Compose User Manual

Contents

Overview

Compose is a visual programming language that uses hyper-pipes between applications to allow componentised applications to be built graphically.

Various components are available to be used with the main Compose application. These components are specially designed applications in their own right. To create a program, components are dragged onto the Compose canvas where they appear as icons. They can be moved around the canvas, and links can be created between components that allow one component to transfer data to another component. Data transfer works in a very similar way to Unix pipes, except that whilst pipes are 1-dimensional, in Compose there can be many links between components. Compose applications are therefore built up by dragging links between components on the two dimensional canvas.

Once components have been linked on the Compose canvas, the whole application can be executed as if it is one program. The Compose application then works as an intermediary to the other components to martial data and send control messages to the components.

Compose is open source software with an MIT style licence.

Licence

Installation

To use the program the !Compose application and the components directory containing all of the components are needed. Simply drag these to whenever you want the program installed. They should also all run from the archive.

The ‘Source’ directory contains the source code for the applications and the ‘Doc’ directory contains information, including this manual. Finally, the ‘Program’ directory contains a number of example programs. None of these are required, but you may want to keep them anyway.

Introduction

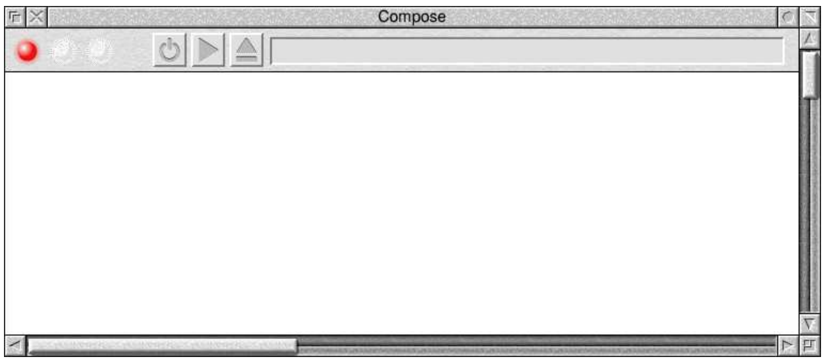
As the overview above explains, Compose is used to create programs by linking together components. This user manual is not intended as a tutorial about visual programming. Instead, we present a brief overview of the features of Compose that can be used as a reference for how to achieve things using it.

For information about how to create Compose components, or other technical details, refer to the Compose technical manual.

The crucial point to bear in mind is that programming with Compose is achieved primarily through joining components together using links that data can flow down. These links are all directed links, so that data flows down them in only one direction.

The Compose Canvas Window

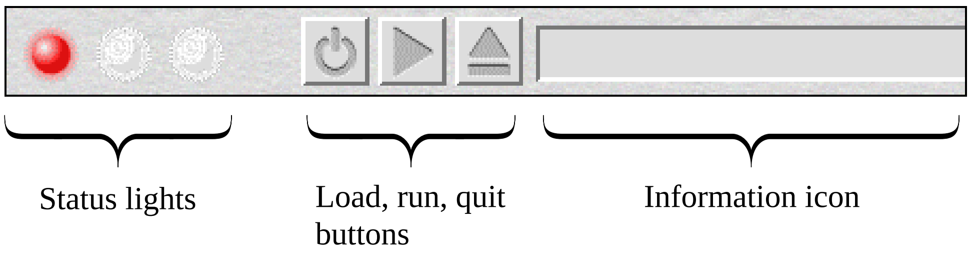
The main Compose window is opened by clicking on the Compose iconbar icon. Only one can currently be open at a time, but in future this may be changed. The window should look much like the image shown below.



The main Compose window

The window is split into two sections: the toolbar along the top of the window and the larger part at the bottom, which is the main canvas area. The canvas area is used for controlling and manipulating the components that make up a Compose program. This will be described in greater detail throughout the remainder of this document.

The toolbar along the top of the window has three main sections (see figure below). On the left are three status lights. Towards the centre are the program controls buttons that load, run and quit the components respectively from left to right. On the right is an information icon that provides details about components and links.

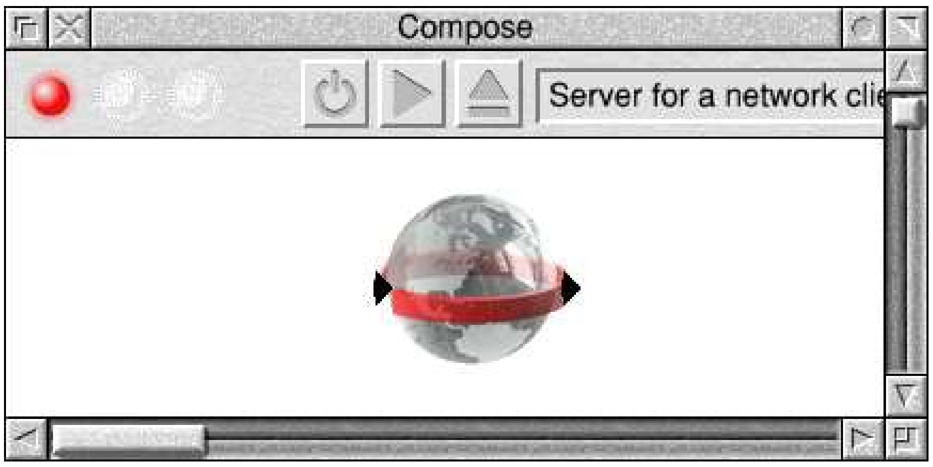


The Compose toolbar

The purpose of all of these will be described specifically in later sections.

Selecting Components

A Compose program will most likely be made up of several components. Components are chosen by dragging the component application from a filer window onto the Compose canvas. If the process is successful, the component will appear as an icon in the main Compose canvas window. The picture below shows how the canvas might look after dragging the 'Server' component onto the window.

