

Plug-In-Entwicklung für die Eclipse Platform

Florian Hawlitzek

V1.0

Informationsquellen im Web



◆ Plug-In-Verzeichnisse

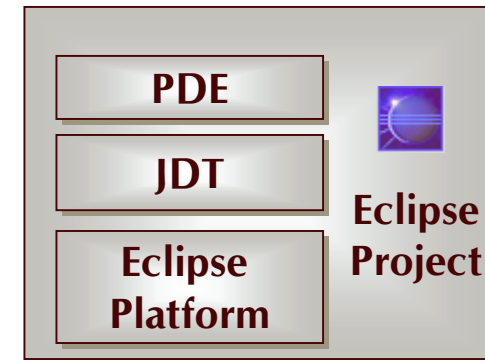
- ▶ <http://eclipse-plugins.2y.net/>
- ▶ <http://www.eclipse.org/community/plugins.html>
- ▶ <http://eclipse-workbench.com/jsp/plugins.jsp>

◆ Mailingliste

- ▶ PDE: <http://www.eclipse.org/mailman/listinfo/pde-dev>

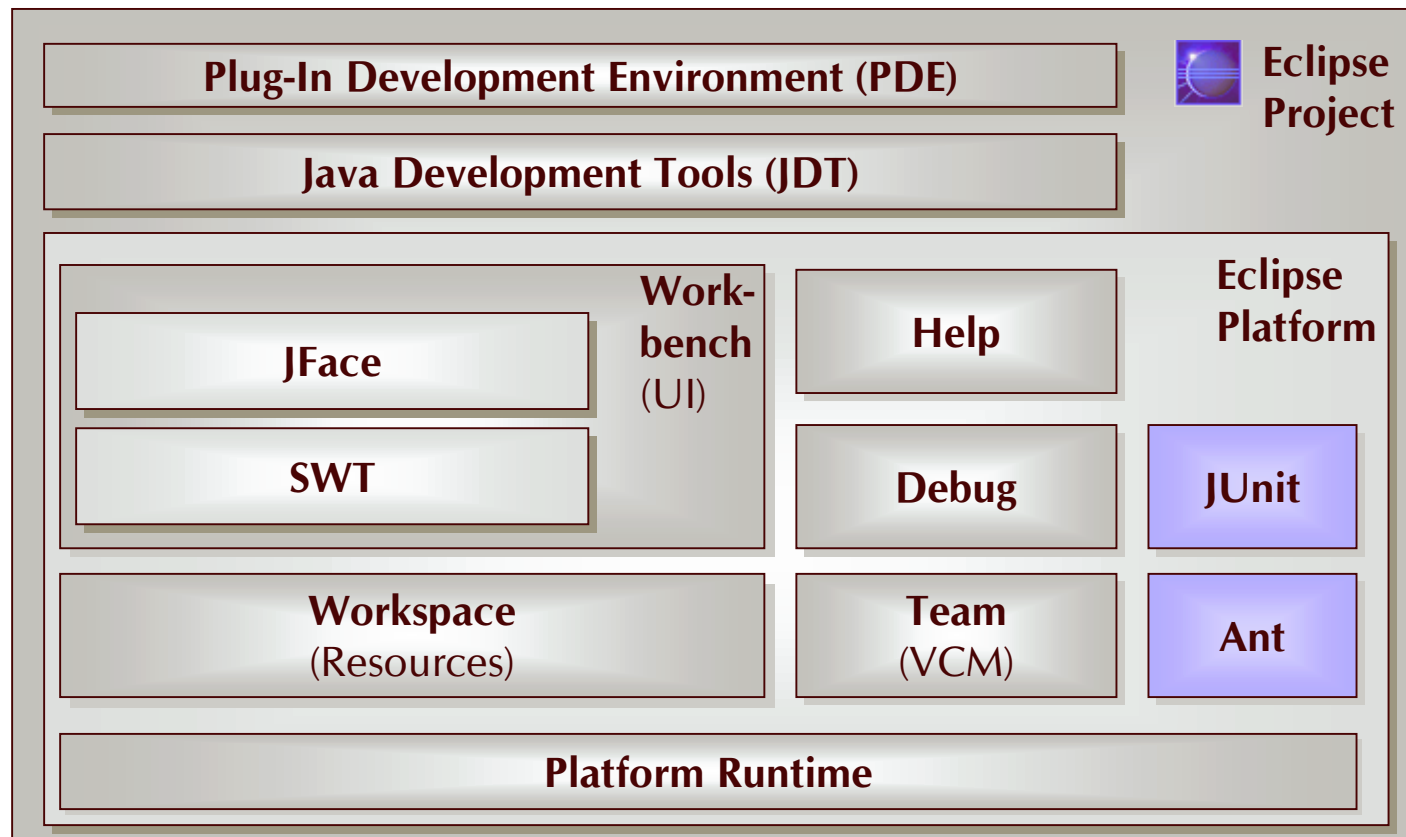
Eclipse-Subprojekte

- ◆ Eclipse Project (→SDK)
 - ▶ Eclipse Platform (IDE-Framework)
 - ▶ JDT (Java Development Tools)
 - ▶ PDE (Plug-in Development Env.)
- ◆ Eclipse Tools
 - ▶ Weitere Werkzeuge auf Eclipse-Basis
 - ▶ C/C++ , COBOL, Grafik, Modellierung...
- ◆ Eclipse Technology
 - ▶ Forschung und Entwicklung
 - ▶ Training



Eclipse Project Aufbau

Eclipse Project



Plug-In-basierter Aufbau



◆ Basis (Core)

- ▶ Platform Runtime und Workspace
- ▶ Ressourcen-Management und Plug-In-Aufbau

◆ Workbench

- ▶ User Interface
 - Perspektiven, Views, Editoren
- ▶ basiert auf
 - SWT (Standard Widget Toolkit): Kontrollelemente
 - JFace: Aktionen, Dialoge, Wizards, Dokumentenmodell

