

Team Development with JUDE and Version Control System

This guide briefly shows how to make your Team development work with JUDE/Professional and Version Control System.

This guide is intended for those who are..

- considering starting a Team Development
- looking for a substitution for JUDE/Server



Change Vision, Inc.
JUDE Development Team
2007.8.9

Potential Problems of Team Development

- Problems the team based developments tend to encounter
 - It is hard for everyone to know who and when is editing what files
 - If the development is carried out with only one project file
 - Only one developer can access the resource file so more than one developer cannot work at a time.
 - The entire project file would be opened all the time, so it could cause performance problems and also make it difficult to keep focusing clearly on which part you need to modify.
 - If the development is carried out by splitting the entire project file and each one taking one's share
 - Merges need to be performed frequently by hand to maintain consistency.
 - Some could modify models of other's share by mistake and could cause instability.

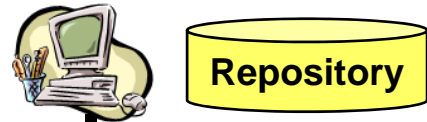
- **What is the solution for these problems?**
 - Use **“Reference Model Management”** Feature of JUDE/Professional
 - Use **Version Control System**

Scenario of Team Development with JUDE+ Version Control System

1. The total image
2. How to import reference files
3. Run Version Control System
4. Making changes and committing
5. Update changes
6. Commit History and Version Control

1. The total image

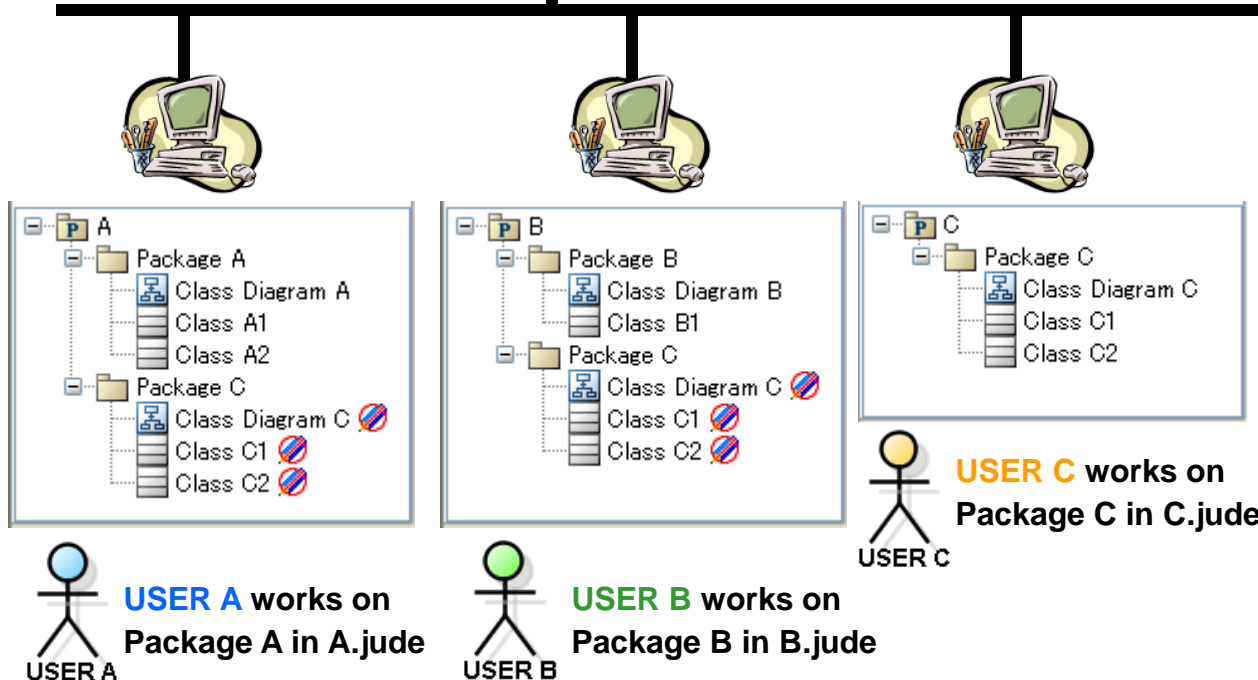
A shared server that has the version control system



