



**WORLD EVENT
2010**

An Integrated Approach to Managing Windchill Customizations

Todd Baltes

Lead PLM Technical Architect

SRAM



WORLD EVENT
2010



Event hashtag is #PTCUSER10

Join the conversation!

Topics

- What is an “Integrated Approach” to Windchill customization?
- Benefits of an Integrated Approach
- Tools for an Integrated Approach
 - Eclipse for Windchill customizations
 - The automated build process
 - Release management
- Demonstration

What is an “Integrated Approach”?

- “Change Management” for customizations
 - Version control and release management
- Support Windchill customization life cycle
 - Team development
 - Enterprise deployment
 - Windchill upgrades
- Comprehensive set of integrated tools to support Windchill customizations

What is an “Integrated Approach”?

- Leverage PTC best practices for customizations
 - Strict “wtSafeArea” approach
 - “wtCustom” resource management
- Leverage Windchill tools
 - Build resources, install site changes, and more
- Automate as much as possible
 - Enable efficiency, consistency and reliability

Benefits of an Integrated Approach

- Efficient development
 - Dynamic compilation and documentation
 - Integrated version control
- Reliable and consistent compilation and deployment
- Integrated version control and release management
 - Only manage what you need to
 - “Time travel”
- Reduced customization costs

Efficient Development

Dynamic Compilation and Integrated Documentation

- Dynamic Compilation
 - Warnings and errors presented visually
 - Uses the Windchill JDK and Windchill APIs
- Integrated Documentation (JavaDoc)
 - Sun Java APIs
 - Windchill APIs
 - Third party and custom APIs

Efficient Development

Integrated Version Control

- Team development – a holistic view
 - Synchronization of incoming and outgoing changes
 - Intelligent merging of concurrent modifications
- File comparison tools
 - File, line, and character level comparison
- File history
 - Version-version comparisons
 - Detailed version-level information

Reliable Compilation and Deployment

- Generate artifacts at time of deployment
 - Java class files and resource bundles
 - Ensure consistency with run-time
- Integrate Windchill deployment tools
 - Ensure consistency with Windchill platform
 - Supports evolution of Windchill tools
- Automated process
 - Reliability and consistency

Version Control and Release Management

Only manage what you need to

- Only version control “source” artifacts
 - Custom Java code
 - Custom JSP
 - Custom configurations
 - Custom resources (.rbInfo)
- Do not version control anything that can be generated
 - Java classes, custom resource bundles
- Instead, leverage automated build
 - Ensure consistency with deployment platform

Version Control and Release Management

“Time Travel”

- Use “tags” to specify significant moments in time
 - Project start or end
 - Release Candidates
 - Final Builds
- Use source control tools to navigate “tags”
 - “Check out” any tag to get contents at that time
 - Easy to deploy new release
 - Easy to revert to any tagged point in time
 - Easy to identify all source changes

Reduced Customization Costs

- Short-term customization cost savings
 - Development – faster, easier coding
 - Reliable deployments
- Long-term customization cost savings
 - Windchill platform upgrades
 - Support of Windchill tools

2010

Tools for an Integrated Environment

- Subversion and Subclipse
 - Source code management tools
- The Eclipse IDE
 - Powerful, extensible development tool
- ANT
 - Facilitates build and deployment activities
- Windchill tools
 - Server management and deployment
 - Custom resource/JAR management

Tools for an Integrated Environment

Subversion

- Open source version control system
 - Full-featured, fast, and efficient
- Enables reliable release management
 - Tags and branches
- Learn more at <http://subversion.apache.org>

2010

Tools for an Integrated Environment

Subclipse

- Subversion plug-in for Eclipse IDE
 - Integrates source control into IDE
 - Full-featured
- Enables effective team development
 - Intuitive comparison tools
 - Provides holistic view of repository changes
 - Enables tagging and branching
- Learn more at <http://subclipse.tigris.org>

