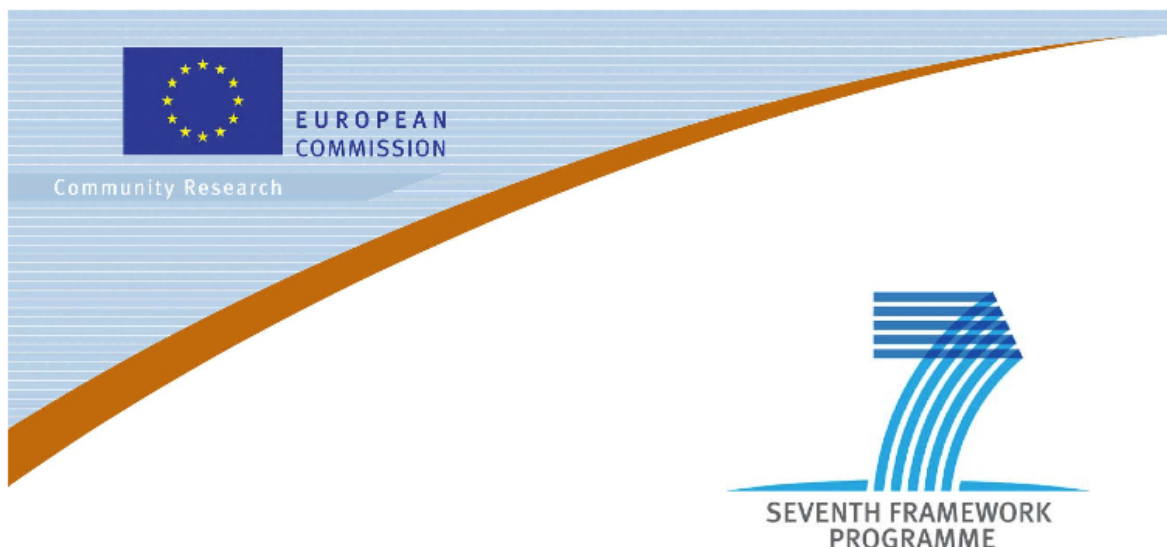


DevComE Basic Framework front page



Large-scale Integrated Project (IP)



D9.1: DevComE Basic Framework

Project acronym: FI-WARE

Project full title: Future Internet Core Platform

Contract No.: 285248

Strategic Objective: FI-ICT-2011.1.7 Technology foundation: Future Internet Core Platform

Project Document Number: ICT-2011-FI-285248-WP9-D9.1b

Project Document Date: 30 May 2012

Deliverable Type and Security: Public

Author: Matteo Melideo (ENG)

Contributors: Davide Dalle Carbonare (ENG), Jesús Gorroñogoitia Cruz (ATOS), Pasquale Vitale (ENG)

Abstract: This deliverable provides a subset of reference tools to support the development of Future Internet applications based on the FI-WARE Generic Enablers and using a collaborative development environment.

Keyword list: FI-WARE, PPP, DevComE, Collaboration, Development, FusionForge, Eclipse, Mylyn

Contents

Articles

DevComE Basic Framework front page	1
DevComE Basic Framework	2
FusionForge connector	3
CDE Management	4
Eclipse Communication Framework	9
Wiki Edit support	14
Source Code Management	15

References

Article Sources and Contributors	18
Image Sources, Licenses and Contributors	19

DevComE Basic Framework

Introduction

The original purpose of the task 9.1 associated to the current deliverable D9.1.b, as stated into the DoW, was to build *a new generation forge specifically designed for managing the development of FI Applications*. Since the first moths of the project this goal has been changed and turned into a supporting environment, integrated into the IDE, that enable developers to interact with a selected forge features directly from the IDE. This change of task 9.1 goal has been discussed with the project coordination, and accepted by the project coordination and the EC.

Software Components

This deliverable is mainly composed by software prototypes that are listed here and briefly described. More information on the tools, together with how to obtain it, are available from each tool-specific section.

FusionForge connector

The FusionForge connector supports the Tasks and Tickets management functionality for the Eclipse Mylyn plug-in.

CDE Management

This Eclipse plug-in offers wizards to create and delete Fusion Forge projects and create users from the FI-WARE IDE.

Eclipse Communication Framework

The scope to adopt this plug-in is to support developer to share the same piece of code and edit it simultaneously.

Wiki Edit support

This component adds to the FI-WARE IDE the support of writing documents using the wiki paradigm.

Source Code Management

This client allows developers to easily interact with the source code repository directly from within the FI-WARE IDE.

FusionForge connector

Introduction

The FusionForge connector provides the Tasks and Tickets management functionality for the Eclipse Mylyn ^[1] plug-in. A detailed documentation is available from the official download site as reported below. Together with the binary packages, also the source code is available for download. Note that the FusionForge connector is released under the open source licence EPL v1.0 ^[2].

For the list of known issues or to provide feedback on this plug-in, use the on-line Bug tracker ^[3] and select "**FusionForge connector**" as Asset option.

Information

Name: FusionForge Connector

Version: 1.0.0

License: EPL v1.0

Documentation: FusionForge Connector User Manual 1.0.0 ^[4]

Download: Binaries and Source Code ^[4]

Scope: Ticket and Task management

References

[1] <http://www.eclipse.org/mylyn>

[2] <http://www.eclipse.org/legal/epl-v10.html>

[3] https://forge.fi-ware.eu/tracker/?func=browse&group_id=15&atid=208

[4] https://forge.fi-ware.eu/frs/?group_id=15

CDE Management

Introduction

This user manual provides information about how to install and use the plugin that provides FI-WARE IDE Wizards for CDE Management. This set of plugins is designed to work within the Eclipse IDE ^[1]. The development is done in the context of the FI-WARE ^[2] project.

