



Extending the JavaScript Development Toolkit

Bradley Childs
IBM Software Group
childs@us.ibm.com
3/19/2008

Agenda

- Overview
- JSDT Feature Highlights
- Benefit of Extending JSDT
- JSDT Platform – What can you extend?
- JSDT Design
- Extension Examples

New User Registration Form

Desired username: This username is in use, or empty username field.

Your name:

Gender: [Select]

Birthday: [Select] - -

E-mail: Invalid e-mail address.

Phone number: Please insert a valid US phone number (xxx-xxx-xxxx).

I've read the Terms of Use

[News](#) [Shopping](#) [Gmail](#) [more](#) ▼

e.g., "10 market st, san francis

[Get directions](#)


My Maps

to your homepage.

My Maps
with your mouse, and double
[our](#) »

[isco](#)

- Video
- Groups
- Books
- Scholar
- Finance
- Blogs
- Calendar
- Photos
- Documents
- Reader
- even more »



JSDT Features

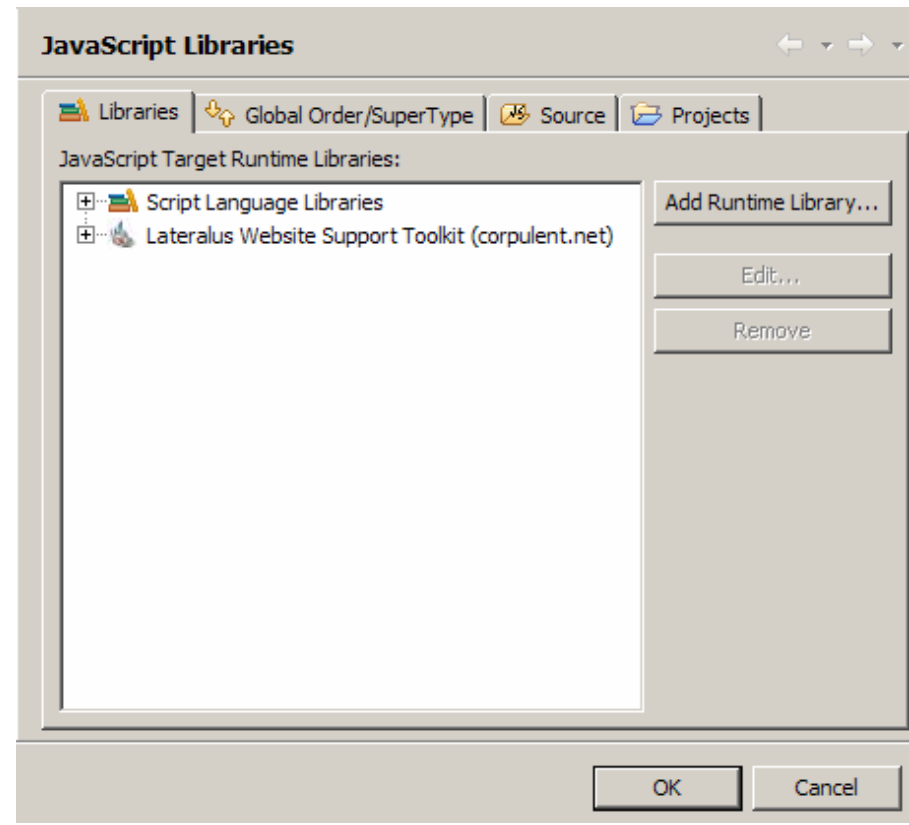
- JavaScript Development Tools (JSDT) is the JavaScript component of WebTools 3.0
 - Replaces old JavaScript editor 2.0
- JavaScript AST Driven Language Model
- JSDT provides rich JavaScript editing capability
 - Content Assit
 - Hover Help
 - Code Completion
 - Validation
 - Refactoring

Benefit of Extending JSDT

- JavaScript as a Platform
- Grow Your JavaScript Technologies With Good Tooling
- Toolkits Dependant on Core JavaScript
- Easy to Leverage Features
- Minimum Effort For Maximum Results

