

# **Control System Studio Training**

-

## **Extension Points: Logbook Example**

**Kay Kasemir**

**ORNL/SNS**

**[kasemirk@ornl.gov](mailto:kasemirk@ornl.gov)**

**Jan. 2013**

# Extension points

## Eclipse:

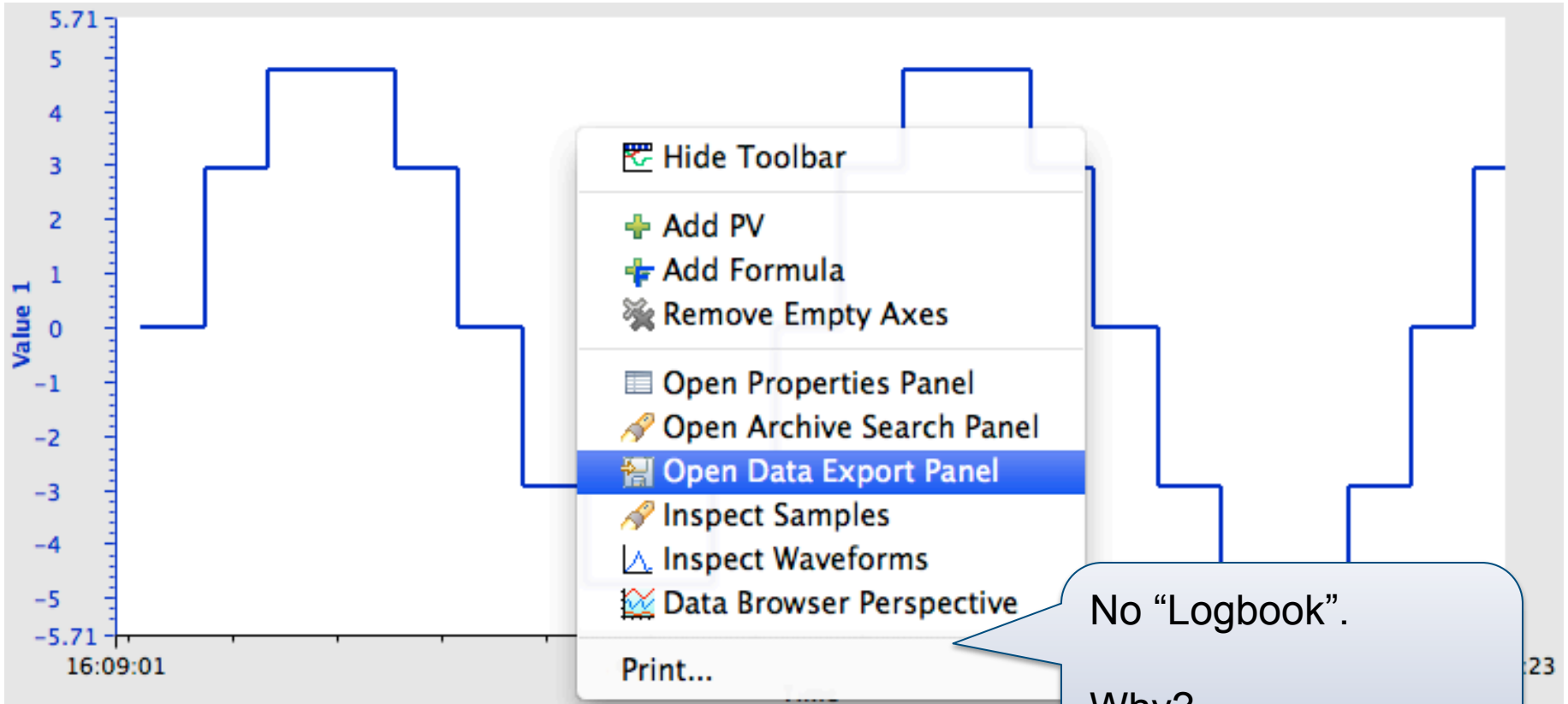
- menus, online help, preferences, views, editors, toolbar buttons, ...

## CSS:

- live data, archived data, logbook, ...