Current release offers wizards to create and delete Fusion Forge projects and create users from the FI-WARE IDE. The functionality to manage existing Fusion Forge projects is on development and will be added to the next release.

Together with the binary packages, also the source code page is available. Note that the FusionForge connector is released under the open source licence EPL v1.0 ^[2].

Information

Name: FI-WARE IDE Wizards for CDE Management

Version: 0.1.0

License: EPL v1.0

Download: Binaries ^[4] Source Code ^[3]

Scope: CDE Management

Installation

This plugin requires to install and configure the Fusion Forge Connector as explained in FusionForge Connector User Manual ^[4]

Copy plugin jar ^[5] into the eclipse plugins directory. In next releases, an Eclipse update site will be provided allowing end users to install the complete FIWARE IDE.

How To

This section explains the usage of the CDE Management wizards shipped within the FIWARE IDE

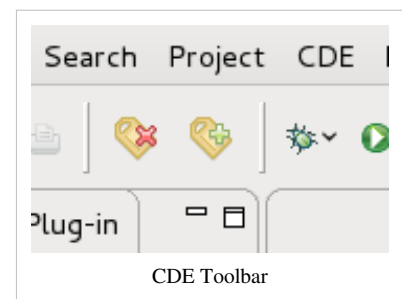
Configure Fusion Forge (FF) Repository

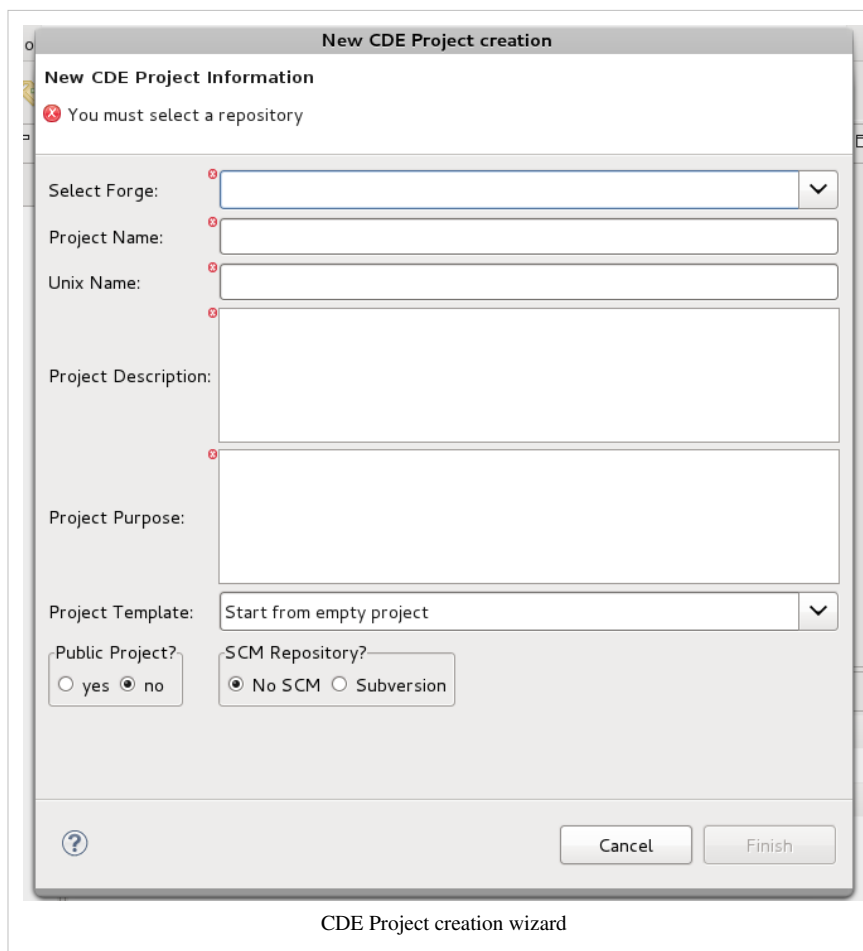
Before using the wizards, a Fusion Forge repository must be defined following instructions provided by Fusion Forge connector as explained in FusionForge Connector User Manual ^[4]

Create a CDE project

CDE wizards are accessible from the Eclipse main top toolbar, or from the Eclipse main top CDE submenu.

To create a new CDE project select that option in the CDE submenu or click on the corresponding action (icon with green + sign) in the toolbar. Then the new CDE project creation wizard pops up.





The image shows a 'New CDE Project creation' dialog box. At the top, it says 'New CDE Project Information'. Below this, there is a red error icon and the text 'You must select a repository'. The form contains several fields: 'Select Forge:' with a dropdown menu, 'Project Name:', 'Unix Name:', 'Project Description:', and 'Project Purpose:', all of which have red error icons next to them. Below these is 'Project Template:' with a dropdown menu showing 'Start from empty project'. At the bottom, there are two groups of radio buttons: 'Public Project?' with 'yes' and 'no' options, and 'SCM Repository?' with 'No SCM' (selected) and 'Subversion' options. At the bottom right, there are 'Cancel' and 'Finish' buttons. A help icon (?) is at the bottom left.

