK notations display. None active notations will be hidden from the current selected INK diagram.

Each INK diagram shall have its own saved filter set of these options.

Every combination is allowed, but relations with hidden source or target elements will be hidden as well, even if marked as visible.

-Elements

🗹 Meta Class

🗹 Class

🗹 Object

- Relations

🗹 Extension

🗹 Instancing

🗹 Referencing

Figure 15 - Diagram filter options tree

## Plugin perspective

The plug in perspective shall set a graphical layout to INK development with a set of inkstone views layout.

It will be called: “INK”, and could be found in the Eclipse perspectives list.

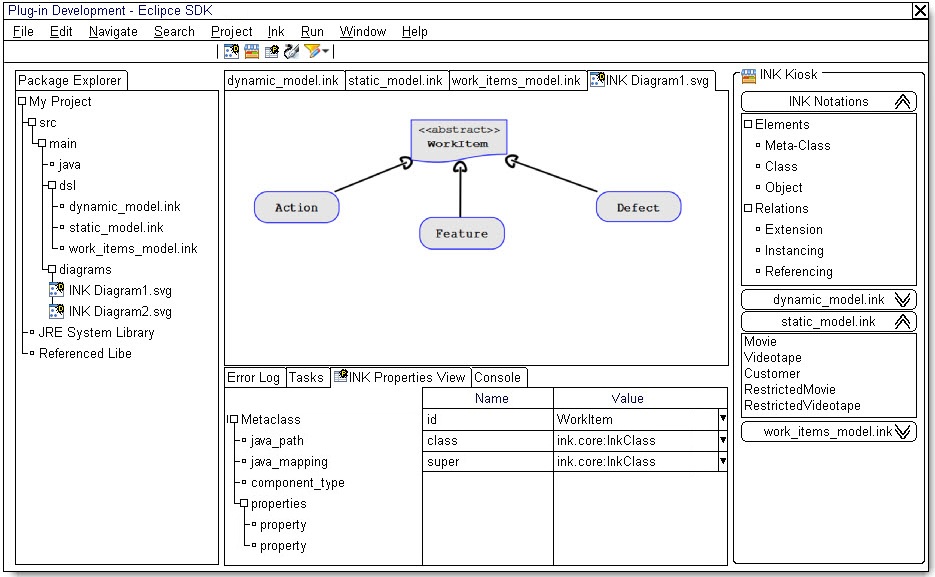


Figure 16 – INK Perspective

# Appendixes

## Appendix A – INK Properties Diagram

The INK properties view shall be used to edit selected INK elements properties.

The next (imbedded) presentation adds more examples that helps illustrate the view requirements at paragraph ‎3.1.1.



## Appendix B – INK Diagram

The INK diagram is a graphic modeling language designed to suite the INK syntax with similarity to UML.

The next (imbedded) presentation describes the different types of INK notations with examples.



## Appendix C – Operational scenarios

These next informal scenarios define operational usage of the inkstone plugin. They should clarify the workflow of the user with the plugin.

### Scenario: Make project-based INK diagram

1. Open existing java project in Eclipse.
2. Create new INK diagram and select one of existing INK files as its default.
3. Drop all needed INK files into the diagram.
4. Wait to the plugin to open INK Kiosk, add all files to the kiosk and add all INK elements from files to the INK diagram.
5. Fix elements locations in the diagram. Remove, or filter out, unnecessary INK elements.
6. Set Diagram zoom and panning.
7. Copy INK diagram image to your office suite application for presentation or documentation needs.

### Scenario: Design INK-based with visual notations.

1. Open existing java project in Eclipse.
2. Open the INK perspective layout.
3. Add new INK Diagram. Select yours new INK file as its default.
4. Drag the relevant INK files, needed, from the Package Explorer view to the INK Kiosk view.
5. Drag (from Kiosk) new elements you design to the newly created INK diagram.
6. Drag existing linked elements (from the Kiosk linked sections) to relate your elements to.
7. Make necessary INK relations to new elements by dragging and assigning new INK relations from the Kiosk to your diagram.
8. Select each new INK elements and their properties types (or values in case of INK objects).
9. Check for errors in the INK properties view, and fix them.
10. The new INK file (diagram default file) is now ready and already saved. You can pass it to peers review and implementation.