## Dummy Logbook: Example for basic mechanism

# Example CSS Product



**In principle, BOY displays and Data Browser support the Logbook ...**

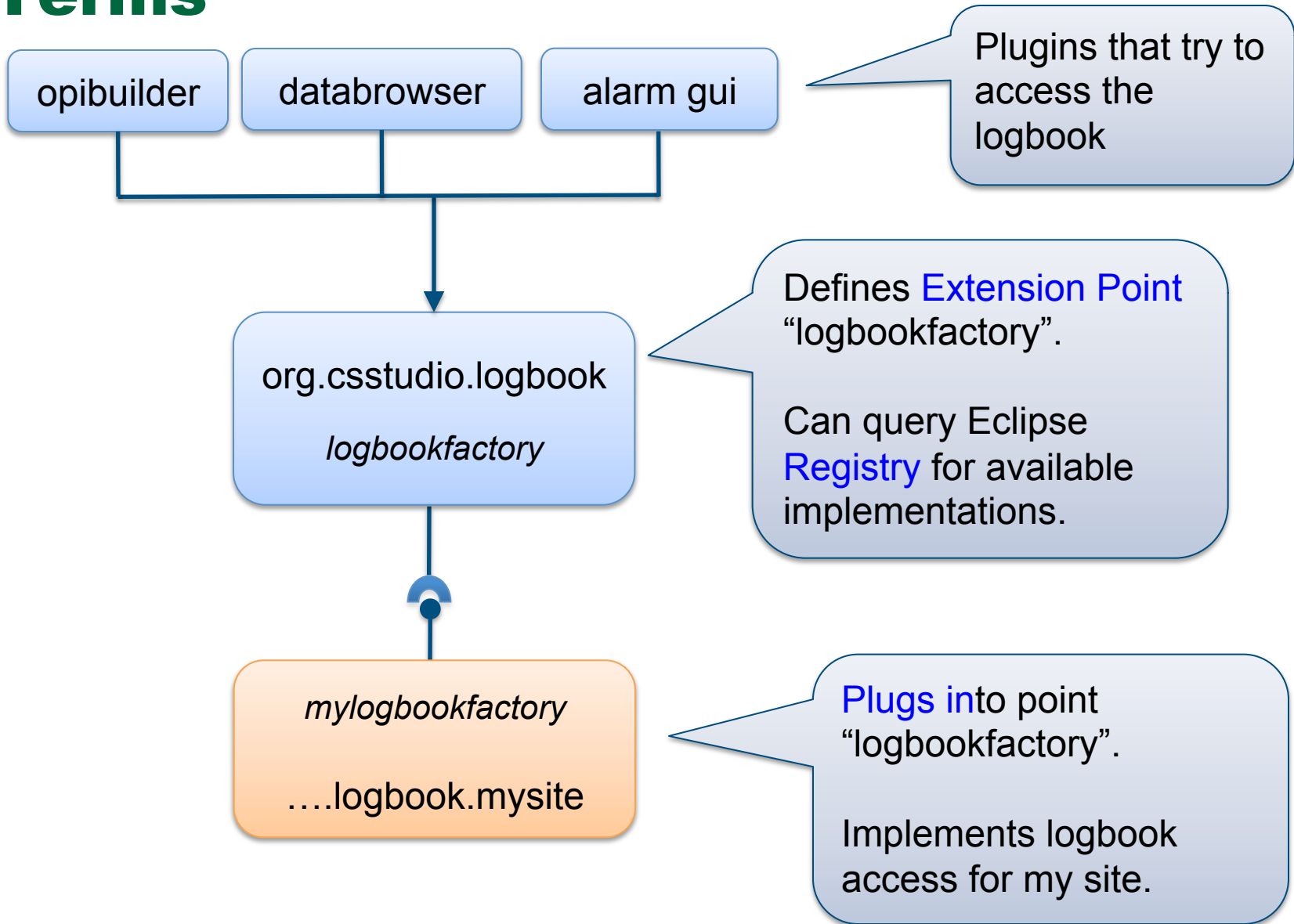
# The only constant is Change...

**Examples refer to a “logbookfactory”, an API for writing simple log entries.**

**BNL/FRIB right now triggering update to a more elaborate “logclientfactory” API for reading and writing more elaborate log entries.**

**The following will soon be outdated, it’s meant as a simple example.**

# Terms



# No Logbook Implementation!

The screenshot shows the Eclipse Run Configurations dialog for a configuration named 'CSS.product'. The 'Plug-ins' tab is selected, showing a list of plug-ins under the 'Workspace' folder. An orange box highlights the following plug-ins:

Plug-in	Start Level	Auto-Start
<input checked="" type="checkbox"/> org.csstudio.logbook (3.1.0.qualifier)	0	default
<input type="checkbox"/> org.csstudio.logbook.nsrrc (1.0.0.qualifier)		
<input type="checkbox"/> org.csstudio.logbook.sns (3.1.1.qualifier)		
<input checked="" type="checkbox"/> org.csstudio.logbook.ui (3.0.0.qualifier)	default	default