New CDE Project creation

New CDE Project Information

✖ You must select a repository

Select Forge: [dropdown]

Project Name: [text box]

Unix Name: [text box]

Project Description: [text box]

Project Purpose: [text box]

Project Template: Start from empty project [dropdown]

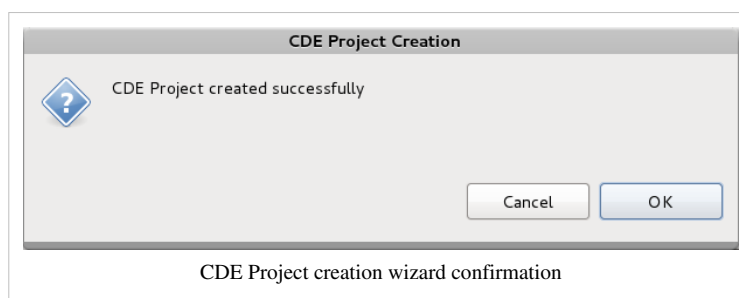
Public Project? ☐ yes ☒ no

SCM Repository? ☒ No SCM ☐ Subversion

Cancel Finish

CDE Project creation wizard

Select Fusion Forge repository and provide a project name, unix name (project identifier), project description and project purpose as mandatory fields. Once all required properties have been fulfilled, the Finish button is enabled. Optionally, the user can select a project template from which the new project will take its configuration, set project visibility (public, private), and the usage of an SCM repository. Click it to finalize. Upon creation, the user is notified.



The image shows a 'CDE Project Creation' confirmation dialog box. It has a blue diamond icon with a question mark. The text inside says 'CDE Project created successfully'. At the bottom right, there are 'Cancel' and 'OK' buttons.

CDE Project Creation

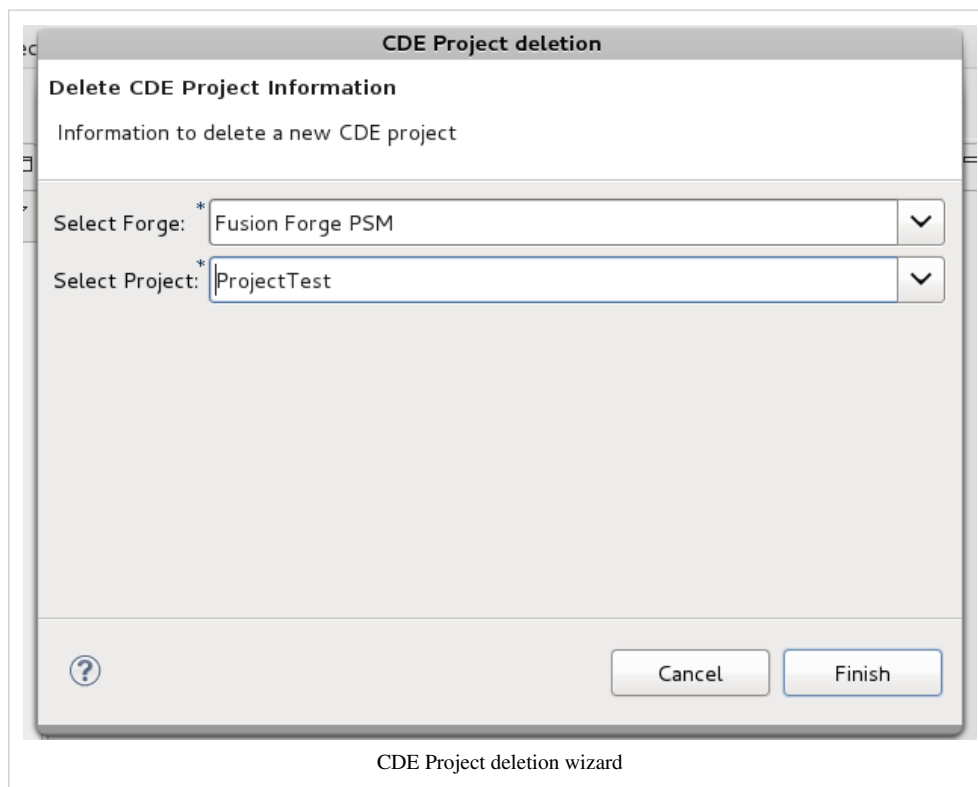
CDE Project created successfully

Cancel OK

CDE Project creation wizard confirmation

Delete a CDE project

To delete an existing CDE project select that option in the CDE submenu or click on the corresponding action (icon with read x sign) in the toolbar. Then the CDE project deletion wizard pops up.



Select the Fusion Forge repository and the project to delete. Confirm clicking of finish button. After a while a confirmation dialog pops up.

Create a CDE user

To create a new CDE user select that option in the CDE submenu. Then the new CDE User creation wizard pops up.

New CDE User creation

New CDE User Information

✖ Login name cannot be empty (only lower case);

Select Forge: * Fusion Forge PSM

Login Name: ✖

Password: ✖

Password (repeat): ✖

Title:

First name: ✖

Last name: ✖

Email Address: ✖

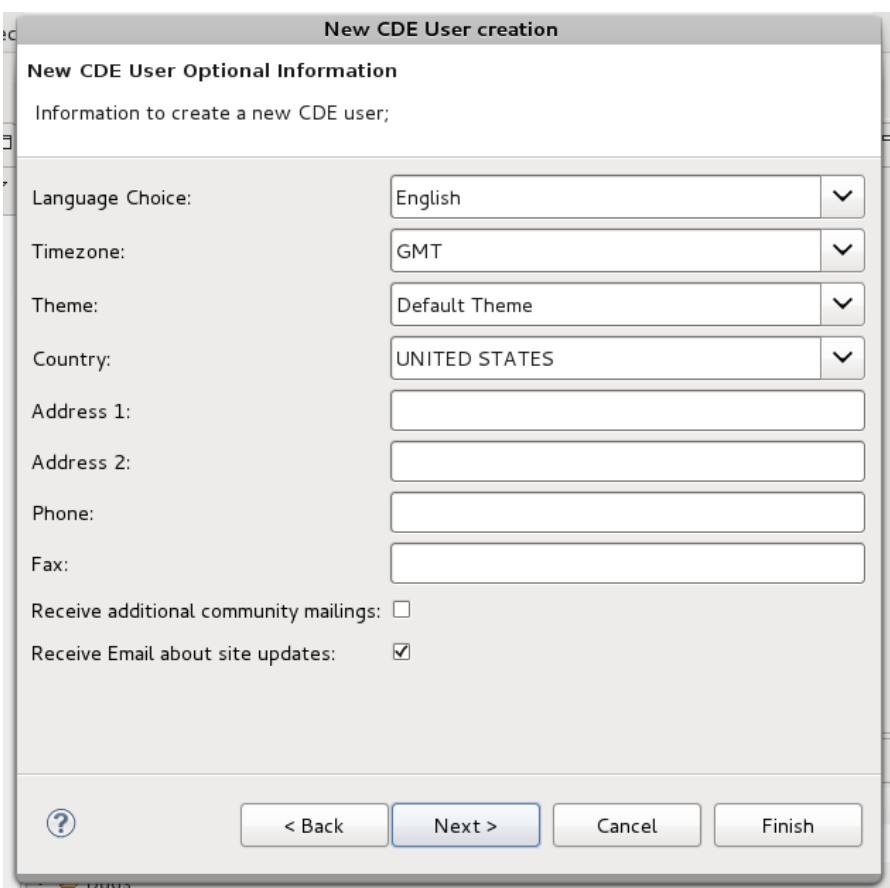
? < Back Next > Cancel Finish

First page of the CDE User creation wizard

Fulfill the mandatory fields in the first page:

- Select the Fusion Forge repository
- unique login name
- password (repeat, minimum 6 characters)
- Title (Mr, Ms, etc) - optional
- First and last name
- Email address

Once all required properties have been provided, the finish button is enabled. The user can either finished the wizard or pass through the next page to fulfill optional properties.



New CDE User creation

New CDE User Optional Information

Information to create a new CDE user;

Language Choice: English

Timezone: GMT

Theme: Default Theme

Country: UNITED STATES

Address 1:

Address 2:

Phone:

Fax:

Receive additional community mailings: ☐

Receive Email about site updates: ☒

? < Back Next > Cancel Finish

Second page of the CDE User creation wizard

Close the wizard by clicking on Finish button. Once the user creation action is complete the user is notified.

References

- [1] <http://www.eclipse.org>
- [2] <http://www.fi-ware.eu>
- [3] <https://forge.fi-ware.eu/scmrepos/svn/tools/trunk/software/eu.fiware.ide.cde/>
- [4] <https://forge.fi-ware.eu/frs/download.php/24/FusionForge%20Connector%20User%20Manual.pdf>
- [5] https://forge.fi-ware.eu/frs/download.php/25/eu.fiware.ide.cde_0.1.0.jar

Eclipse Communication Framework

Introduction

The scope to adopt this plug-in is to support developer to share the same piece of code and edit it simultaneously. The scenarios covered by this feature are for example: collaborative development, where two developers can write on the same file at same time; support/teaching, when it would be easier from a member (of a distributed team) to ask and receive support on a piece of code written by another member; peer review, as supporting tool that improve the effectiveness of this practice. The effective adoption of this plug-in depends then on the specific context in which the project is developed.

Next chapters, briefly, describes the installation steps and gives a more detailed idea of the typical usage of the plug-in. At the end is proposed an extension of the Collaborative Development Environment in order to have a better and more effective integration with this collaborative development service.

Information

Name: Eclipse Communication Framework ^[1]

Version: 3.5.4

License: EPL v1.0

Download: Eclipse Marketplace

Documentation:

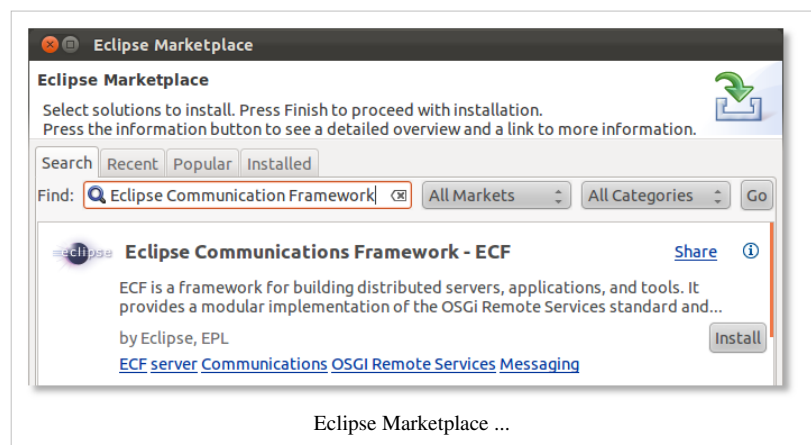
- Official Website ^[2]
- DocShare Plugin ^[3]

Scope: The main goal for adopting this plug-in is to support the developers in sharing the same piece of code and edit it in a real-time and collaborative way.

