A4ALL

Assisi Plugins for Eclipse

V4ALL 

Rapid Application Development for Eclipse
GUI-Design and Codegenerator
Extentable Components (Factory Support)
XML Runtime Support
XML Database
CORBA support (Code Generation)

User Manual

English

Ramin Assisi
Assisi2000@yahoo.com

Release 0.1 last update 10/02/23

Version 0.00002

The english will be reviewed completing the input to this document.

Content

About this manual

Introduction

Preface

Using this Book

-
-

Conventions used in this Book

-

Request for Comments

Feedback

I do everything to explain V4ALL. Though his book will not be fully usable before version 1.1, I made it available to the eclipse community already in a very early date. I that spirit I would like much to hear your comments, corrections or criticism. You can contact me at assisi2000@yahoo.com.

CHAPTER 1
Introduction

Challange

CHAPTER 2
Concepts

Abstract GUI-Design

Goal

GUI-Design-Tools are usually specific to the target API. The set of supported components corresponds there with that API. Many GUI-Developers would like to design a User Interface once and reuse that design for generating code which can be run under different Operating systems and under different target APIs.

The goal of V4ALL is to support this idea of portable code.

Overview

CHAPTER 3
Installation

General

Read carefully. Even if you have already installed before.
This information changes from time to time because of rapid changes in the program during implementation.

Prerequisits

1. Eclipse 2.1 or higher
2. GEF-Version as defined in the downloadpage ww.assisiplugins.com/downloads

Installation of Eclipse

the plugin needs eclipse 2.1 (M4) and higher. It was tested only for Windows (NT/XP). It is clear that it will later also work for LINUX and others and will be tested for them

Installation of GEF

unzip GEF-SDK-I20030128.zip (or the version which is required) to your eclipse-root-directory and start (restart) eclipse. It is very important, that you use the provided version of GEF, because the developers of GEF make still significant changes, which are incompatible to previous versions.

Installation of examples

- unzip v4all_examples_xxx.zip to your eclipse-root/workspace directory.
- import the project: import/existing project and choose eclipse-root/workspace/com.assisi.v4all.examples.

Installation of runtime libraries

All examples for which v4all-XML-runtime-support is defined needs the following Jar-files in theire class-path:

Jdom.jar – This is a special extention of the open-source jdom-API

V4all_runtime.jar

Examples which defines this runtime support have these jar-files in there lib-folder.

Installation of V4ALL

unzip the v4all_xxx.zip file in the eclipse root directory
and start (restart) eclipse.

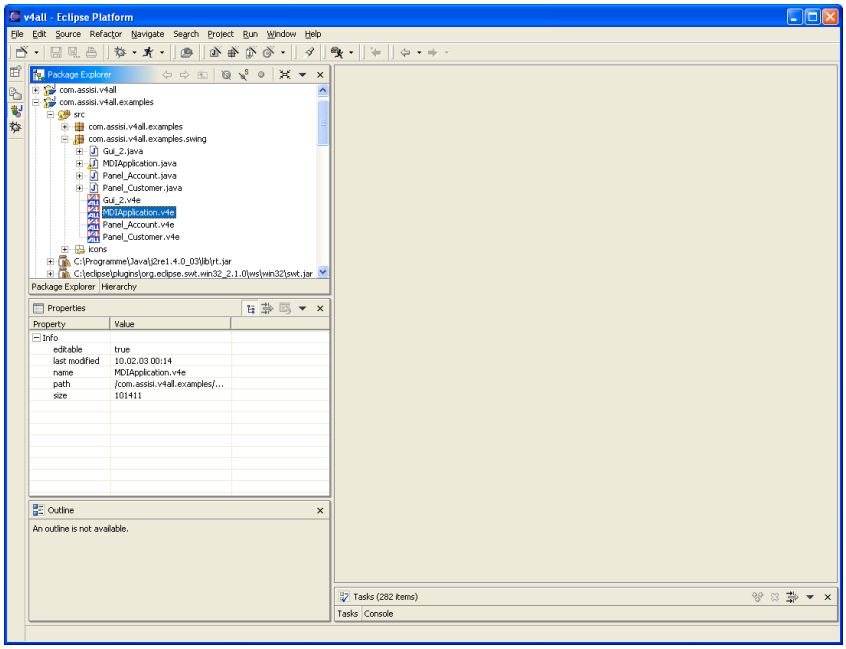
Create perspective

V4ALL will always try to open a perspective with the name „V4ALL“. This perspective should always contain a package explorer, a property view and an outline view. The best way to get such a view is to switch to the standard java perspective.