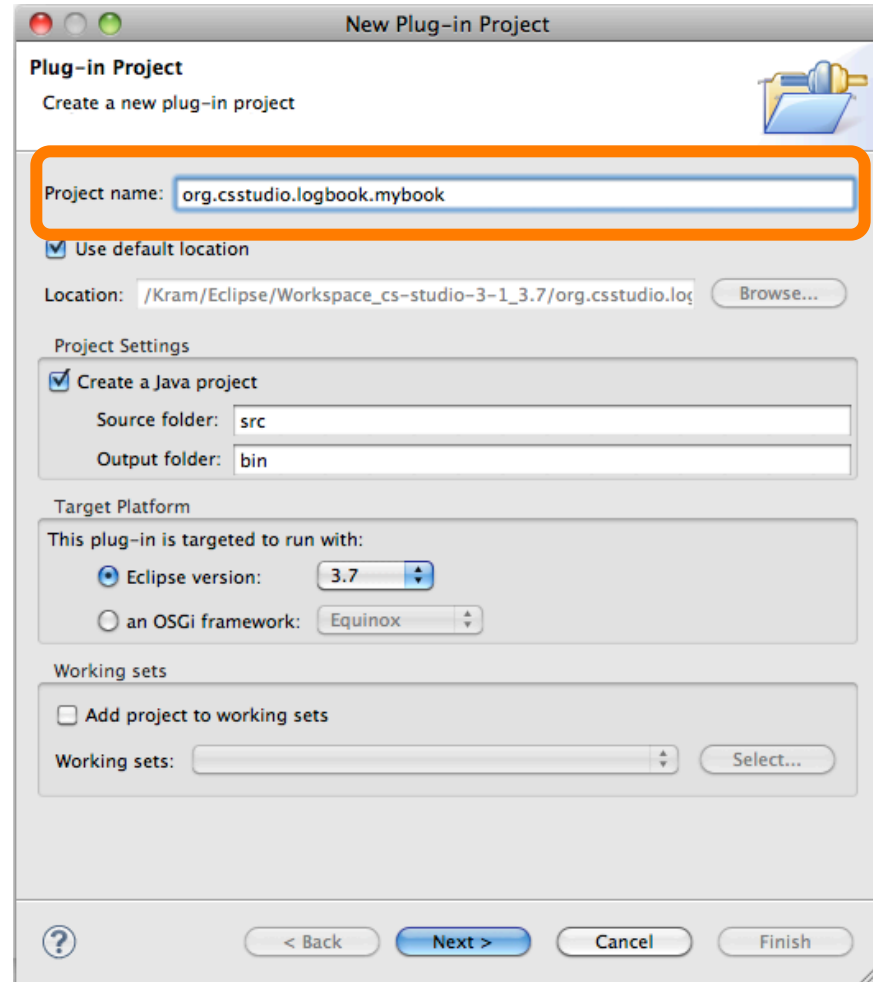
Other details in the dialog include: 'Launch with: plug-ins selected below only', 'Default Start level: 4', 'Default Auto-Start: false', and a 'Validate Plug-ins' button. The left sidebar shows a tree view of configurations, with 'CSS.product' selected. The bottom of the dialog has 'Apply', 'Revert', 'Close', and 'Run' buttons.