Installation

This plug-in can be installed directly from the Eclipse IDE opening the wizard at *Help -> Eclipse Marketplace* ... and searching for *Eclipse Communication Framework* (see image). Install the ECF component following the guided steps.

The official web page of this plug-in is: Eclipse Communication Framework ^[1]



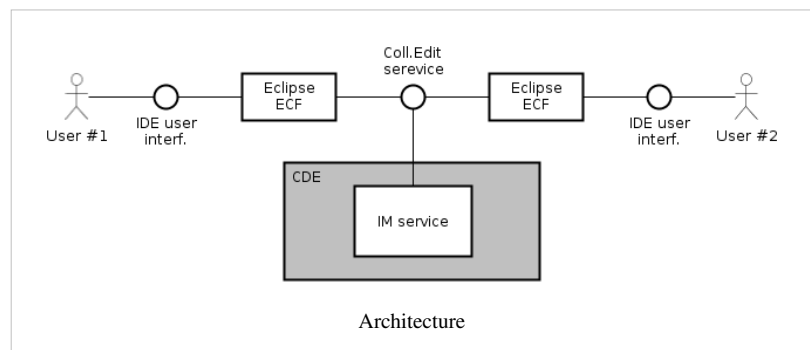
Requirements

The users that want to work with ECF plug-in have to be registered to an Instant Messaging service (IM service) compatible with the ones supported by ECF. A user can start a Collaborative Session with the users on his/her contacts list that have ECF installed.

Architecture

The architecture diagram (see image) represents the main components involved in order to have the collaborative development feature available. The Instant Messaging service (IM service) is considered part of the CDE even if it's provided by third parties.

The two users involved in the collaboration session shares the same code interacting with their own Eclipse IDEs. The information exchange is controlled by the IM service through which the users are connected.



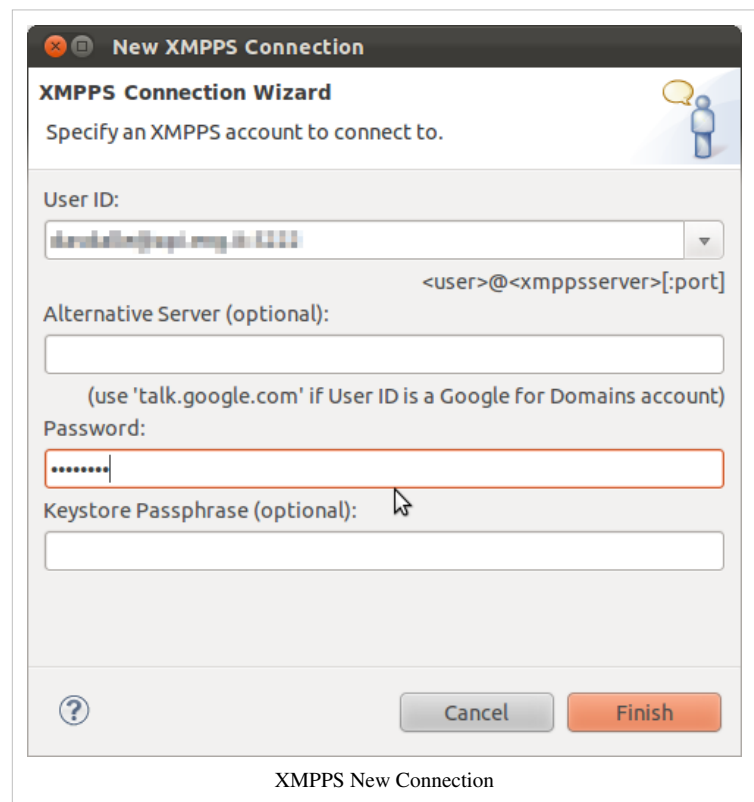
How To

This chapter contains operative indications on how to execute specific tasks.

Configure IM account

The first step to have the plug-in features available (after the installation), is to configure an existing IM account. From the Eclipse IDE open the *Communications* Perspective and again open the *Contacts* View panel.

According to the selected IM type the proper fields are prompted to the user (see image for an example). Press Finish to retrieve the user contact list (see image for an example).



