



NETBEANS PLATFORM

Satyajit Tripathi

Member Technical Staff

ISV-Engineering, Sun Microsystems



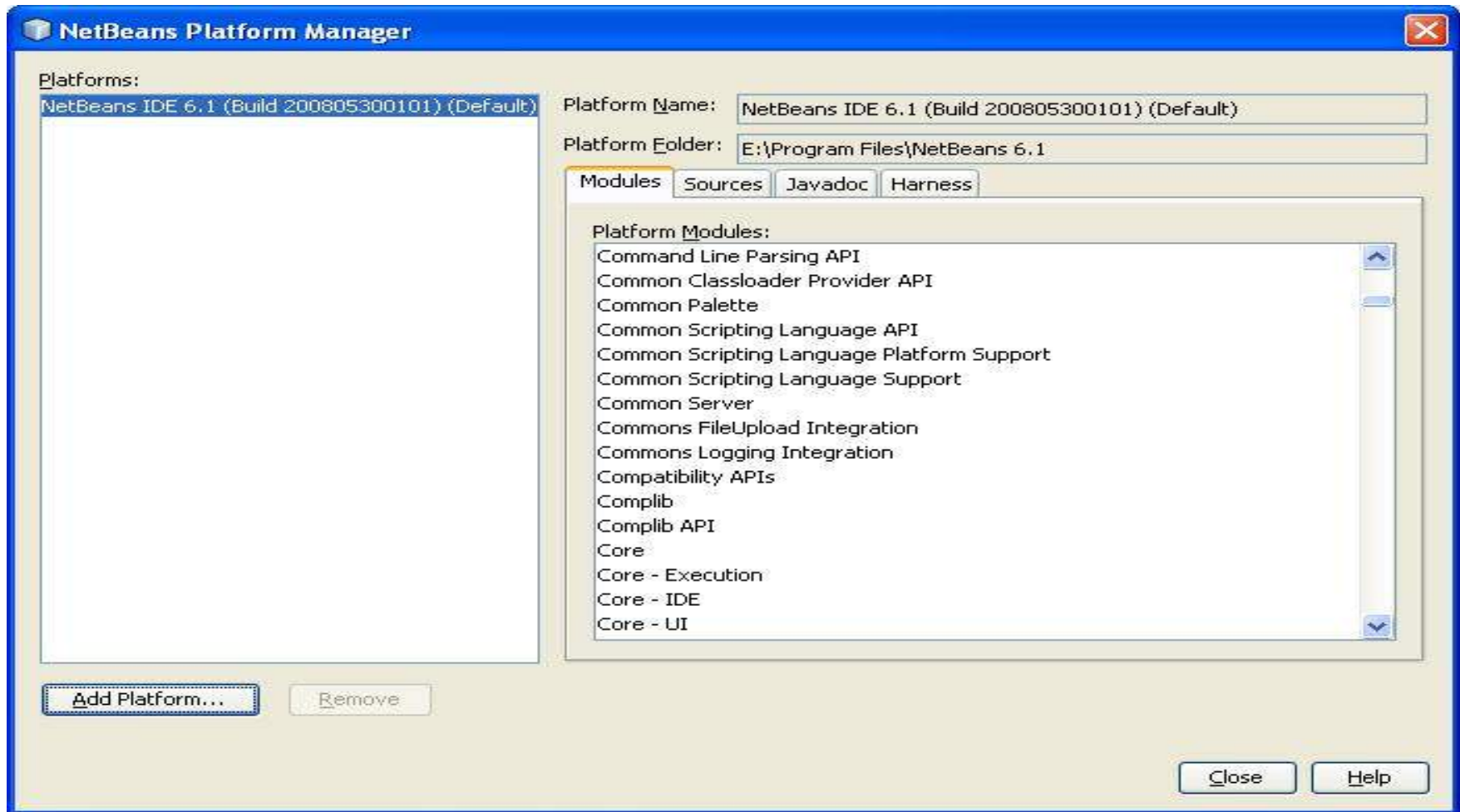
NetBeans Platform

Build new desktop applications without re-inventing the wheel

- NetBeans Platform is a broad SWING-based framework to create rich desktop applications
- Platform provides out-of-the-box APIs to simplify application common requirements such as window management, menus, actions, settings and storage, an update manager, and files access
- Core of NetBeans-IDE is Platform ie. NetBeans-IDE 'minus' IDE specific modules

NetBeans Platform Manager

NetBeans IDE 6.1 (build 200805300101)



NetBeans Module (Plugin)

