New Validation Framework

Gary Karasiuk

2007-12-11

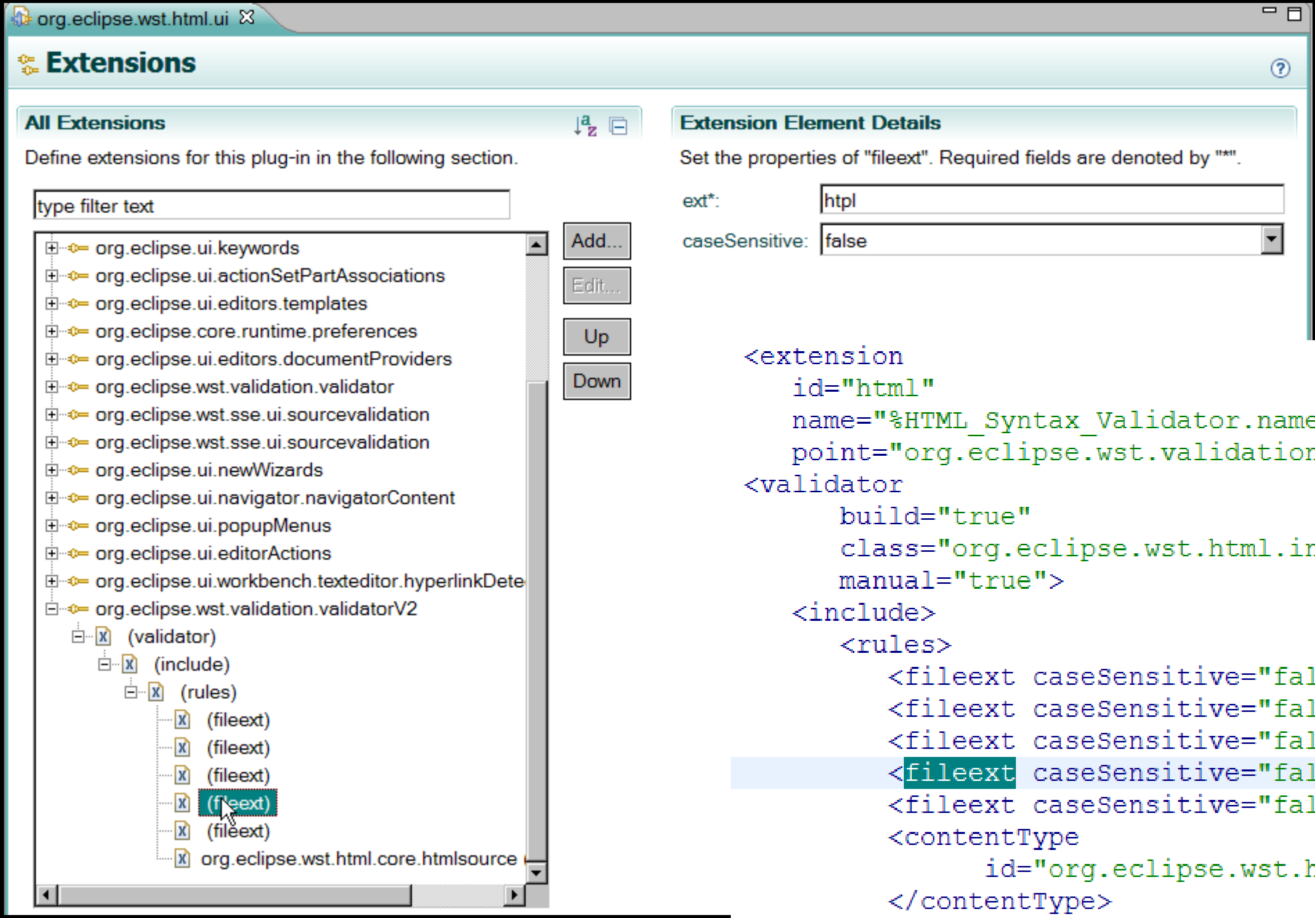
Background

- The current validation framework has a number of known problems and limitations.
- I have been thinking about adding a new validation framework.
- This presentation starts to introduce that new framework.

Approach

- Old validators – work exactly like they used to
  - ✔ They still plug into the old framework
  - ✔ They are still run by the old framework
  - ✔ Good migration story
    - Can't blame new framework if things don't work ✪
    - Migrate at your own pace – possibly never
    - Fine for low volume validators and third party validators
- New Validators
  - ✔ Designed to minimize what validator owners need to do
  - ✔ Use a new extension point
  - ✔ Implementation detail
    - Wrap an old validator
    - Supply a brand new, better validator

Validators get to Pick the Defaults



org.eclipse.wst.html.ui

Extensions

All Extensions

Define extensions for this plug-in in the following section.

type filter text

- org.eclipse.ui.keywords
- org.eclipse.ui.actionSetPartAssociations
- org.eclipse.ui.editors.templates
- org.eclipse.core.runtime.preferences
- org.eclipse.ui.editors.documentProviders
- org.eclipse.wst.validation.validator
- org.eclipse.wst.sse.ui.sourcevalidation
- org.eclipse.wst.sse.ui.sourcevalidation
- org.eclipse.ui.newWizards
- org.eclipse.ui.navigator.navigatorContent
- org.eclipse.ui.popupMenus
- org.eclipse.ui.editorActions
- org.eclipse.ui.workbench.texteditor.hyperlinkDete
- org.eclipse.wst.validation.validatorV2
  - (validator)
    - (include)
      - (rules)
        - (fileext)
        - (fileext)
        - (fileext)
        - (fileext)
        - (fileext)
        - org.eclipse.wst.html.core.htmlsource

Extension Element Details

Set the properties of "fileext". Required fields are denoted by "*".

