

Use of Annotations to Customize EMF Editors

Michael Scharf
Wind River Systems
23 March 2006

EMF Default UI is not user friendly

- Generated editor needs quite some handwork
- Properties View hard to use
- Missing pieces
 - No widget/field enablement
 - No error reporting
 - Not really customizable
- Hand coding is tough

Use EMF Annotations for UI hints

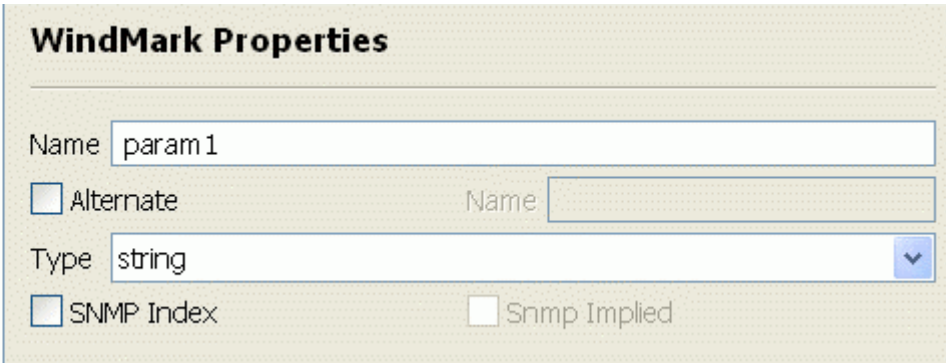
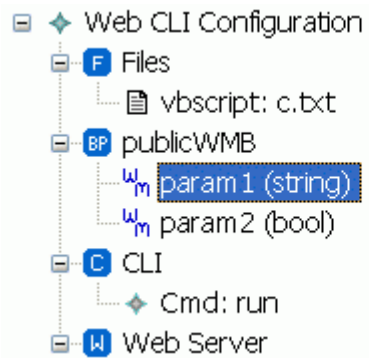
- Rule: Model and UI should be separated
 - That's how I started
- BUT: practically mixing works well
 - Model and UI are in sync
 - All in one place
 - EMFATIC text representation of .emf files

Dynamic versus Generated EMF Objects

- **Dynamic EMF Objects**
 - model is used by external tools
 - Generic UI
 - No changes to standard behavior
- **Generated EMF Objects**
 - Lot of code
 - UI usually works on EMF reflective interfaces
 - Better performance (time and space)
- UIHint is dynamic (no code generation required)

Model Hints

- UIHint
 - Anotation
 - label, text, tooltip..
 - Layout
 - group, separator, tab
 - joinWith, hide
 - Functional
 - listSource
- Enable
 - Condition: OQL query
- Validate
 - OQL query for that attribute



Sample Code

```
@Image (value = "WindMark.gif")
@UIHint (label = "${name} (${type})")
class WindMark {
    @UIHint (banner = "WindMark Properties", separator = "")
    @Validate(condition="value like '^[A-Za-z_][A-Za-z0-9_]*$',
        message="Name must be an identifier!")
    id attr String[1] name;

    @UIHint (label = "Alternate", joinWith = "alternateName")
    attr boolean enableAlternateName = "false";

    @UIHint (label = "Name")
    @Enable (condition = "enableAlternateName")
    attr String alternateName;

    attr WindMarkType type = "string";

    @UIHint (joinWith = "snmpImplied", label = "SNMP Index")
    attr boolean snmpIndex = "false";
    ...
}
```

The screenshot displays the Eclipse IDE interface. On the left, a project tree shows the structure of a 'Web CLI Configuration' project. The tree includes a 'Files' folder, a 'publicWMB' package with two 'WindMark' (Wm) elements named 'param1 (string)' and 'param2 (bool)', a 'CLI' package with a 'Cmd: run' element, and a 'Web Server' package with three elements: 'File: a', 'URL: /NEW/', and another 'URL: /NEW/'.

The main area shows the 'Web Properties' dialog box. It contains several sections for configuring the web server:

- Instantiation:** A checkbox labeled 'Instantiate a standalone Web Server (no management backplane)' is currently unchecked.
- Server Identification:** Includes text boxes for 'DNS Name' (www.target.own), 'Realm' (EnchantedWorld), and 'HTTP URI' (http:).
- WindMark:** Includes text boxes for 'Max Nesting Level' (4) and 'Max Macros In Block' (7).
- Live Control:** A checkbox 'Enable Live Control' is checked. Below it are text boxes for 'Max Clients' (20), 'Max Watch Variables' (40), 'Polling Freq (med)' (2000 msec), and 'Polling Freq (high)' (300 msec).
- HTTP Error Messages:** Includes text boxes for 'Error Header' (<H1>Wind River Web Server Error Report</H1><HR>) and 'Error Footer' (<H3>please mail problems to ???@???.com</H3><HR>).
- HTTP Authentication:** A checkbox 'Enable Javascript Digest Authentication' is checked. Below it are text boxes for 'Login Web Page' (doc/logon.html), 'Authentication Function' (SAMPLE_AuthLevel), 'Idle Timeout' (120 sec), and 'Max Users' (10).
- File System:** Includes a dropdown for 'File Sys Type' (EEnum Literal NVM) and a checked checkbox for 'Compression'.

The screenshot shows the Eclipse IDE's configuration editor for a Wind River Management Backplane. On the left, a tree view shows the project structure: Web CLI Configuration, Files, publicWMB, CLI, and Web Server. The main area is titled 'Backplane Properties' and contains two main sections. The first section, 'Instantiate a Wind River Management Backplane', is selected with a radio button. It includes several text input fields: 'Backplane Name' (publicWMB), 'Deregistration Handler' (NULL), 'Deregistration Cookie' (NULL), 'Request Timeout' (100), 'Initial Max Resources' (1000), and 'Initial Max Components' (12). Below these are three checkboxes: 'Named Pipe Listener' (unchecked), 'Enable Port Listener' (checked), and 'Use Parent Backplane' (unchecked). To the right of the 'Enable Port Listener' checkbox are two input fields: 'Port Number' (0) and 'Name' (empty). The second section, 'Use an External Backplane', is unselected. It contains a 'Non-default Name' checkbox (unchecked) and a 'Name' input field (publicWMB).

The screenshot shows the Eclipse IDE configuration window for the 'Standard Window Manager'. On the left is a tree view under 'Media Library' containing 'System', 'Display', 'BMF Fonts', 'Graphics 0', 'Keyboard /pcConsole/1', 'Pointer /tyCo/0', and 'Standard Window Manager'. The right pane shows settings for 'Window Mgr' (Standard Window Manager), color selection for Active, Inactive, and Root states, and fields for Root Image, Border Width, Min Height, and Min Width. An 'Options' section at the bottom contains four checked items: Enable Splash Screen, Enable Task Bar, Enable Window Captions, and Enable Window Frames.

Media Library

- System
- Display
 - BMF Fonts
 - Graphics 0
 - Keyboard /pcConsole/1
 - Pointer /tyCo/0
 - Standard Window Manager

Window Mgr: Standard Window Manager

Active Color: [Red] Active Text Color: [Yellow]

Inactive Color: [Grey] Inactive Text Color: [White]

Root Color: [Dark Grey]

Root Image: wwmRootImageDib

Border Width: 4

Min Height: 30

Min Width: 60

Options

- Enable Splash Screen
- Enable Task Bar
- Enable Window Captions
- Enable Window Frames