Version Control System Tools:

i.e. CVS, SVN, VSS...etc

- ◆ track various versions of files
- ◆ record histories of source files
- ◆ do the File locking...etc



JUDE :

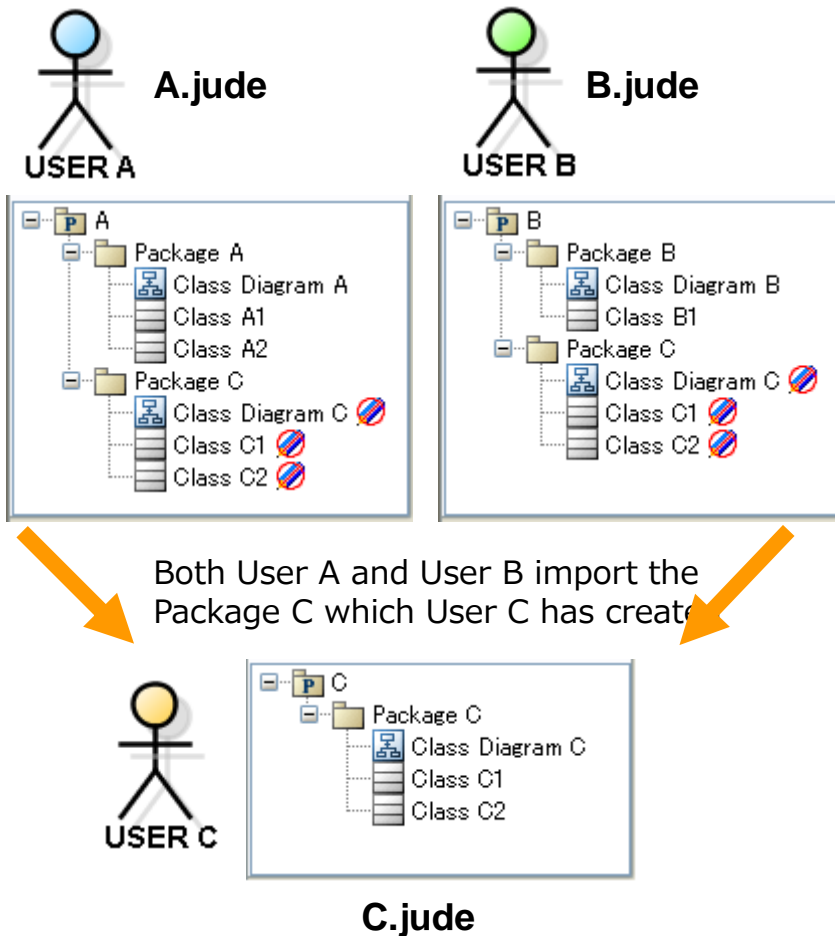
◆ helps you update associated files when they are modified by other members, so everyone can work with the latest and most accurate models always.

◆ Does not allow modification of models if they are not part of your share.

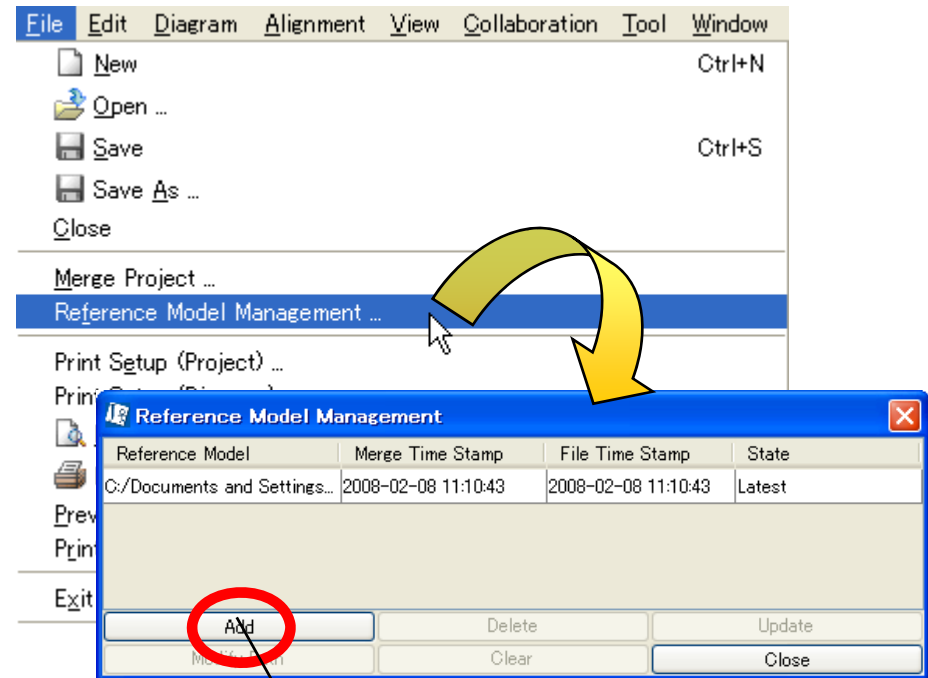
User A works on A.jude and is referring models of C.jude that User C works on and so does User B with B.jude.