JSDT Platform Features - User Experience I

- UI Extension
 - ◆ Hoverhelp
 - ◆ Wizards (platform)
 - ◆ Quickfix
 - ◆ Validation
 - ◆ Refactoring
 - ◆ Toolkit and Library UI



JSDT Platform Features - Core

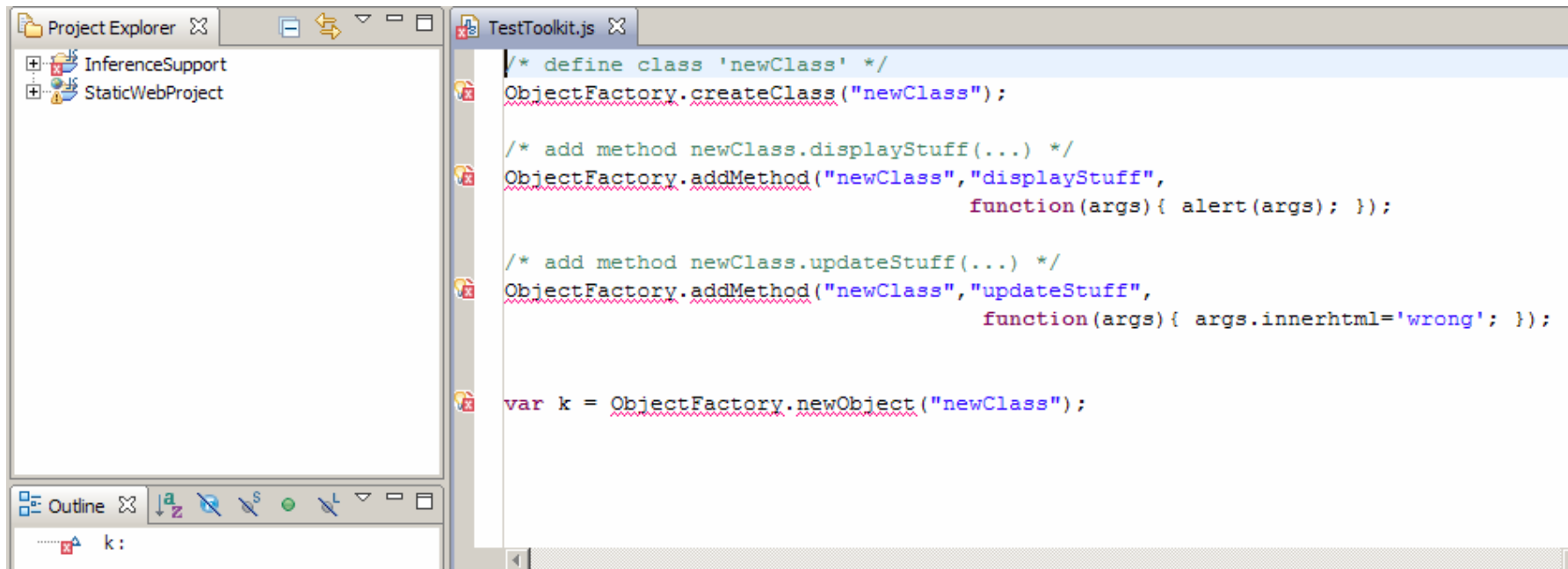
- Core Extensions – **Inferencing**
 - ◆ JavaScript is Ambiguous
 - ◆ JavaScript Flexible to a Fault
 - ◆ Toolkits and Libraries extend JavaScript Language
 - ◆ Proprietary Information about Toolkits Necisary
 - ◆ Class and Type Recognition



Demo Toolkit 'Lateralus'

- **ObjectFactory.createClass(className)**
 - ◆ Defines a new Class
- **ObjectFactory.addMethod(className, functionName, functionBody)**
 - ◆ Adds a method to an already defined class
- **ObjectFactory.newInstance(className)**
 - ◆ Creates a new instance of className