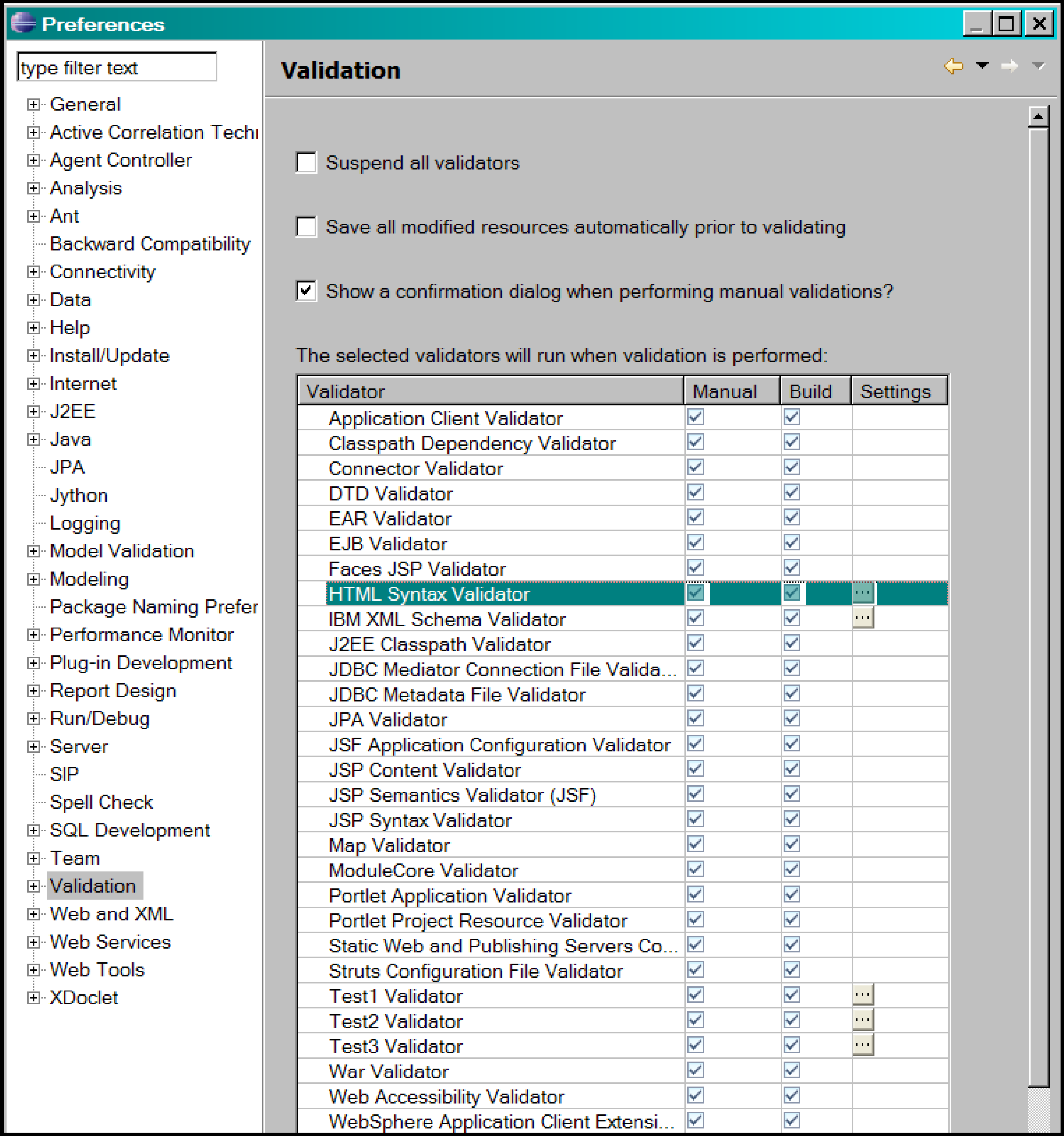
ext*:	htpl
caseSensitive:	false

```
<extension
   id="html"
   name="%HTML_Syntax_Validator.name"
   point="org.eclipse.wst.validation.validatorV2">
<validator
      build="true"
      class="org.eclipse.wst.html.internal.validation.HTMLValidator"
      manual="true">
   <include>
      <rules>
         <fileext caseSensitive="false" ext="html"></fileext>
         <fileext caseSensitive="false" ext="htm"></fileext>
         <fileext caseSensitive="false" ext="xhtml"></fileext>
         <fileext caseSensitive="false" ext="htpl"></fileext>
         <fileext caseSensitive="false" ext="wml"></fileext>
         <contentType
               id="org.eclipse.wst.html.core.htmlsource">
         </contentType>
      </rules>
   </include>
</validator>
```

- But, Users get the ultimate say …

New Validation UI

- Top level UI almost identical to existing UI
  - ✔ The level where you turn on or off validation
- New Validators will have additional configuration options on the “Settings” page:
  - ✔ File extensions
  - ✔ Natures
  - ✔ Facets
  - ✔ Content Type
  - ✔ Folder names



