Hands-On: Using the New Common Builder for Push-Button PDE Builds

Nick Boldt, Red Hat
Andrew Niefer, IBM
Andrew Overholt, Red Hat

Agenda

- **Introduction**

- PDE Build
  - Exercise: Build a Feature
  - Exercise: Generate a p2 Repository

- Common Build Background

- Dash Athena
  - Exercise: Setup
  - Exercise: Configure, Run, Troubleshoot
  - Exercise: Verify Build

- Meet Hudson

- BYO Build Clinic

Your Hosts

- Nick Boldt
  - JBoss, a division of Red Hat
  - Release Engineering (Modeling, Tools, Dash)
- Andrew Niefer
  - IBM Rational
  - PDE Build Maintainer
- Andrew Overholt
  - Red Hat
  - Linux Tools

Tutorial Exercises & Slides

1. Get a USB drive from presenters & copy its contents to your drive. Also available here: http://www.eclipsecon.org/2009/sessions?id=302

2. Unpack **org.eclipse.dash.common.releng.tutorial_*.zip** somewhere on your drive, eg., `c:/tmp`. This zip includes both exercises and slides.

3. Open Eclipse 3.5M6, and select `File > Import > General > Existing Projects into Workspace`

4. Browse to the workspace folder under where you unpacked the zip, eg., `c:/tmp/org.eclipse.dash.common.releng.tutorial/workspace/`

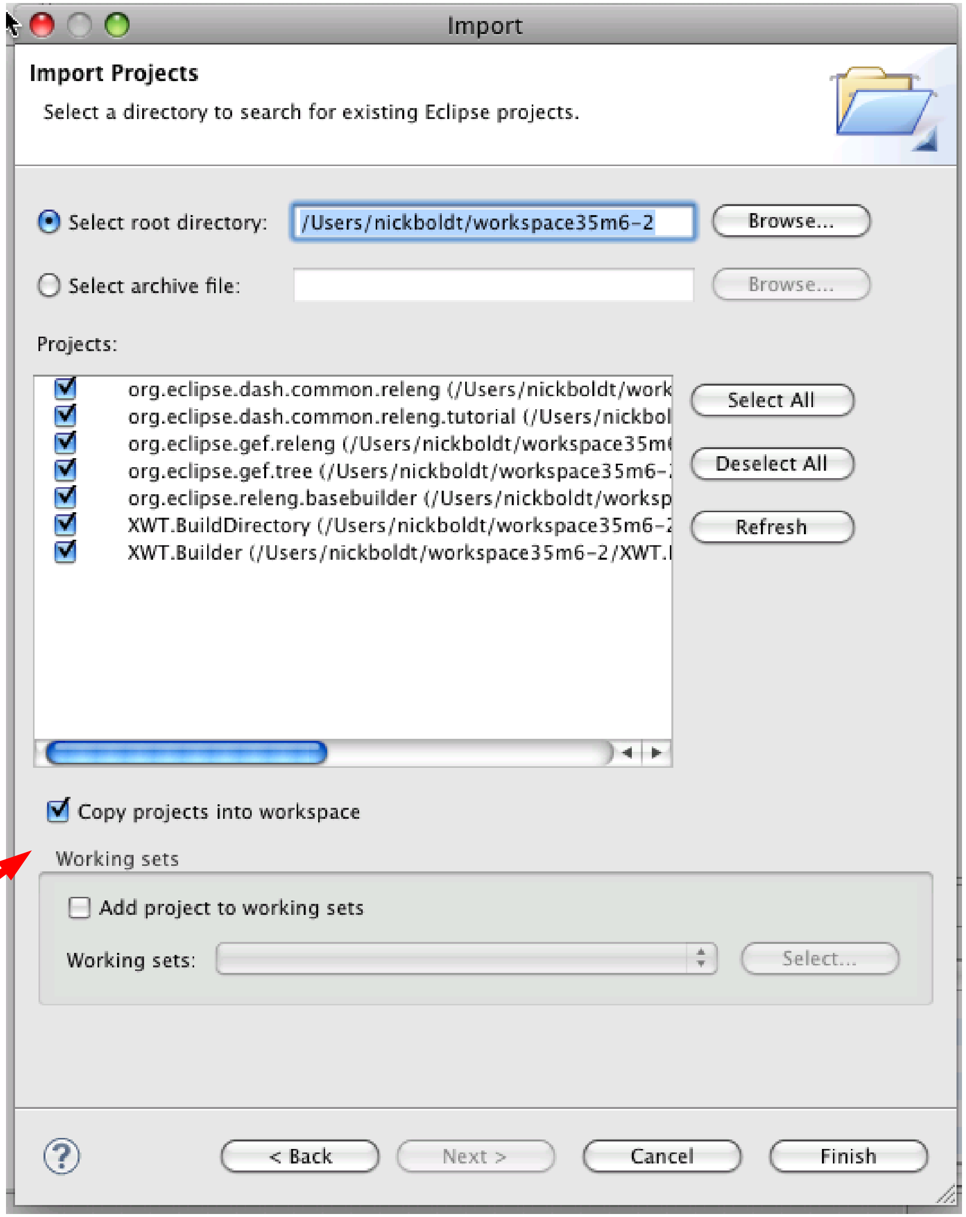
Tutorial Exercises (cont'd)

5. Import the following 6 projects from the workspace folder:

```
org.eclipse.dash.common.releng
org.eclipse.gef.releng
org.eclipse.gef.tree
org.eclipse.releng.basebuilder
XWT.BuildDirectory
XWT.Builder
```

Remember to check `Copy projects into workspace.`

Note that org.eclipse.releng.basebuilder will not compile due to one or more missing requirements. This is not a problem.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: /Users/nickboldt/workspace35m6-2 Browse...

○ Select archive file: Browse...

Projects:

☑ org.eclipse.dash.common.releng (/Users/nickboldt/work
☑ org.eclipse.dash.common.releng.tutorial (/Users/nickbol
☑ org.eclipse.gef.releng (/Users/nickboldt/workspace35m
☑ org.eclipse.gef.tree (/Users/nickboldt/workspace35m6-
☑ org.eclipse.releng.basebuilder (/Users/nickboldt/worksp
☑ XWT.BuildDirectory (/Users/nickboldt/workspace35m6-
☑ XWT.Builder (/Users/nickboldt/workspace35m6-2/XWT.

☑ Copy projects into workspace

Working sets

□ Add project to working sets

Working sets:

Third Party Code

- On the USB drive, there may also be a downloads folder.

- You can choose to download those files manually or use the provided zips/jars **AS-IS** in order to save time re-downloading them. See **README.txt** for more information.

- You **will** need a copy of ant-contrib 1.0b2, and Eclipse (the rest is optional).

  - Copy ant-contrib.jar into your workspace here:
    org.eclipse.dash.common.releng/lib/ant-contrib.jar

  - If not using local files, you can get ant-contrib 1.0b2 here: http://downloads.sourceforge.net/ant-contrib/ant-contrib-1.0b2-bin.zip

Builds ...

- Builds are a TMTOWTDI* problem
- Standardized = better
- Customization always required...

  ...but that leads away from standardization

** There's more than one way to do it*

Like layers of an onion

- Sources in VCS repository

- Packaging details, dependencies, maps
- PDE Build + setup scripts

- X server for running tests
- Web front end for monitoring & auditing

OSGi Bundles

- META-INF/MANIFEST.MF

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: %Plugin.name
Bundle-SymbolicName: org.eclipse.gef; singleton:=true
Bundle-Version: 3.4.0.qualifier
Bundle-Activator: org.eclipse.gef.internal.InternalGEFPlugin
Bundle-Vendor: %Plugin.providerName
Bundle-Localization: plugin
Import-Package: com.ibm.icu.text
Export-Package: org.eclipse.gef,
 org.eclipse.gef.commands,
 [...]
 org.eclipse.gef.util
Require-Bundle: org.eclipse.draw2d;visibility:=reexport;bundle-ver-
sion="[3.2.0,4.0.0)",
 org.eclipse.core.runtime;bundle-version="[3.2.0,4.0.0)",
 org.eclipse.ui.views;resolution:=optional;bundle-
version="[3.2.0,4.0.0)",
 org.eclipse.ui.workbench;bundle-version="[3.2.0,4.0.0)",
 org.eclipse.jface;bundle-version="[3.2.0,4.0.0)"
Bundle-ActivationPolicy: lazy
Bundle-RequiredExecutionEnvironment: J2SE-1.4
```

Bundle dependencies

- META-INF/MANIFEST.MF

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: %Plugin.name
Bundle-SymbolicName: org.eclipse.gef; singleton:=true
Bundle-Version: 3.4.0.qualifier
Bundle-Activator: org.eclipse.gef.internal.InternalGEFPlugin
Bundle-Vendor: %Plugin.providerName
Bundle-Localization: plugin
Import-Package: com.ibm.icu.text
Export-Package: org.eclipse.gef,
 org.eclipse.gef.commands,
 [...]
 org.eclipse.gef.util
Require-Bundle: org.eclipse.draw2d;visibility:=reexport;bundle-ver-
sion="[3.2.0,4.0.0)",
 org.eclipse.core.runtime;bundle-version="[3.2.0,4.0.0)",
 org.eclipse.ui.views;resolution:=optional;bundle-
version="[3.2.0,4.0.0)",
 org.eclipse.ui.workbench;bundle-version="[3.2.0,4.0.0)",
 org.eclipse.jface;bundle-version="[3.2.0,4.0.0)"
Bundle-ActivationPolicy: lazy
Bundle-RequiredExecutionEnvironment: J2SE-1.4
```

Bundle build description

- build.properties

```
bin.includes = about.*,\
               eclipse32.png,\
               plugin.xml,\
               plugin.properties,\
               icons/*.gif,\
               .,\
               META-INF/

bin.excludes = overview.html, **/doc-files/**

source.. = src/
```

Agenda

- Introduction
- **PDE Build**
  - Exercise: Build a Feature
  - Exercise: Generate a p2 Repository
- Common Build Background
- Dash Athena
  - Exercise: Setup
  - Exercise: Configure, Run, Troubleshoot
  - Exercise: Verify Build
- Meet Hudson
- BYO Build Clinic

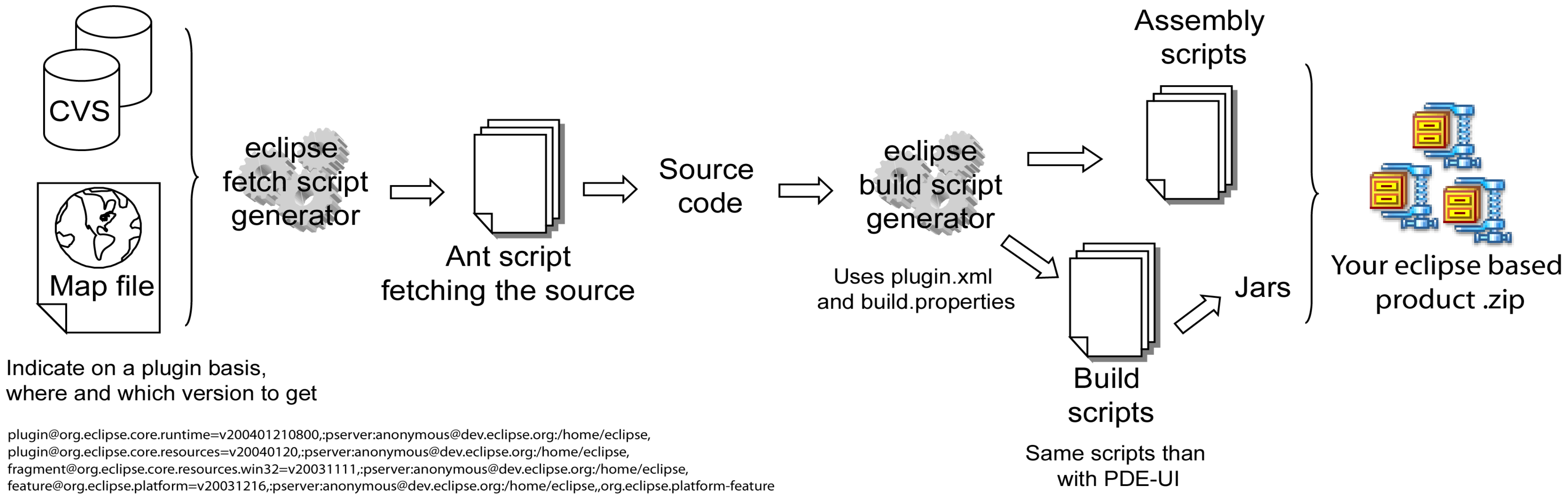
What is PDE Build?

- Ant script generator and build infrastructure
- Manages the compile time classpath based on OSGi dependencies
- The recommended way to build:
  - Plug-ins / Bundles
  - Features
  - RCP apps
- Eclipse itself is built using PDE Build

Overview of the builder

The build uses scripts provided by PDE Build, as well as script generated at build time.

- Fetch source code
- Compile .class files
- Assemble the results into archives.



CVS
Map file
eclipse fetch script generator
Ant script fetching the source
Source code
eclipse build script generator
Uses plugin.xml and build.properties
Assembly scripts
Build scripts
Jars
Your eclipse based product .zip

Indicate on a plugin basis, where and which version to get

```
plugin@org.eclipse.core.runtime=v200401210800,:pserver:anonymous@dev.eclipse.org:/home/eclipse,
plugin@org.eclipse.core.resources=v20040120,:pserver:anonymous@dev.eclipse.org:/home/eclipse,
fragment@org.eclipse.core.resources.win32=v20031111,:pserver:anonymous@dev.eclipse.org:/home/eclipse,
feature@org.eclipse.platform=v20031216,:pserver:anonymous@dev.eclipse.org:/home/eclipse,,org.eclipse.platform-feature
```

Same scripts than with PDE-UI

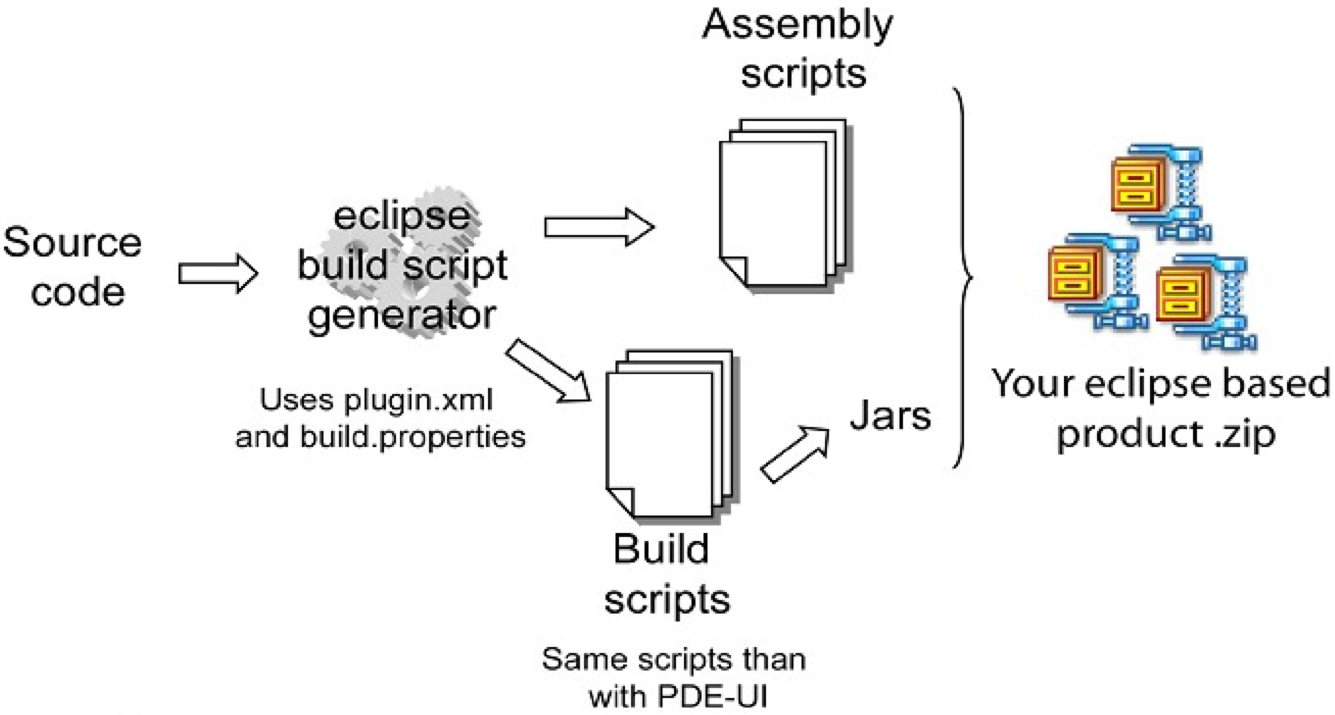
What's new with PDE Build?

