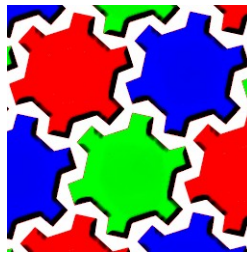


VisualAP 1.2.1¹

A framework for component based design

User guide



¹ 13 March 2011, author: Livio (javalc6@gmail.com)

This document is provided under the open content license: <http://www.opencontent.org/opl.shtml> or read file license.txt
This document has been written using OpenOffice.org Writer.

CONTENTS

Chapter	Page
1 VisualAp.....	3
1.1 Introduction.....	3
1.2 Basic Concepts.....	3
1.3 Release Notes.....	3
2 User Interface.....	4
2.1 Description.....	4
2.2 Details about available menu.....	7
3 Components.....	9
3.1 Introduction.....	9
3.2 Using components.....	9
3.3 Demo components.....	10
4 Installation and start of visualap.....	11
4.1 Windows XP, Vista, Windows 7.....	11
4.2 Linux and other OS.....	11
4.3 Start VisualAp application.....	11
5 Tutorial.....	12
6 Terminology.....	13

1 VisualAp

1.1 Introduction

VisualAp is a visual framework for building application and systems based on lightweight visual components, called proclets. Users can add their own components in order to extend the capability of VisualAp.

The VisualAp application can be used to perform audio processing, image processing, text and other process-driven emulation.

The user can create an application by selecting the components from a toolbox, configuring the parameters (via the Javabeans framework), and connecting the components together in order to set-up communication channels between the components.

In the first release the user runs code execution through the engine included in VisualAp, in interpreted mode. In later releases it will be possible to generate Java code that can be compiled and run in standalone mode.

1.2 Basic Concepts

VisualAp is based on two main concepts: system and components.

A component is the basic element providing some services. A system is build using a number of components that are configured and connected together in order to achieve a complex functionality.

A system is a graph whose nodes are the components.

Note that in VisualAp a component is also called proclet (processing element).

There are three type of components:

- source: any component that generates some sort of data
- sink: any component that consumes data
- processor: any component that process input data into output data