- There's one for ..sns, but not YourSite

# Create org.csstudio.logbook.mybook

## File, New, Plug-In Project

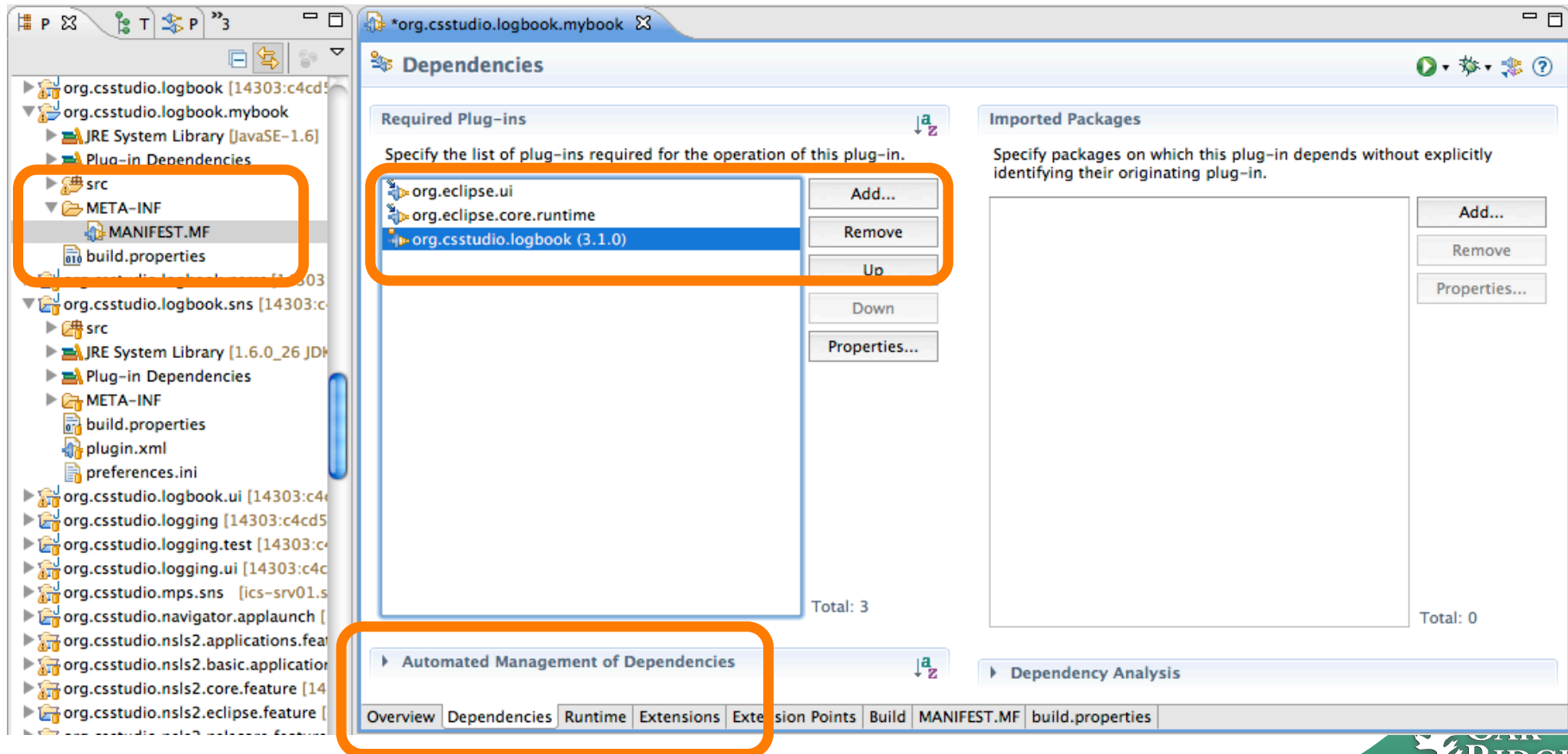
- Enter name
- Next, ... Finish



# Configure org.csstudio.logbook.mybook

## MANIFEST.MF editor should open

- On the “Overview”, check “This plug-in is a singleton”
- Select “Dependencies”
- Add org.csstudio.logbook





# About “Logbookfactory” Extension Point

We could

- Open org.csstudio.logbook/ plugin.xml
- In the “Extension Points” tab, locate the “logbookfactory” and read its description
- Study the ILogbookFactory interface
- Study one of the supplied implementations

Or:  
Just Do It

- And let Eclipse help on the way

The screenshot shows two Eclipse IDE windows. The left window, titled 'org.csstudio.logbook', displays the 'Extension Points' tab. It shows a list of extension points with 'logbookfactory' selected. The 'Extension Point Details' panel for 'logbookfactory' shows its ID, Name, and Schema. The right window, titled 'org.csstudio.logbook.logbookfactory', displays the 'LogbookFactory' configuration markup, including its identifier, since version, description, configuration markup, examples, and API information.

**Extension Points**

All Extension Points  
Edit extension points defined by this plug-in in the following section.

logbookfactory [Add... Delete]

Extension Point Details  
Set the properties of the selected extension point.

ID: logbookfactory  
Name: LogbookFactory  
Schema: schema/logbookfactory.exsd [Browse...]

Show extension point description  
Open extension point schema  
Find references

**LogbookFactory**

**Identifier:** org.csstudio.logbook.logbookfactory  
**Since:** 1.0  
**Description:** Allow other plugins to provide a logbook interface

**Configuration Markup:**

```
<!ELEMENT extension (logbookfactory)>
<!ATTLIST extension
  point CDATA #REQUIRED
  id CDATA #IMPLIED
  name CDATA #IMPLIED
>

<!ELEMENT logbookfactory EMPTY>
<!ATTLIST logbookfactory
  name CDATA #REQUIRED
  class CDATA #REQUIRED
>
```

- **name** - Name of your logbook factory.
- **class** - Java class that implements your logbook factory