eclipseCON™ 2008



The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays a project named 'StaticWebProject' with sub-projects 'InferenceSupport' and 'StaticWebProject'. Below it, the Outline view shows a variable 'k'. The main editor window displays the file 'TestToolkit.js' with the following JavaScript code:

```
/* define class 'newClass' */
ObjectFactory.createClass("newClass");

/* add method newClass.displayStuff(...) */
ObjectFactory.addMethod("newClass", "displayStuff",
    function(args) { alert(args); });

/* add method newClass.updateStuff(...) */
ObjectFactory.addMethod("newClass", "updateStuff",
    function(args) { args.innerHTML='wrong'; });

var k = ObjectFactory.newObject("newClass");
```


eclipseCON™ 2008

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays a project structure with folders for InferenceSupport, JavaScript Support, and StaticWebProject. The InferenceSupport folder contains a file TestToolkit.js, which has a sub-folder newClass containing a file k. The Outline view below shows the newClass folder expanded, listing displayStuff() and updateStuff() methods, and a variable k of type newClass. The main editor window shows the TestToolkit.js file with the following code:

```
/* define class 'newClass' */
ObjectFactory.createClass("newClass");

/* add method newClass.displayStuff(...) */
ObjectFactory.addMethod("newClass", "displayStuff",
    function(args) { alert(args); });

/* add method newClass.updateStuff(...) */
ObjectFactory.addMethod("newClass", "updateStuff",
    function(args) { args.innerHTML='wrong'; });

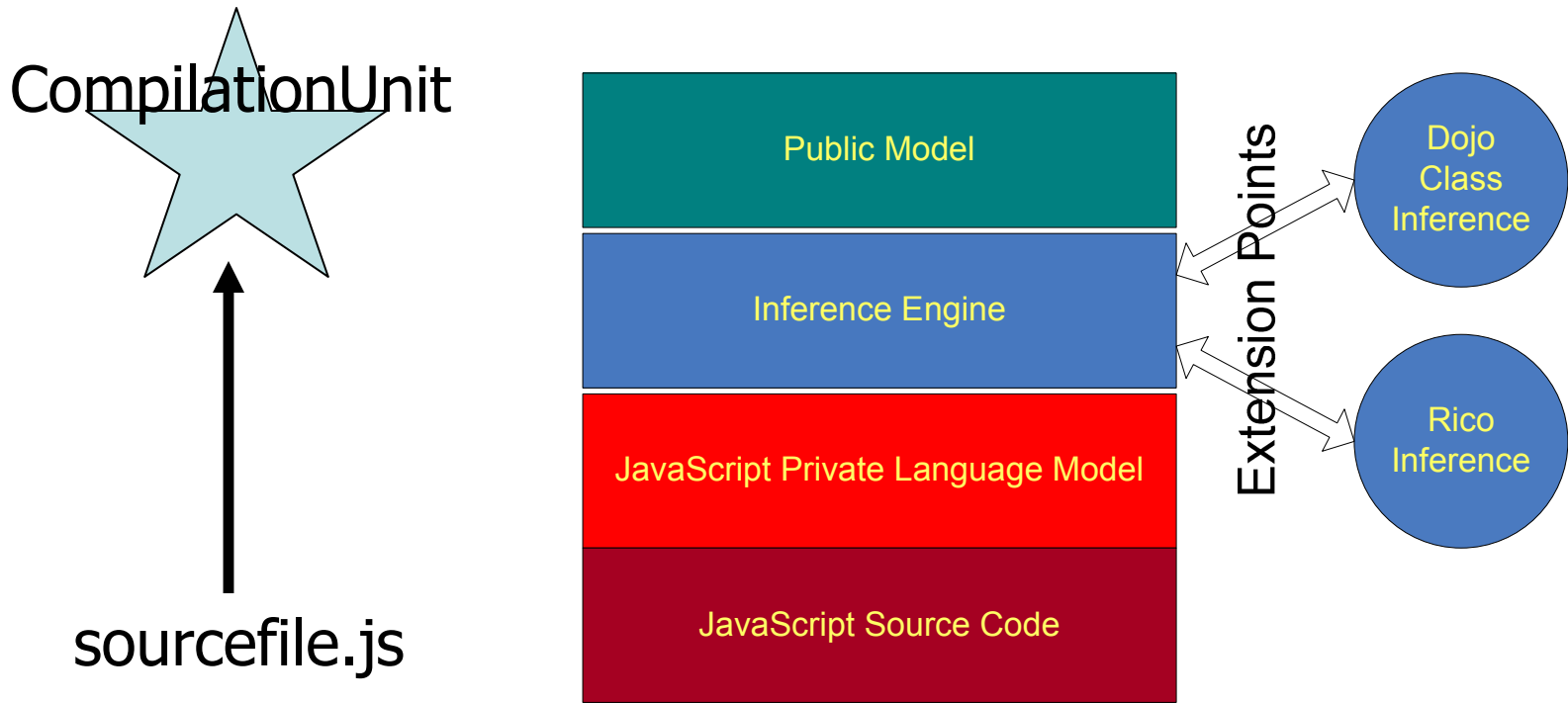
var k = ObjectFactory.newObject("newClass");
```