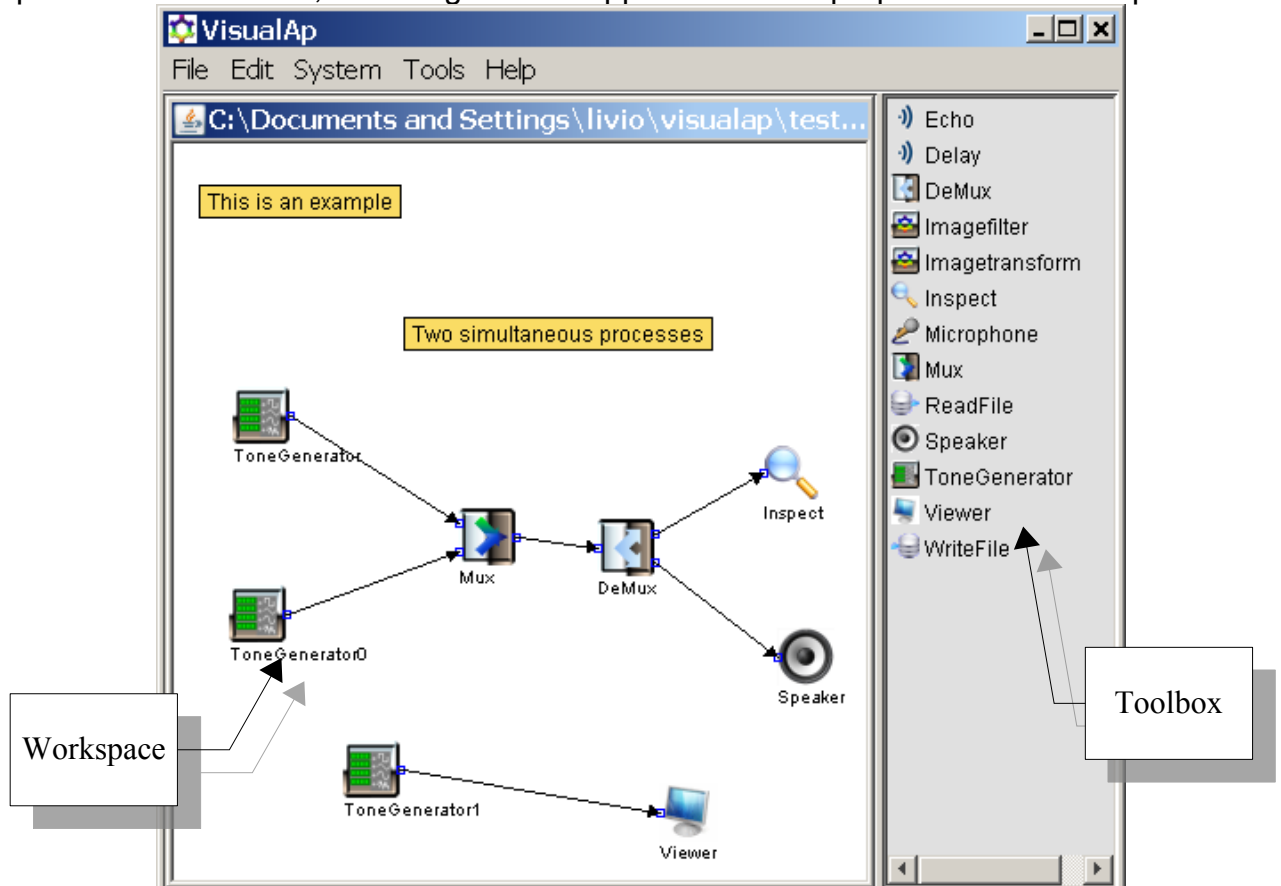
1.3 Release Notes

Version	Description
Version 1.2	Changed: minor enhancements in the user interaction, including drawing of direction arrows Corrected: minor compilation warnings with JDK 1.6, Corrected: integration issues with Windows Vista and Windows 7
Version 1.2.1	New: support for gif and png images, drag&drop support Enhancements: handling of properties, installation procedure Replaced: obsolete com.sun.image.codec.jpeg package

2 User Interface

2.1 Description

This program creates a graphic window with a menu, a workplace panel and a toolbox panel. When needed, a floating window appears to show properties of the components.



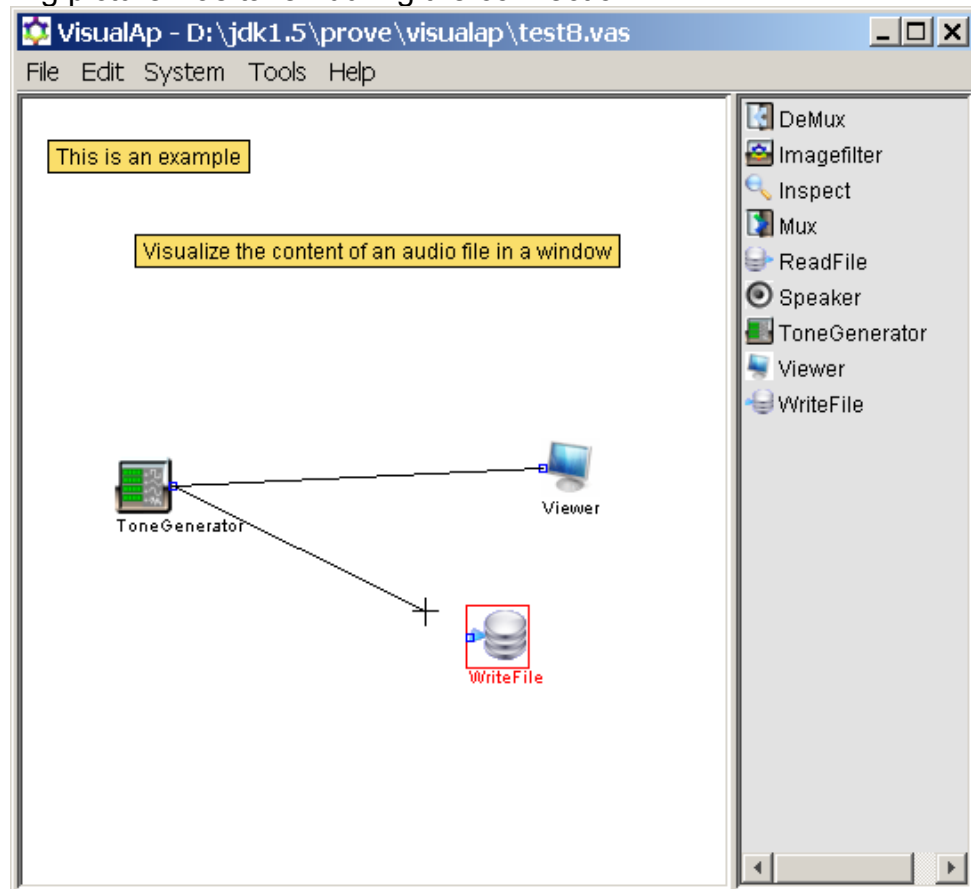
From the application menu it is possible to open a file dialog, print dialog, help window, preference dialog and an about box.