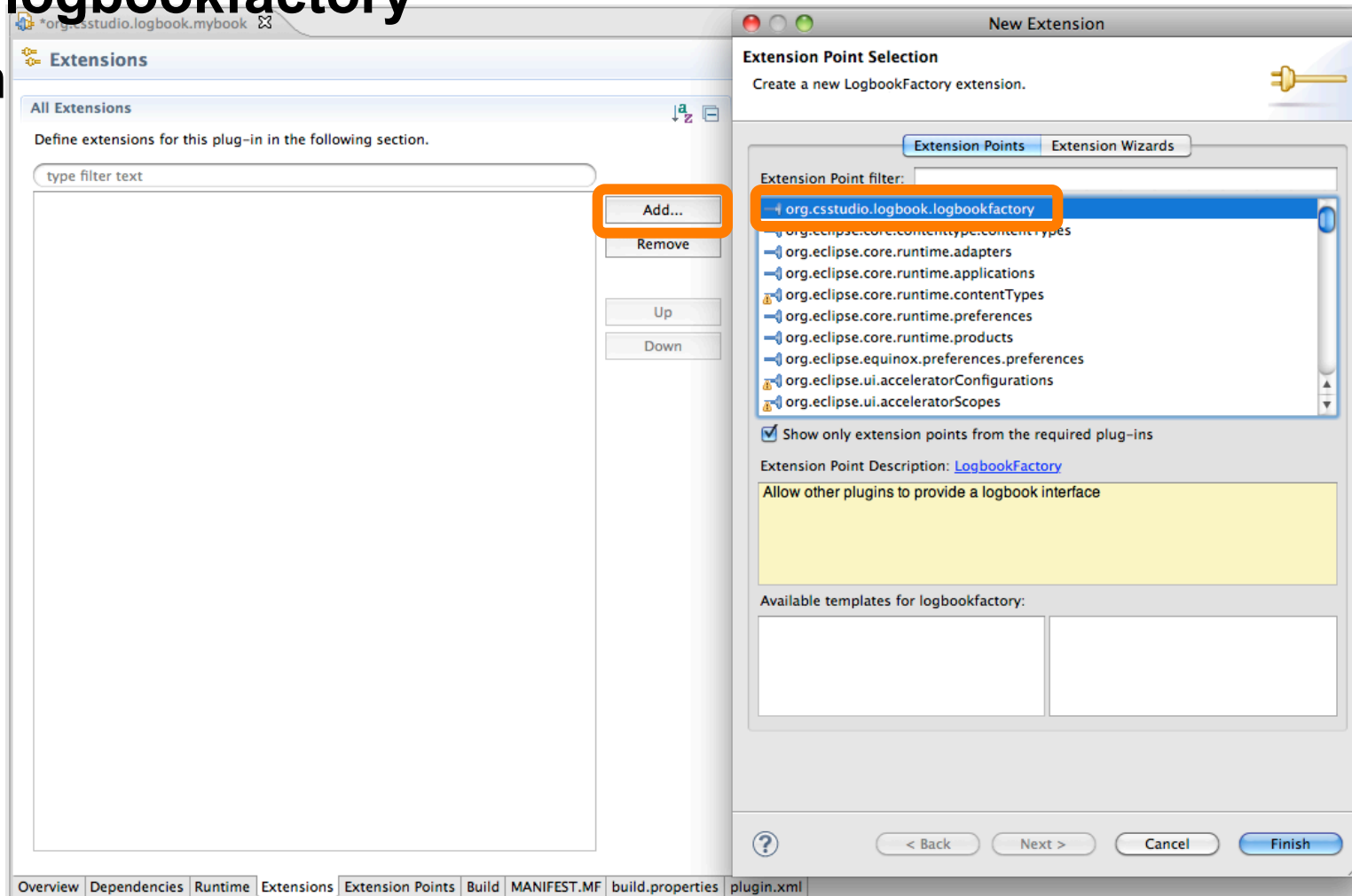
**Examples:**

```
<extension
  point="org.csstudio.logbook.logbookfactory">
  <logbookfactory
    class="org.csstudio.logbook.mine.MyLogbookFactory"
    name="org.csstudio.logbook.mine.mylogbookfactory">
  </logbookfactory>
</extension>
```

**API Information:** Implementors have to provide an org.csstudio.logbook.ILogbookFactory  
**Supplied Implementation:** org.csstudio.logbook.sns is one example

# Make `..mybook` extend `logbookfactory`

- In editor for `..mylogbook/MANIFEST.MF`
  - Select “Extensions”
  - Pick `..logbookfactory`
  - Finish