A tooltip is displayed over the `newObject` method call in the last line of code. The tooltip text is:

[Lateralus Website Support Toolkit]ObjectFactory.newObject(classname)
Defines a new class.
function newObject()
Parameters:
 classname name of class.
Since:
 Version 1.0

At the bottom of the tooltip, it says "Press 'F2' for focus."

Design I – The JavaScript Modeling Process



Extending the JSDT Library Mechanism

- Two Extension Points

- ◆ Libraries Wizard –

org.eclipse.wst.jsdt.ui.JsGlobalScopeContainerPage

- ◆ Resource Management & Global Scope Handling –

org.eclipse.wst.jsdt.core.JsGlobalScopeContainerInitializer





corpulent.net Website Common Library Example I.

- Example of utility JavaScript and common Website headers.
- JavaScript and HTML are copied into the target project. Resources are not shared across projects.
- No Global Scope management. Users must include files in HTML.
- File copy is done in the initializer.



corpulent.net Website Common Library Example II.

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.2"?>
<!-- © Copyright 2008 IBM Corp. All rights reserved. This source code is made available under the terms of the Eclipse Public License,
v1.0. -->

<plugin>
  <!-- JSDT wizard extension -->
  <extension point="org.eclipse.wst.jsdt.ui.JsGlobalScopeContainerPage">
    <JsGlobalScopeContainerPage
      name="corpulent.net common html headers and js"
      class="org.eclipse.wst.jsdt.internal.ui.wizards.buildpaths.CorpulantLibraryWizardPage"
      id="org.eclipse.wst.jsdt.internal.ui.wizards.buildpaths.CorpulantLibrary">
    </JsGlobalScopeContainerPage>
  </extension>

  <!-- JSDT Global Scope Handler -->
  <extension
    point="org.eclipse.wst.jsdt.core.JsGlobalScopeContainerInitializer">
    <JsGlobalScopeContainerInitializer
      class="org.eclipse.wst.jsdt.core.compiler.libraries.CorpulentCommonLibraryInitializer"
      id="org.eclipse.wst.jsdt.launching.corpulant.common">
    </JsGlobalScopeContainerInitializer>
  </extension>
</plugin>
```



corpulent.net Website Common Library Example III.

```
/* © Copyright 2008 IBM Corp. All rights reserved. This source code is made available under the terms of the Eclipse Public License,
v1.0. */
public class CorpulantLibraryWizardPage extends NewElementWizardPage implements IJsGlobalScopeContainerPage,
    IJsGlobalScopeContainerPageExtension, IJsGlobalScopeContainerPageExtension2 {

    private static final String CONTAINER_ID = "org.eclipse.wst.jsdt.launching.corpulant.common";

    public CorpulantLibraryWizardPage() {
        super("CorpulantCommon");
    }

    /* Setup the wizard UI */
    public void createControl(Composite parent) {
        Composite composite = new Composite(parent, SWT.NONE);
        composite.setFont(parent.getFont());
        DialogField field = new DialogField();
        field.setLabelText("corpulent.net common website components added to Project.\n\n - This" +
            "library copies JavaScript elements shared by the corpulent.net website.");
        LayoutUtil.doDefaultLayout(composite, new DialogField[]{field}, false, SWT.DEFAULT, SWT.DEFAULT);
        Dialog.applyDialogFont(composite);
        setControl(composite);
        setDescription("corpulent.net common website components");
    }