- 3.4
  - Required configuration files are reduced to one file: build.properties
  - Added customizability in the assembly and packaging phases of the build
  - Enhanced feature generation capabilities
  - Generation of individual source bundles
- 3.5
  - p2 integration

Exercise: Build a feature

Goals
• Show how to build features • Demystify PDE Build • Introduce key concepts

Relevant Part of PDE Build


Source code
eclipse build script generator
Uses plugin.xml and build.properties
Assembly scripts
Build scripts
Same scripts than with PDE-UI
Jars
Your eclipse based product .zip

Exercise: Build a feature

- Build the e4 XWT feature

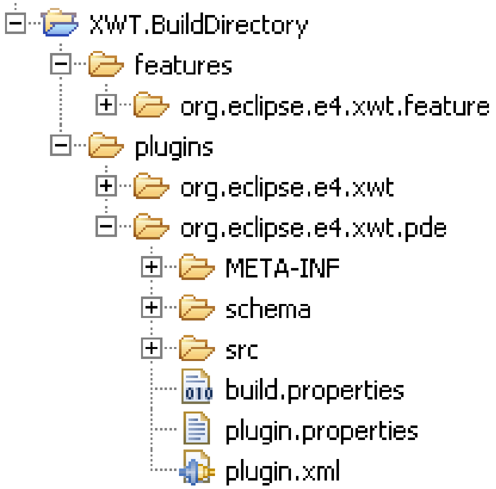
- XWT is a collection of declarative bindings for SWT, used to generate SWT widgets from an XML file

- For setup, see slides 4-6. You will need:
  - Eclipse 3.5M6
  - JDK 5.0+
  - 2 XWT projects – XWT.Builder and XWT.BuildDirectory

Step 1: The Build Directory

- The source to be compiled is in the build directory, structured under features and plugins folder like in a normal Eclipse install.

- This is where the build will run. Generated scripts and build results will end up here.



```
XWT.BuildDirectory
  features
    org.eclipse.e4.xwt.feature
  plugins
    org.eclipse.e4.xwt
    org.eclipse.e4.xwt.pde
      META-INF
      schema
      src
      build.properties
      plugin.properties
      plugin.xml
```

Notes for releng:

- The location is defined by the **${buildDirectory}** ant property

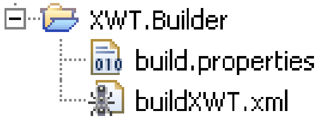
- Make sure build directory and subdirectories are writeable

- This is where build will copy any source it fetches from CVS or other repositories.

Step 2: The Builder

The **builder** directory, aka the build configuration directory, contains the configuration files for the build.

- At a minimum, a build.properties file is required.
- Additional files can be placed here to customize the build.
- Templates of these files are provided in the templates folder of pde build
- The example includes an extra buildXWT.xml for the convenience of running the build in the workspace.



XWT.Builder
build.properties
buildXWT.xml

Notes for releng:

• The location is defined by the **`${builder}`** ant property

• Customization scripts (allElements.xml, customTargets.xml, customAssembly,xml) placed here will be used instead of the defaults – copy from pde build source and adjust as needed

Step 3: Setting Some Properties

- The **build.properties** in the builder folder contains all the switches to configure the overall aspect of the build. The 4 properties that must be set for a build to run are:

topLevelElementId	The feature id to be built
baseLocation	Folder containing the plugins to build against
buildDirectory	Folder containing the source to build (XWT.BuildDirectory)
builder	Folder containing the build configuration (XWT.Builder)

- The **builder** property is specified on the command line for the build

In the example, we are only setting

```
topLevelElementId = org.eclipse.e4.xwt.feature
```

For convenience we are setting the other properties using the buildXWT.xml ant script

The xwt plugins use 1.5 asserts so we also set “javacSource” and “javacTarget” properties here.

Step 4: The `buildXWT.xml` Script

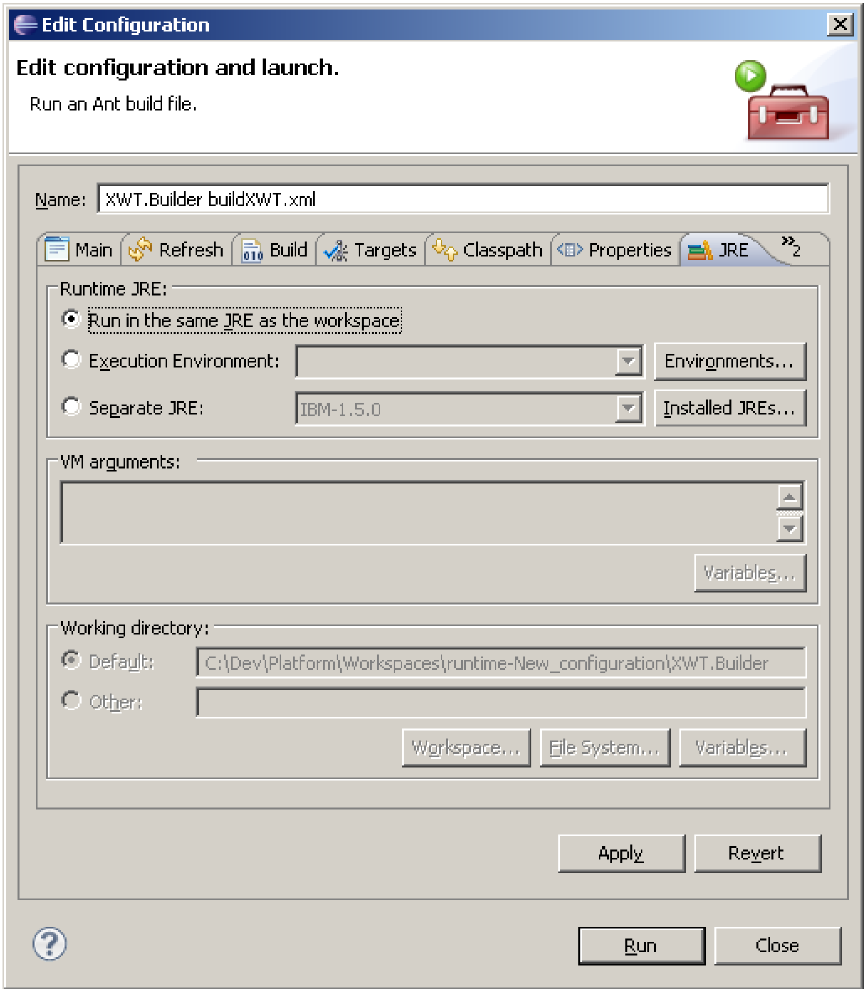
The script is very simple, it sets the properties from Step 3 that we did not put in the build.properties file.

- builder : **`${basedir}`** is the directory containing the script
- buildDirectory : Set to our XWT.BuildDirectory project, relative to the builder
- baseLocation : **`${eclipse.home}`** is the eclipse we are running in
- **`${eclipse.pdebuild.scripts}/build.xml`** is the main PDE Build script

```
<project default="main">
   <target name="main">
      <property name="builder" value="${basedir}" />
      <property name="buildDirectory" value="${basedir}/../XWT.BuildDirectory" />
      <property name="baseLocation" value="${eclipse.home}"/>

      <ant antfile="${eclipse.pdebuild.scripts}/build.xml" />
   </target>
</project>
```

Note: `${eclipse.home}` and `${eclipse.pdebuild.scripts}` are automatically set by Eclipse, see the Ant Runtime Preferences. It is these two properties that make this kind of script a convenient way of running builds.

Step 5: Run The Build



Edit Configuration

Edit configuration and launch.

Run an Ant build file.

Name: XWT.Builder buildXWT.xml

Runtime JRE:

● Run in the same JRE as the workspace

○ Execution Environment:

○ Separate JRE: IBM-1.5.0

VM arguments:

Working directory:

● Default: C:\Dev\Platform\Workspaces\runtime-New_configuration\XWT.Builder

○ Other:

Select the buildXWT.xml script and
“Run As → Ant Build...”

On the JRE tab, make sure that “Run in the same JRE as the workspace” is selected.

Once the build is complete, refresh the XWT.BuildDirectory project to see the results.

Look in the new I.TestBuild folder.

Step 5a: Run The Build

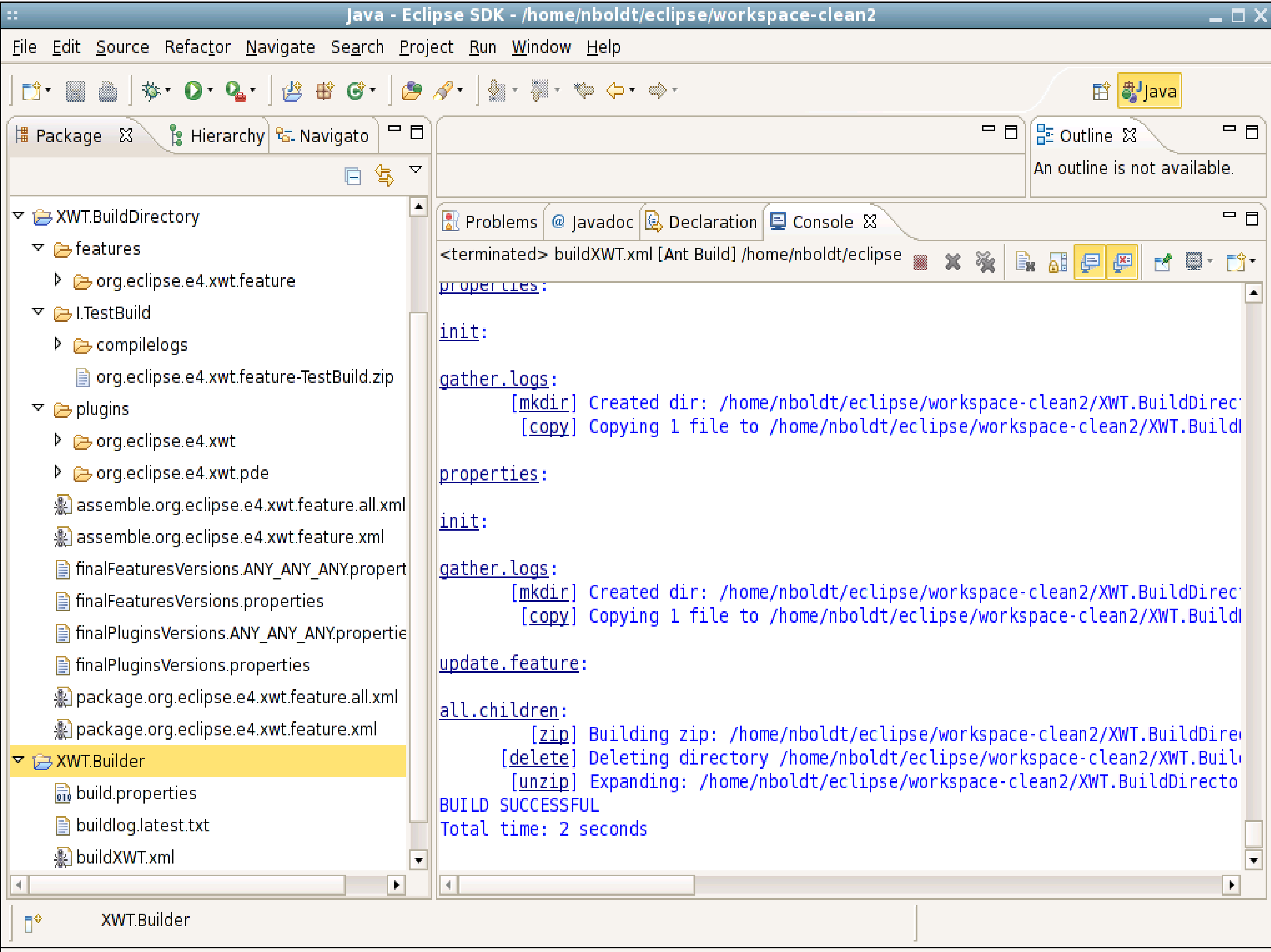
(Command line version)

On the command line, you can start eclipse by running “java -jar” on the equinox launcher.

- -application : the org.eclipse.ant.core.antRunner application.
- -buildfile : the main build.xml script provided by PDE Build
- -Dbuilder : the location of your builder directory

The “baseLocation” and “buildDirectory” properties are normally specified in the builder's build.properties file. Though they can also be specified on the command line:

```
java –jar plugins\org.eclipse.equinox.launcher_<ver>.jar
-application org.eclipse.ant.core.antRunner
-buildfile plugins\org.eclipse.pde.build_<ver>\scripts\build.xml
-Dbuilder=<XWT.Builder> -DbuildDirectory=<XWT.BuildDirectory>
-DbaseLocation=C:\target\eclipse
```


Java - Eclipse SDK - /home/nboldt/eclipse/workspace-clean2
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Package
Hierarchy
Navigato
Outline
An outline is not available.
Problems
@ Javadoc
Declaration
Console
XWT.BuildDirectory
features
org.eclipse.e4.xwt.feature
I.TestBuild
compilelogs
org.eclipse.e4.xwt.feature-TestBuild.zip
plugins
org.eclipse.e4.xwt
org.eclipse.e4.xwt.pde
assemble.org.eclipse.e4.xwt.feature.all.xml
assemble.org.eclipse.e4.xwt.feature.xml
finalFeaturesVersions.ANY_ANY_ANY.propert
finalFeaturesVersions.properties
finalPluginsVersions.ANY_ANY_ANY.propertie
finalPluginsVersions.properties
package.org.eclipse.e4.xwt.feature.all.xml
package.org.eclipse.e4.xwt.feature.xml
XWT.Builder
build.properties
buildlog.latest.txt
buildXWT.xml
XWT.Builder

<terminated> buildXWT.xml [Ant Build] /home/nboldt/eclipse

```
properties:

init:

gather.logs:
       [mkdir] Created dir: /home/nboldt/eclipse/workspace-clean2/XWT.BuildDirec
        [copy] Copying 1 file to /home/nboldt/eclipse/workspace-clean2/XWT.BuildI

properties:

init:

gather.logs:
       [mkdir] Created dir: /home/nboldt/eclipse/workspace-clean2/XWT.BuildDirec
        [copy] Copying 1 file to /home/nboldt/eclipse/workspace-clean2/XWT.BuildI

update.feature:

all.children:
         [zip] Building zip: /home/nboldt/eclipse/workspace-clean2/XWT.BuildDire
      [delete] Deleting directory /home/nboldt/eclipse/workspace-clean2/XWT.Buil
       [unzip] Expanding: /home/nboldt/eclipse/workspace-clean2/XWT.BuildDirecto
BUILD SUCCESSFUL
Total time: 2 seconds
```

Success

- We have an archive!

- See the resulting build

```
XWT.BuildDirectory/I.TestBuild/
   org.eclipse.e4.xwt.feature-TestBuild.zip
```

- See the automatically generated scripts

```
   assemble.org.eclipse.e4.xwt.feature.all.xml
   assemble.org.eclipse.e4.xwt.feature.xml
   package.org.eclipse.e4.xwt.feature.all.xml
   package.org.eclipse.e4.xwt.feature.xml
```

- See the build log in

```
   XWT.Builder/buildlog.latest.txt
```

Exercise: Generate a p2 Repository

3.4 version: Add the following properties to the XWT.Builder `build.properties` file and run the build again to get p2 metadata.