Add the missing functionalities to IDE

- Module can be built to extend the functionality of NetBeans IDE and to add specific features
- NetBeans (Plug-in) Module is a group of Java classes that interacts with NetBeans APIs and provides an application with specific features
- Java classes use the MANIFEST.MF file to declare the module and XML Layer file (layer.xml) to register their functionality
- Modules with non-installer distributions are packaged as NBM files (.nbm extension)

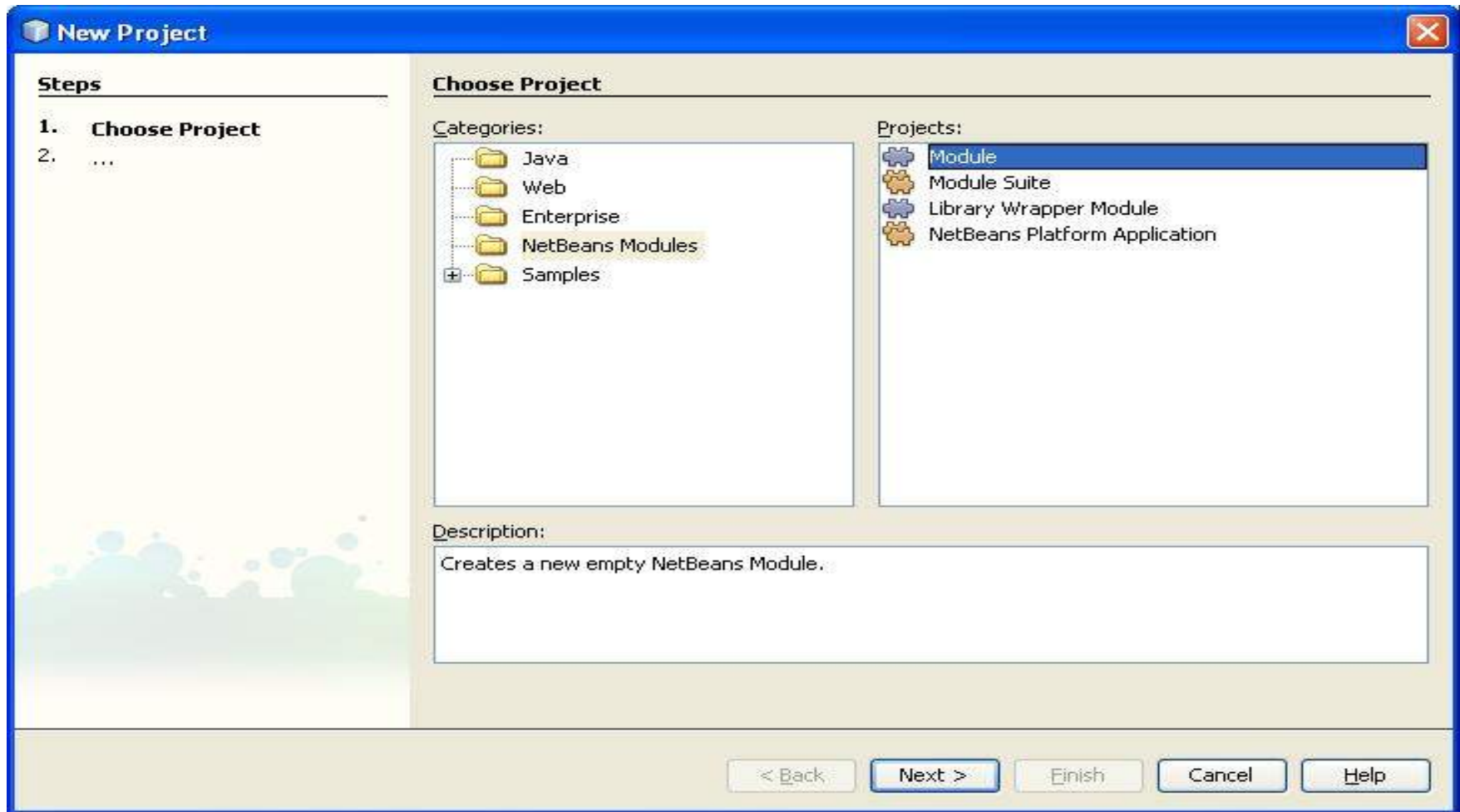
Module (Plugin) Development

Development Life cycle

- Setting up the project
- NetBeans IDE generates Project organization and code
- Coding the module or application
- Building and Running
- Testing, Debugging, and Profiling
- Branding and Distributing