It is possible to create, move, edit any components in the **workplace** panel using the mouse buttons.

Components can be connected together by first pressing the mouse button over a terminal, then move the mouse pointer over another terminal and then release the mouse button: an edge between the two terminals is created.

The **toolbox** shows the available components (proclets) via their icons and names. The user can select a component by clicking the mouse pointer in the toolbox area. After releasing the mouse button, the cursor changes to cross-hair cursor indicating that a component was selected. Then you move the cross-hair cursor in the left panel and click again: when the mouse button is released, the selected component is placed in the workplace panel.

The following picture was taken during the connection:



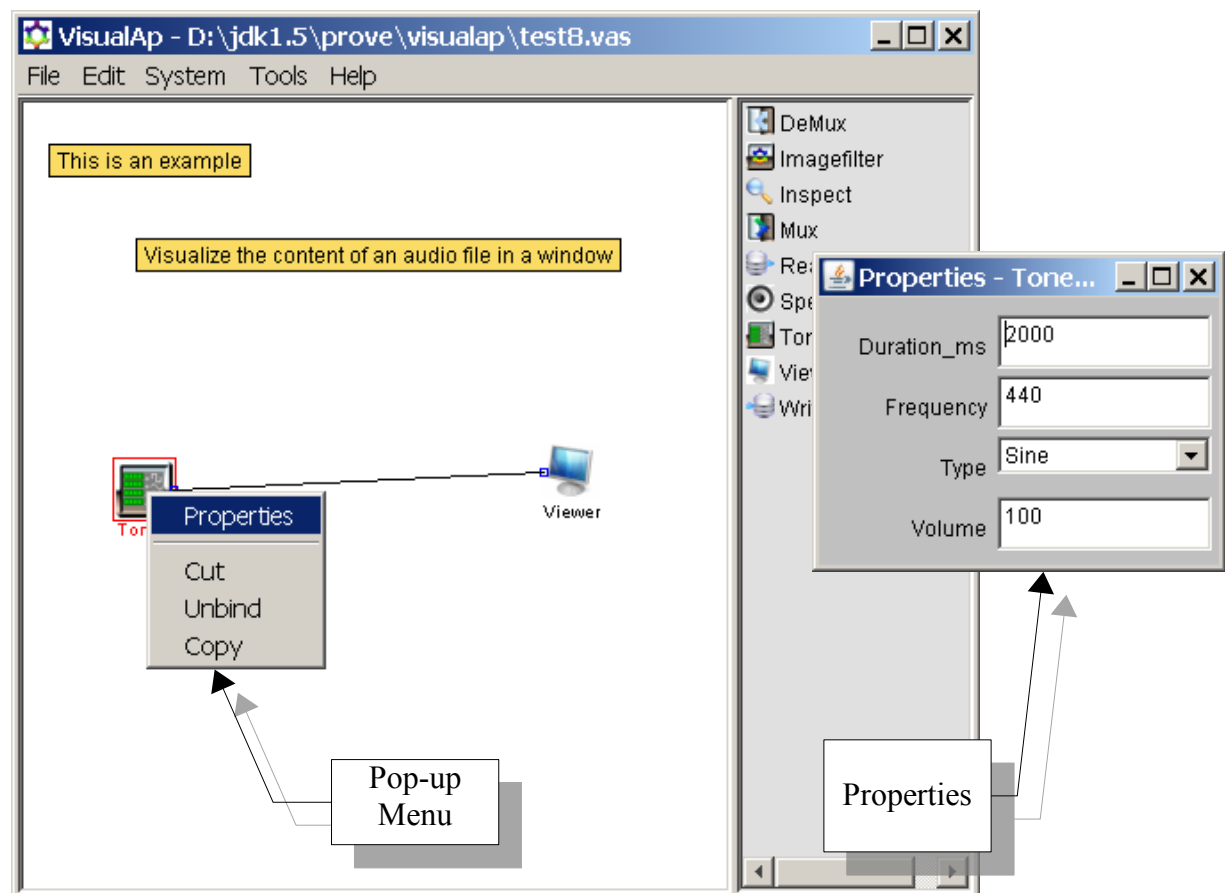
A popup menu is shown when the right button of the mouse is pressed over the workplace panel. Note that mouse cursor changes when specific actions are performed, e.g. during dragging.