Tools for an Integrated Environment

Eclipse IDE

- Powerful, extensible development environment
 - Plug-in framework
 - Specialized text-editors
- Available in several configurations
 - JEE version works well for Windchill
- Learn more at <http://www.eclipse.org>

2010

Tools for an Integrated Environment

ANT

- Facilitates Java build process through XML “script”
- Built in support for many activities
 - File system operations
 - Java compilation
 - Execution of “external” applications
- Extensible
 - Third party and open source extensions
 - If you can code it, you can run it in ANT
- Learn more at <http://ant.apache.org>

Tools for an Integrated Environment

Windchill Tools

- Server management
 - Start and stop Tomcat and Windchill servers
- Build resource files
 - Using “wtCustom” approach
- Generate Windchill JAR files
- Windchill customization installation
 - From “wtSafeArea/siteMod”
- Windchill property management

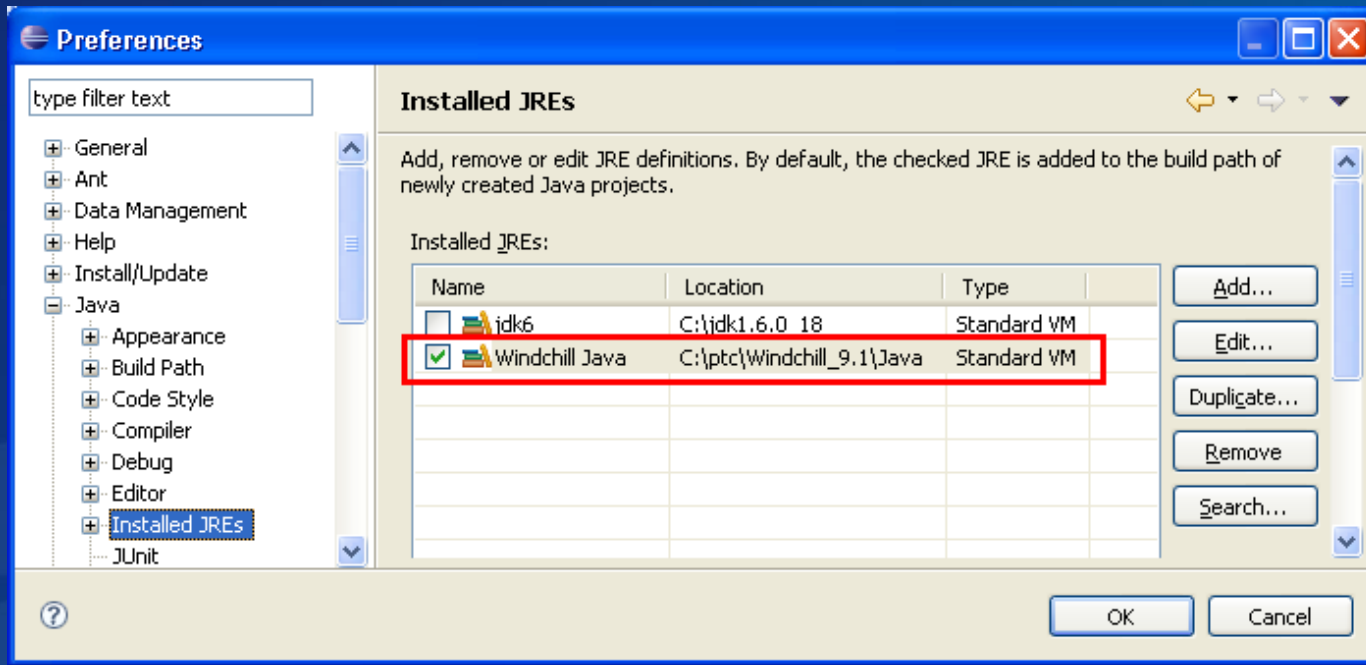
Eclipse for Windchill Customizations

- Download the Eclipse IDE
- Install the Subclipse plug-in for Eclipse
- Configure a Windchill customization project
- Create automated build process using ANT
- The result: an **efficient and productive** development environment for Windchill customizations