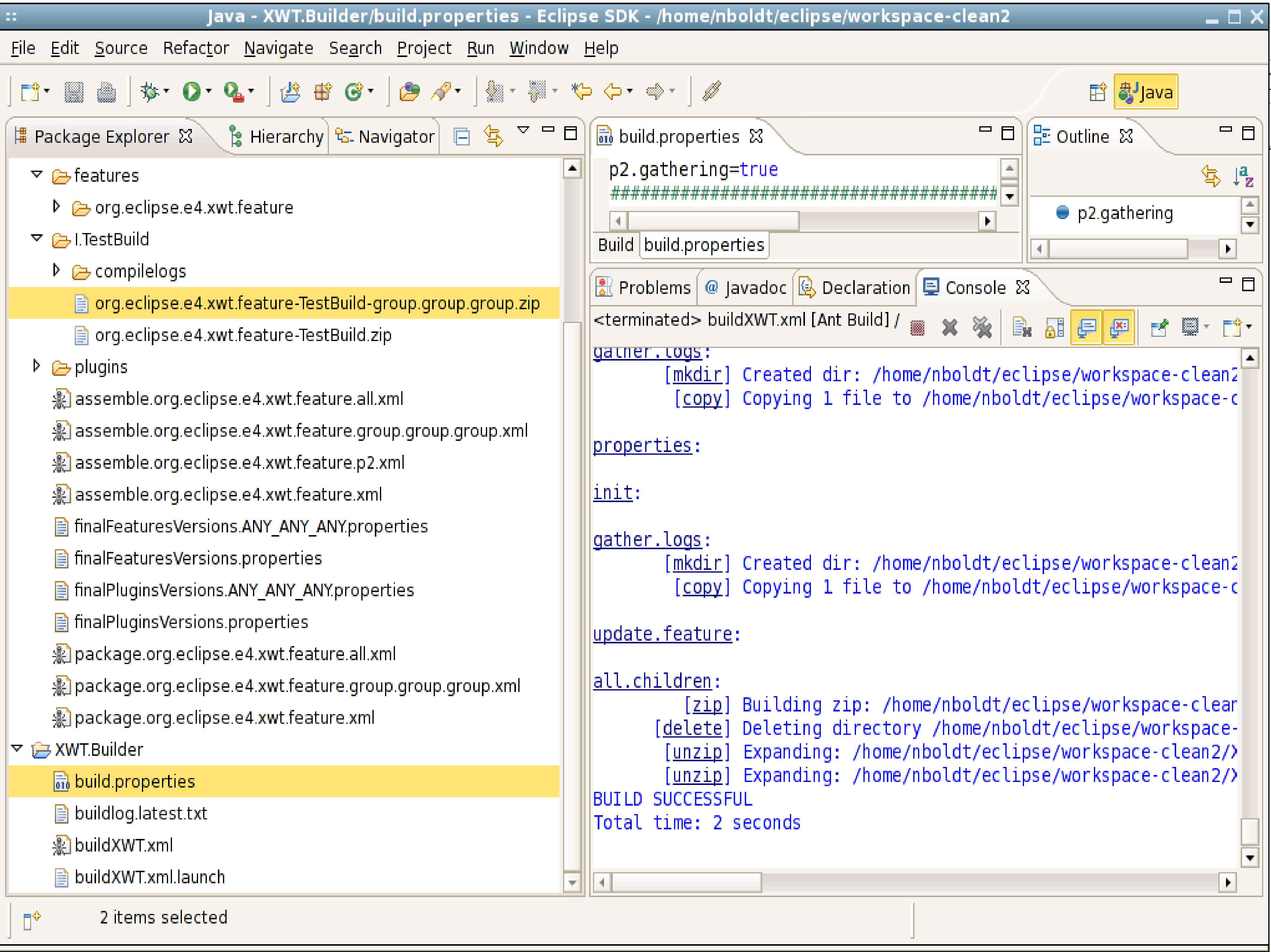
```
generate.p2.metadata=true
p2.metadata.repo=file:${buildDirectory}/repo
p2.artifact.repo=file:${buildDirectory}/repo
p2.publish.artifacts=true
```

3.5M6 version: Add the following property

```
p2.gathering=true
```

And notice the resulting archive is now a p2 repository.

```
org.eclipse.e4.xwt.feature-TestBuild-
  group.group.group.zip
```


Java - XWT.Builder/build.properties - Eclipse SDK - /home/nboldt/eclipse/workspace-clean2
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
Java
Package Explorer
Hierarchy
Navigator
features
org.eclipse.e4.xwt.feature
I.TestBuild
compilelogs
org.eclipse.e4.xwt.feature-TestBuild-group.group.group.zip
org.eclipse.e4.xwt.feature-TestBuild.zip
plugins
assemble.org.eclipse.e4.xwt.feature.all.xml
assemble.org.eclipse.e4.xwt.feature.group.group.group.xml
assemble.org.eclipse.e4.xwt.feature.p2.xml
assemble.org.eclipse.e4.xwt.feature.xml
finalFeaturesVersions.ANY_ANY_ANY.properties
finalFeaturesVersions.properties
finalPluginsVersions.ANY_ANY_ANY.properties
finalPluginsVersions.properties
package.org.eclipse.e4.xwt.feature.all.xml
package.org.eclipse.e4.xwt.feature.group.group.group.xml
package.org.eclipse.e4.xwt.feature.xml
XWT.Builder
build.properties
buildlog.latest.txt
buildXWT.xml
buildXWT.xml.launch
build.properties
Build
build.properties
Outline
p2.gathering
Problems
Javadoc
Declaration
Console
2 items selected

```
p2.gathering=true
########################################
```

```
<terminated> buildXWT.xml [Ant Build] /
gather.logs:
       [mkdir] Created dir: /home/nboldt/eclipse/workspace-clean2
        [copy] Copying 1 file to /home/nboldt/eclipse/workspace-c

properties:

init:

gather.logs:
       [mkdir] Created dir: /home/nboldt/eclipse/workspace-clean2
        [copy] Copying 1 file to /home/nboldt/eclipse/workspace-c

update.feature:

all.children:
         [zip] Building zip: /home/nboldt/eclipse/workspace-clear
      [delete] Deleting directory /home/nboldt/eclipse/workspace-
       [unzip] Expanding: /home/nboldt/eclipse/workspace-clean2/>
       [unzip] Expanding: /home/nboldt/eclipse/workspace-clean2/>
BUILD SUCCESSFUL
Total time: 2 seconds
```

Test Installation of Built Feature

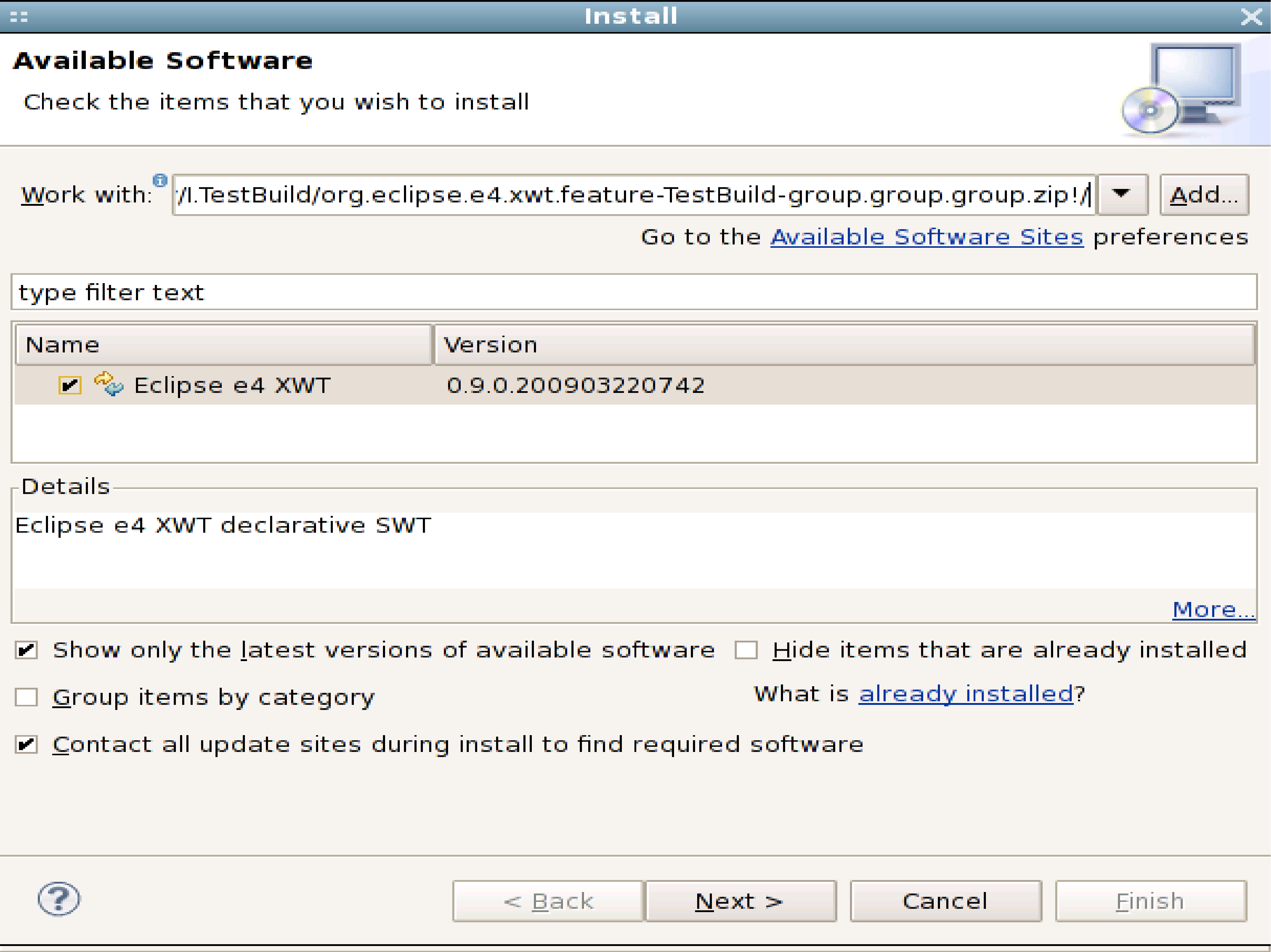
- For normal zip, unpack into `~/eclipse/dropins/` and restart Eclipse using `File > Restart`.
  - `cd ~/workspace/XWT.BuildDirectory/I.TestBuild`

  -
    ```
    unzip org.eclipse.e4.xwt.feature-TestBuild.zip \
      -d ~/eclipse/dropins/
    ```

- Check `Help > About > Installation Details > Installed Software`. Note the version number, eg., **0.9.0.v200903220728.**

Test Installation of Built Repo

- Next, install from the archived p2 repo.
- `Help > Install New Software... > Add`
- Browse for `org.eclipse.e4.xwt.feature-TestBuild-group.group.group.zip`
- Install the XWT feature (you may need to uncheck the `Group items by category` checkbox).
- Check `Help > About > Installation Details > Installed Software`. Note the version number will have increased, eg., **0.9.0.v200903220742.**



Available Software

Check the items that you wish to install

Work with: /I.TestBuild/org.eclipse.e4.xwt.feature-TestBuild-group.group.group.zip!/

Go to the Available Software Sites preferences

type filter text

Name	Version
☑ Eclipse e4 XWT	0.9.0.200903220742

Details

Eclipse e4 XWT declarative SWT

☑ Show only the latest versions of available software

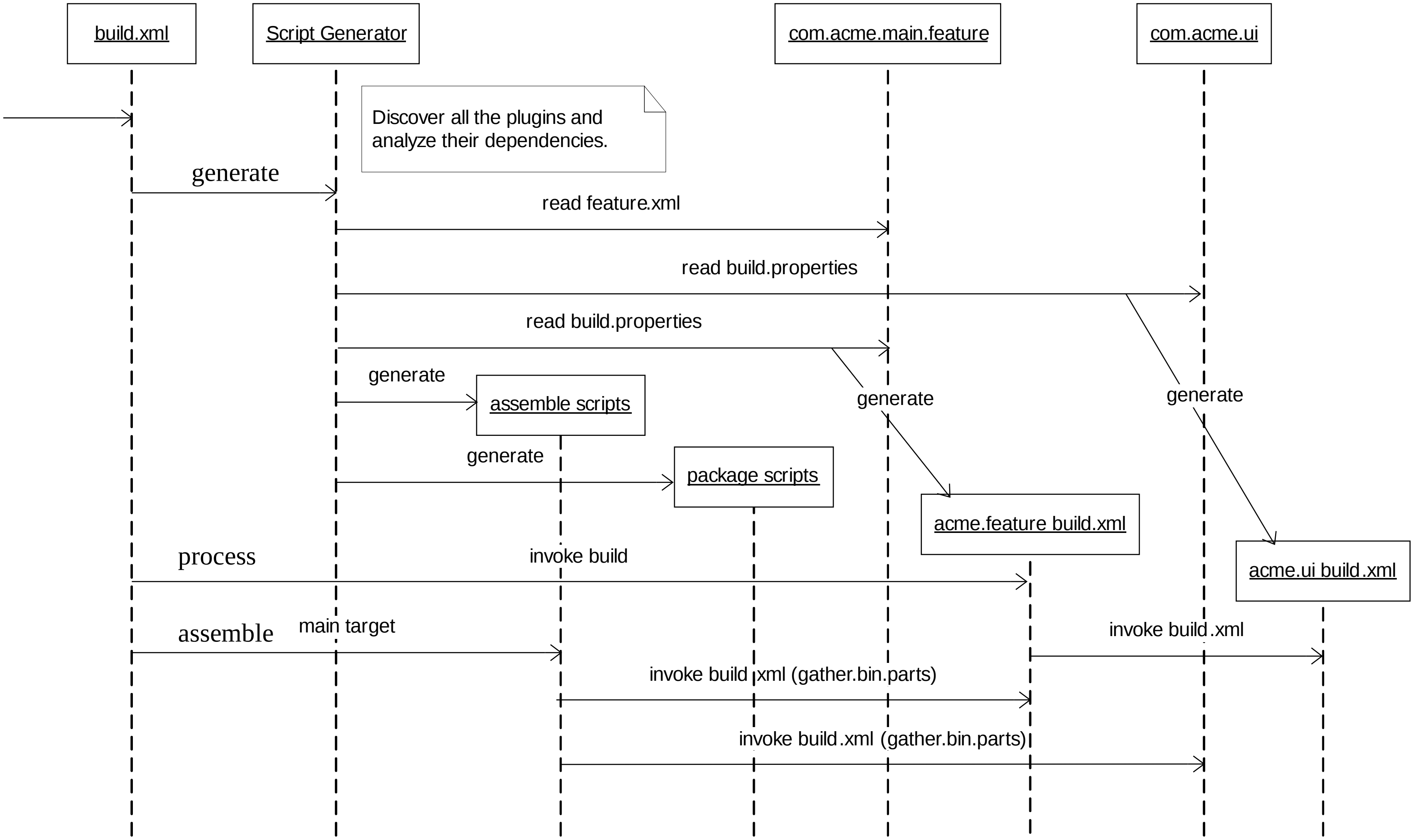
□ Hide items that are already installed

□ Group items by category

What is already installed?

☑ Contact all update sites during install to find required software

Flow of the build



Digging Deeper

- Explore the generated feature & plugin build.xml
  - all.plugins, build.jars
  - @dot, gather.bin.parts

- Import org.eclipse.pde.build into workspace
  - File > Import > Plug-in Development > Plug-ins and Fragments > org.eclipse.pde.build
  - Look in scripts/build.xml: main
  - Look in assemble*.xml and package*.xml
  - Look in allElements.xml

- PDE Build Templates
  - Look in templates/headless-build/customTargets.xml

Digging Deeper (cont'd)

- XWT.Builder/build.properties
  - RunPackager
  - allowBinaryCycles
  - flattenDependencies, parallelCompilation
    - New in Eclipse 3.5M6!
    - No longer need to worry about the order of subfeatures in a given feature.xml
  - outputUpdateJars, generateJnlp
  - forceContextQualifier = v20090323-0800
  - generateFeatureVersionSuffix = hash of contents
  - PluginPath
  - BREEs, bootclasspaths, javacSource, javacTarget

What else can PDE Build do?

- Download zip to use as a target
- Fetch map files and sources
- Hook in custom steps
- Generate source bundles/features
- Build products

Exercise Review

- Review structure of PDE Build process
- Review contents of build.properties, what they do, etc.
- Review important locations
- Locations and how they fit with Athena

Agenda

- Introduction
- PDE Build
  - Exercise: Build a Feature
  - Exercise: Generate a p2 Repository
- **Common Build Background**
- Dash Athena
  - Exercise: Setup
  - Exercise: Configure, Run, Troubleshoot
  - Exercise: Verify Build
- Meet Hudson
- BYO Build Clinic

Common Build Background

- Basebuilder and Releng Projects
- Common Build History
- Build Workshops
- Dash Athena Project

Basebuilder and Releng Projects

- Basebuilder
  - Subset of the Eclipse SDK (CVS, JDT, PDE) required to run PDE Build
  - Includes org.eclipse.build.tools and custom ant tasks
- Releng Project
  - Uses org.eclipse.releng.basebuilder
  - Defines build information
    - Project metadata (name, version)
    - Source locations & branches/tags (map file(s))
    - Optional extra packaging steps
    - Defines 3rd-party requirements
      - eg., Linux Tools requires CDT

Common Build History

- Started with one build, OCL, Nov 2005
- By July 2006, there were 10 divergent builds
  - 3rd party jars
  - multiple upstream dependencies (eg., Eclipse, EMF, UML2, OCL)
  - JDK™ 1.4 & 5.0
- Now, with Xtext, there are more than 20!
  - Some with as many as 10 upstream required builds
- Dash Athena Builder under development
  - 2 projects so far (Linux Tools, GEF)

Build Workshops

- **Sept 2006** http://wiki.eclipse.org/Build Workshop 1: Build Hard
  - Lots of release engineers from Eclipse projects
  - More talk than walk

- **June 2008** http://wiki.eclipse.org/Build Workshop 2: Build Harder
  - Nick, Bjorn, Denis, et al
  - Athena project started with first 3 committers
  - GEF build is ported to run headless on build.eclipse.org

Build Workshops (cont'd)

- **Oct 2008** http://wiki.eclipse.org/Build Workshop 3: Build Hard With A Purpose
  - Nick, Andrew O, Andrew N, Kim, Pascal, et al
  - Athena grows to 6 committers
  - Releng project greatly simplified

- **Feb 2008** http://wiki.eclipse.org/Build Workshop 4: Code's Free So Build Hard
  - Nick, Andrew O, Bjorn
  - Support for building in Eclipse on Linux, Mac and Windows
  - Support for building, testing, and jar signing via Hudson

- **Ongoing**
  - http://wiki.eclipse.org/Build_Workshop_Report
  - http://wiki.eclipse.org/Category:Common_Builder
  - http://www.eclipse.org/projects/project-plan.php?planurl=
    http://www.eclipse.org/dash/athena/project-info/plan.xml&component=Athena

Dash Athena Project

- What it is
  - A work in progress!
  - A wrapper for PDE Build using org.eclipse.releng.basebuilder
  - Ant tasks backed by some shell/exec calls (less over time)

- What it does
  - Standardizes paths and variables
  - Simplifies default offering, while allowing for customizations
  - Uses Ant Contrib, Ant4Eclipse, and PDE SVN Plugin

- Where it runs
  - In Eclipse (Windows, Mac, Linux 32/64)
  - In Hudson & via commandline (Linux 32/64)

Agenda

- **Introduction**

- PDE Build
  - Exercise: Build a Feature
  - Exercise: Generate a p2 Repository

- Common Build Background

- **Dash Athena**
  - Exercise: Setup
  - Exercise: Configure, Run, Troubleshoot
  - Exercise: Verify Build

- Meet Hudson

- BYO Build Clinic

Exercise: Setup

(Network-based config; skip if projects already in workspace)

1. Download Team Project Set File (PSF):
   http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.dash/athena/org.eclipse.dash.commonbuilder/org.eclipse.dash.common.releng.tutorial/psf/gef.psf?revision=1.1.2.1&root=Technology_Project

2. Import PSF using `File > Import > Team > Team Project Set`

3. You should now have these projects:
   - `org.eclipse.releng.basebuilder`
   - `org.eclipse.dash.common.releng`
   - `org.eclipse.gef.releng`

4. Download ant-contrib-1.0b2 (not 1.0b3!):
   http://downloads.sourceforge.net/ant-contrib/ant-contrib-1.0b2-bin.zip

5. Unpack it, and put the renamed jar here:
   - `org.eclipse.dash.common.releng/lib/ant-contrib.jar`

Building In Eclipse Using Athena

- Modify properties for your machine
  - Paths for Java, Ant, workspace dir, build dir
- Use local source checkout if you prefer
  - Subsequent builds are faster, consume less bandwidth
- Right-click releng project's build.xml
  - Run As > Ant Build
- Console log stored in releng project
  - buildlog.latest.txt

Exercise: Configure & Run

- You should already have the example project `org.eclipse.gef.releng` checked out from Athena repository (see previous exercise)

- Open `build.properties`

- Right-click releng project's `build.xml`
  `Run As > Ant Build`

- Check console log output in
  `buildlog.latest.txt`

Athena build.properties

- What & how to build:

```
projectid=tools.gef
zipPrefix=GEF
version=3.5.0
buildType=N
mainFeatureToBuildID=org.eclipse.gef.all
testFeatureToBuildID=org.eclipse.gef.test

JAVA_HOME=/path/to/java
JAVA14_HOME=/path/to/java
JAVADOC14_HOME=/path/to/java/bin

dependencyURLs=http://download.eclipse.org/\
  eclipse/downloads/drops/\
  S-3.5M5-200902021535/\
  eclipse-SDK-3.5M5-linux-gtk.tar.gz
```

Athena build.properties (cont'd)

- Where to get sources:
  - If building from CVS or SVN, use map file(s)
  - If building locally, use:
    ```
    localSourceCheckoutDir=\
      C:/workspace/org.eclipse.gef.tree
    ```
- To get local cache, import using `gef-tree.psf`
- Or via command line:

  -
    ```
    cd ~/workspace; cvs -d :pserver:
       anonymous@dev.eclipse.org:\
    /cvsroot/tools -q co -d org.eclipse.gef.tree \
    org.eclipse.gef
    ```

Troubleshooting

- Missing classpath or bootclasspath entries
  - `ant-contrib.jar (wrong version!)`
  - `org.eclipse.pde.build.svn_*.jar`
  - `dt.jar, rt.jar, classes.jar`
- Map file problems
  - Missing file? Missing entries in file?
  - Wrong feature or plugin id to path mapping?
  - Invalid cvs/svn credentials?
- Testing problems
  - Need Xvnc or Xvfb to run headless tests
  - Missing org.eclipse.test or org.eclipse.ant.optional.junit?
  - Note: tests do not currently run headless on Mac or Windows

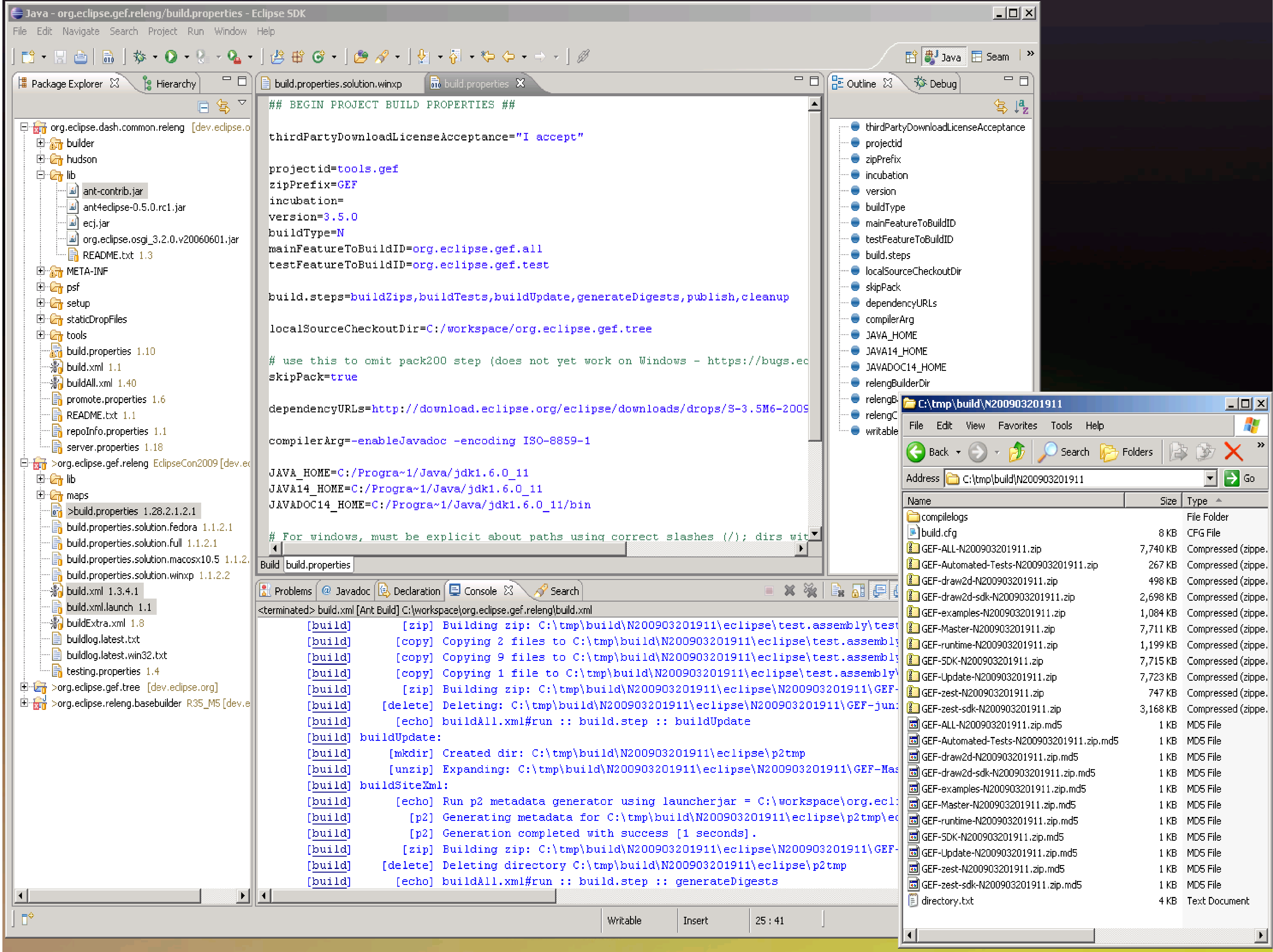
Troubleshooting (cont'd)

- Building from SVN sources
  - Need an SVN client installed; PDE SVN does not include its own SVN implementation
- Platform-specific configuration tips

http://wiki.eclipse.org/Common_Build_Infrastructure/Virtual_Server_Setup/Fedora

http://wiki.eclipse.org/Common_Build_Infrastructure/Virtual_Server_Setup/MacOSX

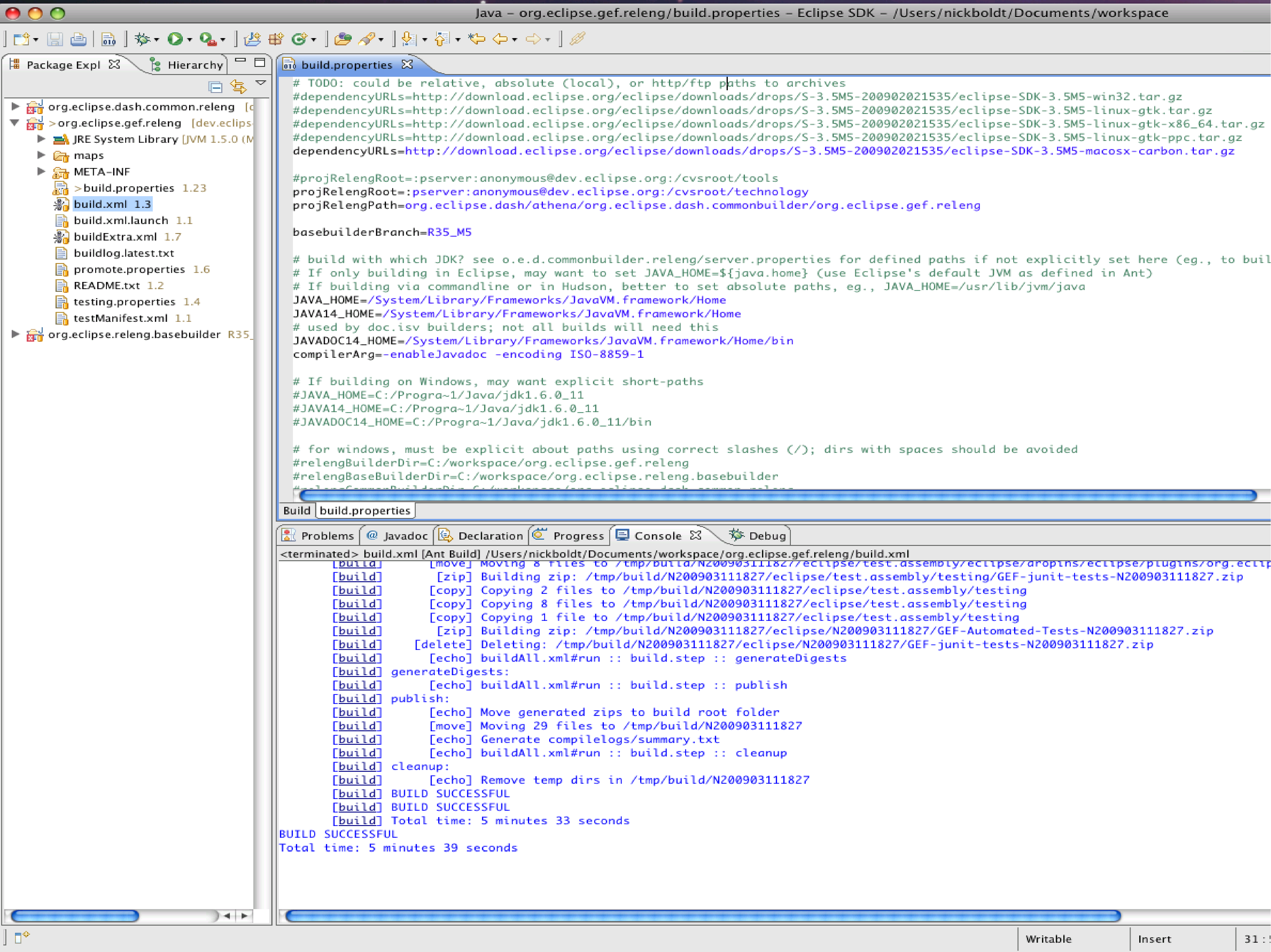
http://wiki.eclipse.org/Common_Build_Infrastructure/Virtual_Server_Setup/WinXP


Java - org.eclipse.gef.releng/build.properties - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Java
Seam
Package Explorer
Hierarchy
org.eclipse.dash.common.releng [dev.eclipse.o
builder
hudson
lib
ant-contrib.jar
ant4eclipse-0.5.0.rc1.jar
ecj.jar
org.eclipse.osgi_3.2.0.v20060601.jar
README.txt 1.3
META-INF
psf
setup
staticDropFiles
tools
build.properties 1.10
build.xml 1.1
buildAll.xml 1.40
promote.properties 1.6
README.txt 1.1
repoInfo.properties 1.1
server.properties 1.18
>org.eclipse.gef.releng EclipseCon2009 [dev.ec
lib
maps
>build.properties 1.28.2.1.2.1
build.properties.solution.fedora 1.1.2.1
build.properties.solution.full 1.1.2.1
build.properties.solution.macosx10.5 1.1.2.
build.properties.solution.winxp 1.1.2.2
build.xml 1.3.4.1
build.xml.launch 1.1
buildExtra.xml 1.8
buildlog.latest.txt
buildlog.latest.win32.txt
testing.properties 1.4
>org.eclipse.gef.tree [dev.eclipse.org]
>org.eclipse.releng.basebuilder R35_M5 [dev.e
build.properties.solution.winxp
build.properties
Outline
Debug
thirdPartyDownloadLicenseAcceptance
projectid
zipPrefix
incubation
version
buildType
mainFeatureToBuildID
testFeatureToBuildID
build.steps
localSourceCheckoutDir
skipPack
dependencyURLs
compilerArg
JAVA_HOME
JAVA14_HOME
JAVADOC14_HOME
relengBuilderDir
relengB
relengC
writable
Build
build.properties
Problems
Javadoc
Declaration
Console
Search
Writable
Insert
25 : 41
C:\tmp\build\N200903201911
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\tmp\build\N200903201911
Go

```
## BEGIN PROJECT BUILD PROPERTIES ##

thirdPartyDownloadLicenseAcceptance="I accept"

projectid=tools.gef
zipPrefix=GEF
incubation=
version=3.5.0
buildType=N
mainFeatureToBuildID=org.eclipse.gef.all
testFeatureToBuildID=org.eclipse.gef.test

build.steps=buildZips,buildTests,buildUpdate,generateDigests,publish,cleanup

localSourceCheckoutDir=C:/workspace/org.eclipse.gef.tree

# use this to omit pack200 step (does not yet work on Windows - https://bugs.ec
skipPack=true

dependencyURLs=http://download.eclipse.org/eclipse/downloads/drops/S-3.5M6-2009

compilerArg=-enableJavadoc -encoding ISO-8859-1

JAVA_HOME=C:/Progra~1/Java/jdk1.6.0_11
JAVA14_HOME=C:/Progra~1/Java/jdk1.6.0_11
JAVADOC14_HOME=C:/Progra~1/Java/jdk1.6.0_11/bin

# For windows, must be explicit about paths using correct slashes (/); dirs wit
```

```
<terminated> build.xml [Ant Build] C:\workspace\org.eclipse.gef.releng\build.xml
      [build]        [zip] Building zip: C:\tmp\build\N200903201911\eclipse\test.assembly\test
      [build]       [copy] Copying 2 files to C:\tmp\build\N200903201911\eclipse\test.assembly
      [build]       [copy] Copying 9 files to C:\tmp\build\N200903201911\eclipse\test.assembly
      [build]       [copy] Copying 1 file to C:\tmp\build\N200903201911\eclipse\test.assembly\
      [build]        [zip] Building zip: C:\tmp\build\N200903201911\eclipse\N200903201911\GEF-
      [build]     [delete] Deleting: C:\tmp\build\N200903201911\eclipse\N200903201911\GEF-jun
      [build]       [echo] buildAll.xml#run :: build.step :: buildUpdate
      [build] buildUpdate:
      [build]      [mkdir] Created dir: C:\tmp\build\N200903201911\eclipse\p2tmp
      [build]      [unzip] Expanding: C:\tmp\build\N200903201911\eclipse\N200903201911\GEF-Mas
      [build] buildSiteXml:
      [build]       [echo] Run p2 metadata generator using launcherjar = C:\workspace\org.ecl
      [build]         [p2] Generating metadata for C:\tmp\build\N200903201911\eclipse\p2tmp\ec
      [build]         [p2] Generation completed with success [1 seconds].
      [build]        [zip] Building zip: C:\tmp\build\N200903201911\eclipse\N200903201911\GEF-
      [build]     [delete] Deleting directory C:\tmp\build\N200903201911\eclipse\p2tmp
      [build]       [echo] buildAll.xml#run :: build.step :: generateDigests
```