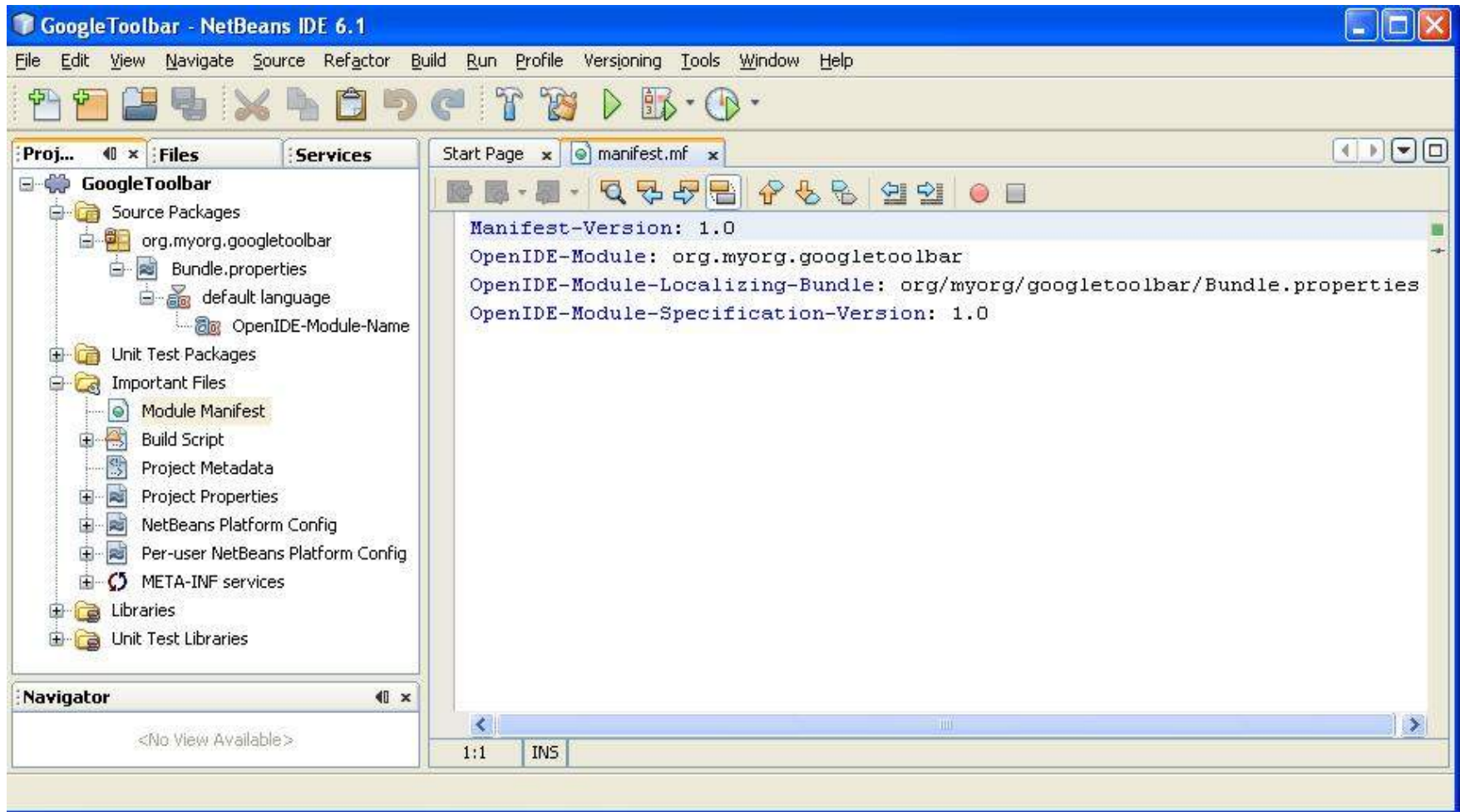
NetBeans Module Projects

NetBeans IDE 6.1



Module Project Organization

NetBeans IDE 6.1



NetBeans APIs

Complete API set is quite extensive!

- Action APIs
- File Systems APIs
- Loaders APIs
- Nodes APIs
- Windows APIs

- Palette APIs
- Refactoring APIs
- MultiView APIs
- JavaHelp Integration APIs

Action APIs

Package : org.openide.actions

- Standard SWING Actions
For installation of global, always-enabled actions, register javax.swing.Action in appropriate folder of System Filesystem. Not necessary to subclass one of NetBeans action class. Actions are presented in popup menus or attached to a component such as window, node, data object, or filesystem, or installed globally in the main menu or toolbars
- Standard NetBeans API Actions
Action subclasses available in NetBeans API are subclass of SystemAction. SystemAction objects should be Singleton.
 - > Callback actions
 - > Context-sensitive (base class org.openide.util.actions)
 - > Presenters
- UseCallableSystemAction (Always enabled)
- Use CookieAction (Conditionally enabled) One or Multiple Nodes

