

# Control System Studio Training

-

## Hierarchical Preferences

**Kay Kasemir**

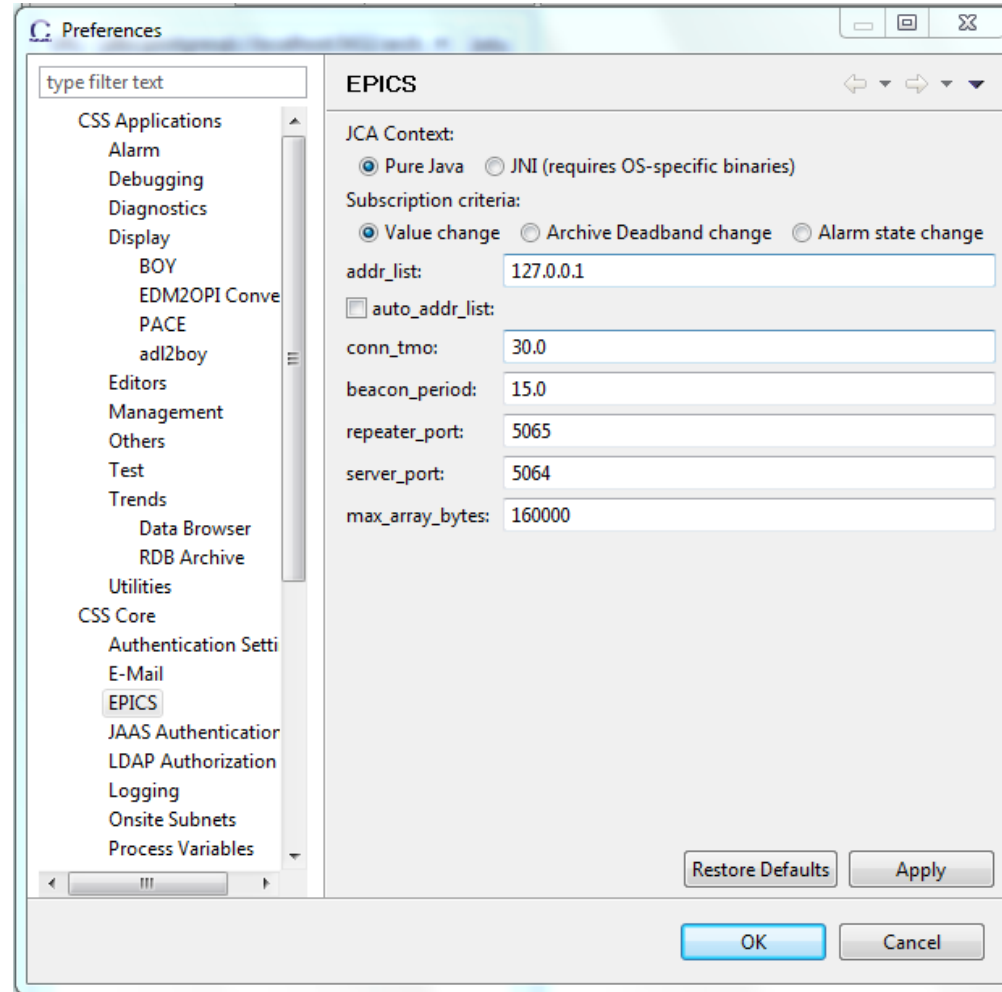
**ORNL/SNS**

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

**Jan. 2013**

# Preference GUI

- **Menu *Edit, Preferences***
  - **Not too hard to find**
  - **Maybe consult associated *Help, Help Content***