Name	Size	Type
compilelogs		File Folder
build.cfg	8 KB	CFG File
GEF-ALL-N200903201911.zip	7,740 KB	Compressed (zippe
GEF-Automated-Tests-N200903201911.zip	267 KB	Compressed (zippe
GEF-draw2d-N200903201911.zip	498 KB	Compressed (zippe
GEF-draw2d-sdk-N200903201911.zip	2,698 KB	Compressed (zippe
GEF-examples-N200903201911.zip	1,084 KB	Compressed (zippe
GEF-Master-N200903201911.zip	7,711 KB	Compressed (zippe
GEF-runtime-N200903201911.zip	1,199 KB	Compressed (zippe
GEF-SDK-N200903201911.zip	7,715 KB	Compressed (zippe
GEF-Update-N200903201911.zip	7,723 KB	Compressed (zippe
GEF-zest-N200903201911.zip	747 KB	Compressed (zippe
GEF-zest-sdk-N200903201911.zip	3,168 KB	Compressed (zippe
GEF-ALL-N200903201911.zip.md5	1 KB	MD5 File
GEF-Automated-Tests-N200903201911.zip.md5	1 KB	MD5 File
GEF-draw2d-N200903201911.zip.md5	1 KB	MD5 File
GEF-draw2d-sdk-N200903201911.zip.md5	1 KB	MD5 File
GEF-examples-N200903201911.zip.md5	1 KB	MD5 File
GEF-Master-N200903201911.zip.md5	1 KB	MD5 File
GEF-runtime-N200903201911.zip.md5	1 KB	MD5 File
GEF-SDK-N200903201911.zip.md5	1 KB	MD5 File
GEF-Update-N200903201911.zip.md5	1 KB	MD5 File
GEF-zest-N200903201911.zip.md5	1 KB	MD5 File
GEF-zest-sdk-N200903201911.zip.md5	1 KB	MD5 File
directory.txt	4 KB	Text Document



```
# TODO: could be relative, absolute (local), or http/ftp paths to archives
#dependencyURLs=http://download.eclipse.org/eclipse/downloads/drops/S-3.5M5-200902021535/eclipse-SDK-3.5M5-win32.tar.gz
#dependencyURLs=http://download.eclipse.org/eclipse/downloads/drops/S-3.5M5-200902021535/eclipse-SDK-3.5M5-linux-gtk.tar.gz
#dependencyURLs=http://download.eclipse.org/eclipse/downloads/drops/S-3.5M5-200902021535/eclipse-SDK-3.5M5-linux-gtk-x86_64.tar.gz
#dependencyURLs=http://download.eclipse.org/eclipse/downloads/drops/S-3.5M5-200902021535/eclipse-SDK-3.5M5-linux-gtk-ppc.tar.gz
dependencyURLs=http://download.eclipse.org/eclipse/downloads/drops/S-3.5M5-200902021535/eclipse-SDK-3.5M5-macosx-carbon.tar.gz

#projRelengRoot=:pserver:anonymous@dev.eclipse.org:/cvsroot/tools
projRelengRoot=:pserver:anonymous@dev.eclipse.org:/cvsroot/technology
projRelengPath=org.eclipse.dash/athena/org.eclipse.dash.commonbuilder/org.eclipse.gef.releng

basebuilderBranch=R35_M5

# build with which JDK? see o.e.d.commonbuilder.releng/server.properties for defined paths if not explicitly set here (eg., to buil
# If only building in Eclipse, may want to set JAVA_HOME=${java.home} (use Eclipse's default JVM as defined in Ant)
# If building via commandline or in Hudson, better to set absolute paths, eg., JAVA_HOME=/usr/lib/jvm/java
JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Home
JAVA14_HOME=/System/Library/Frameworks/JavaVM.framework/Home
# used by doc.isv builders; not all builds will need this
JAVADOC14_HOME=/System/Library/Frameworks/JavaVM.framework/Home/bin
compilerArg=-enableJavadoc -encoding ISO-8859-1

# If building on Windows, may want explicit short-paths
#JAVA_HOME=C:/Progra~1/Java/jdk1.6.0_11
#JAVA14_HOME=C:/Progra~1/Java/jdk1.6.0_11
#JAVADOC14_HOME=C:/Progra~1/Java/jdk1.6.0_11/bin

# for windows, must be explicit about paths using correct slashes (/); dirs with spaces should be avoided
#relengBuilderDir=C:/workspace/org.eclipse.gef.releng
#relengBaseBuilderDir=C:/workspace/org.eclipse.releng.basebuilder
#relengCommonBuilderDir=C:/workspace/org.eclipse.dash.common.releng
```

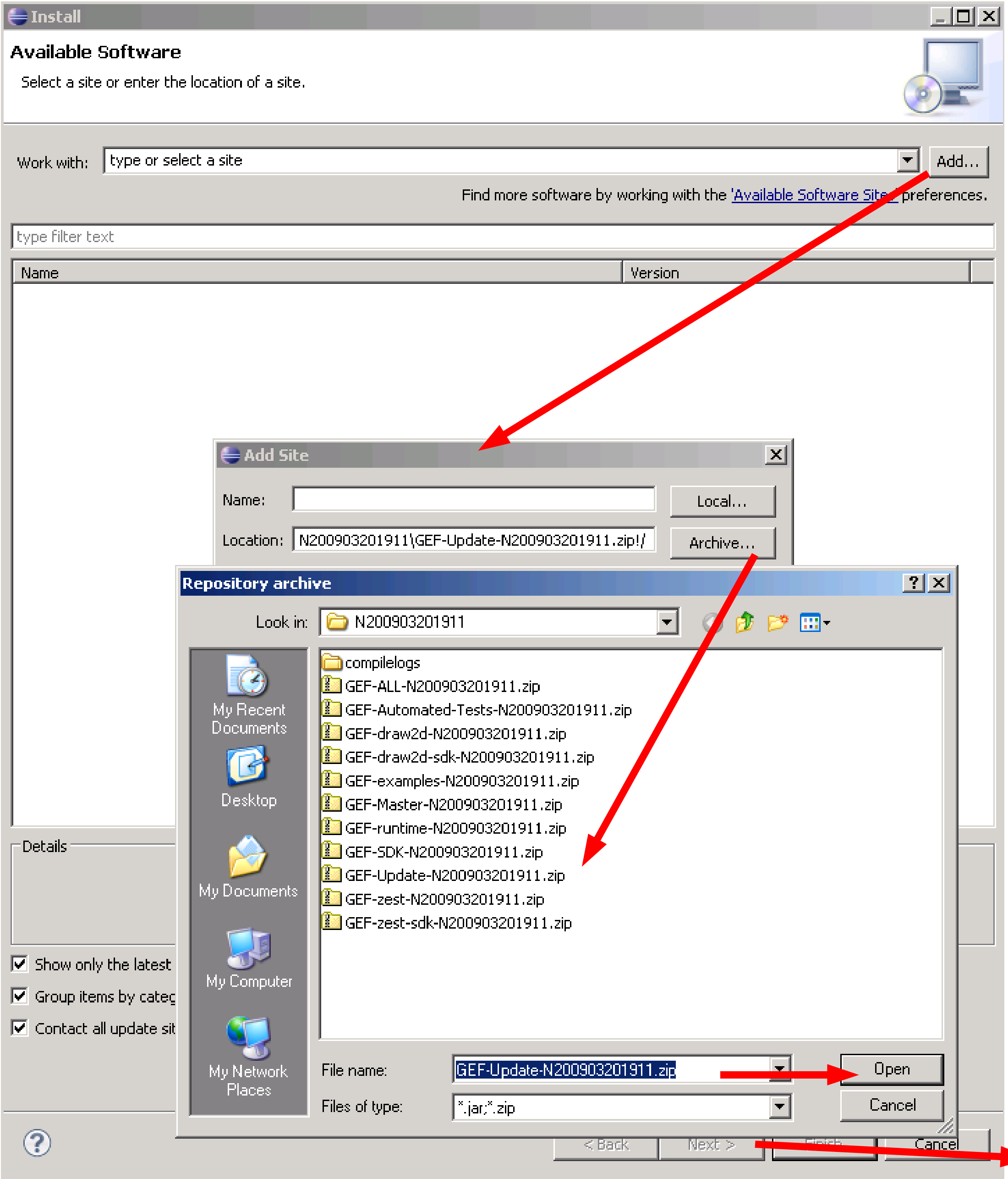
```
<terminated> build.xml [Ant Build] /Users/nickboldt/Documents/workspace/org.eclipse.gef.releng/build.xml
     [build]      [move] Moving 8 files to /tmp/build/N200903111827/eclipse/test.assembly/eclipse/dropins/eclipse/plugins/org.eclip
     [build]       [zip] Building zip: /tmp/build/N200903111827/eclipse/test.assembly/testing/GEF-junit-tests-N200903111827.zip
     [build]      [copy] Copying 2 files to /tmp/build/N200903111827/eclipse/test.assembly/testing
     [build]      [copy] Copying 8 files to /tmp/build/N200903111827/eclipse/test.assembly/testing
     [build]      [copy] Copying 1 file to /tmp/build/N200903111827/eclipse/test.assembly/testing
     [build]       [zip] Building zip: /tmp/build/N200903111827/eclipse/N200903111827/GEF-Automated-Tests-N200903111827.zip
     [build]    [delete] Deleting: /tmp/build/N200903111827/eclipse/N200903111827/GEF-junit-tests-N200903111827.zip
     [build]      [echo] buildAll.xml#run :: build.step :: generateDigests
     [build] generateDigests:
     [build]      [echo] buildAll.xml#run :: build.step :: publish
     [build] publish:
     [build]      [echo] Move generated zips to build root folder
     [build]      [move] Moving 29 files to /tmp/build/N200903111827
     [build]      [echo] Generate compilelogs/summary.txt
     [build]      [echo] buildAll.xml#run :: build.step :: cleanup
     [build] cleanup:
     [build]      [echo] Remove temp dirs in /tmp/build/N200903111827
     [build] BUILD SUCCESSFUL
     [build] BUILD SUCCESSFUL
     [build] Total time: 5 minutes 33 seconds
BUILD SUCCESSFUL
Total time: 5 minutes 39 seconds
```

Exercise: Verify Build

- Exercise: Install GEF SDK from Archived p2 Repository (an update site with metadata)
  - `Help > Install New Software... > Add > Archive...`
  - Browse for GEF-Update-*.zip
  - Uncheck '`Group items by category`' checkbox
  - Select GEF SDK feature
  - Restart when prompted
  - Verify GEF SDK is installed
    `Help > About > Installation Details`

- Exercise: Install from SDK zip (Optional)
  - Uninstall previously installed GEF SDK; restart when prompted
  - Verify GEF is no longer installed. Close Eclipse
  - Unpack GEF SDK zip into `eclipse/dropins/gef/`
  - Restart Eclipse
  - Verify GEF SDK is installed
    `Help > About > Installation Details`

Exercise:
Install GEF SDK
From Archived p2 Repo



Available Software

Select a site or enter the location of a site.

Work with: type or select a site

Find more software by working with the 'Available Software Sites' preferences.

☑ Show only the latest

☑ Group items by categ

☑ Contact all update sit

Name:

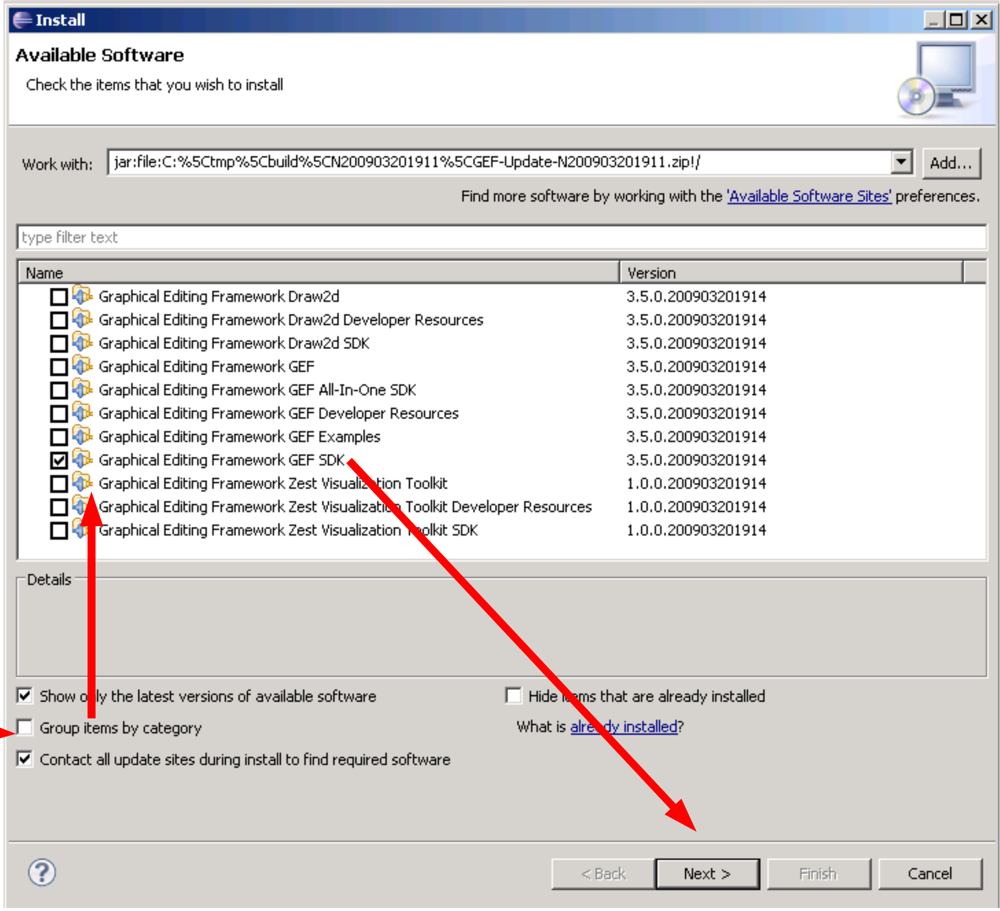
Location: N200903201911\GEF-Update-N200903201911.zip!/

Look in: N200903201911

- compilelogs
- GEF-ALL-N200903201911.zip
- GEF-Automated-Tests-N200903201911.zip
- GEF-draw2d-N200903201911.zip
- GEF-draw2d-sdk-N200903201911.zip
- GEF-examples-N200903201911.zip
- GEF-Master-N200903201911.zip
- GEF-runtime-N200903201911.zip
- GEF-SDK-N200903201911.zip
- GEF-Update-N200903201911.zip
- GEF-zest-N200903201911.zip
- GEF-zest-sdk-N200903201911.zip

File name: GEF-Update-N200903201911.zip

Files of type: *.jar;*.zip



Available Software

Check the items that you wish to install

Work with: jar:file:C:%5Ctmp%5Cbuild%5CN200903201911%5CGEF-Update-N200903201911.zip!/

Find more software by working with the 'Available Software Sites' preferences.

Name	Version
☐ Graphical Editing Framework Draw2d	3.5.0.200903201914
☐ Graphical Editing Framework Draw2d Developer Resources	3.5.0.200903201914
☐ Graphical Editing Framework Draw2d SDK	3.5.0.200903201914
☐ Graphical Editing Framework GEF	3.5.0.200903201914
☐ Graphical Editing Framework GEF All-In-One SDK	3.5.0.200903201914
☐ Graphical Editing Framework GEF Developer Resources	3.5.0.200903201914
☐ Graphical Editing Framework GEF Examples	3.5.0.200903201914
☑ Graphical Editing Framework GEF SDK	3.5.0.200903201914
☐ Graphical Editing Framework Zest Visualization Toolkit	1.0.0.200903201914
☐ Graphical Editing Framework Zest Visualization Toolkit Developer Resources	1.0.0.200903201914
☐ Graphical Editing Framework Zest Visualization Toolkit SDK	1.0.0.200903201914

☑ Show only the latest versions of available software

☐ Hide items that are already installed

☐ Group items by category

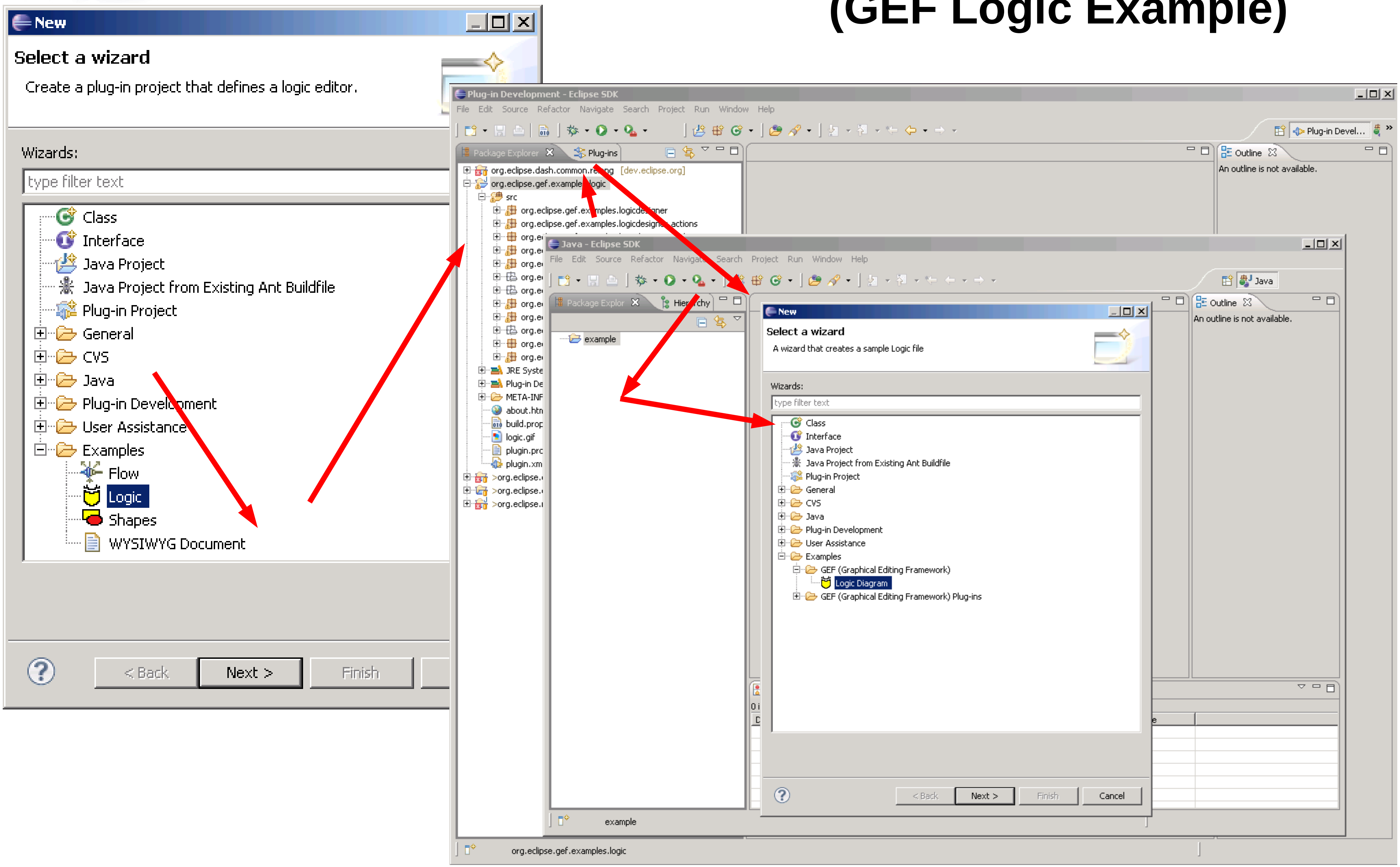
What is already installed?