Eclipse for Windchill Customizations

Add the Windchill JRE to Eclipse

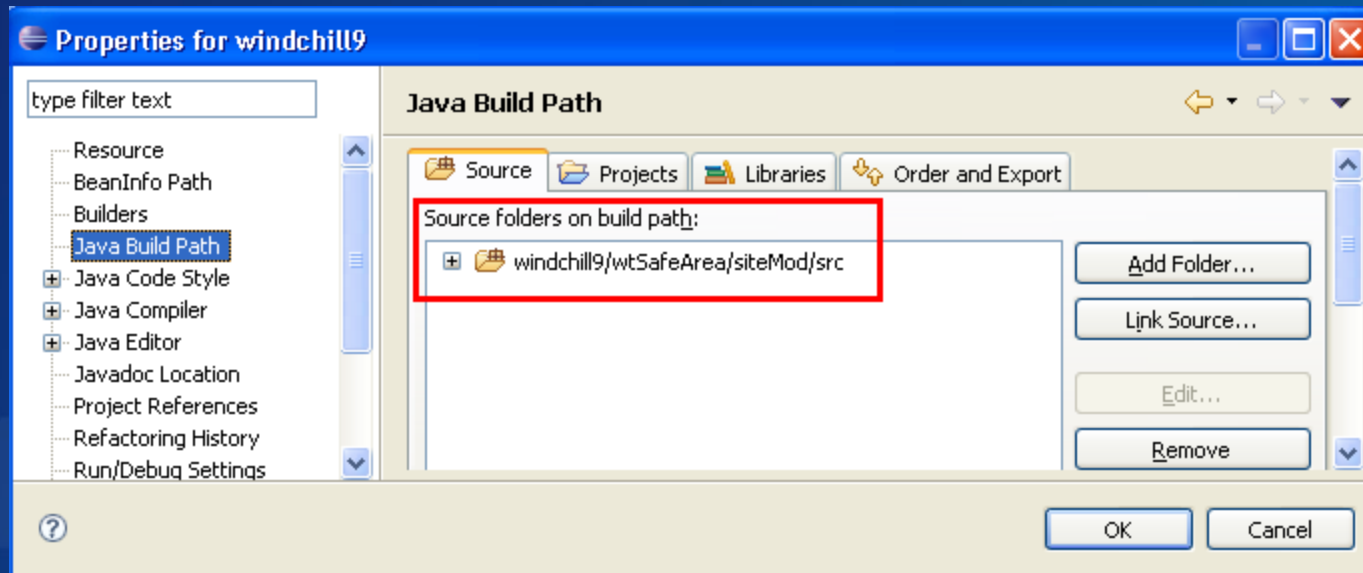
- Preferences → Java → Installed JREs → Add...