# Issue 1: What's best for each user?

**1. Download CSS**

**2. Run**

**1. Download CSS**

**2. Set EPICS address list**

**3. Configure email server**

**4. Disable BOY editing**

**5. Add archive data sources**

**6. ...**

**7. Run**

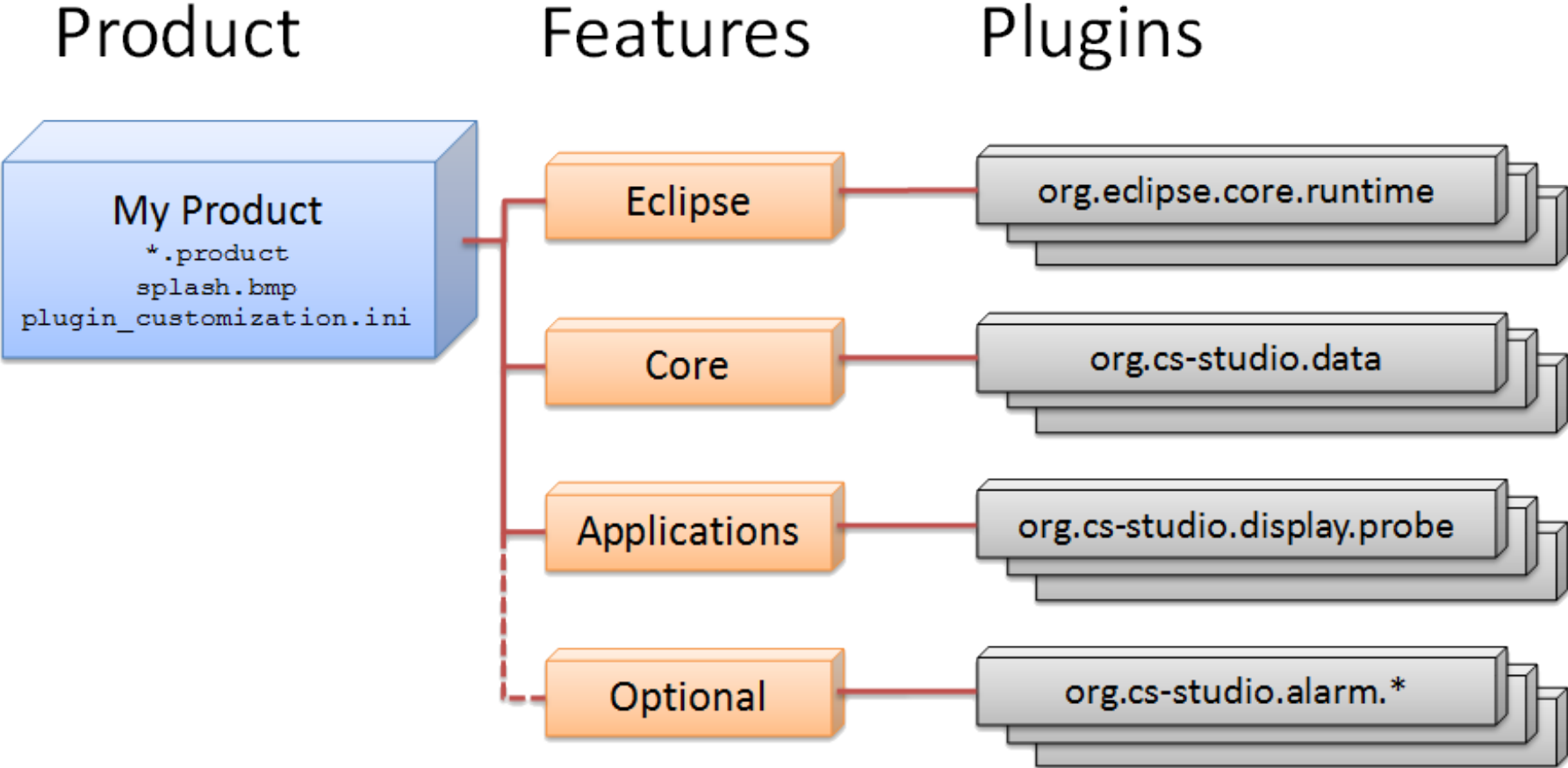
# Issue 2: What about GUI-less tools?