Validation

☐ Suspend all validators

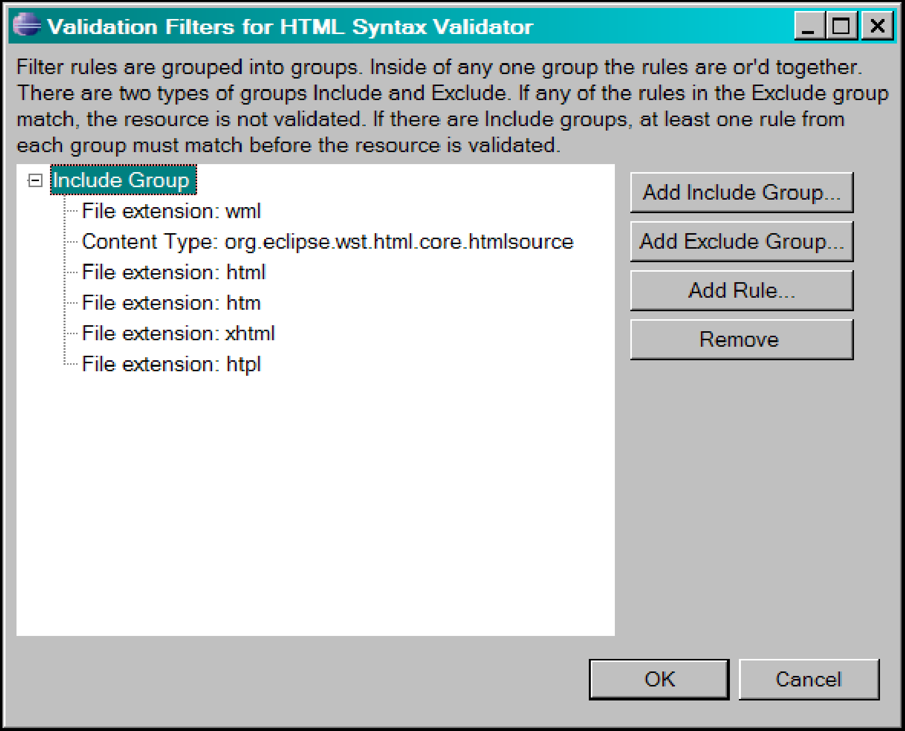
☐ Save all modified resources automatically prior to validating

☑ Show a confirmation dialog when performing manual validations?

The selected validators will run when validation is performed:

Validator	Manual	Build	Settings
Application Client Validator	✓	✓	
Classpath Dependency Validator	✓	✓	
Connector Validator	✓	✓	
DTD Validator	✓	✓	
EAR Validator	✓	✓	
EJB Validator	✓	✓	
Faces JSP Validator	✓	✓	
HTML Syntax Validator	✓	✓	...
IBM XML Schema Validator	✓	✓	...
J2EE Classpath Validator	✓	✓	
JDBC Mediator Connection File Valida...	✓	✓	
JDBC Metadata File Validator	✓	✓	
JPA Validator	✓	✓	
JSF Application Configuration Validator	✓	✓	
JSP Content Validator	✓	✓	
JSP Semantics Validator (JSF)	✓	✓	
JSP Syntax Validator	✓	✓	
Map Validator	✓	✓	
ModuleCore Validator	✓	✓	
Portlet Application Validator	✓	✓	
Portlet Project Resource Validator	✓	✓	
Static Web and Publishing Servers Co...	✓	✓	
Struts Configuration File Validator	✓	✓	
Test1 Validator	✓	✓	...
Test2 Validator	✓	✓	...
Test3 Validator	✓	✓	...
War Validator	✓	✓	
Web Accessibility Validator	✓	✓	
WebSphere Application Client Extensi...	✓	✓	

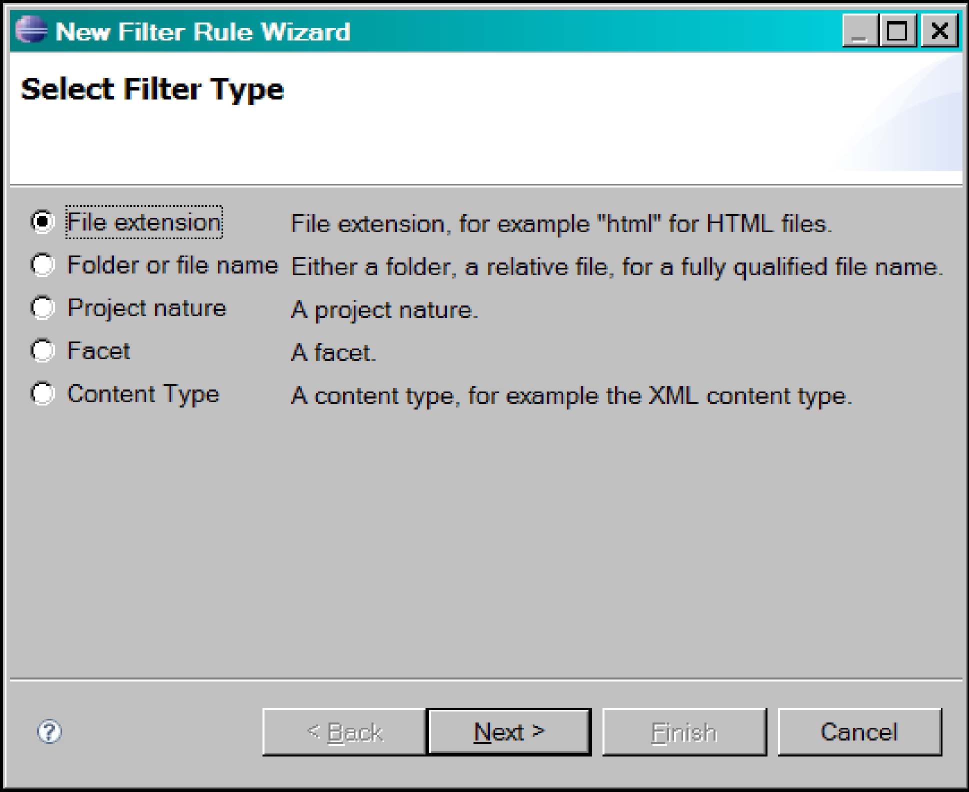
Users get more control over Filters



Validation Filters for HTML Syntax Validator

Filter rules are grouped into groups. Inside of any one group the rules are or'd together. There are two types of groups Include and Exclude. If any of the rules in the Exclude group match, the resource is not validated. If there are Include groups, at least one rule from each group must match before the resource is validated.

