Project Migration CDT Enhancement

1 Introduction

1.1 What is Project Migration?

A Project Migration is a variation of a project import. The primary inputs to a Project Migration are an initial project directory (source and project specific files like .project and .cproject) and a destination project directory (source but no project-specific files like .project or .cproject files) The migration consists of copying the .project and other project-specific files from the initial directory into the destination directory and then importing the destination directory into the user's workspace.

The primary use case scenario for a Project Migration could involve an administrator that sets up a reference C/C++ project directory and stores the exported Team Shared Index. One or more developers can then create their own destination directories and "migrate" the reference project into their own destination directory. They only have read access to the reference directory and the project source is too large to efficiently copy from the reference directory into their own. Instead, a source code control system like ClearCase can be used to efficiently produce another copy of the source code for the developers.

In addition to just copying the project-specific .project and .cproject files, there is some facility in this implementation for making changes to path names used for Eclipse linked folders. If there are path names in the reference project which are "outside" of the project directory and need to be different in the developer's copy of the project, then these transformations can be taken care of.

To allow for the case where a single developer may want to migrate the same reference project multiple times, the facility for renaming the project is also provided. A developer may need to migrate the same project more than once if they have multiple defects they are working on and each needs to be submitted separately.

1.2 What is a Trusted Project Migration?

One of the main goals of a Project Migration is to reduce the time it takes to get a developer up and running with a copy of the reference project. For a C/C++ project this includes having an Index available so the developer can browse the code and start working on the desired changes.

While there is a facility for reusing a Team Shared Index, for large projects, there is still a considerable amount of time spent on doing a Project Refresh at the time of doing an import. In the primary use scenario, where the developer project directory is assured of being the same as the reference project directory (through the magic of ClearCase), the time spent on checking the time-stamps of thousands of source files is wasted time. A trusted migration enhances the notion of the Team Shared Index to include the resource time-stamp information. When the developer does the Project Migration, this saved information is used to greatly reduce the Project Refresh time (since the actual file system does not need to be traversed). In addition, since the developer source files are assumed to be the same as the reference source files, the checksums stored with the Team Shared Index do not need to be checked.

The Trusted Project Migration saves a considerable amount of time.

1.3 Primary Obstacles

The primary obstacles that needed to be overcome in order to have an efficient Trusted Project Migration include the following:

1. Symbolic links need to be properly handled. This in not a CDT issue but is more of core eclipse issue. (see http://bugs.eclipse.org/233939) For the case that motivated this work, this was extremely important. Most of the resources in the project were reached through symbolic links. Without a fix for this problem, all of these files were represented in the CDT Index as External files. The Index took more than 2 hours to generate. As External files, they have absolute path names in the index. It doesn’t really hurt the developer to have absolute path names to the reference project. But this doesn’t actually happen since the absolute path names are removed from the index when the Team Shared Index is exported. This means that when the project is imported into the developer’s project, the index needed to be regenerated since most of the files were “missing”. Fixing this problem is critical to getting greater reuse of the shared index.

2. Workspace relative names were being stored in the index. This means the project names were being stored in the index and this effectively prevents the project from being easily renamed at the time of the project migration. This problem has been addressed by (http://bugs.eclipse.org/239472)

3. The Project Refresh times for the case that motivated this work was greater than 10 minutes. This is a considerable amount of time for a developer to sit and wait before they can start working.

4. When a Team Shared Index is exported, a set of file checksums are also exported. These are used at the time of the import to verify that the source files are the same as when the export was done. For very large files, the time needed to verify the checksums can be several minutes.

2 Case Study Results

For the case that motivated this work, the initial conditions were that they had very large projects consisting of more than 50,000 files. These were in ClearCase and most of the project tree consisted of symbolic links to several ClearCase directories. The time it took to recalculate the CDT Index was greater than 2 hours. The time to refresh a project was greater than 10 minutes. In short, they had no effective way for a developer to get a copy of a reference project and become productive. If a developer wanted to create a new project and use the CDT browsing capabilities, they were looking at nearly 2-3 hours before all could be ready.

The changes described above were put in place. These changes addressed all the obstacles and provided a Project Migration capability. The developers are now able to create a new project and are able to use the CDT browsing capabilities in less than a minute. This is a dramatic improvement.

3 Implementation

3.1 Project Migration

The Project Migration consists of a new operation that copies the project files from the reference project to the developer's destination project and then does a Project Import of the destination project into the current workspace. The project files can be modified based on a simple path transformation that allows something like eclipse linked folders to be remapped. The transformation implemented is very simple and may not be sufficient in some cases.

3.1.1 Project Migration Preparation

To prepare for a Project Migration, a new Export Wizard has been added. This wizard creates a .projmig.zip file in the project directory. This files contains the snapshot of the project refresh information and a list of files that should be copied to the new project directory when the migration is done. (e.g. .project and .cproject). The list of files to copy is made by an extension point which particular projects like CDT can make use of. The extension point also allows additional files to be stored in the zip file. CDT uses this capability to store the Team Shared Index for the project.