Seeing is understanding.

2. How to import reference files



1. Open A.jude in JUDE/Professional
2. Select [File] – [Reference Model Management] then add “C.jude”



Add “C.jude”

User B can add “C.jude” in the same way.

Seeing is understanding.

3. Run Version Control System



1. Set up a version control environment
 - CVS, SVN, VSS..etc
 - Server and Clients
2. Add all JUDE project files to version control
3. Setup file locking or reserved checkouts

When more than one person works on a project, often two people try to edit the same file simultaneously. Also since JUDE project files (.jude) are in binary file format, it needs to have a solution to avoid having multiple people modify the same file at the same time.

Note) The solutions are different depending on what system you use.

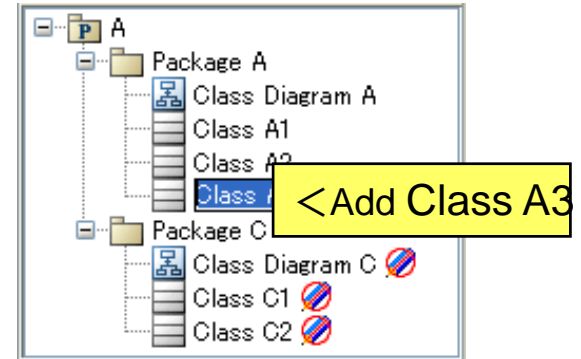
4. Checkout the project file

* Please see the instruction of the Version Control System you use for details.

4. Making changes and committing

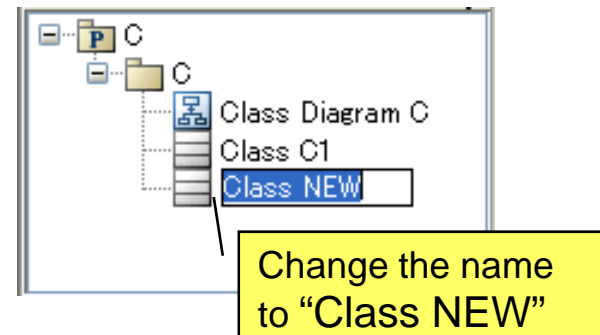
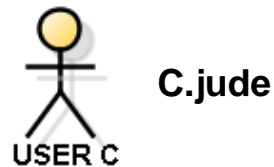
User A opens “A.jude” and edits as described below.

1. Obtain a lock to edit “A.jude” from Version Control Tool.
2. Open “A.jude” in JUDE, then add a new class “Class A3” then save the file.
 - ✧ Since reference models are read-only, it avoids having User A accidentally modify reference models (models under Package C).
 - ✧ User A can easily set up Attribute Types etc. by choosing reference models for new class.
3. Commit the updated file of A.jude to the repository.
 - ✧ Enter descriptive comments when committing.



While User A is making changes, User C edits their part at the same time, like this:

- ✧ Change name of one of Classes to “Class New” in C.jude then commit.