- Include Group
  - File extension: wml
  - Content Type: org.eclipse.wst.html.core.htmlsource
  - File extension: html
  - File extension: htm
  - File extension: xhtml
  - File extension: htpl

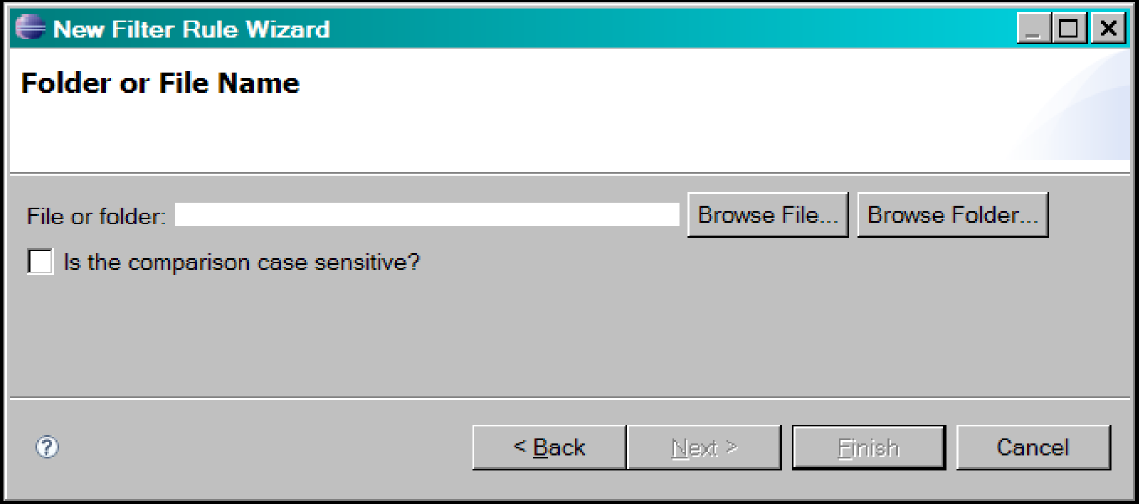


New Filter Rule Wizard

Select Filter Type

[✓ File extension]	File extension, for example "html" for HTML files.
Folder or file name	Either a folder, a relative file, for a fully qualified file name.
Project nature	A project nature.
Facet	A facet.
Content Type	A content type, for example the XML content type.

Adding Filters

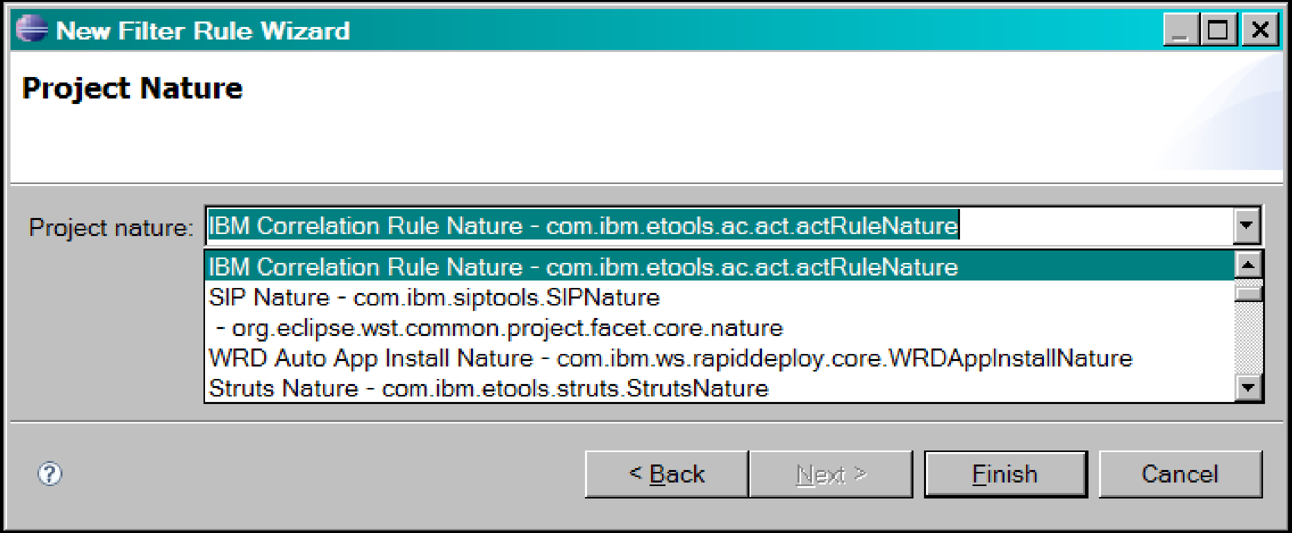


New Filter Rule Wizard

Folder or File Name

File or folder:

Is the comparison case sensitive?