Plug-In-basierter Aufbau (Forts.)



- ◆ Team (VCM)
 - ▶ Schnittstelle zu Version and Control Management
 - ▶ Team-Repository, Default: CVS
- ◆ Debug
 - ▶ GUI + allg. Debug-Schnittstelle
- ◆ Help
 - ▶ Erweiterbares Hilfesystem, Suche (Lucene)
- ◆ Integrierte Open-Source-Tools:
 - ▶ Apache Ant (Build), Lucene (Suche), Xerces (XML)
 - ▶ JUnit (Unit-Tests)

Plug-In-basierter Aufbau (Forts.)



- ◆ Java Development Tools (JDT)
 - ▲ GUI: Java-Perspektive(n)
 - Package-, Gliederungs-, Hierarchie-Views
 - Java-Editor, -Debugger
 - Formatter, Refactoring Tools
 - ▲ inkrem. Compiler, Debugger, Java-Syntaxmodell
- ◆ Plug-In Development Environment (PDE)
 - ▲ GUI: Plug-In Dev.-Perspektive
 - Wizards, Manifest-Editor, Testumgebung

Eclipse Plug-Ins

Definition

Was ist ein Plug-In?



◆ Plug-In

- ▶ Funktionseinheit in Eclipse
- ▶ besteht aus Plug-In-Manifest (plugin.xml) und Code/Ressourcen
- ▶ besitzt eigenes Unterverzeichnis

◆ Beispiele

- ▶ Vergleichsfunktion: `org.eclipse.compare`
- ▶ SWT: `org.eclipse.swt`

Komplexe Plug-Ins



◆ Feature

- ▶ Aus mehreren Plug-Ins zusammengesetzt
- ▶ Installationseinheit
- ▶ Beispiele: JDT, PDE