# Configure your logbookfactory

If we had read the extension point description, we would know that we need an

- ID
- Name of our Java class

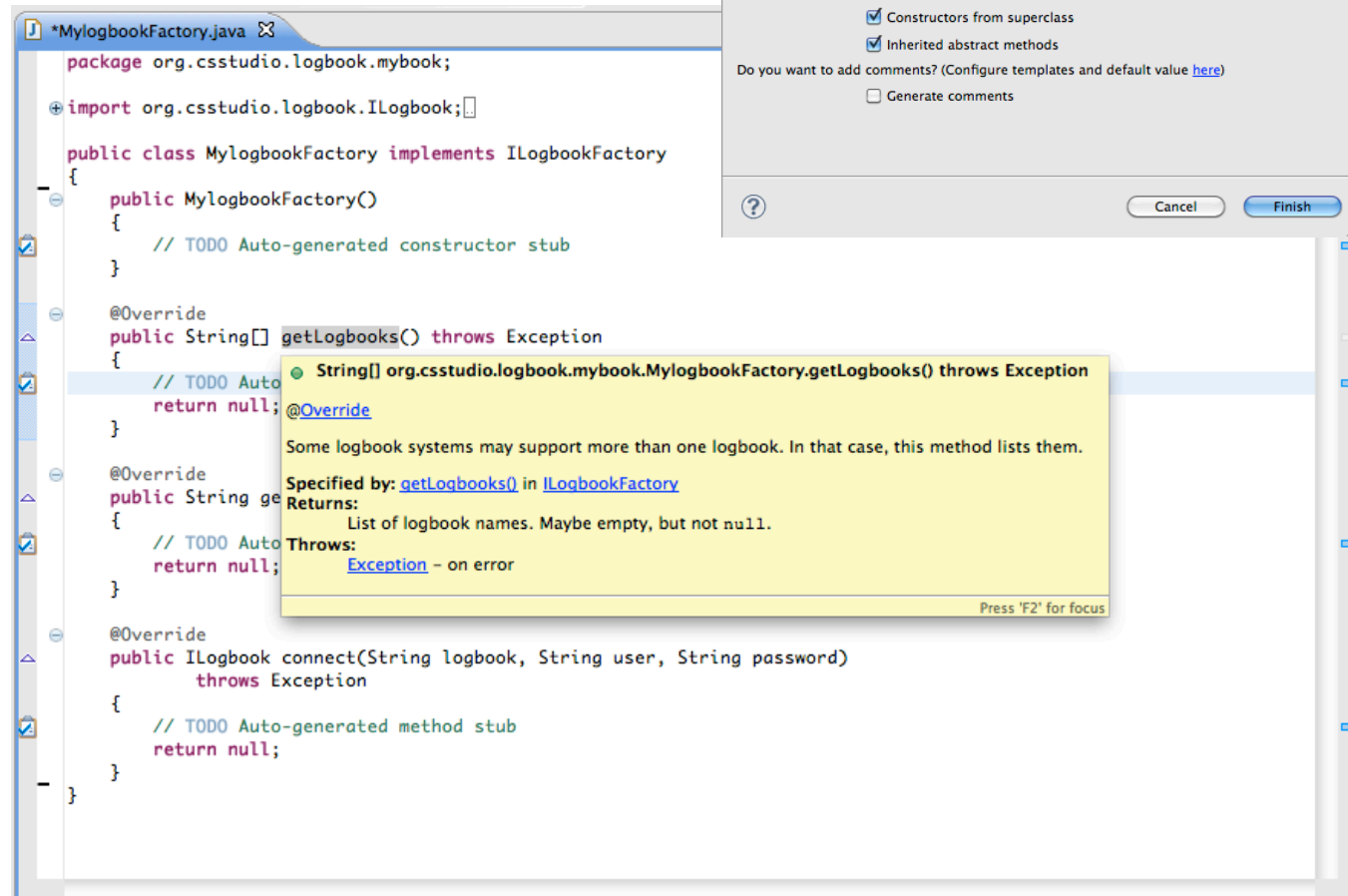
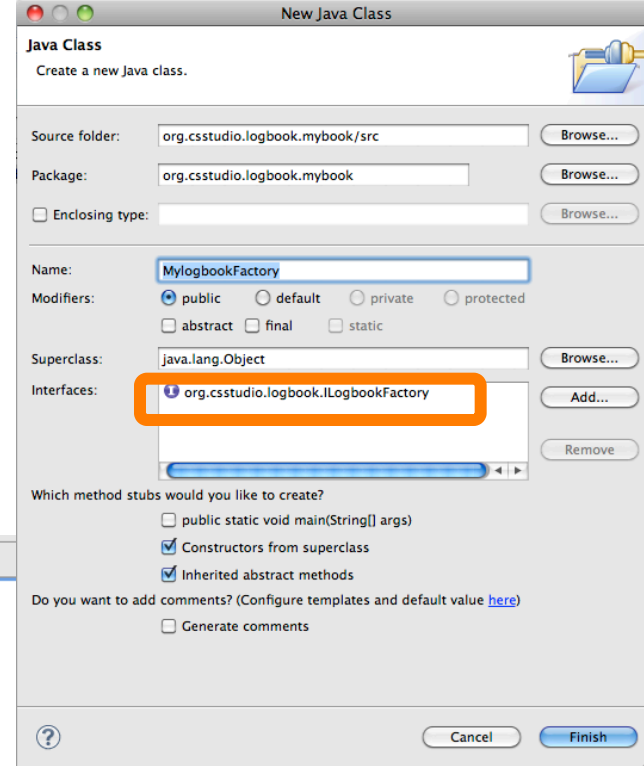
The extensions editor helps us to provide these.

Finally, click on the “**class\*:**” link

The screenshot shows the Eclipse IDE's Extensions editor for the plug-in `org.csstudio.logbook.mybook`. The "All Extensions" list contains one entry: `org.csstudio.logbook.mybook.mylogbookfactory (logbookfactory)`. The "Extension Element Details" panel is open, showing the configuration for the "logbookfactory" extension point. The "name\*" field is set to `org.csstudio.logbook.mybook.mylogbookfactory` and the "class\*" field is set to `org.csstudio.logbook.mybook.MylogbookFactory`. A tooltip for the "class\*" field indicates it is a "Java class that implements your logbook factory".