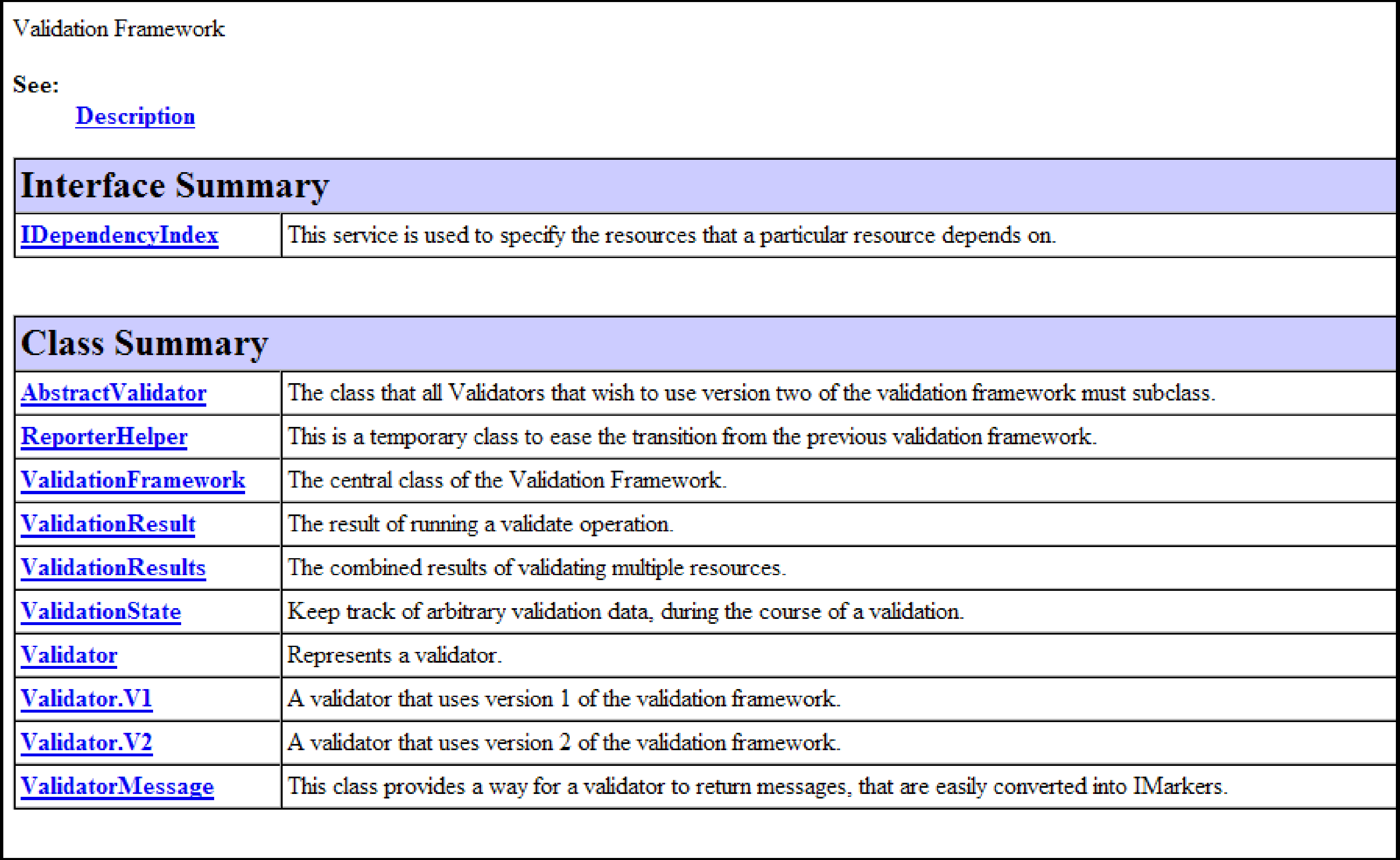
New Filter Rule Wizard

Project Nature

Project nature: IBM Correlation Rule Nature - com.ibm.etools.ac.act.actRuleNature

IBM Correlation Rule Nature - com.ibm.etools.ac.act.actRuleNature
SIP Nature - com.ibm.siptools.SIPNature
 - org.eclipse.wst.common.project.facet.core.nature
WRD Auto App Install Nature - com.ibm.ws.rapiddeploy.core.WRDAppInstallNature
Struts Nature - com.ibm.etools.struts.StrutsNature