**Archive Engine, Alarm Server, Scan Server, ...  
don't have a GUI.**

***No Edit/Preferences...***

**So how do the CSS (Eclipse)  
preferences work?**

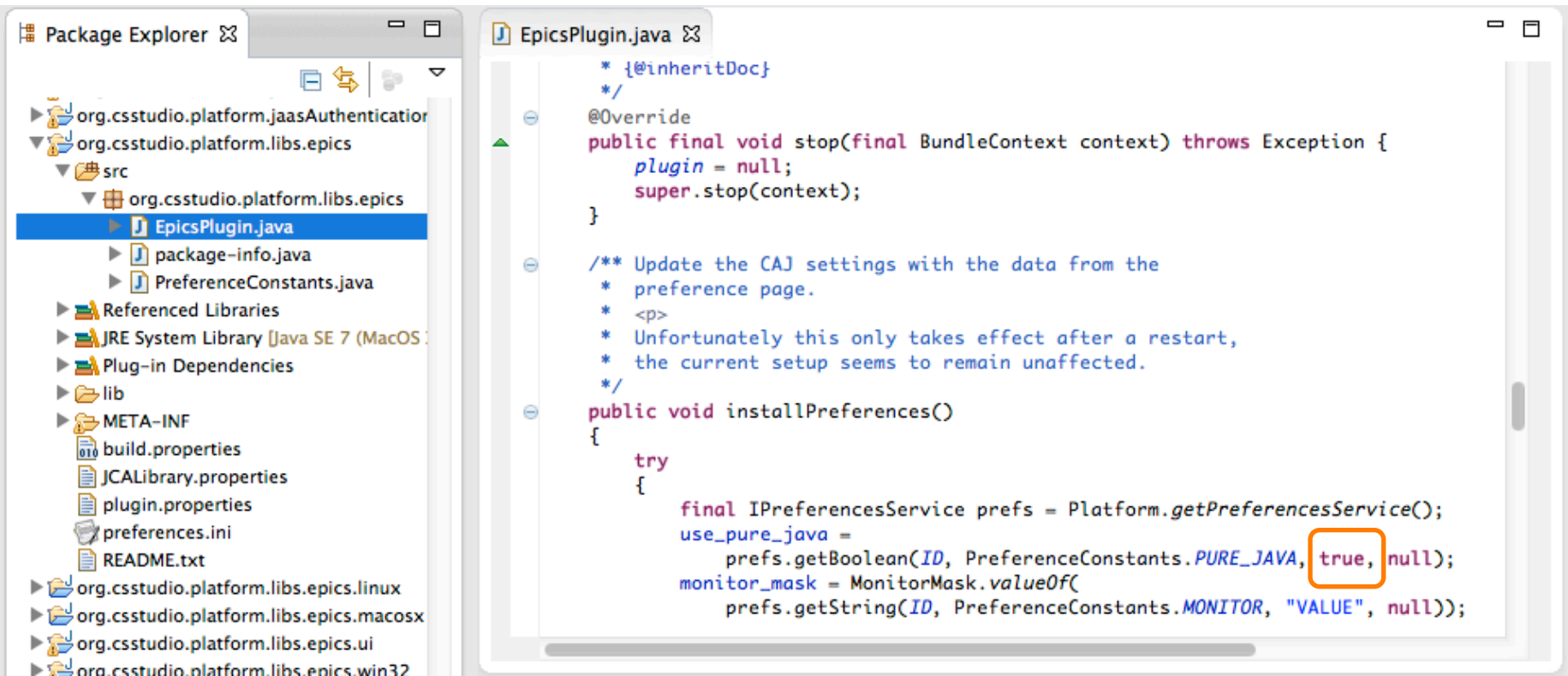
# Eclipse “Product”



# Hierarchical Preferences

1. Code inside plugin
2. Plugin's preferences.ini
3. Product's plugin\_customization.ini
4. Command-line

# 1. Code inside plugin



The screenshot shows an IDE window with two panes. The left pane is the Package Explorer, showing a project structure with the following folders and files:

- org.csstudio.platform.jaasAuthentication
- org.csstudio.platform.libs.epics
  - src
    - org.csstudio.platform.libs.epics
      - EpicsPlugin.java** (selected)
      - package-info.java
      - PreferenceConstants.java
    - Referenced Libraries
    - JRE System Library [Java SE 7 (MacOS ...)]
    - Plug-in Dependencies
    - lib
    - META-INF
      - build.properties
      - JCALibrary.properties
      - plugin.properties
      - preferences.ini
      - README.txt
  - org.csstudio.platform.libs.epics.linux
  - org.csstudio.platform.libs.epics.macosx
  - org.csstudio.platform.libs.epics.ui
  - org.csstudio.platform.libs.epics.win32

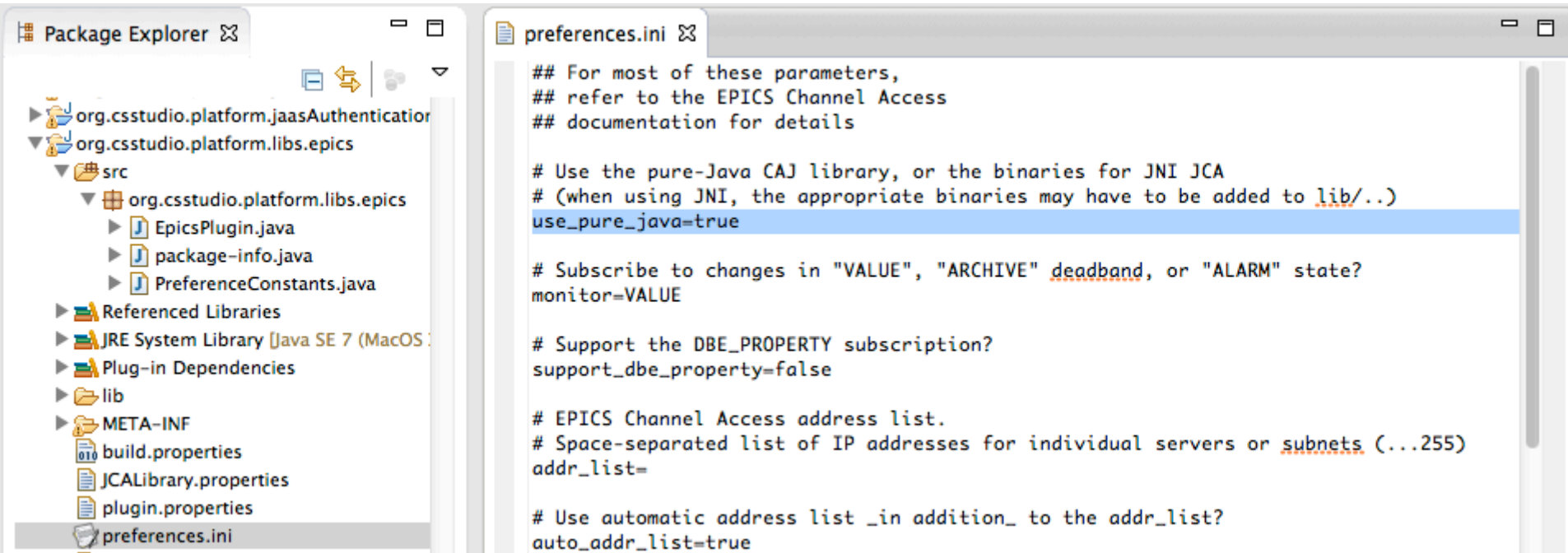
The right pane shows the source code of `EpicsPlugin.java`:

```
    * {@inheritDoc}
    */
    @Override
    public final void stop(final BundleContext context) throws Exception {
        plugin = null;
        super.stop(context);
    }

    /** Update the CAJ settings with the data from the
     * preference page.
     * <p>
     * Unfortunately this only takes effect after a restart,
     * the current setup seems to remain unaffected.
     */
    public void installPreferences()
    {
        try
        {
            final IPreferencesService prefs = Platform.getPreferencesService();
            use_pure_java =
                prefs.getBoolean(ID, PreferenceConstants.PURE_JAVA, true, null);
            monitor_mask = MonitorMask.valueOf(
                prefs.getString(ID, PreferenceConstants.MONITOR, "VALUE", null));
        }
    }
}
```