    /* return the container id that does the file handling for this library */
    public IClasspathEntry[] getNewContainers() {
        IClasspathEntry library = JavaCore.newContainerEntry(new Path(CONTAINER_ID));
        return new IClasspathEntry[]{library};
    }
}
```

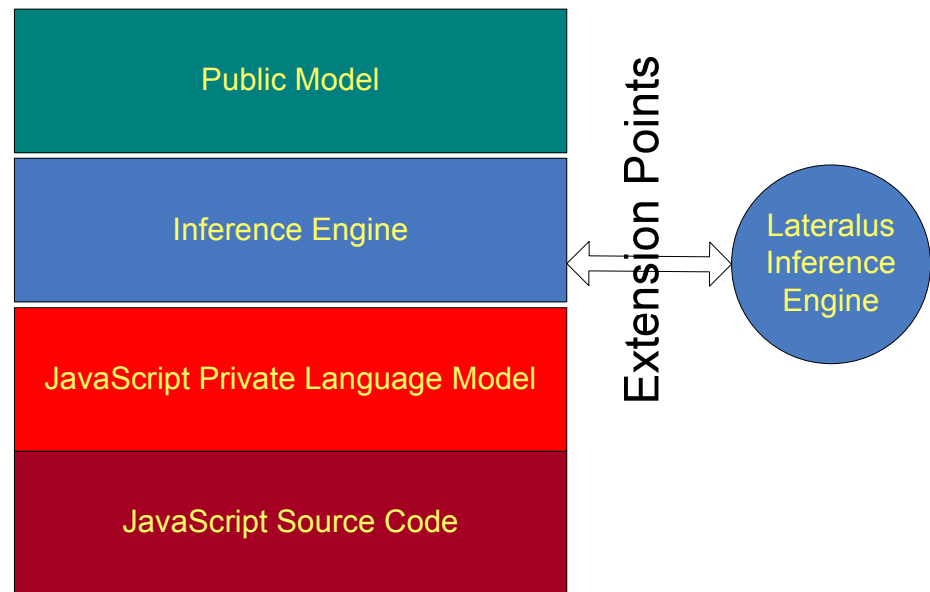


corpulent.net Website Common Library Example IV.

```
/* © Copyright 2008 IBM Corp. All rights reserved. This source code is made available under the terms of the Eclipse Public License, v1.0. */  
public class CorpulentCommonLibraryInitializer extends JsGlobalScopeContainerInitializer implements IJsGlobalScopeContainerInitializerExtension {  
  
    public void initialize(IPath containerPath, IJavaProject project) throws CoreException {  
        copyFiles(project);  
        JavaCore.setJsGlobalScopeContainer(containerPath, new IJavaProject[] { project },  
                                           new IJsGlobalScopeContainer[] { getContainer(containerPath,  
                                                                                   project) }, null);  
    }  
    public void removeFromProject(IJavaProject project) {  
        removeFiles(project);  
    }  
    public int getKind() {  
        return IJsGlobalScopeContainer.K_APPLICATION;  
    }  
    public String getDescription(IPath containerPath, IJavaProject project) {  
        return CorpulentCommonLibraryInitializer.ContainerDescription;  
    }  
    public String getDescription() {  
        /* ... */  
    }  
    public ImageDescriptor getImage(IPath containerPath, String element, IJavaProject project) {  
        /*... */  
    }  
}
```

Advanced Toolkit – Lateralus Toolkit I.

- Similar UI and resource handling as previous examples.
- Provides class inference.



JSDT Core / Inference Support

- Class / Type Inference –
org.eclipse.wst.jsdt.core.inferenceSupport



Advanced Toolkit – Lateralus Toolkit II.

<!-- © Copyright 2008 IBM Corp. All rights reserved. This source code is made available under the terms of the Eclipse Public License, v1.0. -->

```
<!-- JSDT wizard extension -->
<extension point="org.eclipse.wst.jsdt.ui.JsGlobalScopeContainerPage">
  <JsGlobalScopeContainerPage
    name="Lateralus Toolkit"
    class="org.eclipse.wst.jsdt.internal.ui.wizards.buildpaths.LateralusLibraryWizardPage"
    id="org.eclipse.wst.jsdt.internal.ui.wizards.buildpaths.LateralusLibrary">
  </JsGlobalScopeContainerPage>
</extension>