API

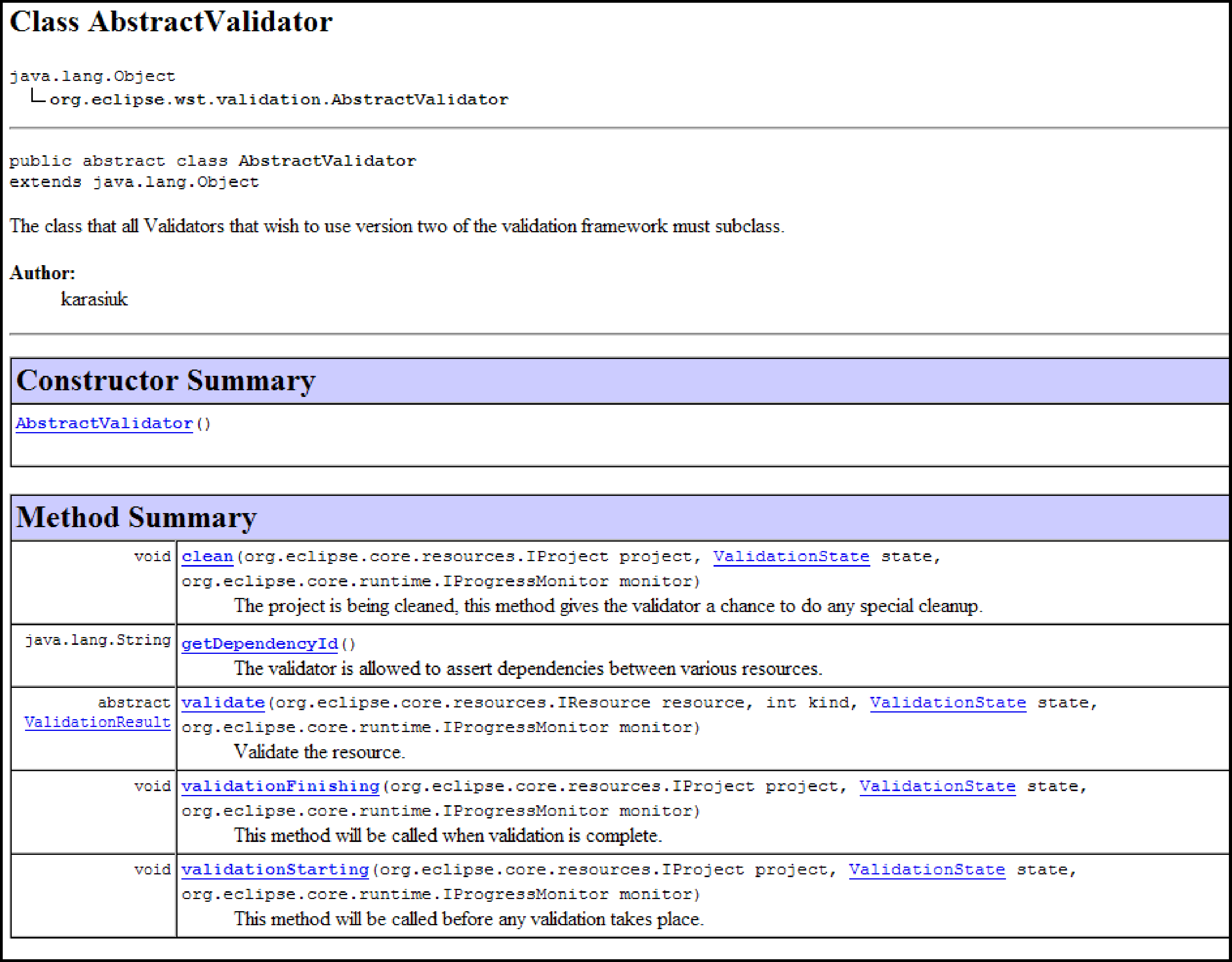


Validation Framework

See:
Description

Interface Summary	
IDependencyIndex	This service is used to specify the resources that a particular resource depends on.

Class Summary	
AbstractValidator	The class that all Validators that wish to use version two of the validation framework must subclass.
ReporterHelper	This is a temporary class to ease the transition from the previous validation framework.
ValidationFramework	The central class of the Validation Framework.
ValidationResult	The result of running a validate operation.
ValidationResults	The combined results of validating multiple resources.
ValidationState	Keep track of arbitrary validation data, during the course of a validation.
Validator	Represents a validator.
Validator.V1	A validator that uses version 1 of the validation framework.
Validator.V2	A validator that uses version 2 of the validation framework.
ValidatorMessage	This class provides a way for a validator to return messages, that are easily converted into IMarkers.



Class AbstractValidator

```
java.lang.Object
  └org.eclipse.wst.validation.AbstractValidator
```

```
public abstract class AbstractValidator
extends java.lang.Object
```

The class that all Validators that wish to use version two of the validation framework must subclass.

Author:

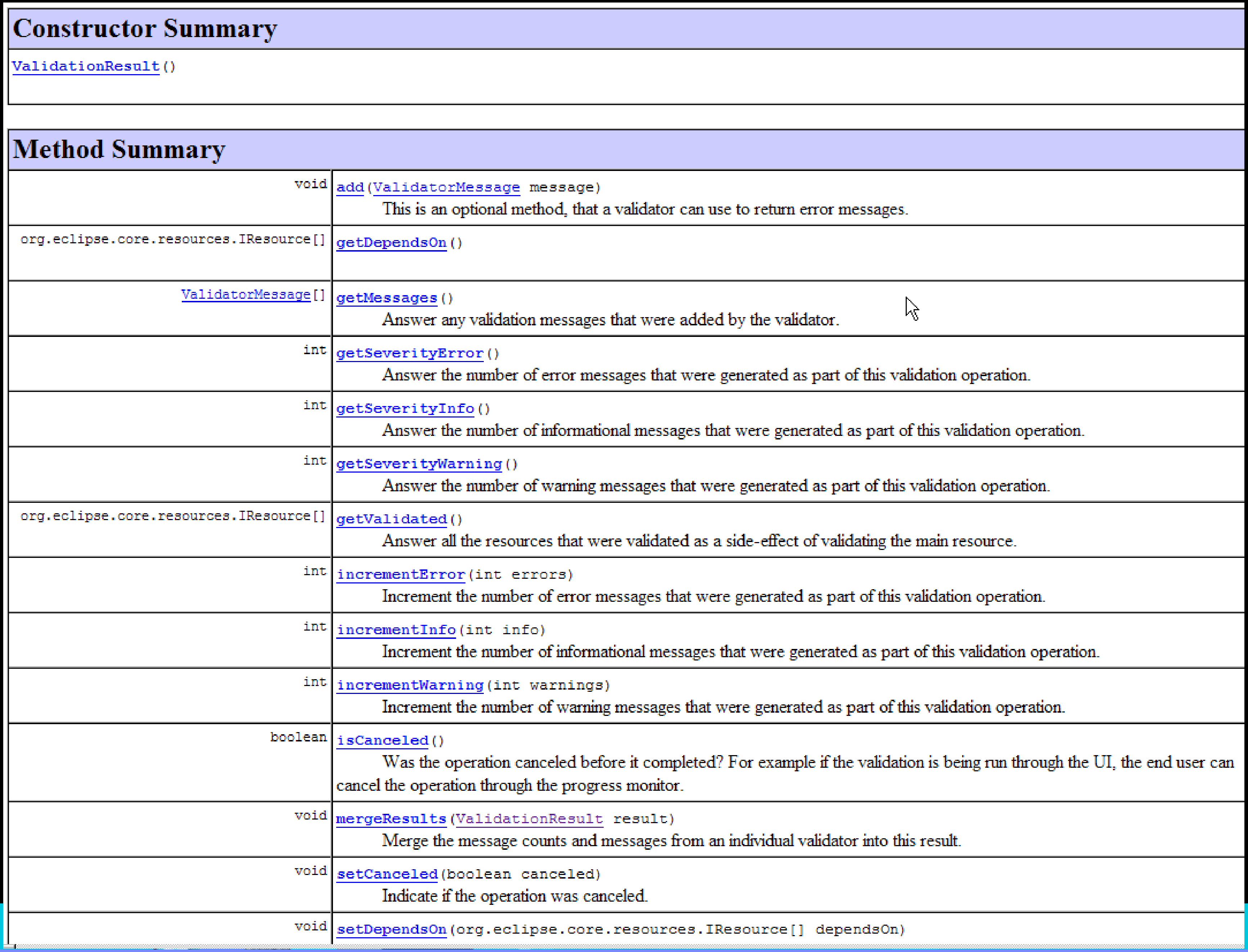
karasiuk

Constructor Summary

`AbstractValidator()`

Method Summary

void	`clean(org.eclipse.core.resources.IProject project, ValidationState state, org.eclipse.core.runtime.IProgressMonitor monitor)` The project is being cleaned, this method gives the validator a chance to do any special cleanup.
java.lang.String	`getDependencyId()` The validator is allowed to assert dependencies between various resources.
abstract ValidationResult	`validate(org.eclipse.core.resources.IResource resource, int kind, ValidationState state, org.eclipse.core.runtime.IProgressMonitor monitor)` Validate the resource.
void	`validationFinishing(org.eclipse.core.resources.IProject project, ValidationState state, org.eclipse.core.runtime.IProgressMonitor monitor)` This method will be called when validation is complete.
void	`validationStarting(org.eclipse.core.resources.IProject project, ValidationState state, org.eclipse.core.runtime.IProgressMonitor monitor)` This method will be called before any validation takes place.



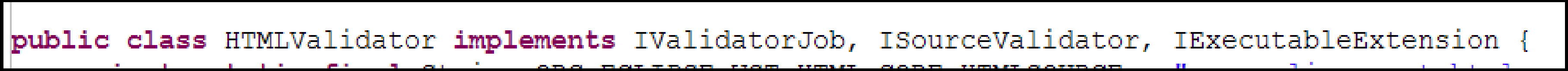
Constructor Summary
`ValidationResult()`

Method Summary	
`void`	`add(ValidatorMessage message)` This is an optional method, that a validator can use to return error messages.
`org.eclipse.core.resources.IResource[]`	`getDependsOn()`
`ValidatorMessage[]`	`getMessages()` Answer any validation messages that were added by the validator.
`int`	`getSeverityError()` Answer the number of error messages that were generated as part of this validation operation.
`int`	`getSeverityInfo()` Answer the number of informational messages that were generated as part of this validation operation.
`int`	`getSeverityWarning()` Answer the number of warning messages that were generated as part of this validation operation.
`org.eclipse.core.resources.IResource[]`	`getValidated()` Answer all the resources that were validated as a side-effect of validating the main resource.
`int`	`incrementError(int errors)` Increment the number of error messages that were generated as part of this validation operation.
`int`	`incrementInfo(int info)` Increment the number of informational messages that were generated as part of this validation operation.
`int`	`incrementWarning(int warnings)` Increment the number of warning messages that were generated as part of this validation operation.
`boolean`	`isCanceled()` Was the operation canceled before it completed? For example if the validation is being run through the UI, the end user can cancel the operation through the progress monitor.
`void`	`mergeResults(ValidationResult result)` Merge the message counts and messages from an individual validator into this result.
`void`	`setCanceled(boolean canceled)` Indicate if the operation was canceled.
`void`	`setDependsOn(org.eclipse.core.resources.IResource[] dependsOn)`

Case Study

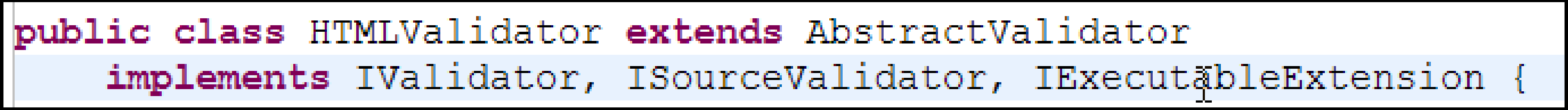
- What did it take to convert the HTML Syntax Validator?
- Not much
  - ✓ Two files changed
    - Plugin.xml – to use the new extension point
    - HTMLValidator

Before

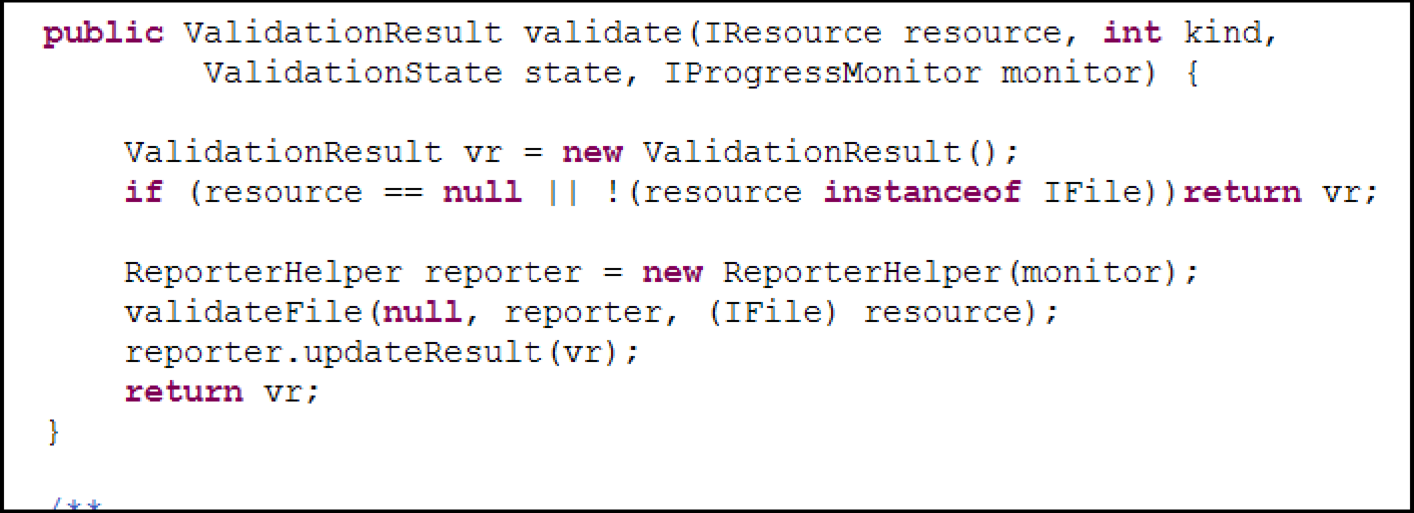


```
public class HTMLValidator implements IValidatorJob, ISourceValidator, IExecutableExtension {
```