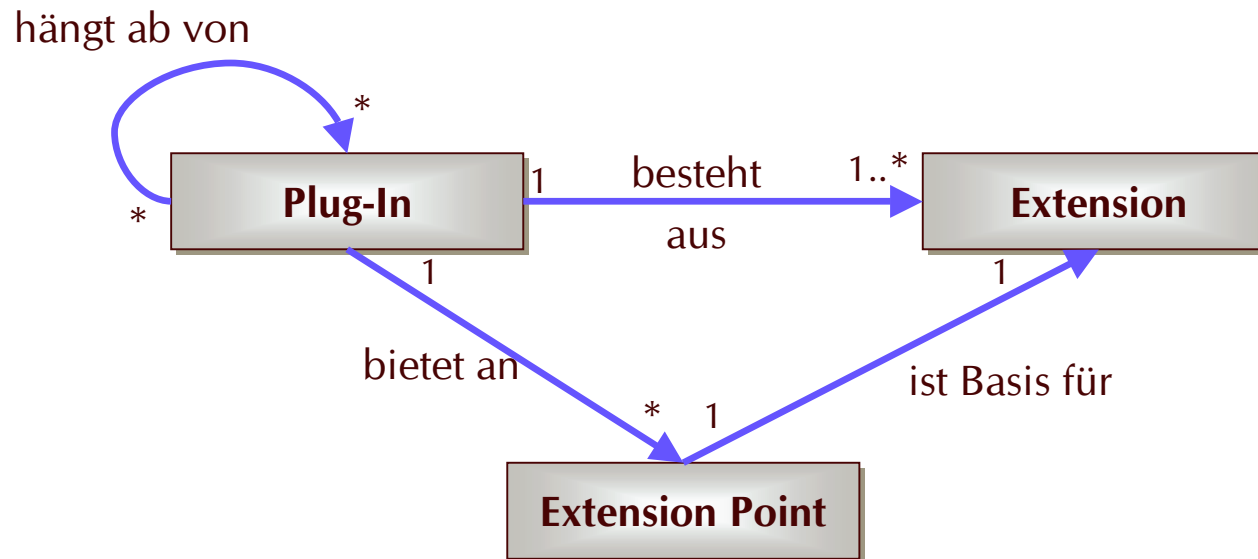
◆ Fragment

- ▶ Erweiterung, Zusatz zu einem existierenden Plug-In
- ▶ Wird in angegebenes Plug-In gemerged
- ▶ Beispiele: Sprachpakete (i18n), Betriebssystemspezifische Teile

Abhängigkeiten



- ◆ Plug-Ins hängen voneinander ab



Abhängigkeiten



◆ Dependencies

- ▶ ein Plug-Ins hängt zur Laufzeit von n anderen Plug-Ins ab

◆ Extension Point

- ▶ benannter Erweiterungspunkt eines Plug-Ins
- ▶ Beispiel: Menü

◆ Extension

- ▶ fügt Eclipse neues Verhalten hinzu
- ▶ Definition, welche Extension Points anderer Plug-Ins das Plug-In benutzt

plugin.xml



- ◆ enthält die Konfiguration eines Plug-Ins
- ◆ IDs, Version
- ◆ Name der Runtimes (Klassen, Ressourcen)
- ◆ Abhängigkeiten von anderen Plug-Ins
- ◆ Extensions
 - ▲ Aktionen
 - ▲ Basis (benutzte Extension Points), Filter, Menüs...
- ◆ bereitgestellte neue Extension Points