Now you can save this perspective with the Mneuitem Windows/Save Perspective As/ and typing "v4all".

Add to this perspective a Property page. And save it again under „v4all“.

The result



Choosing an existing V4ALL-resource

By double clicking on a v4all resource icon in the package explorer the editor will be launched

Change the Target API

If you click on the white board if the GUI editor, you see in the Property-window two properties (API type and router). You can switch between the different API´s.

Adding a component

You can add components to the white board by dragging the appropriate symbol from the palette view to the white board or to a already existing container component.

Trying the example

- download the folder "examples/com.assisi.v4e" to yout worspace-folder.
- import/import existing project/ -> choose eclipse/workspace/com.assisi.v4e
- change the perspective to java

CHAPTER 4
Configuration

CHAPTER 5
User manual

Setup a V4all Project

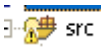
Create a java project

To use V4ALL you need first a java project. You can choose an existing one ore you crate a new one with the New Wizard.

V4ALL resources must always in a named package. The default package is not allowed.

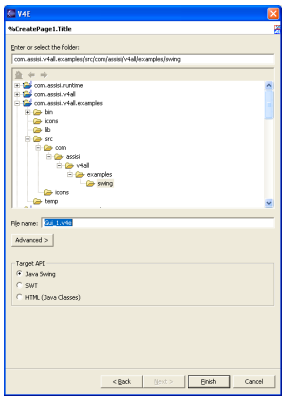
Another requirement is to have a Source folder with the name src. If you have not already one, you must create it by choosing New/folder from the context menu in the package browser.

This source folder must be defined in the java-build path: Choose properties of your java project from the context menu in the package explorer view. Choose java build path. On the Source Tab choose Add folder and add the src folder of your project. If you have confirmed, the icon of the src-folder should change to



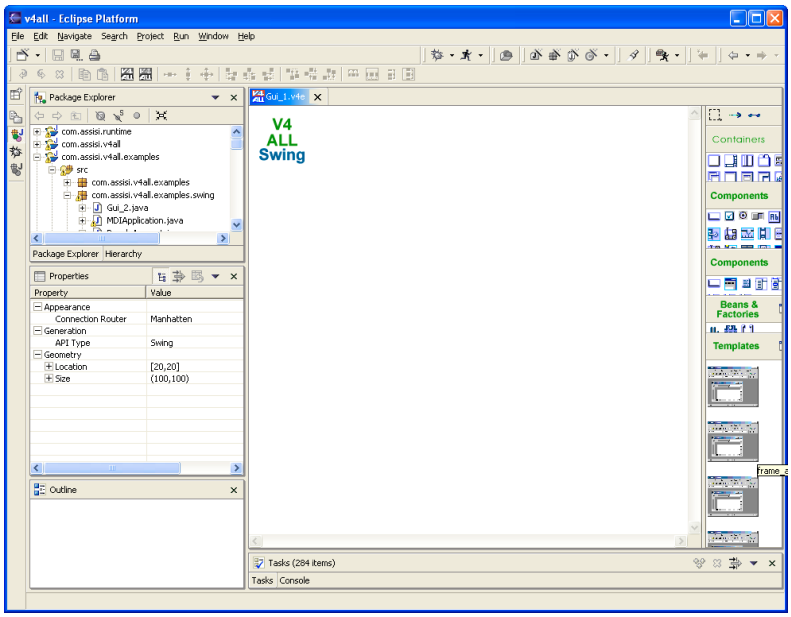
Create a new V4ALL-Resource

After selecting the package where you want to create the new V4ALL-resource, you can start the New Wizard (File/New/Project). You find a submenu item Java/V4E Gui Editor. After activation you get a Wizard, where you can define the name of the resource you want to create.



You can define here also, for which API you want to generate the code.

After you have confirmed your entries you should see the following sreen:



Choosing an existing V4ALL-resource

By double clicking on a v4all resource icon in the package explorer the editor will be launched

Change the Target API

If you click on the white board if the GUI editor, you see in the Property-window two properties (API type and router). You can switch between the different API´s.

Adding a component

You can add components to the white board by dragging the appropriate symbol from the palette view to the white board or to a already existing container component.

Create a „src“ Folder

Right Mouse Click in the package explorer, New / Folder

Add the „src“ Folder to your build path

- project/properties/Java Build Path/Source/Add existing folder/
- Use source folders in your project/
- Now choose your src folder

Switch to Java Perspective

Switch to the Java perspectice (not resource perspective!).
The package explorer must always be active during the GUI-session.