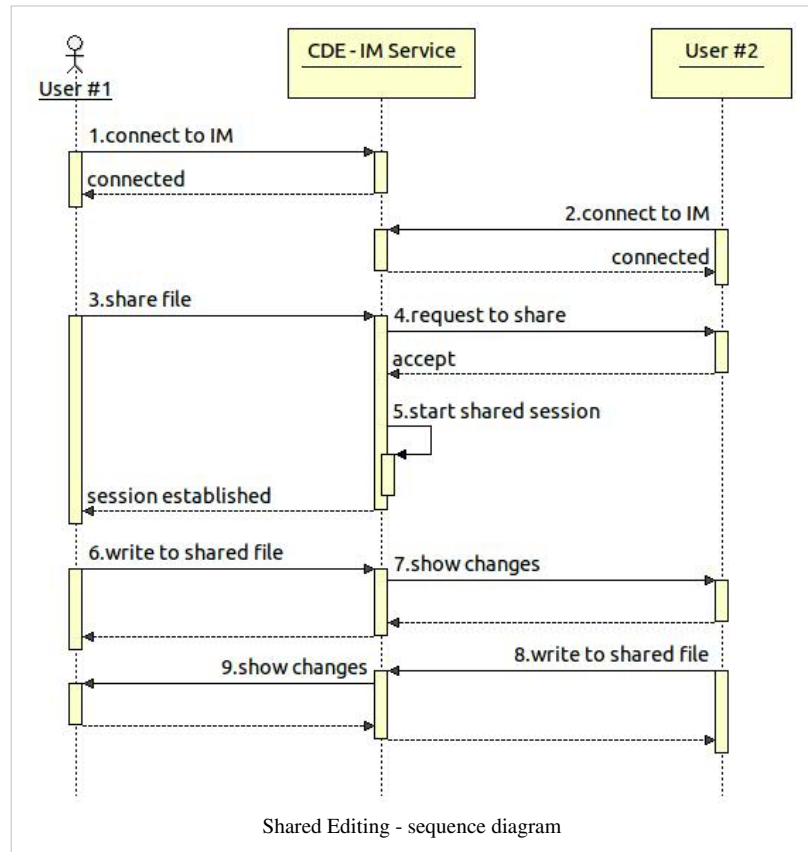
Shared editing

The Eclipse Communication Framework allows to share the editing of a file with another person. This feature comes out of the box once the plug-in is correctly installed and configured. The preconditions to enable the shared editing of the same file are:

- both parties
 - are connected to the same IM service;
 - are running Eclipse and have the ECF properly installed and configured;
- each one belongs to the contact list of the other;
- have the same project on the workspace.

User #1 open the file to share with User #2. From the contextual menu (on the opened file) select *Share Editor With ...* and from the sub-menus select

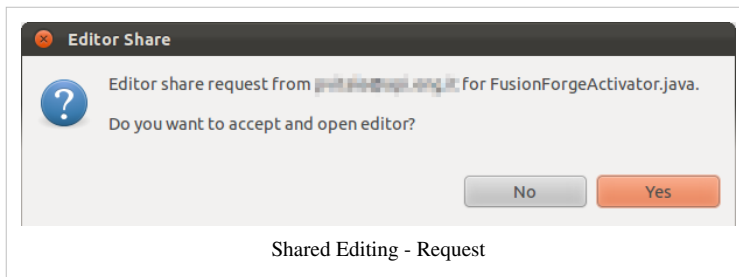
the contact person you what to share the edit with. Once selected, the contact, User #2, receives a notification message to accept or reject the request from the User #1. If the User #2 accepts, a temporary copy of the shared file is opened on User #2 editor, and both, User #1 and User #2, can start typing simultaneously. All the changes on the text that are done by one party are reflected in real time to the other party, included cursor position and text selections. The sequence diagram Shared Editing represents the interaction steps between the users and the IM service.



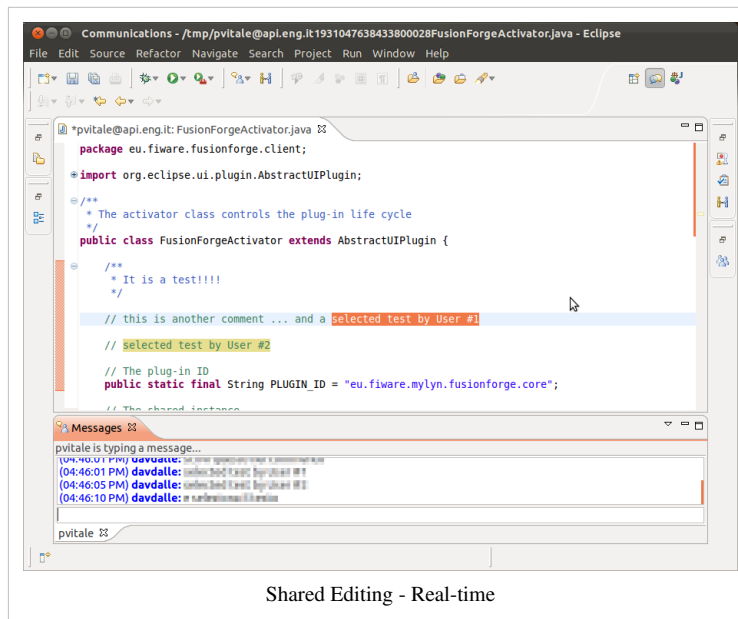
Once the connection is established, the two users can chat using the Message View. (see the image below)



When User #1 requests for a shared editing session, User #2 receives a pop-up message with that request. (see the image below)



When the User #2 accepts the request to share the editing of a file, the two users can write on the same file and receive the updates in real-time, cursor position and text highlight, included. (see the image below)



Improvements

This chapter proposes how to improve the Collaborative Editing integration into the DevComE. The sequence diagram represents a complete process that goes from the user subscription (to a project) to sharing a file for collaborative editing with another member of the project.

The entire diagram can be split in two main parts (referring to step numbers):