<!-- JSDT Global Scope Handler -->
<extension
  point="org.eclipse.wst.jsdt.core.JsGlobalScopeContainerInitializer">
  <JsGlobalScopeContainerInitializer
    class="org.eclipse.wst.jsdt.core.compiler.libraries.LateralusLibraryInitializer"
    id="org.eclipse.wst.jsdt.launching.LateralusLibrary">
  </JsGlobalScopeContainerInitializer>
</extension>

<!-- Inference Support -->
<extension
  point="org.eclipse.wst.jsdt.core.inferenceSupport">
  <inferenceProvider class="org.eclipse.wst.jsdt.core.infer.LateralusInferenceProvider"/>
</extension>
```

Advanced Toolkit – Lateralus Toolkit III.

/* © Copyright 2008 IBM Corp. All rights reserved. This source code is made available under the terms of the Eclipse Public License, v1.0. */

```
public class LateralusInferenceProvider implements InferenceProvider {  
    public int applyTo(IScriptFileDeclaration scriptFile) {  
        return InferenceProvider.MAYBE_THIS;  
    }  
  
    public InferEngine getInferEngine() {  
        InferEngine engine = new LateralusInferEngine();  
        engine.inferenceProvider=this;  
        return engine;  
    }  
    public String getID() {  
        return "org.eclipse.wst.jsdt.infer";  
    }  
}
```



Advanced Toolkit – Lateralus Toolkit IV.

*/** © Copyright 2008 IBM Corp. All rights reserved. This source code is made available under the terms of the Eclipse Public License, v1.0. **/*

```
public class LateralusInferEngine extends InferEngine{  
    // goto demo in eclipse  
}
```

The FireFox Browser Library - Example

- JSDT Browser Library provides all the types + objects from the FireFox Library.
- Objects are defined with *.prototype* structure in .js files.
- Defines *Window()* object for inheritance.
- When the FireFox Browser Support Library is added, the source file is copied to the workspace **working** directory. All projects reference the same sourcefile.
- No inference support needed. Only UI and Global Scope management are used –
 - Global Scope Handling - **FireFoxLibInitializer.java**
 - Library Wizard UI - **FireFoxLibraryWizardPage.java**

Global Scope

- All CompilationUnits within a project share the same Global Scope.
- Depending on the project's library configuration, high level types + fields in source CU's are added back to the projects Global Scope.
- Library plug-ins may contribute JavaScript source for inclusion in a project's Global Scope.
- The Global Scope is managed through the *Library Configuration* page.

Conclusion and Additional Resources

- CVS Repository Locations
 - ◆ /cvsroot/webtools
- As Standalone JavaScript Editor
 - ◆ sourceediting/plugins/org.eclipse.wst.jsdt.core
 - ◆ org.eclipse.wst.jsdt.core
 - ◆ org.eclipse.wst.jsdt.manipulation
 - ◆ org.eclipse.wst.jsdt.ui
- Integrated with Web Tools
 - ◆ sourceediting/plugins/org.eclipse.wst.jsdt.core
 - ◆ org.eclipse.wst.jsdt.web.core
 - ◆ org.eclipse.wst.jsdt.web.ui
- See Wiki page for more information
 - ◆ <http://http://wiki.eclipse.org/JSDT>

Legal Notices

- Java and all Java-based marks, among others, are trademarks or registered trademarks of Sun Microsystems in the United States, other countries or both.
- Eclipse and the Eclipse logo are trademarks of Eclipse Foundation, Inc.
- IBM and the IBM logo are trademarks or registered trademarks of IBM Corporation, in the United States, other countries or both.
- Other company, product and service names may be trademarks or service marks of others.

- THE INFORMATION DISCUSSED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, AND IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, SUCH INFORMATION. ANY INFORMATION CONCERNING IBM'S PRODUCT PLANS OR STRATEGY IS SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.
- Mozilla Firefox® is a trademark of the Mozilla Foundation
- Dojo® is a trademark of the The Dojo Foundation