After



```
public class HTMLValidator extends AbstractValidator
    implements IValidator, ISourceValidator, IExecutableExtension {
```



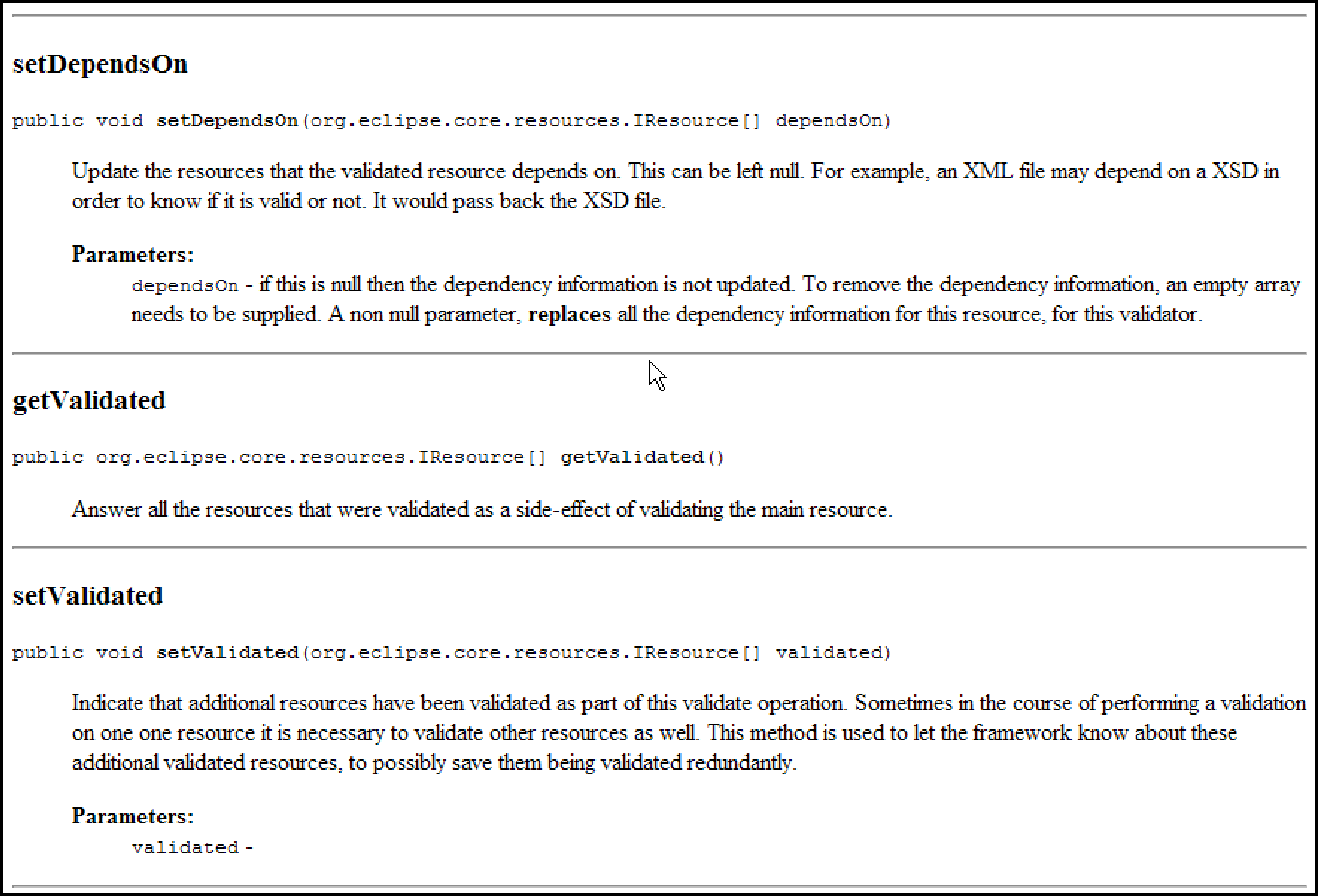
```
public ValidationResult validate(IResource resource, int kind,
        ValidationState state, IProgressMonitor monitor) {

    ValidationResult vr = new ValidationResult();
    if (resource == null || !(resource instanceof IFile))return vr;

    ReporterHelper reporter = new ReporterHelper(monitor);
    validateFile(null, reporter, (IFile) resource);
    reporter.updateResult(vr);
    return vr;
}

/**
```

Some methods to note:



setDependsOn

```
public void setDependsOn(org.eclipse.core.resources.IResource[] dependsOn)
```

Update the resources that the validated resource depends on. This can be left null. For example, an XML file may depend on a XSD in order to know if it is valid or not. It would pass back the XSD file.

Parameters:
`dependsOn` - if this is null then the dependency information is not updated. To remove the dependency information, an empty array needs to be supplied. A non null parameter, **replaces** all the dependency information for this resource, for this validator.

getValidated

```
public org.eclipse.core.resources.IResource[] getValidated()
```

Answer all the resources that were validated as a side-effect of validating the main resource.

setValidated

```
public void setValidated(org.eclipse.core.resources.IResource[] validated)
```

Indicate that additional resources have been validated as part of this validate operation. Sometimes in the course of performing a validation on one one resource it is necessary to validate other resources as well. This method is used to let the framework know about these additional validated resources, to possibly save them being validated redundantly.

Parameters:
`validated` -

Why move to the New Framework?

- Performance
  - ✔ Caching
  - ✔ Job Control
  - ✔ Content Type
  - ✔ Fine Grain Control
- Dependency Support
- UI Dialog
- Simplicity and Documentation
- Framework Collateral
- Logging and Auditing