Eclipse for Windchill Customizations

Create a project and define the source directory

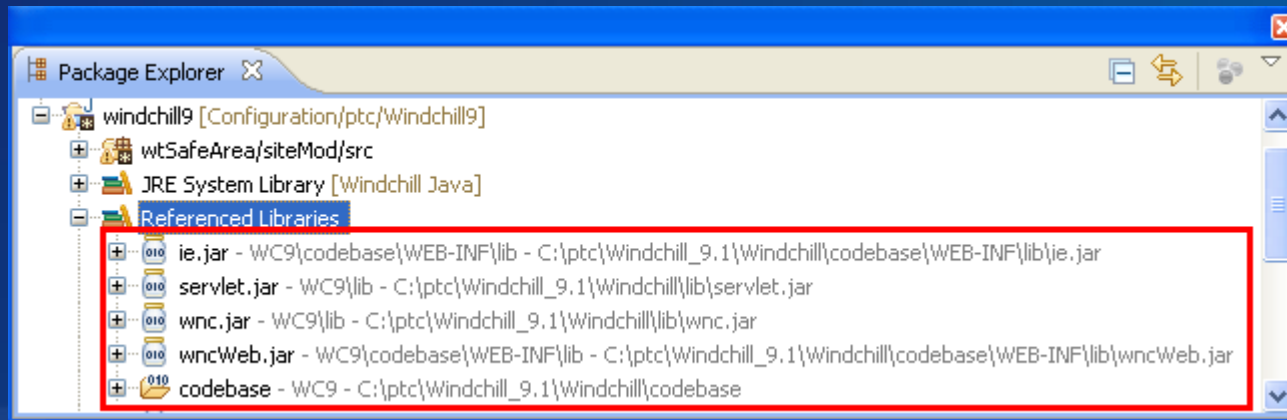
- Use wtSafeArea/siteMod/src



Eclipse for Windchill Customizations

Add Windchill reference libraries to project

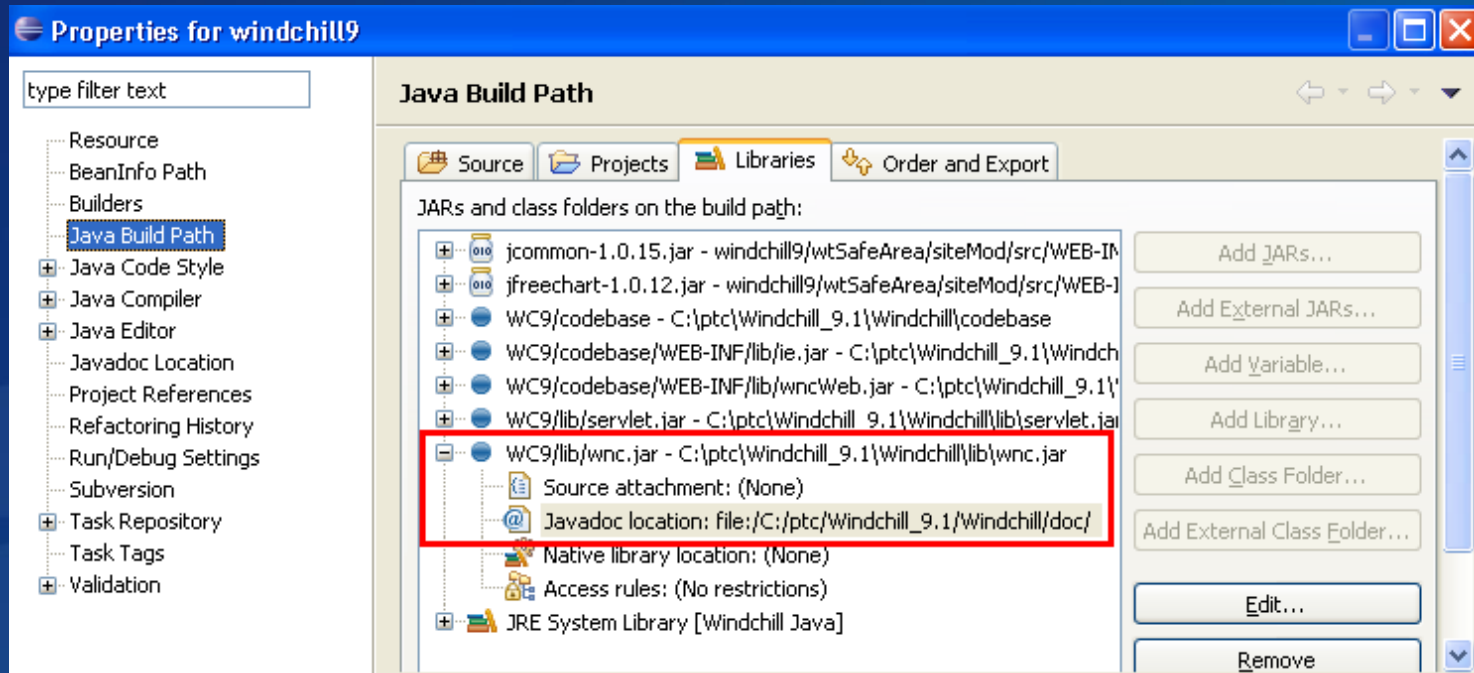
- Include JARs as well as “codebase” directory