from 1 to 3

additional steps to extend the process of joining a project

from 4 to 12

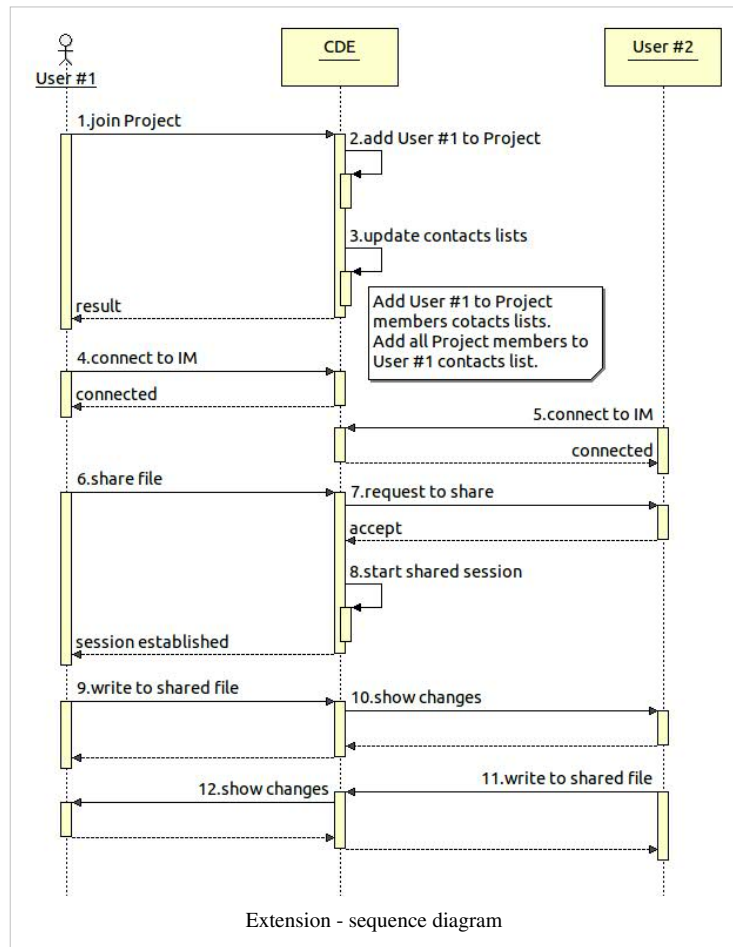
usual process steps to share and edit a file with a contact.

The proposed improvement consists in some additional steps executed when a requests, to join a project on the CDE (e.g. forge) from a user, is accepted. These additional steps, as reported on the sequence diagram, consist on the automatic subscription of a new member to the contact lists of the other team members. The new member contact list will be also updated with the contacts of the projects' members already subscribed. If necessary the IM service account for the user is created.

The main benefits drawn from this extension are basically:

- all team members of a project receive an IM account;
- all team members can directly get in contact with each other;
- define one contact group per project (*);
- use one single IM service avoiding different accounts and tools;
- share a chat room dedicated to the project (*);
- keep the log of all the conversation as additional knowledge base;
- include into the cross search service the conversation logs.

(*) Depending on the specific implementation of the IM service it's possible to include into the extension process the automatic set up of a chat-room shared by all the team members and a new contact group that contains all the project members.



References

- [1] <http://www.eclipse.org/ecf>
- [2] <http://www.eclipse.org/ecf/documentation.php>
- [3] http://wiki.eclipse.org/DocShare_Plugin

Wiki Edit support

Introduction

In the context of the project it's proposed to adopt a specific plug-in, Mylyn WikiText, that adds to the Future Internet project under development the support of writing documents using the wiki paradigm. At the moment a relevant set of wiki dialects are already supported (MediaWiki, Textile, Confluence, TracWiki and TWiki), anyway, the architecture of the plug-in allows to develop additional ones and extend the list.

Information

Name: WikiText ^[1]

Version: 1.6.0

License: EPL v1.0

Download: Eclipse Marketplace

Documentation: Mylyn WikiText User Guide ^[2]

Scope: To enable the FIApp project the writing of documentation using the wiki paradigm.

Installation

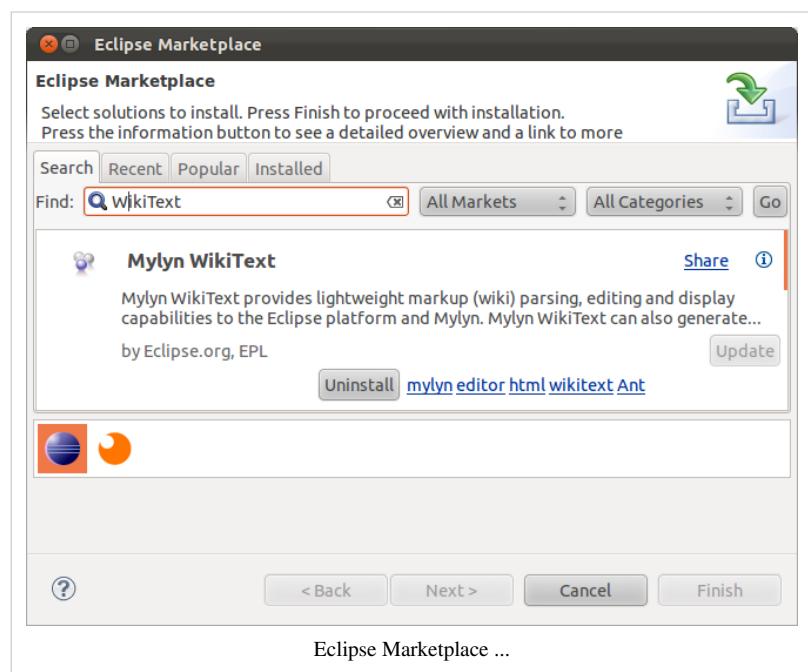
This plug-in can be installed (or updated) directly from the Eclipse IDE opening the wizard at Help -> Eclipse Marketplace ... and searching for Subclipse (see image). Install the Subclipse component following the guided steps.

Requirements

The WikiText requires some Mylyn components that are automatically resolved and installed by the guided install procedure.

How To

All the operative information on how to start using WikiText for writing wiki contents can be found at the Getting Started section of the User Guide ^[2].