It is possible to use keyboard accelerators like Ctrl+C, Ctrl+X, Ctrl+V, Ctrl+S, Ctrl+N, Ctrl+O, Ctrl+P, F1.

The application supports selections of multiple objects by dragging the mouse cursor. Copy and paste are supported.

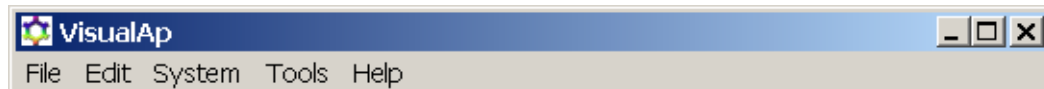
It is possible to view/modify the configuration of a component by double-click over a component in the workplace panel.

Properties of components are shown in the "properties" floating window. It is possible to change each property of a component by changing data inside the "properties" windows.



In case you set data that are not allowed by the component, a dialog box will pop-up explaining the limitation.

2.2 Details about available menu



2.2.1 File Menu

New: creates a new system. Accelerator: Ctrl+N

Open: open an existing system (.vas is the filetype for VisualAp systems). Accelerator: Ctrl+O

Save: save the current system (noname.vas in case no filename has been defined). Accelerator: Ctrl+S

Save as: save the current system using a specified file name

Print: print the current system. Accelerator: Ctrl+P

Quit: quits VisualAp

2.2.2 Edit menu

Properties: displays the selected component properties (the same command is available by right-clicking the mouse over the component in the workspace panel)

Cut: delete the selected component (the same command is available by right-clicking the mouse over the component in the workspace panel). Accelerator: Ctrl+X

Unbind: removes all the edges connected to the selected component (the same command is available by right-clicking the mouse over the component in the workspace panel)

Copy: copy the selected component (the same command is available by right-clicking the mouse over the component in the workspace panel). Accelerator: Ctrl+C

Paste: insert a copy of the clipboard in the workspace panel (the same command is available by right-clicking the mouse in a free area of the workspace panel). Accelerator: Ctrl+V

Insert: insert a new component in the workspace panel (the same command is available by right-clicking the mouse in a free area of the workspace panel)

2.2.3 System menu

Check: performs a check of the correctness of the current system

Run: run process simulation of the current system

Properties: Set properties of the current system

2.2.4 Tools menu

For future use.