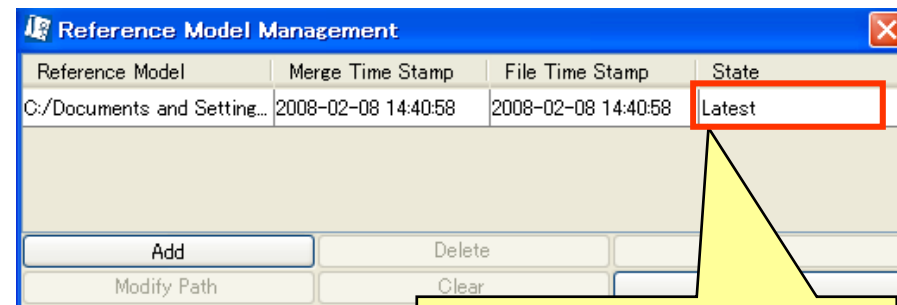
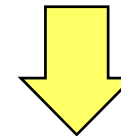
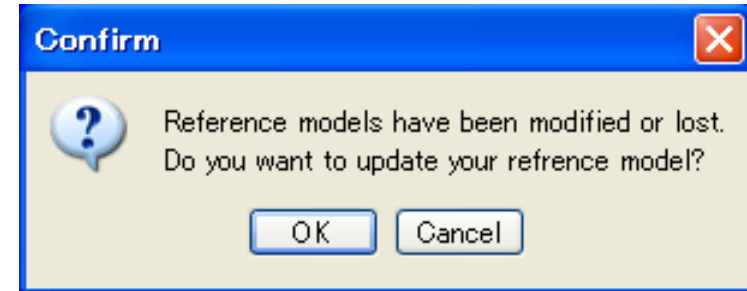


Seeing is understanding.

5. Update changes

Now User B updates the changes that User A and User C made.

1. Check if there are any changes in other member's files by using the Version Control tool.
2. Update A.jude and C.jude to the latest.
3. Open "B.jude" in JUDE
 - ✧ A [Confirm] dialog shows up to inform that Reference models have been modified.
4. Update Reference models in JUDE
 - ✧ Select [Update] on the [Reference Model Management] dialog.
 - ✧ Contents of package C in B.jude are updated to the latest. (One of class's name is now shown as [Class New])
 - ✧ All attributes of models in B.jude will be updated also.



If the reference model is updated to the latest, the status says "Latest". If not, it says "Need Update".

6. Commit History and Version Control



By using Version Control System, you are able to...

- See the history of source files
- Restore previous versions of source files

Date	Revision Number	User	Comments	Modified files
2007.7.25 14:10	5	User B	Reflect the change of User C.	B.jude
2007.7.25 12:05	4	User C	Rename ClassC1 to ClassC2	C.jude
2007.7.23 9:30	3	User A	Add a new Class, "Class A3"	A.jude
2007.7.9 15:20	2	User X	Xxxxxx	Xxxxx
2007.7.7 10:00	1	User Z	Add new Project	Xxxxx

Seeing is understanding.

Summary Table



Functions	Merge Feature of JUDE/Professional	Reference Model Management Feature of JUDE/Professional + Version Control System Tool	Collaboration with JUDE/Server
Can I use multiple files for one project?	YES	YES	No, there must be only one file for one Project
Does it support Users/Groups Management?	N/A	Yes, by the Version Control Tool	YES
Can I import other member's files as read-only?	N/A	YES	N/A
Does it inform me when other member's files have been updated?	N/A	YES	N/A
What is the unit of checkout/checkin?	N/A	Per file	Per Package
What is the unit of Administrator?	Per file	Per Project and per file	Per Project and Per Package
What is the unit for File Locking?	Per file (.jude)	Per file by the Version Control Tool	Per package
Can I view the history of source files?	N/A	YES	YES
Can I restore previous versions of source files?	N/A	YES	N/A
Does it have snapshot backups?	N/A	N/A	YES
Remarks	Technical Support is provided. It's hard to maintain consistency.	Technical Support is provided.	JUDE/Server is a non-support product

Seeing is understanding.

Reference Information

- Merge Utility for hierarchy Reference Projects
 - Total Merge Utility
 - Update Reference models with BAT or SH file.
- Compare differences between two project files via XMI
 - Export project files into XMI files, then compare them
 - Compare differences from [Flexible Merge] Dialog with detailed merge
- Team modeling without using Version Control tools
 - You can make your Team development work by using shared files and setting Administrations to them