Improvements

A possible improvement to be implemented into the Mylyn WikiText plug.in is the possibility to connect to a remote wiki installation and edit those wiki pages directly from the Eclipse IDE. A relationship with the Mylyn community of developers is under development.

References

[1] <http://wiki.eclipse.org/Mylyn/Incubator/WikiText>

[2] <http://help.eclipse.org/helios/index.jsp?topic=%2Forg.eclipse.mylyn.wikitext.help.ui%2Fhelp%2FMylyn+WikiText+User+Guide.html>

Source Code Management

Introduction

One crucial aspect to be addressed when dealing with collaborative development is the management of the source code. To help on this, it's possible to adopt a solution identified as Source Code Management (SCM) system. There are various solutions available on the market, at the right moment (e.g. Subversion, Git); the one that has been selected, as first reference implementation, is Subversion ^[1] together with one of its Eclipse clients, Subclipse ^[2]. This client allows developers to easily interact with the source code repository directly from within the Eclipse IDE. Apart from the basic features (check out, commit, update), also the most complex ones (manual conflict resolution, branch, tag, merge) are supported, out of the box, with graphical and intuitive user interface.

Starting from the assumption that the selected reference implementation of the forge solution is FusionForge, we consider that the Subversion server is already available (see FusionForge and Subversion installation manuals respectively).

Information

Name: Subclipse ^[2]

Version: 1.6.x

License: EPL v1.0

Download: Eclipse Marketplace

Documentation: Official Website ^[2]

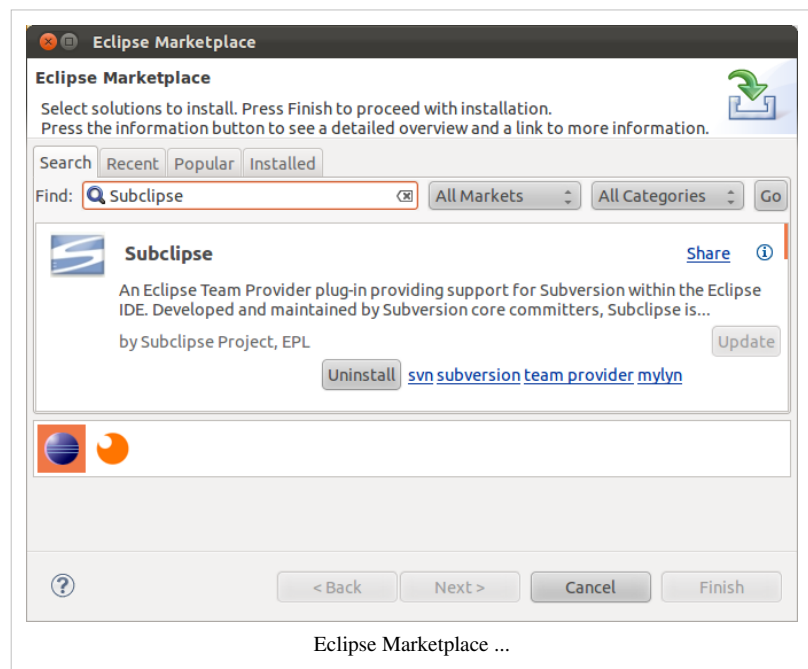
Scope: Source Code Management

Installation

Before to proceed with plug-in installation it's important to obtain the version of the Subversion server that will be used. This is useful in order to select the proper version of the Subclipse client.

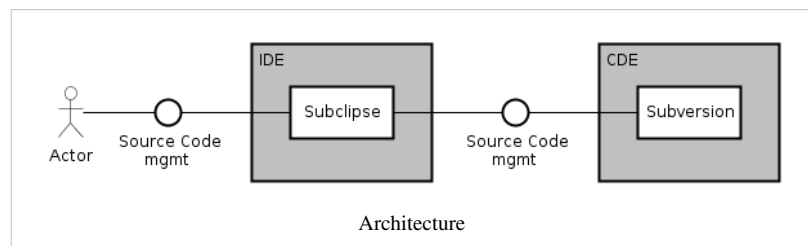
This plug-in can be installed (or updated) directly from the Eclipse IDE opening the wizard at *Help -> Eclipse Marketplace ...* and searching for *Subclipse* (see image). Install the Subclipse component following the guided steps.

After the installation it may be useful to take a look at the FAQ ^[3] page for additional information on specific usage or constraints (e.g. svn+ssh protocol).



Architecture

The architecture diagram (see image) represents the main components involved and their interactions. The plug-in provides to the developer a single contact point to interact with the source code manager system.



How To

Here are reported the features that are mostly used during a traditional development process. Additional features and details are reported on the official documentation ^[2].

Connect to SVN Repository

Once the plug-in is installed, the first action to do is to configure an SVN Repository to connect to. This allows to browse the repository content and checkout projects or contents in general.

Synchronize the code

Update and Commit operations are well supported by the plug-in. Thanks to the Team Synchronizing perspective, and the Synchronization view, it's easy to check what are the resources that has been modified locally and remotely (by other users). It highlights the conflicts detected in order to take care of those and solve them before committing the updates.

Tag, Branch, Merge

All these three basic operations are fully supported by the Subclipse plug-in and it's possible to execute them with dedicated wizards. It's worth to note that the Merge operation can benefit from the "Revision graph" graphical representation in order to simplify the identification of the right revisions and ranges to merge.

References

- [1] <http://subversion.apache.org>
- [2] <http://subclipse.tigris.org>
- [3] <http://subclipse.tigris.org/wiki/PluginFAQ>