



Make Me an Eclipse View (with less Plumbing): The PTP External Tools Framework Feedback View

Beth R. Tibbitts
IBM Corp.
tibbitts@us.ibm.com
March, 2010

EclipseCon 2010, Santa Clara

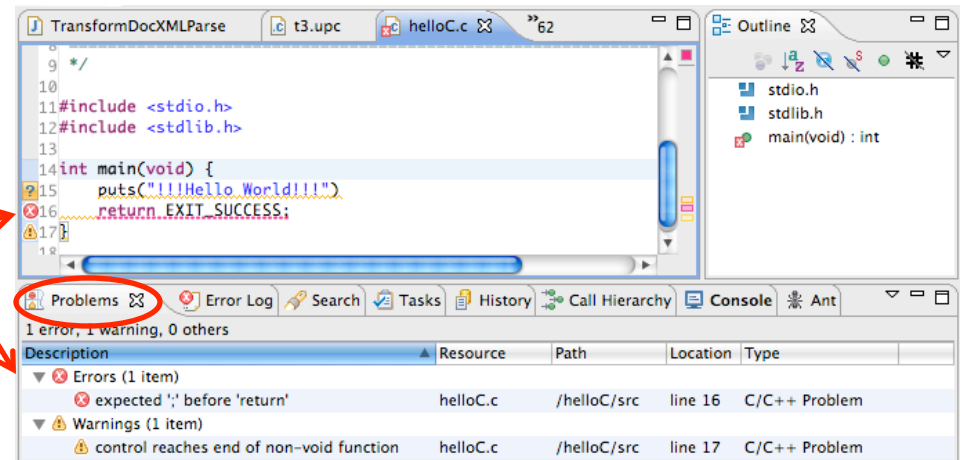
This material is partly based upon work supported by the Defense Advanced Research Projects Agency under its Agreement No. HR0011-07-9-0002



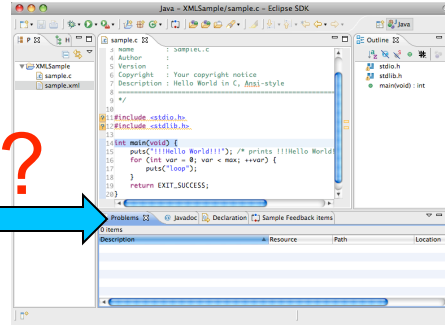
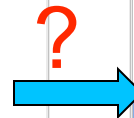
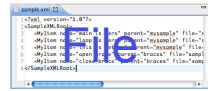
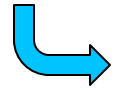
Background/Motivation

- You have a tool that generates information about a (source code) file, including line numbers
- You want to display the information to the end-user in a way that makes it easy to navigate to that src code line
 - Like the “Problems” view
- You don’t want to write a lot of Eclipse “plumbing” code