Create a Package

Right Mouse Click in the Package Explorer View

Creating & Editing a component

Adding components

You can drag the components from the palette view to the white board to add new components

Changing properties

You can change the properties of a components by selecting the component and change the value of a property in the property view.

Adding and editing an external bean

Adding

If you drag and drop a bean from the palette view to the white board, a type dialog opens and you have to choose an existing class. If the class was created by V4all and the v4e-resource was found, the program checks if there exists a main class. This indicates, that the choosen class is a visual one. In that case the bean will be viewed visually at the drop location.

If the component was not visually, or nor designed by V4all or it has no main component, the bean will be dsiplayed a bean symbol.

Using Layout Managers

Choosing a layout manager

In the property view you can choose a layout manager under the menu item container.

Defining the Main Component

Purpose

The generated class is considered as a visual component if one component is defined as the main class. This is the case, if it is the first component which was dropped onto the white board. It can also be manually defined or changed in the property view under the menu item Generation/Main Component. If no component has this property set to true the whole bean is considered as a non-visual component.

Adding User Code

Purpose

You might want to add user code to the generated code. This is possible, if you place it beween special tags:

```
// user code begin [target api]
// user code end
```

Dependend on your target API you must place it in the appropriate tag. During generation the other user codes, which do not belong to the current target API will be uncommented. The advantage is, that you can switch later to other target API´s and you can reactivate your user code belonging to that API.

The following example shows the generation for Swing

```
// user code begin {1} Swing
   System.out.println("Swing 1");
           // user code end
           // user code begin {1} SWT
   // V4ALL System.out.println("SWT 1");
           // user code end
```

Some user code areas do not have the target api specifier. For instance the exception handling. There a possible user code is for all target API´s.

The user code for fields and methods must placed in the following tags

```
/**
 * user code fields
 */
// user code begin FIELDS {1} Swing
// user code end
// user code begin FIELDS {1} SWT
// user code end
/**
 * user code methods
 */
// user code begin METHODS {1} Swing
// user code end
// user code begin METHODS {1} SWT
// user code end
```

Defining XML runtime support

Goal

If you intend to create a whole application you usually must solve a lot of technical problems, which are not directly a part of your business requirements.

- Easy and extentable Model-View-Controler implementation (MVC).
- „Intelligent“ Component for instance a special TextField which accepts only a valid security number of a person and it do display such a number in an appropriate manner.
- Writing code repeatingly for the same task, e.g. Buttons actions must be linked to a method in another class as part of the business logic.
- Programming of a progress bar.
- Manipulating the shape of components for all of them or for groups of them
- Displaying the object hierarchy in a tree component (navigator)
- …

All of these tasks become much more easier with the use of the v4all runtime library which is based on the XML technology.

Installation

To get the v4all runtime-support you must define a class-path to the v4all_runtime library. This library is provided with the v4all_example in its lib-folder.

Activation

To choose XML-runtime support you must click to the white board and change the property XML Runtime Support/Rutime Support to true.

During generation the generated class will be linked to the v4all_runtime.jar resource.

Features

Actions

If a component has a ActionCommand defined, during runtime the GUI will be scanned and a method with that name in a class which is derived from `com.assisi.v4all.bo.common.ProcessManager` will be called (See examle MDIApplication).

Domain Objects

Object Hirachies

Link to a Tree Component

Print Support

MSWord

Staroffice

CHAPTER 6

Tutorial

The MDIApplication-example

The MDIApplication-example

Appendix A

FAQ & Known problems

Nothing works	Check, that you have the Versions of Eclipse and GEF according the information of the download page
Swing does not generate correctly	solution: wait until final release
SWT does not generate, what I expect	SWT works in the moment only for very simple design: a frame with a panel with a button. It is only a study, to show the principal of work. SWT will have the second priority after Swing. Solution: wait until final release.
will v4all generate everything from the design?	No, because the goal is to generate from one design as much target APIs as possible. Sometime you have to add your user code in a simliar manner like in the "old" good VAJ. The difference will be, that tose code will be separated per API, so you can later switch between the targets without loosing user code. This is another approach as in other GUI-designers.
grid-bag layout shows in the desiner not every property (weight etc.)	Solution: run the generated code. Later versions will try to show already in the designer as realistic as possible. It has for the moment a lower priority. Code generation and principal working of functionallity is more important now.