plugin.xml - Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin id="de.hawlitzek.eclipse.jdt.constructor"
  name="HawConstructor Plug-in"
  version="1.0.0"
  class="de.hawlitzek.jdt.ext.ConstructorPlugin">
<runtime><library name="constr.jar"/></runtime>
<requires>
  <import plugin="org.eclipse.core.resources"/>
</requires>
<extension
  id="de.hawlitzek.jdt.ext.editorMenuExtensions"
  point="org.eclipse.ui.editorActions">
...

```

Eclipse Plug-Ins

PDE 1/2

Plug-In Development Environment



HawConstructor Plug-in

Alerts and Action Items

There are 0 problem(s), 1 task(s) associated with this plug-in.
This plug-in contains unresolved and/or cyclical references to other plug-ins

General Information

This section describes general information about this plugin

Plug-in Id: `de.hawlitzeK.eclipse.jdt.constructor`

Plug-in Name: `HawConstructor Plug-in`

Version: `1.0.0`

Provider Name: `HawlitzeK IT-Consulting GmbH`

Class: `de.hawlitzeK.eclipse.jdt.constructor.C`

Runtime Libraries

The following libraries will be added to the classpath of this plug-in:

- `constructor.jar`

Required Plug-ins

The following plug-ins must be installed for this plug-in to work properly:

- `org.eclipse.core.resources`
- `org.eclipse.core.runtime`
- `org.eclipse.core.boot`
- `org.eclipse.ui`
- `org.eclipse.jdt`
- `org.eclipse.jdt.core`
- `org.eclipse.jdt.ui`

Extensions

The following extensions are defined in this plug-in:

- `org.eclipse.ui.editorActions`
- `org.eclipse.ui.popupMenus`

Extension Points

Other plug-ins can use the following extension points to extend this plug-in:

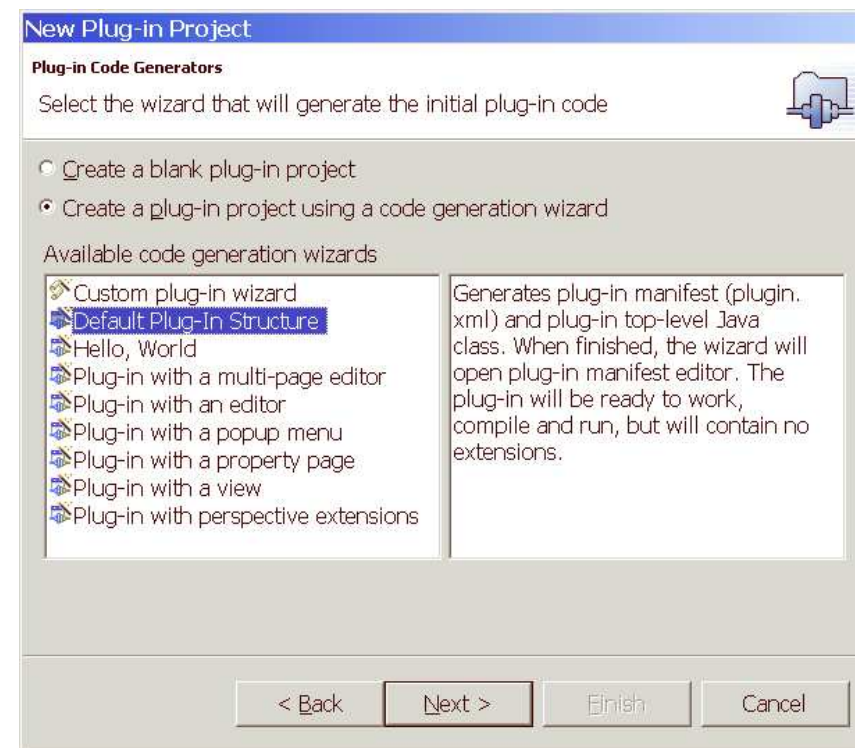
-

Outline

- Welcome
- Overview
- Dependencies
 - `org.eclipse.core.resources`
 - `org.eclipse.core.runtime`
 - `org.eclipse.core.boot`
 - `org.eclipse.ui`
 - `org.eclipse.jdt`