☑ Contact all update sites during install to find required software

Exercise: Verify Build

- Exercise: Smoke Test (GEF Logic Example)
  - Install GEF SDK as in one of previous exercises
  - `File > New > Example... > Logic`
  - Select generated project
  - `Run > Run As > Eclipse Application`
  - **In second Eclipse instance**
    - `New > Project... > General > Project`
    - `New > Example... > GEF > Logic Diagram`
      - Select `Four-bit Adder Model`
    - Review generated diagram

Exercise: Smoke Test (GEF Logic Example)



New

Select a wizard

Create a plug-in project that defines a logic editor.

Wizards:

type filter text

Class
Interface
Java Project
Java Project from Existing Ant Buildfile
Plug-in Project
General
CVS
Java
Plug-in Development
User Assistance
Examples
Flow
Logic
Shapes
WYSIWYG Document
< Back
Next >
Finish
Plug-in Development - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Plug-ins
Plug-in Devel...
org.eclipse.dash.common.rel[...]ng [dev.eclipse.org]
org.eclipse.gef.examples.logic
src
org.eclipse.gef.examples.logicdesigner
org.eclipse.gef.examples.logicdesigner.actions
JRE Syste
Plug-in De
META-INF
about.htm
build.prop
logic.gif
plugin.prc
plugin.xm
>org.eclipse.
Outline
An outline is not available.
org.eclipse.gef.examples.logic
Java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explor
Hierarchy
Java
example
Outline
An outline is not available.

New

Select a wizard

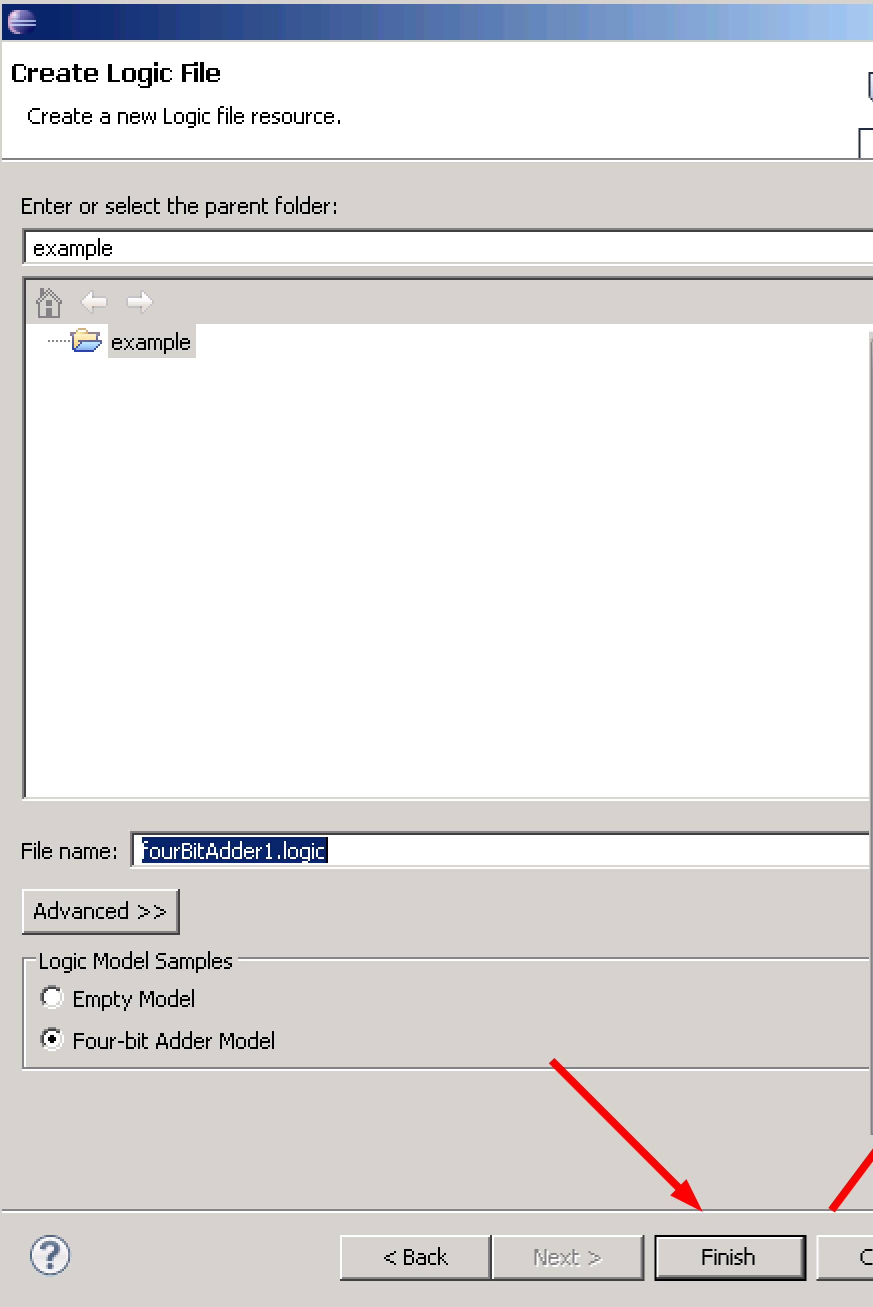
A wizard that creates a sample Logic file

Wizards:

type filter text

Class
Interface
Java Project
Java Project from Existing Ant Buildfile
Plug-in Project
General
CVS
Java
Plug-in Development
User Assistance
Examples
GEF (Graphical Editing Framework)
Logic Diagram
GEF (Graphical Editing Framework) Plug-ins
< Back
Next >
Finish
Cancel
example

Exercise: Smoke Test (GEF Logic Example) (cont'd)



Create Logic File

Create a new Logic file resource.

Enter or select the parent folder: example

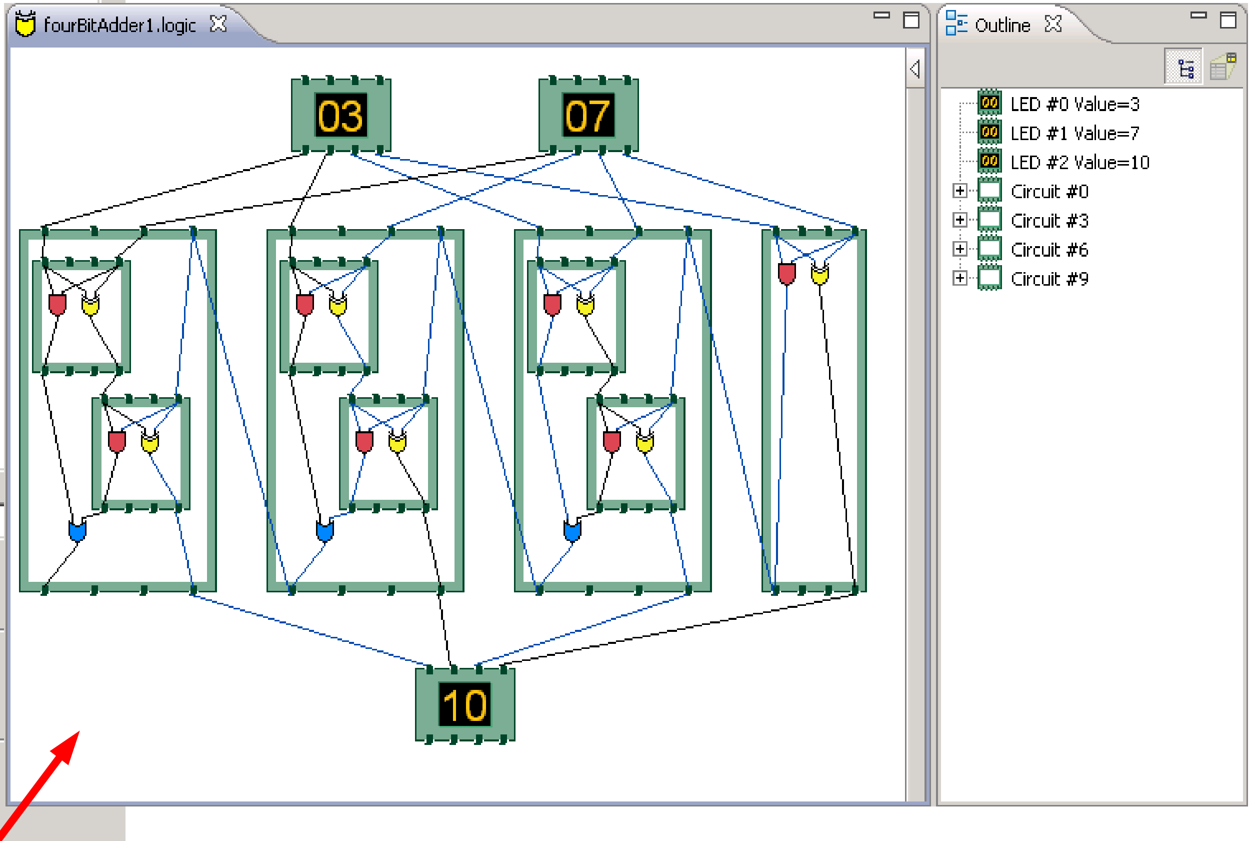
example

File name: fourBitAdder1.logic

Logic Model Samples

○ Empty Model

● Four-bit Adder Model



fourBitAdder1.logic

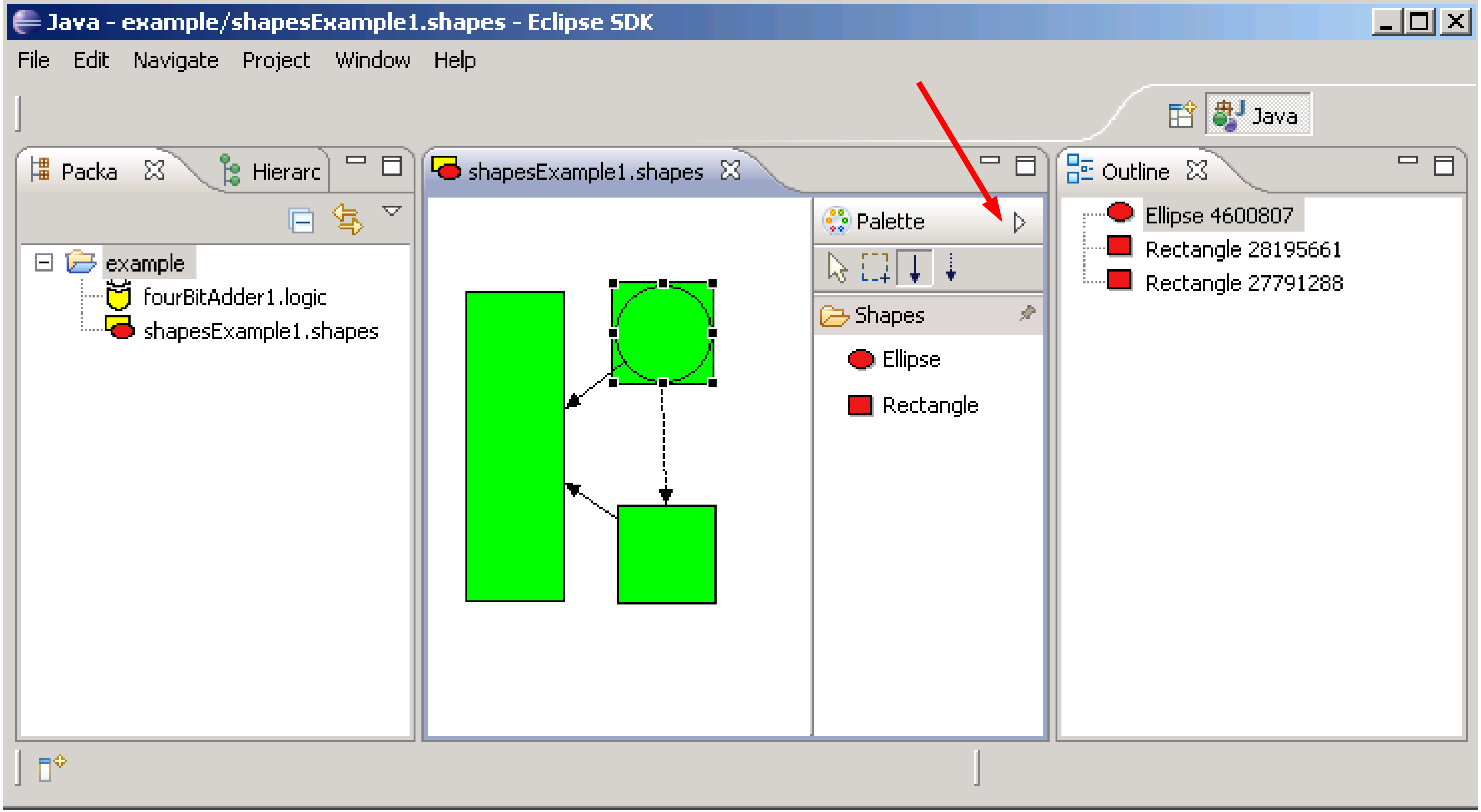
Outline

- LED #0 Value=3
- LED #1 Value=7
- LED #2 Value=10
- Circuit #0
- Circuit #3
- Circuit #6
- Circuit #9

Exercise: Verify Build

- Exercise: Smoke Test (GEF Shapes) (Optional)
  - Install GEF SDK as in one of previous exercises
  - `File > New > Example... > Shapes`
  - Select generated project
  - `Run > Run As > Eclipse Application`
  - **In second Eclipse instance**
    - `New > Project... > General > Project`
    - `New > Example... > GEF > Shapes Diagram`
    - Open palette on right side of editor; create a diagram

Exercise: Smoke Test (GEF Shapes Example)



Agenda

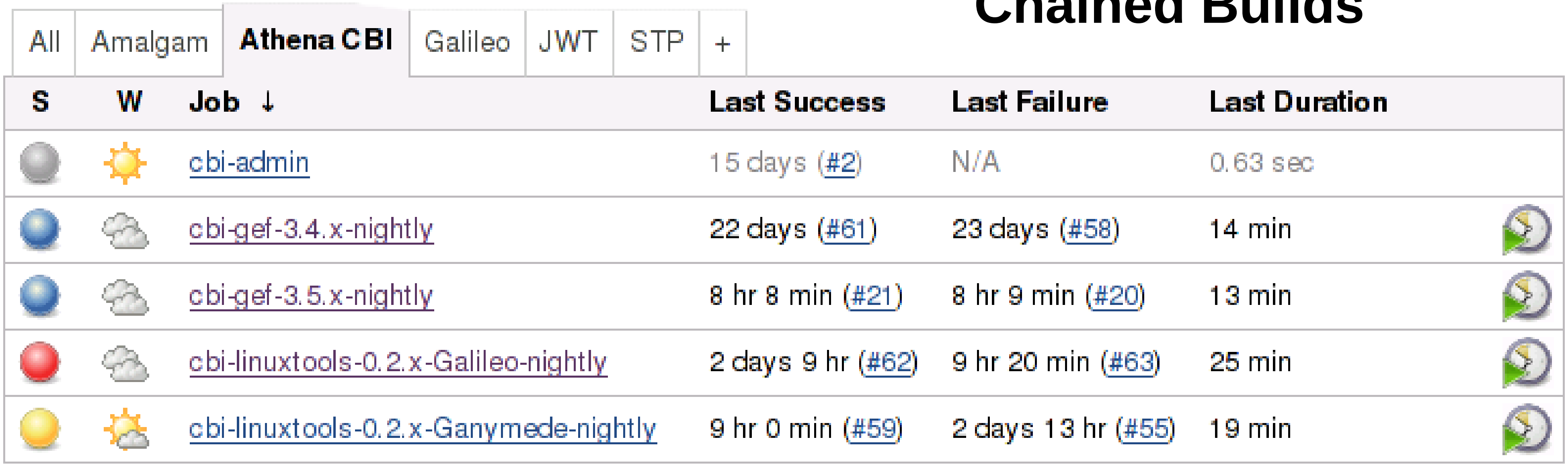
- Introduction
- PDE Build
  - Exercise: Build a Feature
  - Exercise: Generate a p2 Repository
- Common Build Background
- Dash Athena
  - Exercise: Setup
  - Exercise: Configure, Run, Troubleshoot
  - Exercise: Verify Build
- **Meet Hudson**
- BYO Build Clinic