2.2.5 Help menu

Help Topics: display the Help window. Accelerator: F1

Help on beans: display the help information related to available beans

Check Version: checks for new version. If needed it opens the download page

About VisualAp: Displays the version number of VisualAp

3 Components

3.1 Introduction

A component (or proclet) is the building block of the VisualAp environment.

A component is shown with its icon plus a number of terminals.

Terminals are represented by a small blue square. Terminals are also called pins.

Input terminals are placed in the left side of a component, while output terminals are placed in the right side of a component.

3.2 Using components

It is possible to view/modify the configuration of a component by double-click over a component in the workplace panel.

Components can be connected together by first pressing the mouse button over a terminal, then move the mouse pointer over another terminal and then release the mouse button: an edge between the two terminals is created.

The following rules must be followed when connecting a terminal to another one:

- it is not possible to connect together two or more output terminals
- any input terminal must be connected to an output terminal
- loops and cycles are not allowed in a system

These errors are detected only when the system is checked (System->Check) or run (System->Run)

It is possible to design and release new components for VisualAp. please read the related developer guide.

New components shall be placed in the <beans> directory under the VisualAp installation directory.

3.3 Demo components

The following components are provided with VisualAp 1.2:

- Viewer: shows the incoming data in a floating window
- ReadFile: read a file, contains a property "file" -> a custom editor is used for file property
- WriteFile: write a file, supported type: text, audio, image
- Mux: generate stereo audio from two mono audio inputs
- DeMux: split stereo audio in two mono audio
- Inspect: shows the type of the incoming data
- Speaker: plays an audio stream
- ToneGenerator: generate a simple audio tone, contains properties that are checked against max values, contains a property "type" -> a custom editor is used to select a specific value
- Microphone: records an audio file

Other included components:

- Delay: introduces a delay in audio stream
- Echo: implements echo effect
- Imagetransform: transforms an image

4 Installation and start of visualap

Note: VisualAp requires either JRE or JDK, version 1.6 or later. Please download it before installing VisualAp. Further details are available here: [Free Java Download](#)

Tested with Windows XP, Vista, Windows 7 and Ubuntu.

4.1 Windows XP, Vista, Windows 7

Download VisualAp-Setup.exe file and execute it in windows environment. The installation program will create a directory, installing needed files, adding windows registry keys and short-cuts in Windows start menu.

4.2 Linux and other OS

Download the visualap-setup.jar file. Create a directory, e.g. Visualap, put the visualap-setup.jar in that directory and execute the following command:

```
java -jar visualap-setup.jar
```

Such installation does not set windows registry keys nor short-cuts, in case you need it, use the file VisualAp-Setup.exe.

The file visualap.jar includes all java sources.

4.3 Start VisualAp application

How to run VisualAp using java command:

```
java -jar visualap.jar [-fast] [-run] [-report] [-uniqueID] [-help]
<filename>
```

Command line options:

-fast	fast startup
-run	automatic run
-report	print a report about available beans
-uniqueID	print uniqueID
-help	this help

If you are behind a firewall, you may want to configure the proxy:

```
java -Dhttp.proxyHost=proxyhost
    [-Dhttp.proxyPort=portNumber] -jar visualap.jar
```

The simple way to launch the program is to run visualap.bat in a DOS window. You may modify the batch file with proxy configuration.