**Would have to edit source code, recompile, ...**

## 2. Plugin's preferences.ini

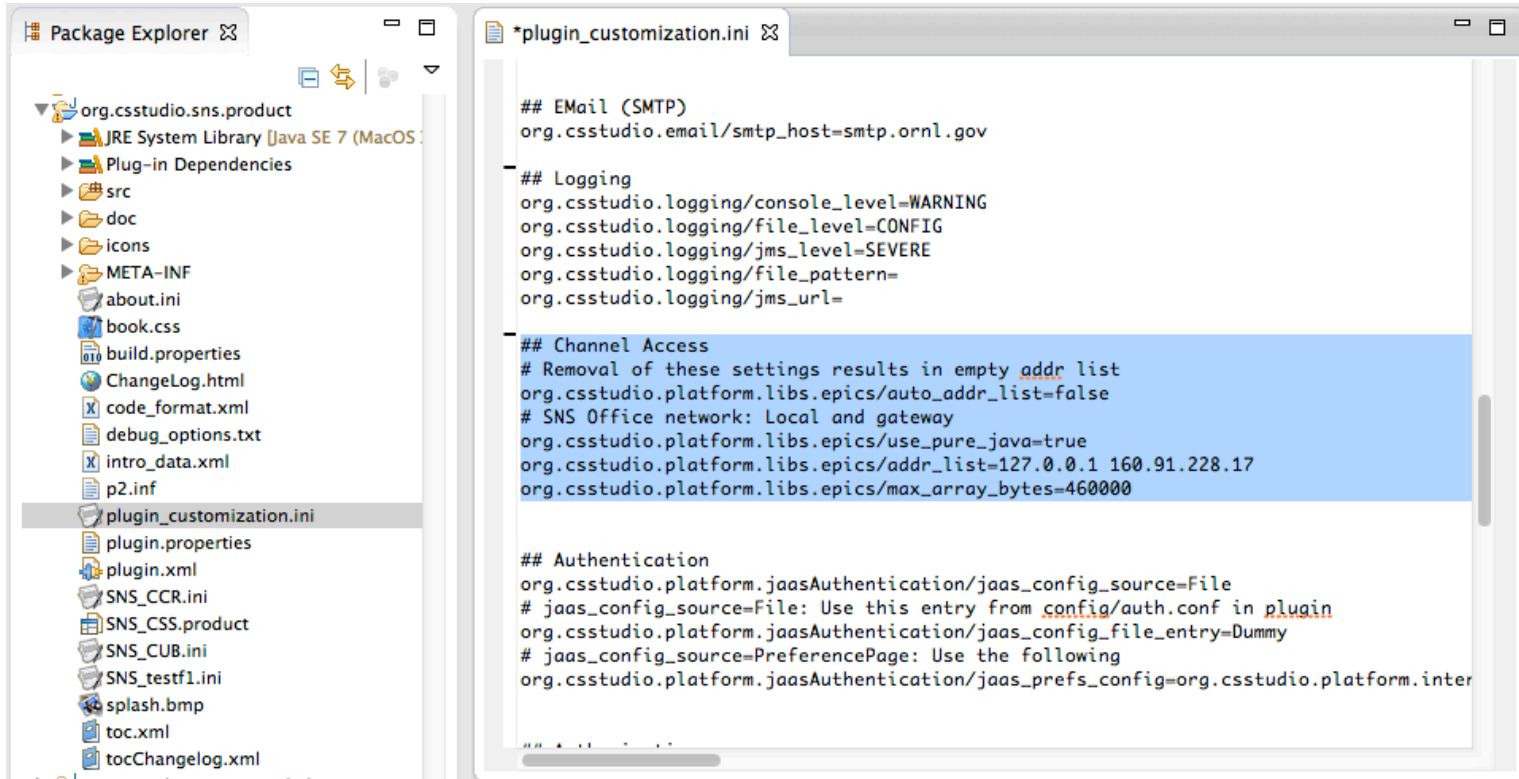


Documents the available preference options, lists defaults.