Meet Hudson

- Continuous Integration
  (build when sources change)
- Change Tracking (CVS, SVN, ...)
- Automated Jar Signing
- Extensible Framework (Plug-ins)
- Parameterized Builds, Chained Builds

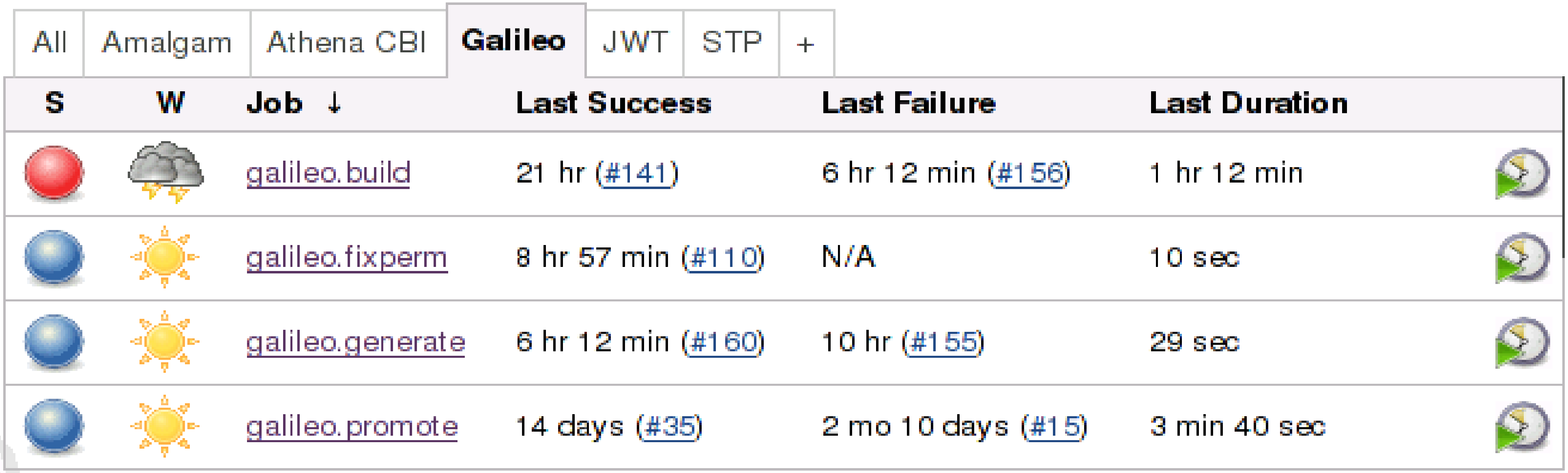
https://hudson.dev.java.net/

Build Groups & Chained Builds



All | Amalgam | **Athena CBI** | Galileo | JWT | STP | +

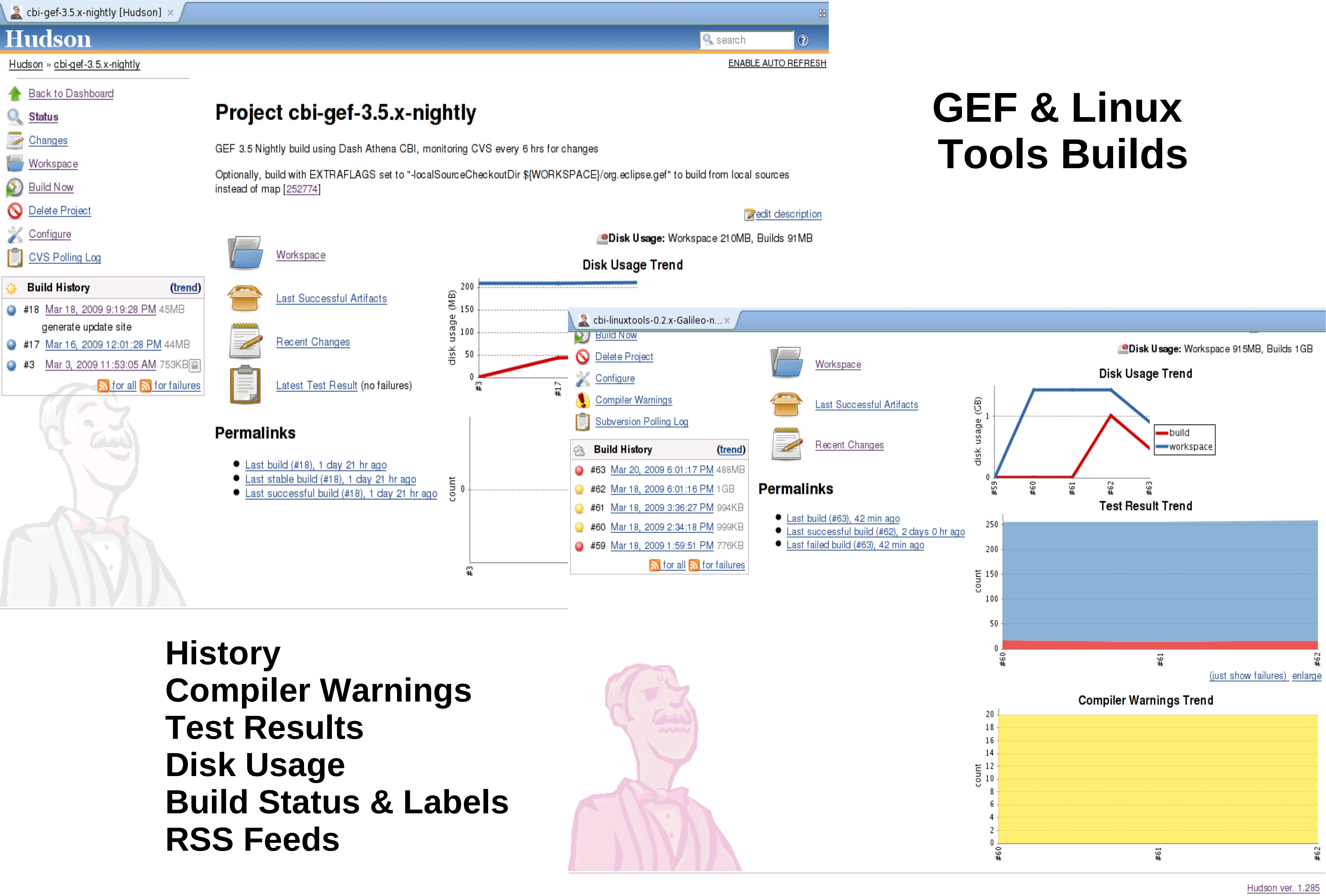
S	W	Job ↓	Last Success	Last Failure	Last Duration
⚪	☀	cbi-admin	15 days (#2)	N/A	0.63 sec
🔵	☁	cbi-gef-3.4.x-nightly	22 days (#61)	23 days (#58)	14 min
🔵	☁	cbi-gef-3.5.x-nightly	8 hr 8 min (#21)	8 hr 9 min (#20)	13 min
🔴	☁	cbi-linuxtools-0.2.x-Galileo-nightly	2 days 9 hr (#62)	9 hr 20 min (#63)	25 min
🟡	⛅	cbi-linuxtools-0.2.x-Ganymede-nightly	9 hr 0 min (#59)	2 days 13 hr (#55)	19 min



All | Amalgam | Athena CBI | **Galileo** | JWT | STP | +

S	W	Job ↓	Last Success	Last Failure	Last Duration
🔴	⛈	galileo.build	21 hr (#141)	6 hr 12 min (#156)	1 hr 12 min
🔵	☀	galileo.fixperm	8 hr 57 min (#110)	N/A	10 sec
🔵	☀	galileo.generate	6 hr 12 min (#160)	10 hr (#155)	29 sec
🔵	☀	galileo.promote	14 days (#35)	2 mo 10 days (#15)	3 min 40 sec

GEF & Linux Tools Builds



cbi-gef-3.5.x-nightly [Hudson]

Hudson

search

Hudson » cbi-gef-3.5.x-nightly

ENABLE AUTO REFRESH

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure
- CVS Polling Log

Build History (trend)

#18 Mar 18, 2009 9:19:28 PM 45MB
generate update site
#17 Mar 16, 2009 12:01:28 PM 44MB
#3 Mar 3, 2009 11:53:05 AM 753KB

for all for failures

Project cbi-gef-3.5.x-nightly

GEF 3.5 Nightly build using Dash Athena CBI, monitoring CVS every 6 hrs for changes

Optionally, build with EXTRAFLAGS set to "-localSourceCheckoutDir ${WORKSPACE}/org.eclipse.gef" to build from local sources instead of map [252774]

edit description

Disk Usage: Workspace 210MB, Builds 91MB

Workspace

Last Successful Artifacts

Recent Changes

Latest Test Result (no failures)

Permalinks

- Last build (#18), 1 day 21 hr ago
- Last stable build (#18), 1 day 21 hr ago
- Last successful build (#18), 1 day 21 hr ago

cbi-linuxtools-0.2.x-Galileo-n...

- Build Now
- Delete Project
- Configure
- Compiler Warnings
- Subversion Polling Log

Build History (trend)

#63 Mar 20, 2009 6:01:17 PM 488MB
#62 Mar 18, 2009 6:01:16 PM 1GB
#61 Mar 18, 2009 3:36:27 PM 994KB
#60 Mar 18, 2009 2:34:18 PM 999KB
#59 Mar 18, 2009 1:59:51 PM 776KB

for all for failures

Workspace

Last Successful Artifacts

Recent Changes

Permalinks

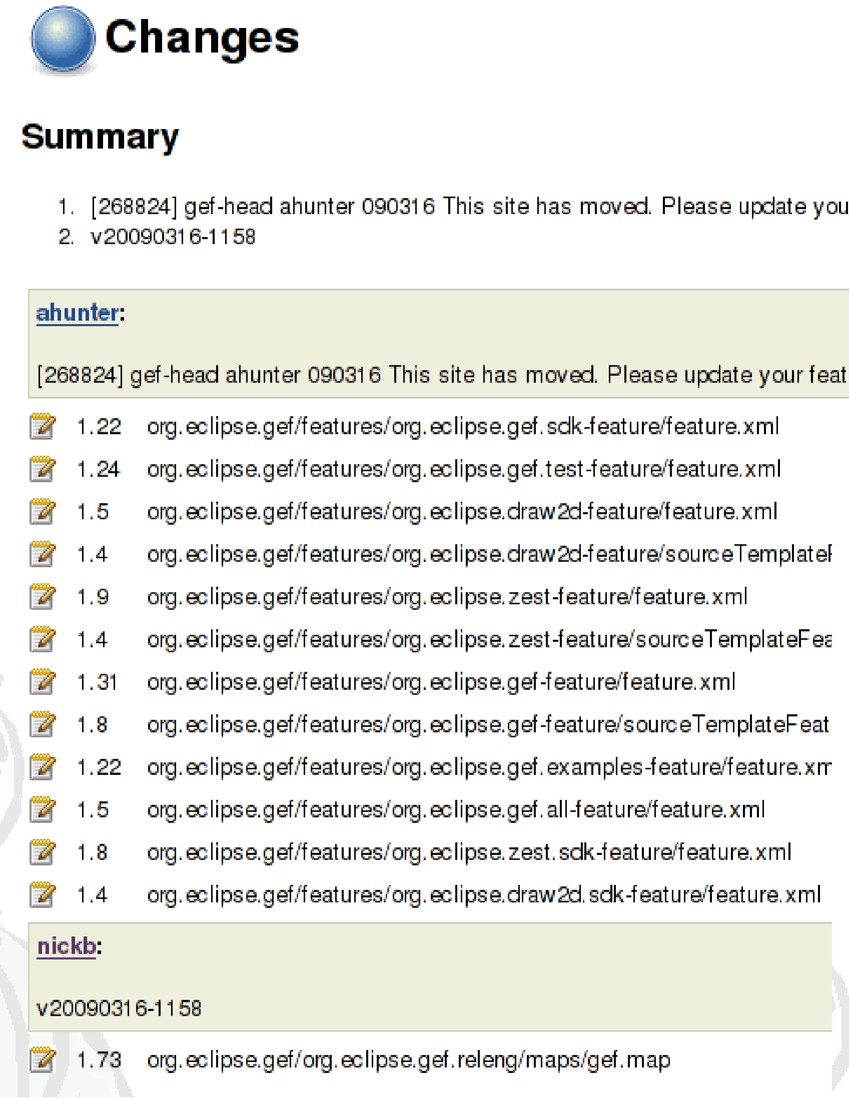
- Last build (#63), 42 min ago
- Last successful build (#62), 2 days 0 hr ago
- Last failed build (#63), 42 min ago

Disk Usage: Workspace 915MB, Builds 1GB

Hudson ver. 1.285

- History
- Compiler Warnings
- Test Results
- Disk Usage
- Build Status & Labels
- RSS Feeds

Source Change Tracking: CVS & SVN



Changes

Summary

1. [268824] gef-head ahunter 090316 This site has moved. Please update your features.
2. v20090316-1158

ahunter:

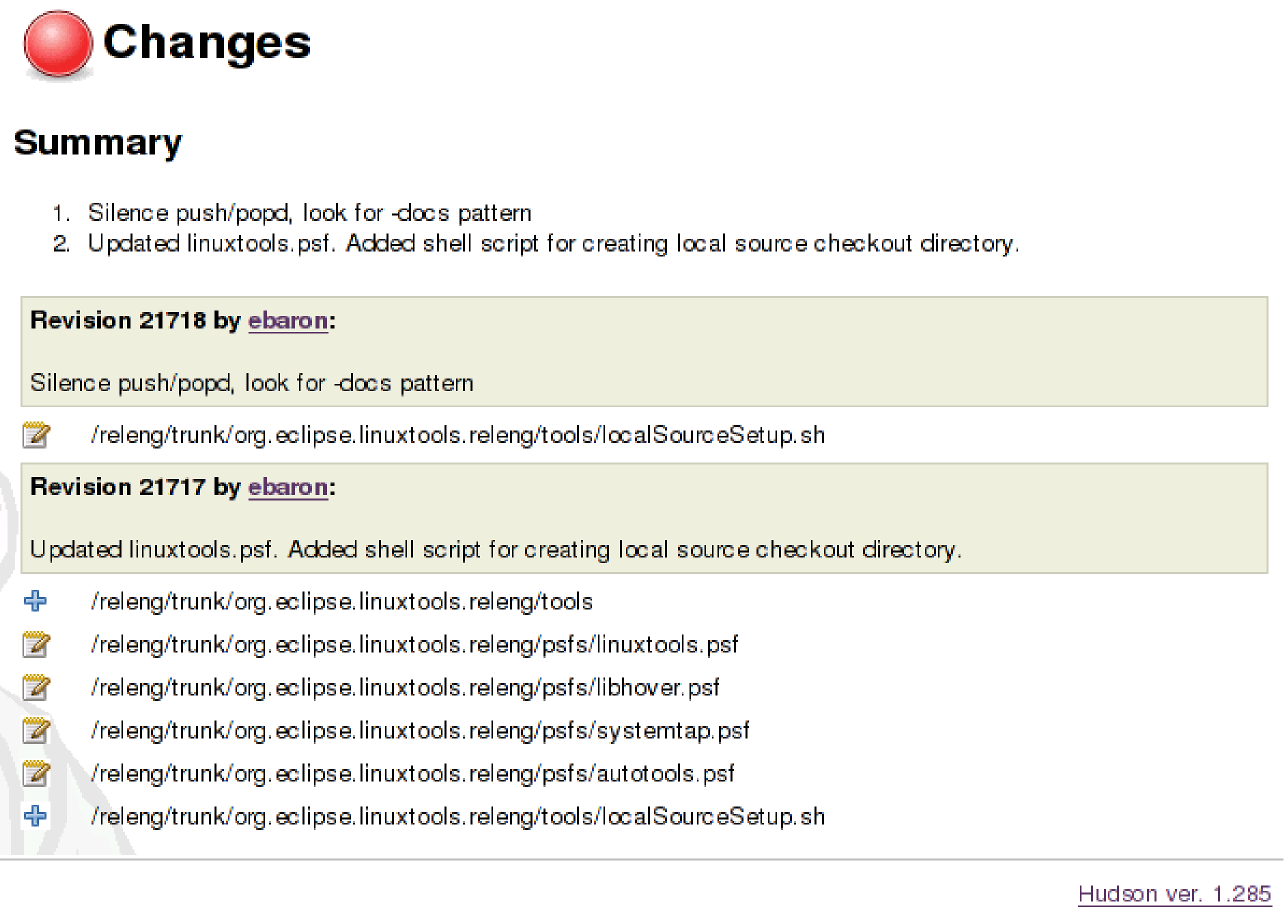
[268824] gef-head ahunter 090316 This site has moved. Please update your features.