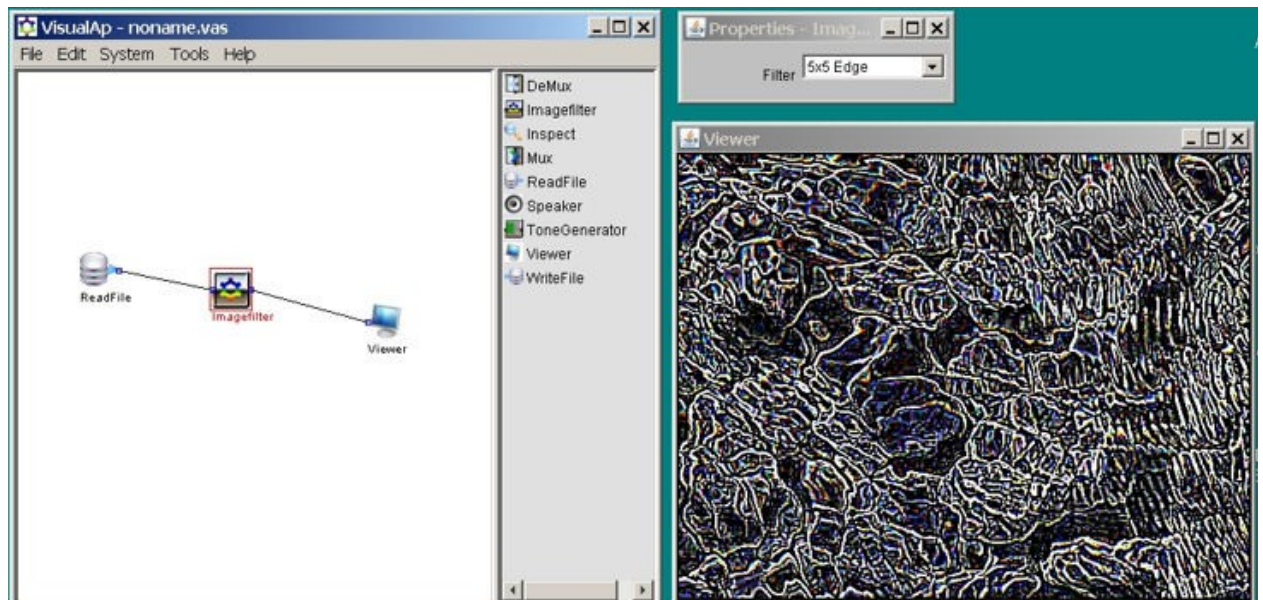
Execute the file build.bat in a DOS window to compile VisualAp.

Execute the file buildtest.bat in a DOS window to compile demo beans.

5 Tutorial

The scope of this paragraph is learn how to build a simple system, step by step:

1. Launch the application VisualAp, and select File->New to start the workspace.
2. Select in the toolbox the *ReadFile* component, move and click the mouse pointer in the left side of the workspace: the component *ReadFile* is placed in the left side of the workplace
3. Double-click the *ReadFile* component in the workspace: the properties window for *ReadFile* appears.
4. In the properties window press the “Choose Filter” button, select the file “sassi.jpg” and press “Done”.
5. Now select in the toolbox the *ImageFilter* component, move and click the mouse pointer in the center of the workspace: the component *ImageFilter* is placed in the workplace.
6. Connect the output pin of *ReadFile* to the input pin of the *ImageFilter* component.
7. Now select in the toolbox the *Viewer* component, move and click the mouse pointer in the right side of the workspace: the component *Viewer* is placed in the workplace.
8. Connect the output pin of *ImageFilter* to the input pin of the *Viewer* component.
9. Now check the system: System->Check, you should get a pop-up with “System Check Passed” answer.
10. Eventually you can run the system: System->Run, a new window pops-up with the inverted image.
11. Double-click the *ImageFilter* component in the workspace: the properties window for *ImageFilter* appears. You can change the effect to 5x5Edge.
12. Run the system, again System->Run, the image will change due to the new filter.



6 Terminology

Term	Meaning
System	A set of components that are interconnected and interacting. A system is represented as a graph, without cycles/loops, whose nodes are the components of the system.
Component	The basic building block of a system, performing some kind of simple processing.
Sink	A component that consumes data.
Source	A component that produces data.
Procllet	A Java component performing data processing in the VisualAp framework.