Eclipse for Windchill Customizations

Add Windchill JavaDoc to referenced libraries

- Download and extract Windchill JavaDoc from PTC
- Add JavaDoc location to referenced libraries



The Automated Build Process

- Build Process defined as an ANT project
 - Typically in a file named “build.xml”
- Each Process “task” defined as an ANT target
- Each target can
 - Execute one or more ANT tasks
 - Define dependencies on other targets
- Targets can be run individually
 - Allows partial-build capabilities
 - Dependencies ensure proper execution

Build Process Details – Install target

- “Install” target executes a complete build
 - Executes all relevant targets in the proper order
- “Install” can be run from within Eclipse
 - Efficient, Consistent deployment of code changes
- Can also be run from command line
 - Enterprise server deployment

2010

Build Process Details – Install

- Set Build Properties
- Stop Windchill Servers
- Clean up generated artifacts
- Compile and copy customizations
- Build Windchill resource files
- Build Windchill JAR files
- Install Site Changes
- Propagate Windchill properties
- Start Windchill servers

Build Process Details

Set Build Properties

- “Initialize” the build
- Define the build environment
 - Windchill installation directory
 - Other relevant directories
- Initialize Logging
 - Record each activity for build auditing

2010

Build Process Details

Stop Windchill Servers

- Stop Windchill
 - <exec> `${windchill.home}/bin/Windchill.exe`
 - <arg> “stop”
- Stop Tomcat
 - <exec>
`${windchill.home}/../Tomcat/bin/wttomcat_stop.bat`

2010

Build Process Details

Clean Up Generated Artifacts

- Dependency on “Initialization”
- <delete> artifacts from previous build
 - Windchill installation directory
- <delete> previous JavaDoc
 - Consistency and reliability...

2010

Build Process Details

Compile and Copy Customizations

- Dependency on “Clean Up”
- `<javac>` to compile the customizations
 - Provide “classpath” and compiler details to facilitate command line execution
- `<copy>` files to “wtSafeArea/siteMod/codebase”
 - Compiled “.class” files
 - Non-java source files (JSP, properties, etc)

Build Process Details

Build Windchill Resource Files

- Dependency on “Compile”
- “wtCustom” approach
 - All resource modifications in “wtCustom”
 - Merge customizations into OOTB using “src” files
 - Includes modified OOTB and new (custom) resources
- `<ant> {windchill.home}/bin/tools.xml`
 - Target = “bundle_custom”

Build Process Details

Build Windchill JAR Files

- Dependency on “Resources”
- Ensures consistency between server and client resource bundles
- `<ant> ${windchill.home}/codebase/MakeJars.xml`

2010

Build Process Details

Install Site Changes

- Dependency on “Make Jars”
- Deploys ALL customizations from “wtSafeArea/siteMod”
 - Includes codebase, loadfiles, tasks, etc.
- This process supports Windchill upgrades
 - Guarantees that “wtSafeArea” is consistent with deployed customizations
 - Reduces long-term costs of customizations
- `<ant> ${windchill.home}/bin/swmaint.xml`
 - Target = installSiteChanges

Build Process Details

Propagate Windchill Properties

- Run Windchill “xconfmanager” process
- `<java>`
`${windchill.home}/codebase/WEB-INF/lib/install.jar`
 - `<arg> “-r”`
 - `<arg> ${windchill.home}`
 - `<arg> “-p”`

2010

Build Process Details

Restart the Windchill Servers

- Start Windchill
 - <exec> `${windchill.home}/bin/Windchill.exe`
 - <arg> “start”
- Start Tomcat
 - <exec>
`${windchill.home}/../Tomcat/bin/wttomcat_start.bat`

2010

“The road to learning by precept is long, but by example short and effective.”

– Seneca

2010



WORLD EVENT
2010

2010