# Implement LogbookFactory

- Eclipse
  - suggests required Java Interface
  - Provides skeleton
  - JavaDoc



# Finally: Do need to write code

```
MylogbookFactory.java
package org.csstudio.logbook.mybook;

import org.csstudio.logbook.ILogbook;

public class MylogbookFactory implements ILogbookFactory
{
    @Override
    public String[] getLogbooks() throws Exception
    {
        return new String[]
        {
            "Operation Logbook",
            "Control System Logbook"
        };
    }

    @Override
    public String getDefaultLogbook()
    {
        return "Operation Logbook";
    }

    @Override
    public ILogbook connect(String logbook, String user, String password)
        throws Exception
    {
        // Ignore user, password
        return new MyLogbook(logbook);
    }
}
```

```
MyLogbook.java
package org.csstudio.logbook.mybook;

import org.csstudio.logbook.ILogbook;

public class MyLogbook implements ILogbook
{
    final private String logbook;

    public MyLogbook(String logbook)
    {
        this.logbook = logbook;
    }

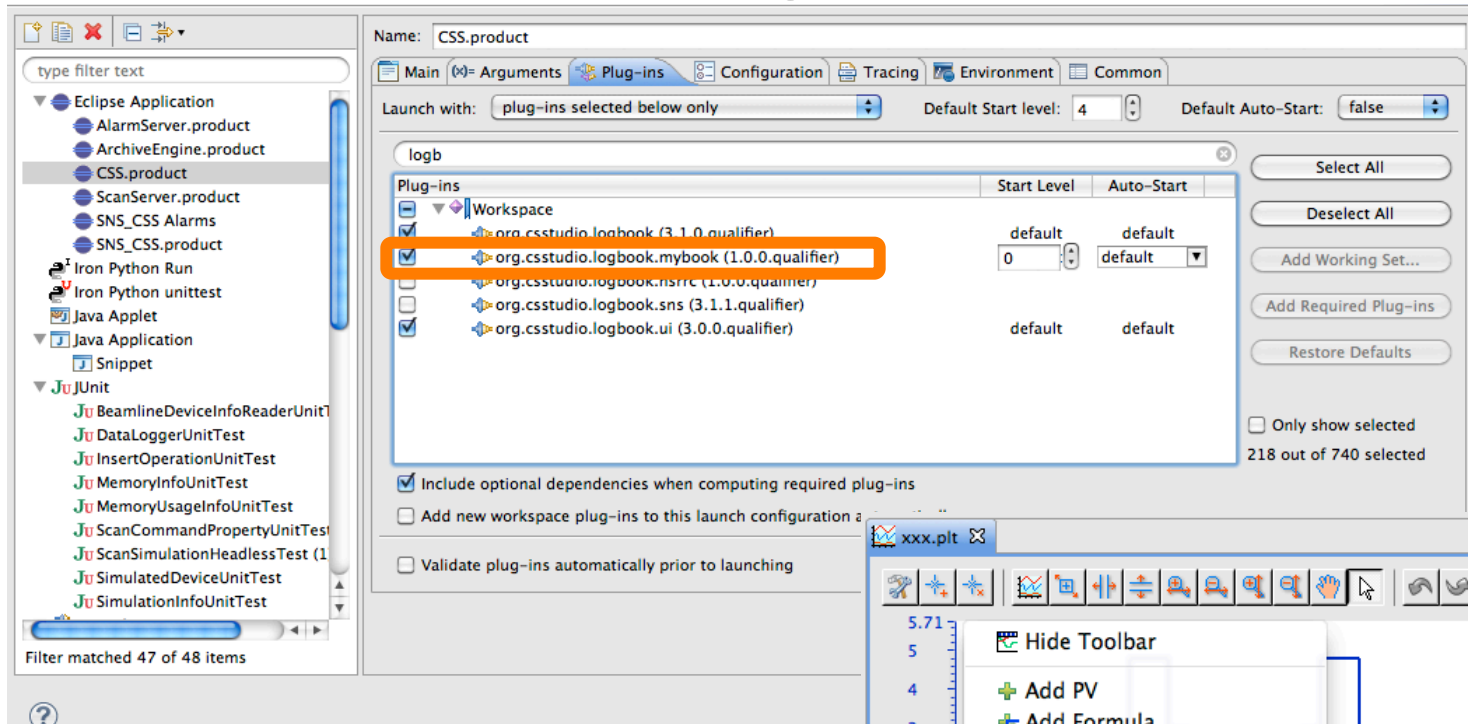
    @Override
    public void createEntry(String title, String text, String... file_names)
        throws Exception
    {
        // Simply print, no actual logbook entry created
        System.out.println("Logbook: " + logbook);
        System.out.println("Title: " + title);
        System.out.println("Text: " + text);
    }

    @Override
    public void createEntry(String title, String text, String[] filenames,
        String[] captions) throws Exception
    {
        // Ignore captions
        createEntry(title, text, filenames);
    }

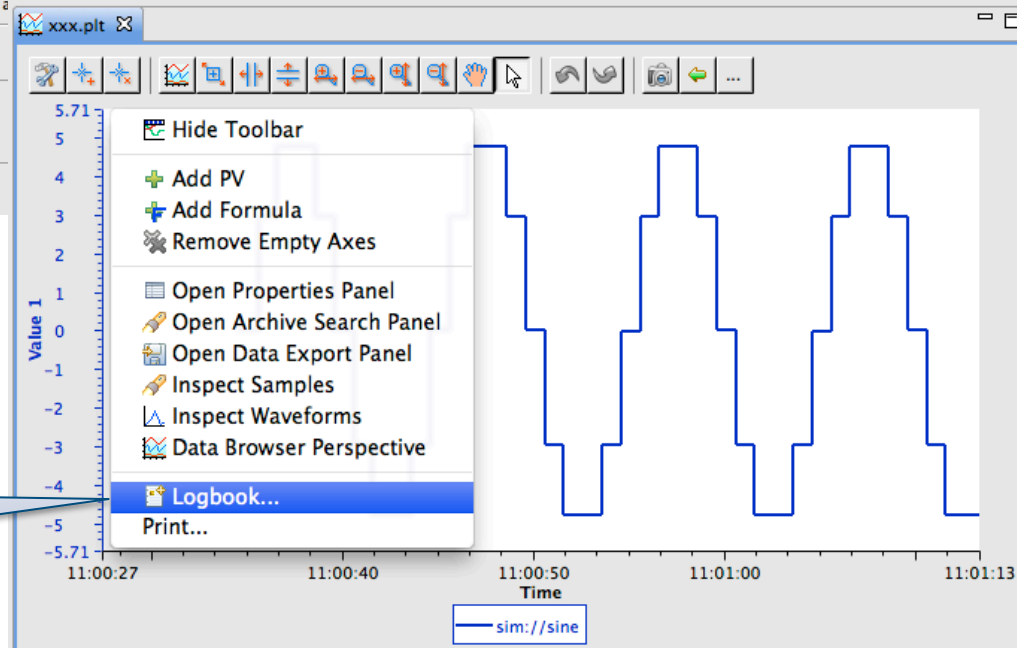
    @Override
    public void close()
    {
        // Nothing to close
    }
}
```