3.1.2 Project Migration

A new Import Wizard has been added to perform a Project Migration. This operation copies the .projmig.zip file to the destination directory and copies the list of files specified in the zip file (like .project and .cproject) and then the destination project directory is imported into the current workspace. Note that when the files like .project and .cproject are copied, the extension point is used to allow the files to be changed. This is used to implement the project name change.

The CDT project import has been changed so that it can look in the .projmig.zip file in addition to .settings/cdt-index.zip for an exported pdom.

3.2 Command Line Utilities

For the case that motivated this work, it was also requested that a number of operations be available as commands. Primarily, these were intended to be used in nightly jobs for the creation of the reference projects. They have about 4000 reference projects that need to be automatically set up. The commands that have been supplied include the following:

1. Migration Preparation. This command can prepare the migration information for a specific project.
2. Project import. This command can import a project or a set of preferences into a workspace. This command has been most useful for importing the set of preferences.
3. Index Project. This command can cause the CDT Index to be rebuilt for a project.

3.3 Callable APIs

Two capabilities were needed to be callable from other Eclipse plug-ins. The capabilities included:

1. Project Migration Preparation (Export) for a project.
2. Project Migration

4 Changes

4.1 Plugins

From a plugin perspective, the following changes were made:

4.1.1 New CDT Plugins

Plugin name	Purpose
org.eclipse.cdt.cli	This plugin contains utility functions needed by the command line capabilities. This plugin is not dependent on anything in CDT. It isn't clear that being in CDT is the best place for this plugin.
org.eclipse.cdt.cli.commands	This plugin contains the code needed to implement the specific commands. This plugin

	depends on org.eclipse.cdt.cli and other cdt plugins.

4.1.2 Modified CDT Plugins

Plugin name	Changes
org.eclipse.cdt.core	The biggest change is the addition of the Project Migration Helper extension. This helper assists with the Migration Preparation and the actual migration. A change to the import was also done to get the exported pdom from either the .projmig.zip or from .settings/cdt-index.zip. Changes were also made to display things in more detail in the ProgressMonitors for some of the lengthy operations.

4.1.3 Eclipse Platform Changes

There were two Eclipse Platform changes needed for this work. See http://bugs.eclipse.org/233939 and https://bugs.eclipse.org/254492. The first deals with a better handling of symbolic links. An initial implementation is attached to the bug and here. The discussion in the bug is looking for a more complete solution. The second deals with access to an internal API that was not made public. This is only needed in the Project Import command and is not critical to the Project Migration capability. It is not quite clear why the API's that support zip files are public but the API's that support tar files are not but that is a discussion and/or debate that does not need to happen here.

Plugin	Description
org.eclipse.core.resources	Definition of the ProjectMigrationHelpers extension point. Addition of a ProjectMigrationPreparation API to do the export of needed information. An API to do the actual Project Migration is also added. Change in 233939 is needed for this capability.
org.eclipse.ui.ide	Two wizards were added. One is an Export for the Project Migration Preparation. The other is an Import for the actual Project Migration. Both use public API functions that were added to org.eclipse.core.resources.

4.2 Files

This is a slightly more detail look at the new and changed files and the primary reasons for the changes.

Plugin	File	Purpose
org.eclipse.core.resources	META-INF/MANIFEST.MF	Additional packages
	plugin.properties	Strings for extension definition
	plugin.xml	New extension
	src/org/eclipse/core/resources/ResourcesPlugin.java	New extension support
	src/org/eclipse/core/internal/resources/projectmigration	New snap filestore support, extension support
	src/org/eclipse/core/resource/projectmigration	New package containing Public API for migration preparation, migration operation, and migration helper extension
org.eclipse.ui.ide	src/org/eclipse/ui/internal/wizards/projectmigration	New package containing 2 wizards for migration preparation (export) and actual migration (import)

Plugin	File	Purpose
org.eclipse.cdt.core	parser/org/eclipse/cdt/internal/core/pdom/indexer/Messages.java	New messages for progress monitors
org.eclipse.cdt.core	parser/org/eclipse/cdt/internal/core/pdom/indexer/messages.properties	New messages for progress monitors
org.eclipse.cdt.core	parser/org/eclipse/cdt/internal/core/pdom/Messages.java	New messages for progress monitors
org.eclipse.cdt.core	parser/org/eclipse/cdt/internal/core/pdom/messages.properties	New messages for progress monitors
org.eclipse.cdt.core	parser/org/eclipse/cdt/internal/core/pdom/PDOMManager.java	Progress monitor changes. Changed a job from System to User for better visibility in progress view.
org.eclipse.cdt.core	parser/org/eclipse/cdt/internal/core/pdom/TeamPDOMImportOperation.java	Added ability to use the pdom in a .projmig.zip information on import and avoid the checksum verification for a trusted project migration
org.eclipse.cdt.core	src/org/eclipse/cdt/internal/projectmigration	Project Migration Helper extension that helps with migration preparation and with the actual migration. Takes care of CDT specific details.
org.eclipse.cdt.platform-feature	feature.xml	Added the new plugins. org.eclipse.cdt.cli, org.eclipse.cdt.cli.commands.