✎	1.22	org.eclipse.gef/features/org.eclipse.gef.sdk-feature/feature.xml
✎	1.24	org.eclipse.gef/features/org.eclipse.gef.test-feature/feature.xml
✎	1.5	org.eclipse.gef/features/org.eclipse.draw2d-feature/feature.xml
✎	1.4	org.eclipse.gef/features/org.eclipse.draw2d-feature/sourceTemplateF
✎	1.9	org.eclipse.gef/features/org.eclipse.zest-feature/feature.xml
✎	1.4	org.eclipse.gef/features/org.eclipse.zest-feature/sourceTemplateFea
✎	1.31	org.eclipse.gef/features/org.eclipse.gef-feature/feature.xml
✎	1.8	org.eclipse.gef/features/org.eclipse.gef-feature/sourceTemplateFeat
✎	1.22	org.eclipse.gef/features/org.eclipse.gef.examples-feature/feature.xm
✎	1.5	org.eclipse.gef/features/org.eclipse.gef.all-feature/feature.xml
✎	1.8	org.eclipse.gef/features/org.eclipse.zest.sdk-feature/feature.xml
✎	1.4	org.eclipse.gef/features/org.eclipse.draw2d.sdk-feature/feature.xml

nickb:

v20090316-1158

✎	1.73	org.eclipse.gef/org.eclipse.gef.releng/maps/gef.map



Changes

Summary

1. Silence push/popd, look for -docs pattern
2. Updated linuxtools.psf. Added shell script for creating local source checkout directory.

Revision 21718 by ebaron:

Silence push/popd, look for -docs pattern

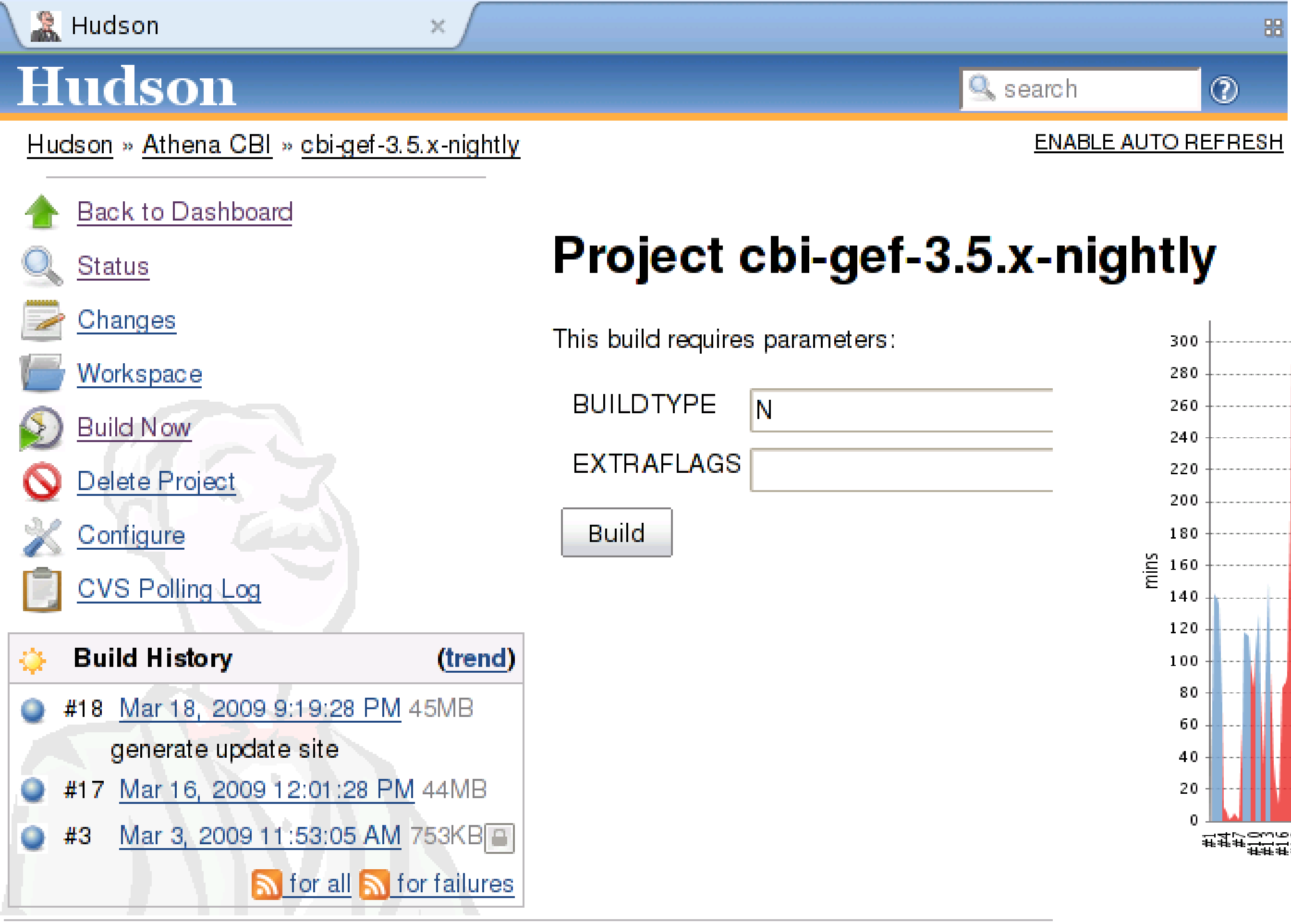
✎	/releng/trunk/org.eclipse.linuxtools.releng/tools/localSourceSetup.sh

Revision 21717 by ebaron:

Updated linuxtools.psf. Added shell script for creating local source checkout directory.

+	/releng/trunk/org.eclipse.linuxtools.releng/tools
✎	/releng/trunk/org.eclipse.linuxtools.releng/psfs/linuxtools.psf
✎	/releng/trunk/org.eclipse.linuxtools.releng/psfs/libhover.psf
✎	/releng/trunk/org.eclipse.linuxtools.releng/psfs/systemtap.psf
✎	/releng/trunk/org.eclipse.linuxtools.releng/psfs/autotools.psf
+	/releng/trunk/org.eclipse.linuxtools.releng/tools/localSourceSetup.sh

Hudson ver. 1.285

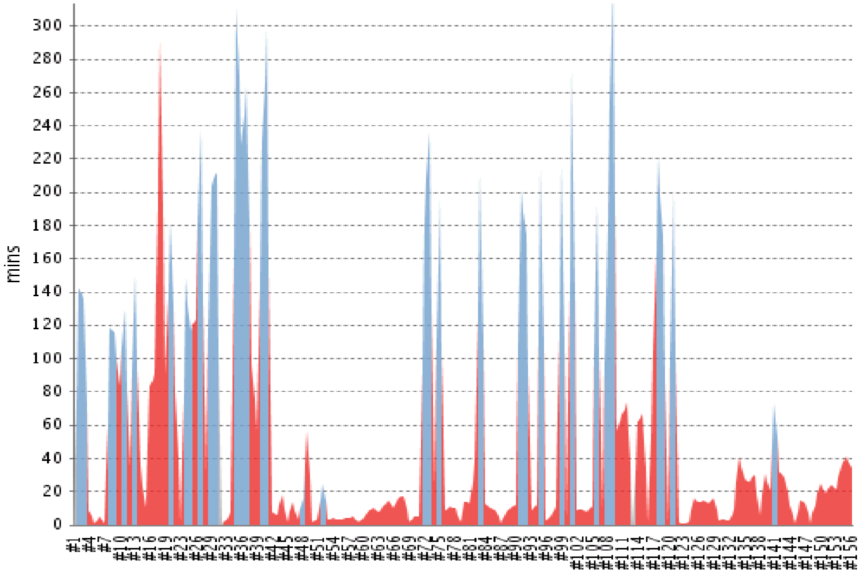


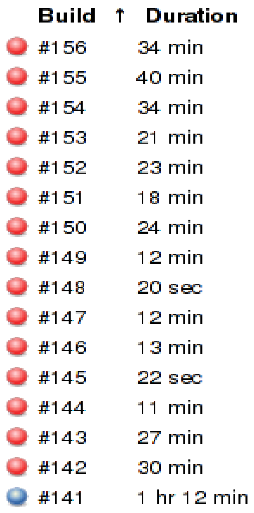
Project cbi-gef-3.5.x-nightly

This build requires parameters:

BUILDTYPE	N
EXTRAFLAGS	

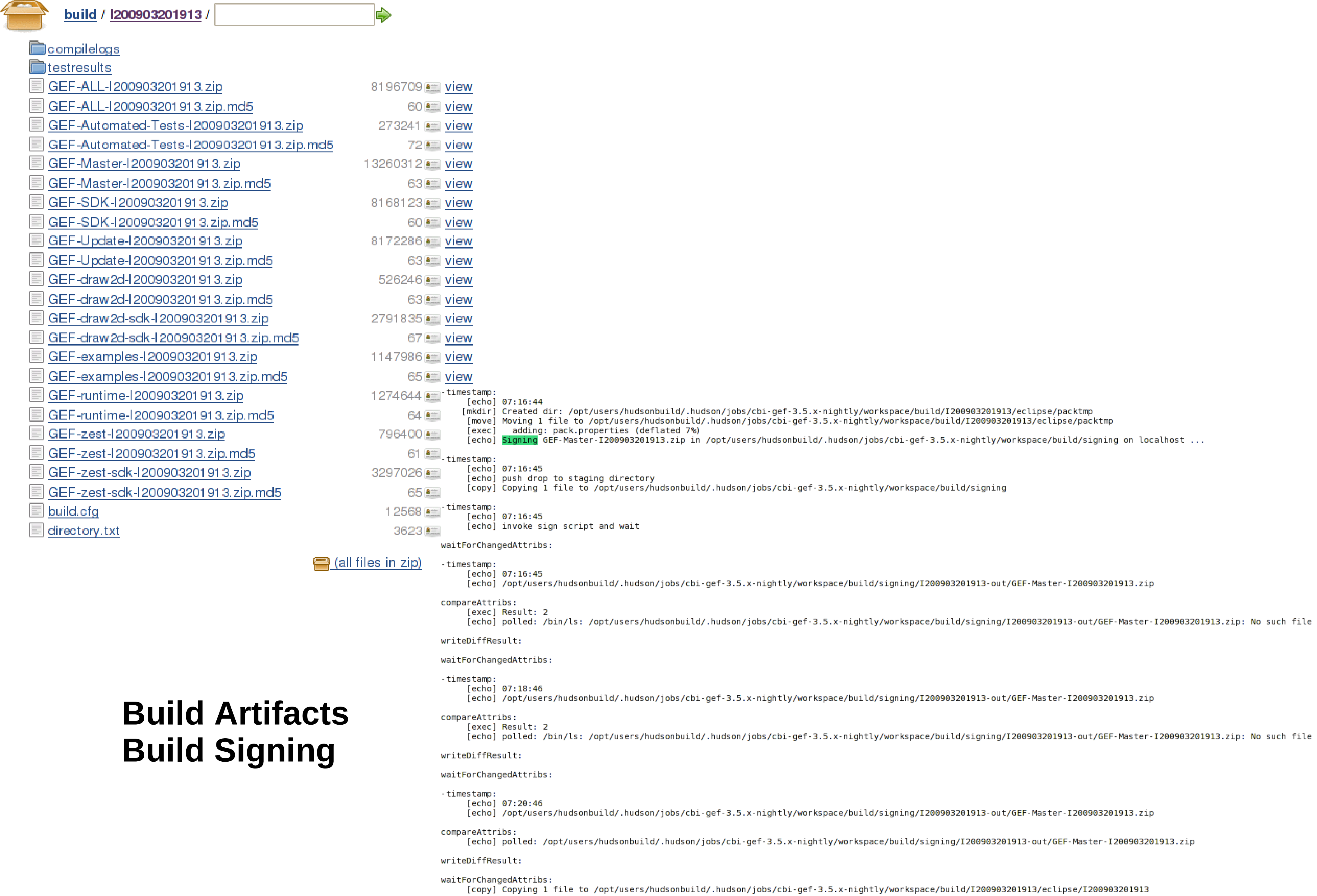
#18	Mar 18, 2009 9:19:28 PM	45MB	
	generate update site		
#17	Mar 16, 2009 12:01:28 PM	44MB	
#3	Mar 3, 2009 11:53:05 AM	753KB	





	Build ↑	Duration
●	#156	34 min
●	#155	40 min
●	#154	34 min
●	#153	21 min
●	#152	23 min
●	#151	18 min
●	#150	24 min
●	#149	12 min
●	#148	20 sec
●	#147	12 min
●	#146	13 min
●	#145	22 sec
●	#144	11 min
●	#143	27 min
●	#142	30 min
●	#141	1 hr 12 min

Parameterized Builds
Build Duration Trending



build / I200903201913 /

compilelogs			
testresults			
GEF-ALL-I200903201913.zip	8196709		view
GEF-ALL-I200903201913.zip.md5	60		view
GEF-Automated-Tests-I200903201913.zip	273241		view
GEF-Automated-Tests-I200903201913.zip.md5	72		view
GEF-Master-I200903201913.zip	13260312		view
GEF-Master-I200903201913.zip.md5	63		view
GEF-SDK-I200903201913.zip	8168123		view
GEF-SDK-I200903201913.zip.md5	60		view
GEF-Update-I200903201913.zip	8172286		view
GEF-Update-I200903201913.zip.md5	63		view
GEF-draw2d-I200903201913.zip	526246		view
GEF-draw2d-I200903201913.zip.md5	63		view
GEF-draw2d-sdk-I200903201913.zip	2791835		view
GEF-draw2d-sdk-I200903201913.zip.md5	67		view
GEF-examples-I200903201913.zip	1147986		view
GEF-examples-I200903201913.zip.md5	65		view
GEF-runtime-I200903201913.zip	1274644		
GEF-runtime-I200903201913.zip.md5	64		
GEF-zest-I200903201913.zip	796400		
GEF-zest-I200903201913.zip.md5	61		
GEF-zest-sdk-I200903201913.zip	3297026		
GEF-zest-sdk-I200903201913.zip.md5	65		
build.cfg	12568		
directory.txt	3623		

(all files in zip)

```
-timestamp:
     [echo] 07:16:44
    [mkdir] Created dir: /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/I200903201913/eclipse/packtmp
     [move] Moving 1 file to /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/I200903201913/eclipse/packtmp
     [exec]   adding: pack.properties (deflated 7%)
     [echo] Signing GEF-Master-I200903201913.zip in /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/signing on localhost ...

-timestamp:
     [echo] 07:16:45
     [echo] push drop to staging directory
     [copy] Copying 1 file to /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/signing

-timestamp:
     [echo] 07:16:45
     [echo] invoke sign script and wait

waitForChangedAttribs:

-timestamp:
     [echo] 07:16:45
     [echo] /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/signing/I200903201913-out/GEF-Master-I200903201913.zip

compareAttribs:
     [exec] Result: 2
     [echo] polled: /bin/ls: /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/signing/I200903201913-out/GEF-Master-I200903201913.zip: No such file

writeDiffResult:

waitForChangedAttribs:

-timestamp:
     [echo] 07:18:46
     [echo] /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/signing/I200903201913-out/GEF-Master-I200903201913.zip

compareAttribs:
     [exec] Result: 2
     [echo] polled: /bin/ls: /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/signing/I200903201913-out/GEF-Master-I200903201913.zip: No such file

writeDiffResult:

waitForChangedAttribs:

-timestamp:
     [echo] 07:20:46
     [echo] /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/signing/I200903201913-out/GEF-Master-I200903201913.zip

compareAttribs:
     [echo] polled: /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/signing/I200903201913-out/GEF-Master-I200903201913.zip

writeDiffResult:

waitForChangedAttribs:
     [copy] Copying 1 file to /opt/users/hudsonbuild/.hudson/jobs/cbi-gef-3.5.x-nightly/workspace/build/I200903201913/eclipse/I200903201913
```

Build Artifacts
Build Signing

Running Hudson

- Download the latest hudson.war or .jar from https://hudson.dev.java.net/
- `java -jar hudson.war`
- (That's it!)

Agenda

- Introduction
- PDE Build
  - Exercise: Build a Feature
  - Exercise: Generate a p2 Repository
- Common Build Background
- Dash Athena
  - Exercise: Setup
  - Exercise: Configure, Run, Troubleshoot
  - Exercise: Verify Build
- Meet Hudson
- **BYO Build Clinic**

BYO Build Clinic / Q&A

- Bring us your checked out sources, we'll try to get you running w/ an Athena build
- Any questions?
- Want more? For extra credit, see the next two slides.

Exercise: Linux Tools (Optional)

- Try building the Linux Tools project!
- Get org.eclipse.linuxtools.releng project
  - See psf/linuxtools.psf file in tutorial zip
- Source tree can be fetched from
  - http://dev.eclipse.org/svnroot/technology/org.eclipse.linuxtools/
- Configure releng/build.properties to include required dependencies: JDK 6.0, Eclipse, EMF, GEF, CDT & BIRT.
- Run build.xml!

Exercise: Build in Hudson (Optional)

- Try building with Hudson, either locally or on a virtual server. See

  http://wiki.eclipse.org/Common_Build_Infrastructure/Virtual_Server_Setup
- Start Hudson. Open http:/localhost:8080
- Create a new job
- Launch a GEF or Linux Tools build using
  - org.eclipse.dash.common.releng/tools/scripts/start.sh
  - (Run script without options for examples and summary of commandline flags)