Problems view shows error
Double-click on error
to go to src line



ETFw Feedback view

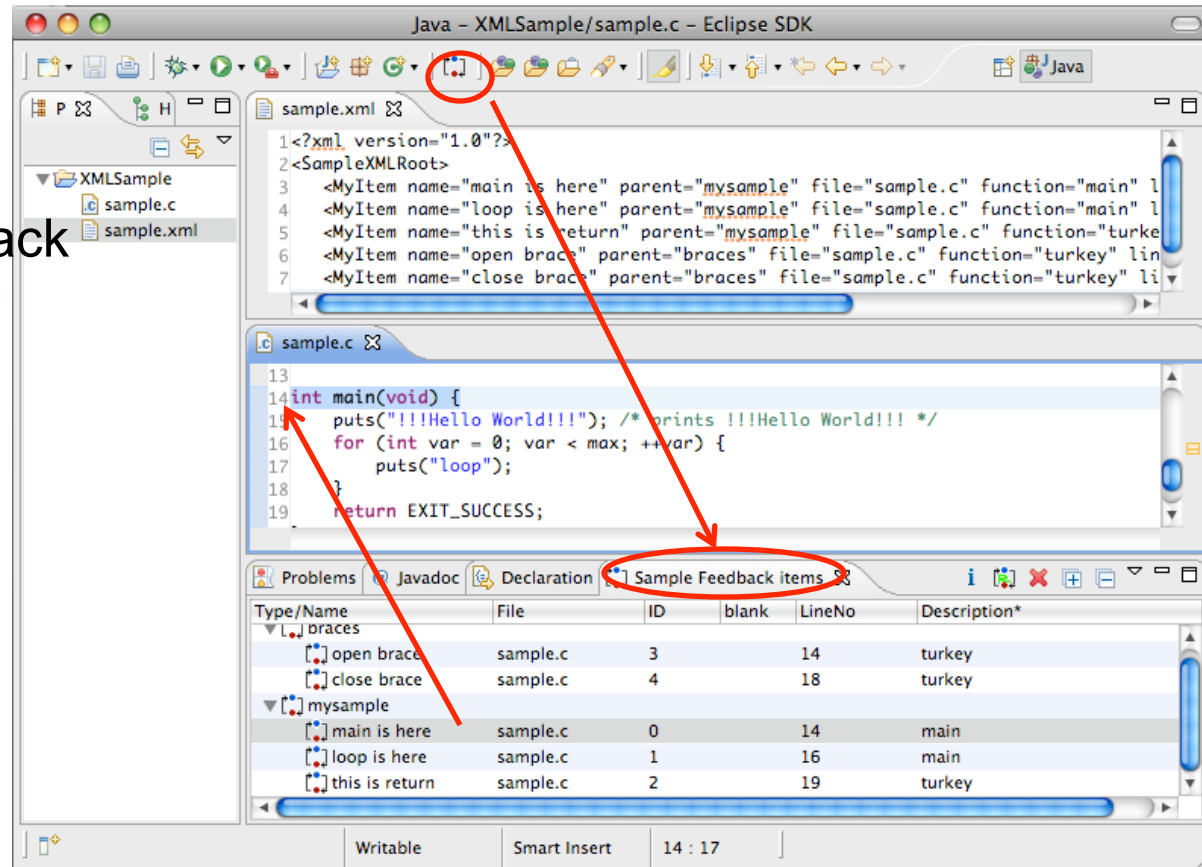


PTP External Tools Framework (ETFw)

- Provides simple way to integrate execution of existing (command-line?) tools into eclipse workbench
- But if your tool (launched from ETFw or not) has results, how to show user src lines that are relevant?
- Answer: *ETFw Feedback view*
 - You provide:
 - output file (e.g. xml file with info about src files and line numbers)
 - A simple Parser that creates IFeedback items from your file (xml parser basics are in the sample)
 - ETFw Feedback view provides:
 - View with your items in a TreeTable view, mapped to src lines

ETFw Feedback view – What it looks like

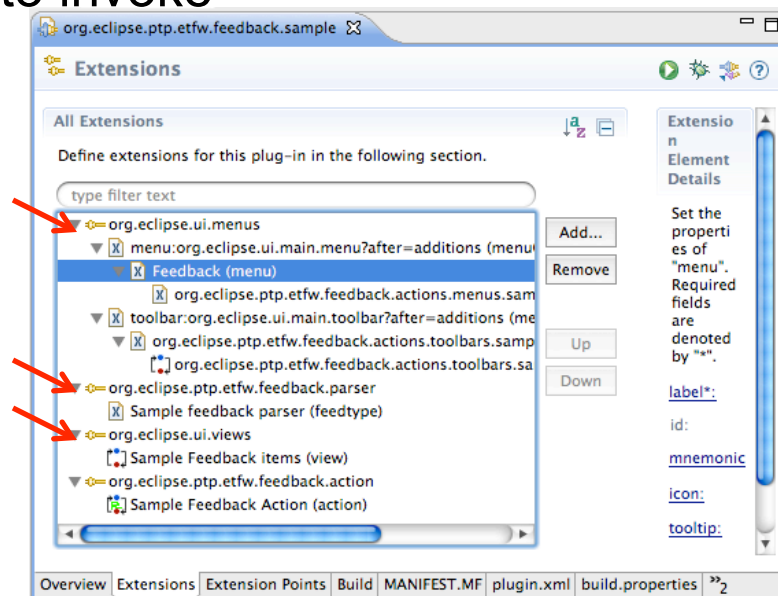
- Select side file
- Select feedback action
- Show view
- Double-click line in view to highlight line of src



ETFw Feedback view – how to use

- See sample project:
dev.eclipse.org / tools / org.eclipse.ptp / tools
Project: org.eclipse.ptp.feedback.sample
- Add menus/actions, as desired – to invoke
- Name your parser recognized by initial node in side file (e.g. root XML node)
- Name your view
- Implement the action

Most of the above classes are simple extensions of the base classes. Except..