# Try it

- Include in run configuration

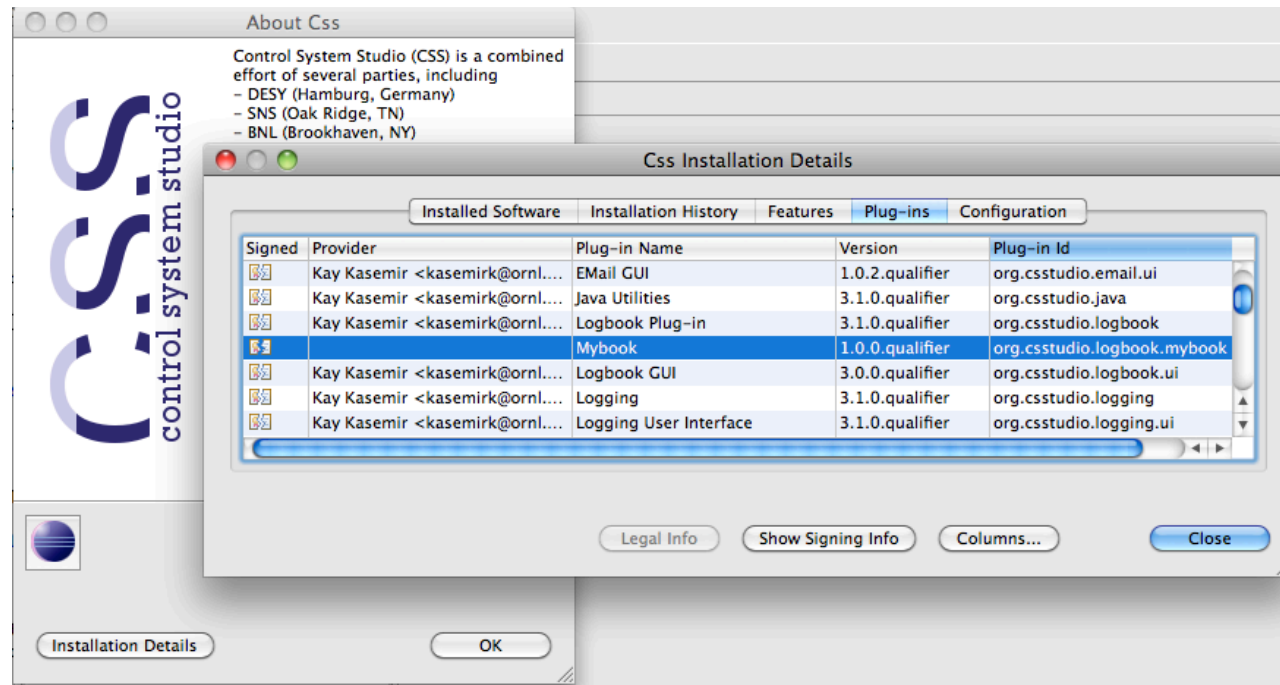


There it is!



# Troubleshooting

- Was the plugin included in the run configuration?
- In product, check Help, About, Installation Details, Plug-Ins
  - Is it really loaded?



# Extension Points

- **There are many!**
  - **Read Eclipse RCP Books**
  - **See Eclipse IDE online help**
  - **<http://cs-studio.sourceforge.net/docbook/>**
  - **Study CSS Sources**
- **Code is nicely decoupled and “packaged”**