 - `org.eclipse.jdt.core`
 - `org.eclipse.jdt.ui`
- Runtime
 - `constructor.jar`
- Extensions
 - `org.eclipse.ui.editorActions`
 - `org.eclipse.ui.popupMenus`
- Extension Points

PDE Wizard

- ◆ Erstellung neuer Plug-Ins und Fragmente
- ◆ generiert plugin.xml
- ◆ Template für typische Anwendungsfälle



Plug-In Aktivierung

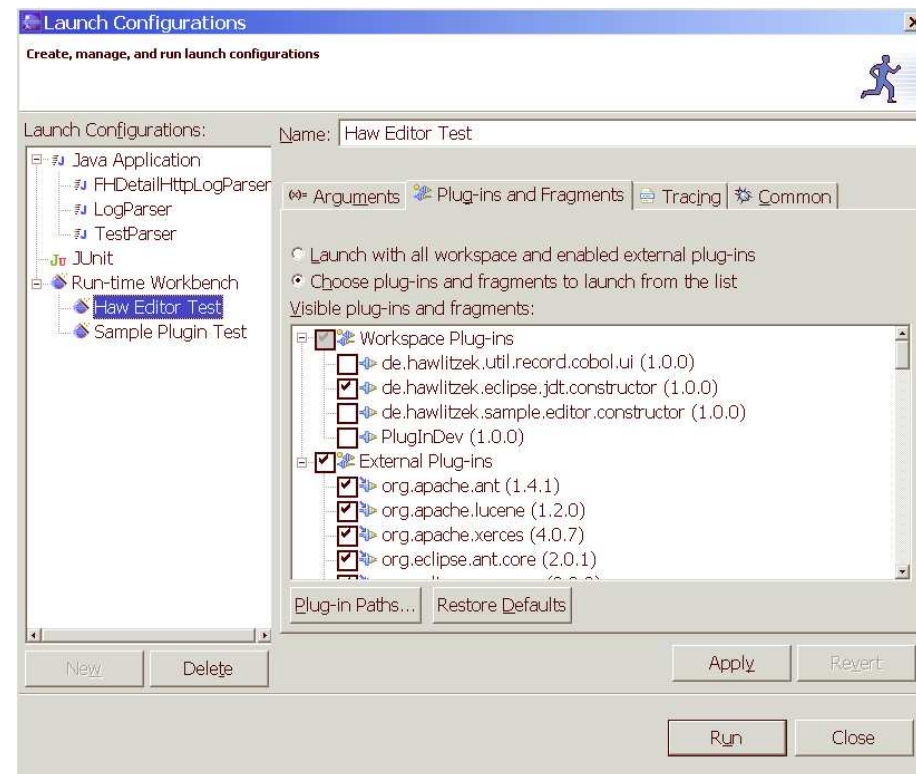


- ◆ Eclipse erzeugt beim Start eine Liste der installierten Plug-Ins
- ◆ Plug-Ins werden erst beim Aufruf geladen
- ◆ Jedes Plug-In besitzt eigenen Classloader
- ◆ Plug-In Registry wird gecacht

Eclipse Plug-Ins Run-time Workbench

Testumgebung

- ◆ Zweite Eclipse-Instanz
- ◆ Getrennter Workspace
- ◆ Auswahl darin aktiver Plug-Ins
- ◆ Debugging möglich



Workspace/Resource API



- ◆ `org.eclipse.core.*`
 - ▲ Zugriff auf Projekte, Verzeichnisse, Dateien
 - ▲ Zustandsverwaltung, Build-Prozess
- ◆ `org.eclipse.ui.*`
 - ▲ Standarddialoge, Import/Export
 - ▲ Aktionen (Erzeugen, Löschen, Kopieren...), Tasks
 - ▲ Texteditor

APIs für Java-Tools



- ◆ org.eclipse.jdt.*
 - ▲ Verwaltung von Java-Projekten (IJavaProject)
 - ▲ Syntaxmodell von Java-Klassen, Binaries
 - ▲ Compiler, Debugger
 - ▲ UI: Editor, Views, Vergleich, Java-spez. Dialoge
 - ▲ Refactoring, Content Assist
 - ▲ JUnit-Einbindung
 - ▲ Launch Configurations

APIs für grafische Plug-Ins



◆ org.eclipse.swt.*

- ▶ AWT-ähnlich, aber moderne Kontrollelemente
- ▶ Grafische Kontrollelemente
- ▶ Layout

◆ org.eclipse.jface.*

- ▶ Wizards, Views, Menüs
- ▶ Aktionen
- ▶ Dokumentenmodell für Editoren

Hawlitzek IT-Consulting IBM Business Partner



◆ Anschrift:

Hawlitzek IT-Consulting GmbH
Landsberger Str. 155
80687 München
info@hawlitzek-consulting.de



<http://www.hawlitzek.de>

◆ Anbieter von:

- ✦ Schulungen und Consulting
zu Java, Eclipse und WebSphere