ETFw Feedback view parser – you implement

...Except for the parser that interprets your side file and creates objects that can be shown in the view

```
package org.eclipse.ptp.etfw.feedback.sample;

import java.io.File;

public class SampleFeedbackParser extends AbstractFeedbackParser {

    public List<IFeedbackItem> getFeedbackItems(IFile file) {
        String xmlfile = file.getLocation().toOSString();
        List<IFeedbackItem> items=new ArrayList<IFeedbackItem>();
        try {
            items.parse(xmlfile);
        } catch (XPathExpressionException e) {
            // TODO Auto-generated catch block
            ...etc.
        }
        return items;
    }
}
```



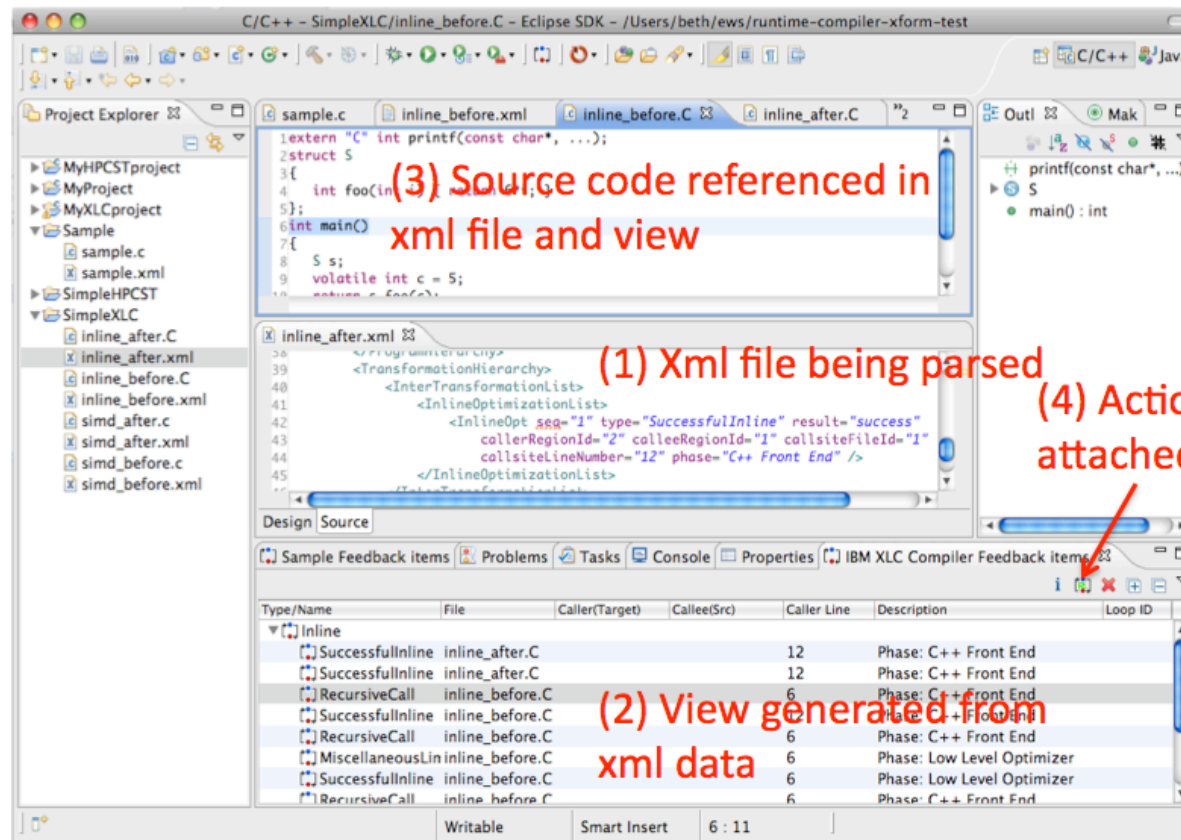
Sample XML file

```
<?xml version="1.0"?>
<SampleXMLRoot>   Root XML node identifies type of file
  <MyItem name="main is here" parent="mysample" file="sample.c"
function="main" lineNo="14" isFunction="true" id="0"/>
  <MyItem name="loop is here" parent="mysample" file="sample.c"
function="main" lineNo="16" isFunction="true" id="1"/>
  <MyItem name="this is return" parent="mysample" file="sample.c"
function="turkey" lineNo="19" isFunction="true" id="2"/>
  <MyItem name="open brace" parent="braces" file="sample.c"
function="turkey" lineNo="14" isFunction="true" id="3"/>
  <MyItem name="close brace" parent="braces" file="sample.c"
function="turkey" lineNo="18" isFunction="true" id="4"/>
</SampleXMLRoot>
```