Can change w/o recompilation:  
Unzip the plugin .jar,  
Edit the preferences.ini.



# 3. Product's plugin\_customization.ini



Site-specific Product sets preferences for each plugin

name.of.plugin/setting = value

# 4. Command Line

```
css -pluginCustomization /path/to/settings.ini
```

**Format: Same as plugin\_customization.ini**

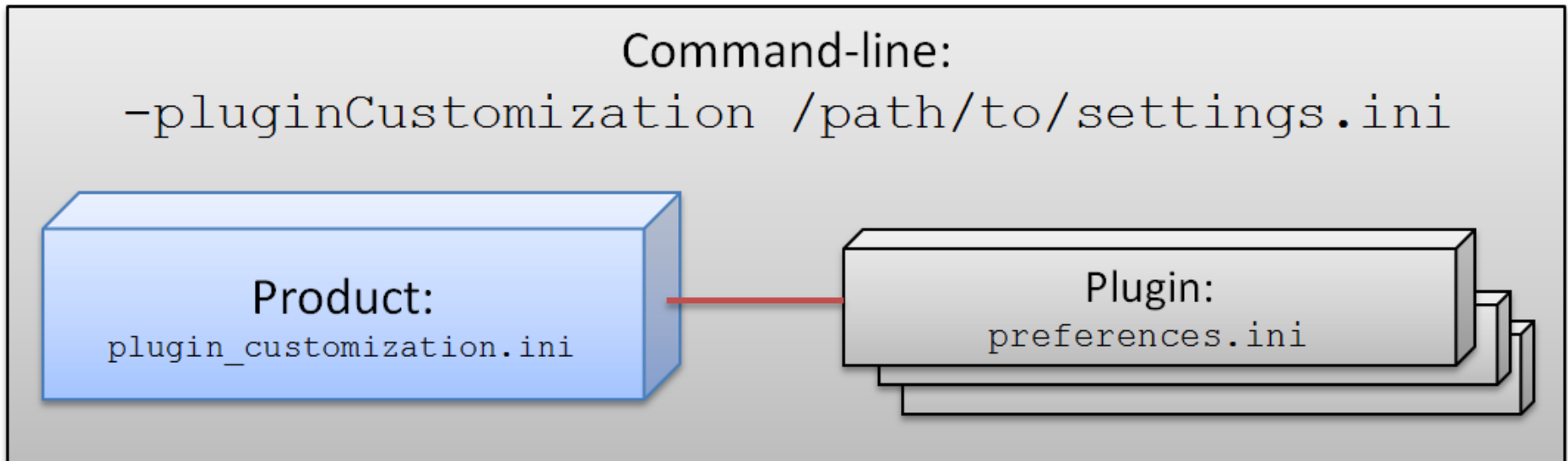
```
name.of.plugin/setting = value
```

**To avoid pulling your hair out:**

- **Do include the /path/to/...!**
  - File not found: No error message
- **Double-check name.of.plugin, name-of-setting!**
  - Typo: Just won't work

# Hierarchical Preferences

- **org.csstudio.platform.libs.epics preferences.ini:**  
addr\_list=
- **Product's plugin\_customization.ini:**  
org.csstudio.platform.libs.epics/addr\_list=127.0.01
- **Command-line**



# Scenarios

## Site-specific product:

- SNS CSS, NSLS-II CSS, ...

**Best, but also more work**

## Use existing product

- Edit product's `plugin_customization.ini` before passing on to users at your site

**Quite practical for a while**

## Command-line `–pluginCustomization /.../x.ini`

**To get started, or to support different setups for 'Office', 'Control Room', 'Home', ...**