File System APIs

Package : `org.openide.filesystems`

- Manipulating files on disk, and used indirectly by Loaders API
- Common Tasks
 - > Finding files and folders
 - > Creating, deleting and renaming files and folders
 - > Reading and writing files
 - > Listening on file events
 - > Determining MIME Content type

Loaders (Datatypes) APIs

Package : `org.openide.loaders`

- Datasystems are the logical layer between a filesystem and higher level file oriented operations. The API works on top of file objects and gives each file a logical behavior – icon, name, operation, etc. Each file types recognized by NetBeans has a corresponding DataObject subclass provided by the module that add support for the file type
- There are three packages in this API
 - > **`org.openide.loaders`**
Handles cluster of files together into groups and assign types to data
 - > **`org.openide.cookies`**
Provides a design pattern for attachable behaviors to data objects and nodes. Cookies provides a way for both data objects and nodes to indicate in flexible and dynamic fashion the operation supported
 - > **`org.openide.util.datatransfer`**
Implements some extensions to the clipboard, and enhances the functions provided in `java.awt.datatransfer`

Nodes APIs

Package : org.openide.nodes

- The Nodes API controls the usage and creation of nodes, which are a variant of Java Beans having adjustable property sets; provide cookies and actions; formed into a hierarchical tree.
- Common node types
 - > Data nodes
 - > Data folder nodes
 - > A user-developed Java Bean on disk
 - > A node derived from a Java Bean , not representing one
 - > The Component Palette
 - > A breakpoint in the debugger is a node
 - > A project desktop node (various files associated with a project)

Window System APIs

Package : `org.openide.windows`

- The implementation takes care of the main window, all multi-tabbed, and split frames, toolbars and MDI/SDI support
- As a rule, modules should not create their own top-level windows (eg. `java.awt.Window`). Those would otherwise not be managed by NetBeans window manager
- Window API allows module to provide window-like components, mainly through embeddable visual components called topcomponents. A top component is a SWING component and need to be docked to be manipulated by the window system
- Window manager is capable of manipulating application windows including docking and window configuration persistent across session
- The window system layout in NetBeans is based on modes. API provides access to modes but cannot create directly. This can be achieved by specifying in the XML layer

Component Palette APIs

Package : `org.netbeans.spi.palette`

- The API provides access to Common Component Palette. The palette client can define content to displayed in the common palette TopComponent when their editors are active
- The API includes support for the clients writing palette content insertable into the text editor
- Allows users to Drag and Drop text into the palette to create new custom code clips. This can be achieved by subclassing DragAndDropHandler
- Palette visibility is defined per document-type
- Palette provider needs to define own actions to reset its default state
- Allows associating palette content with document MIME type
- Allows displaying name and tooltip to be defined directly in item's XML

MultiView APIs

Package : `org.netbeans.core.api.multiview`
`org.netbeans.core.spi.multiview`

- The API allows to access the multiview component's content. The SPI handles the lifecycle of a multiview component
- Multi views enable modules to display several perspectives, or views of data, visually represented as data document or object in unified manner at one specific place in the system
- A marker interface `SourceViewMarker` has been added to the `MultiViewDescription` instance to identify as containing source code
- Factory methods to create multiview components
- MultiViews project depends on Window System API (`org.openide.windows.TopComponent`) and some new APIs in `openide` `NbDocument.CustomToolbar` and `CloneableEditorSupport.Pane`

JavaHelp Integration APIs

Package : `org.netbeans.api.javahelp`

- JavaHelp integration API wraps the standard JavaHelp extension library (jh.jar version 1.1.3 or on Solaris `/usr/j2se/opt/javahelp/lib/jhall.jar`)
- It also provides a small additional API for NetBeans modules to supply help sets to the system, add GUI menu items, and request that particular help topics be displayed
- A singleton instance of `org.netbeans.api.javahelp.Help` is registered into lookup. An instance of a handler for the `nbdocs` URL protocol is also registered for the module code to find it

NetBeans 6.1 Platform Resources

- NetBeans Platform Site : platform.netbeans.org
 - > Learning: overview, tutorials, quick start, FAQ
 - > Community: news, mailing-list, nbusers@netbeans.org, source
- NetBeans API List : [Javadoc](#)
- NetBeans 6.1 API Changes : [api-changes](#)
- NetBeans Platform wiki : wiki.netbeans.org
 - > Release and planning, answers and support, tutorial and api-docs
- Expert Presentation Series : [netbeans_platform](#)
- NetBeans Platform Certified Training : [nbplatform-certified-training](#)



NETBEANS 6.1

Satyajit Tripathi

satyajit.tripathi@sun.com