Parent node can be used to group items in Feedback view Tree

ETFw Feedback view with compiler reports

- Compiler attempts at inlining etc.
- Action can be attached. Refactoring?



The screenshot shows the Eclipse IDE interface with several views and annotations:

- Project Explorer:** Shows a project structure with files like `sample.c`, `inline_after.C`, `inline_after.xml`, `inline_before.C`, `simd_after.xml`, `simd_before.c`, and `simd_before.xml`.
- Editor:** Displays the source code of `inline_before.C` with a red annotation: "(3) Source code referenced in xml file and view".
- XML View:** Shows the contents of `inline_after.xml` with a red annotation: "(1) Xml file being parsed".
- Feedback View:** Displays a table of compiler feedback items with a red annotation: "(2) View generated from xml data".
- Properties View:** Shows the properties of the selected feedback item with a red annotation: "(4) Action can be attached to view".

Type/Name	File	Caller(Target)	Callee(Src)	Caller Line	Description	Loop ID
SuccessfulInline	inline_after.C			12	Phase: C++ Front End	
SuccessfulInline	inline_after.C			12	Phase: C++ Front End	
RecursiveCall	inline_before.C			6	Phase: C++ Front End	
SuccessfulInline	inline_before.C			6	Phase: C++ Front End	
RecursiveCall	inline_before.C			6	Phase: C++ Front End	
MiscellaneousLin	inline_before.C			6	Phase: Low Level Optimizer	
SuccessfulInline	inline_before.C			6	Phase: Low Level Optimizer	
RecursiveCall	inline_before.C			6	Phase: C++ Front End	

ETFw Feedback view with Bottleneck reports



The screenshot shows the Eclipse IDE interface with the HPCST Feedback view. The Project Explorer on the left shows a project structure with files like sample.c, sample.xml, and hpcs_report.209332.262.xml. The main editor displays the source code of mhd.F, which includes comments about Lattice Boltzmann Code and Octagonal Grid Scalar-Vector Model. The HPCST Feedback view at the bottom shows a table of bottleneck reports.

Type/Name	File	Caller(Target)	Callee(Src)	Caller Line	Description
Hotspot	mhd.F			9	description
SmallDataBlockSize parent=hotspot0	mhd.F			9	description
Solution: Large IO Page	mhd.F			9	Suggestion: sugges
Slow IO parent=hotspot0	mhd.F			9	description
Stalls by LSU parent=hotspot0	mhd.F			9	description
Hotspot	mhd.F			355	description
GlobalTranspose parent=hotspot1	mhd.F			355	description
Solution: qtranspose	mhd.F			355	Suggestion: reshape



Summary

- Feedback view provides simple/easy way to view results of external tools
- Good reviews from users
- Can attach other actions... to leverage data from the tools
- PTP 3.0.x contains framework/abstract code for Feedback view
 - PTP 4.0 in Helios too
 - No concrete views; Sample project in CVS
- Simple Implementation now
- Some limitations now
- Open to suggestions on extensions / other uses



References

- Parallel Tools Platform: <http://eclipse.org/ptp>
- ETFw wiki page: http://wiki.eclipse.org/PTP/ETFw/PTP_External_Tools_Framework
- ETFw Feedback view specifically:
http://wiki.eclipse.org/PTP/ETFw/PTP_External_Tools_Framework_Feedback_View