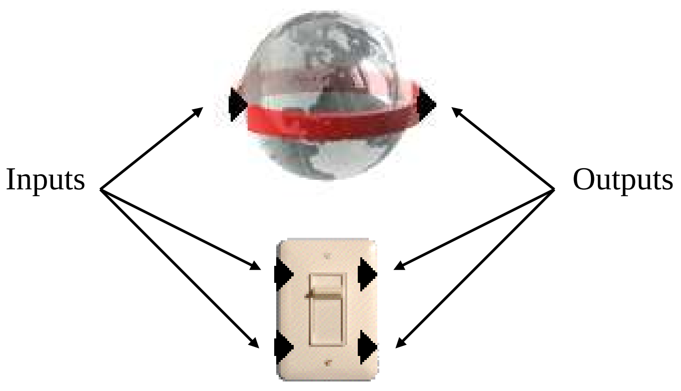


The Compose window with the 'Server' component

Component Features

Components that appear in the Compose canvas window can be manipulated in various ways and share a number of common features.

On the left and right hand side of the component icon you'll often see some black arrows (see the image below). A component can have any number of these arrows, and may not have any at all. However the arrows are very important. They represent the inputs to and outputs from the component and constitute the main means of programming in Compose.



Component inputs and outputs

The arrows on the left hand side of a component each represent an input to the component. The arrows on the right hand side of a component represent outputs from the component. Links can be created between components, but they will always link *from* an output and *to* an input. In effect, the links move from left to right across the canvas.

The position of a component on the canvas doesn't in itself affect the way a Compose program will execute, only the links between components. The components can therefore be arranged on the canvas in any way desired.

Components can be arranged by dragging them around the canvas using the mouse. Just click in the central area of the component icon and drag it to its new desired position.

Removing a Component

If you wish to remove a component from the canvas simply drag it onto the toolbar at the top of the window. If a component that has input or output links connected to it is removed in this way, then all of the links attached to it will also be removed.

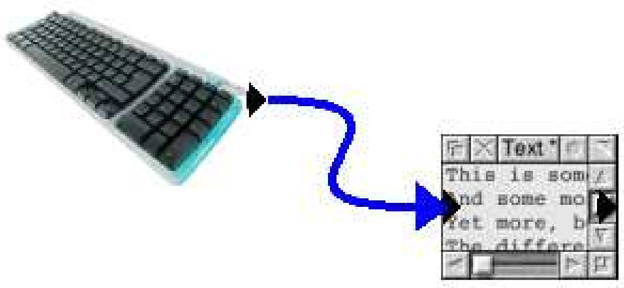
Getting Information About a Component

Components are constructed using application directories. Therefore, they may contain help files accessible from the filer menu just like any other application.

In addition to this, you can obtain a brief one-line description of a component by hovering the mouse pointer over the component's icon in the main canvas window. The description will be displayed in the information icon on the toolbar at the top of the main window. By hovering over the input and output arrows on the left and right hand side of a component you can also obtain information about the data expected in, or that will be sent out, of a particular component link. Again, this will be displayed in the information icon of the toolbar at the top of the main window.

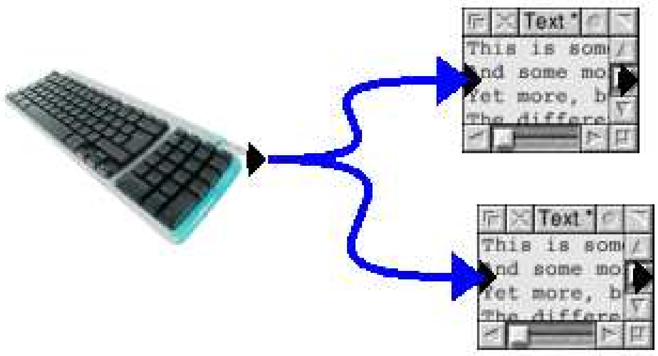
Creating Links Between Components

To create a directed link between any two components, simply drag a link from the output of one component to the input of another. In other words, click and hold on one of the output arrows of a component (on the right hand side), drag the link to an input arrow of another component (on the left hand side) until it 'clicks' into place and then release the mouse button to fix the link. If successful a blue arrow will be drawn to indicate the existence of a link between the two components. An example link is shown below.



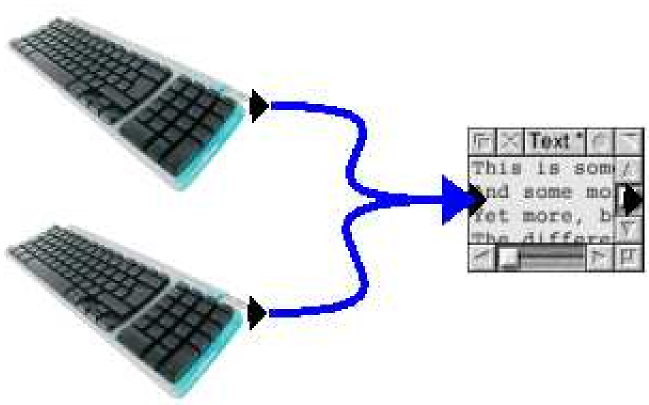
Linking two components

It is perfectly okay to create more than one link from a single output. An example of this is shown below.



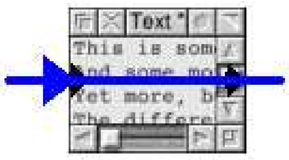
Multiple links from a single output

You may also send more than one link to a single input. An example of this is shown in the image below.



Multiple links to a single input

Finally, you may also send the output of a component to an input from the same component. Again, you can see an example of this below.



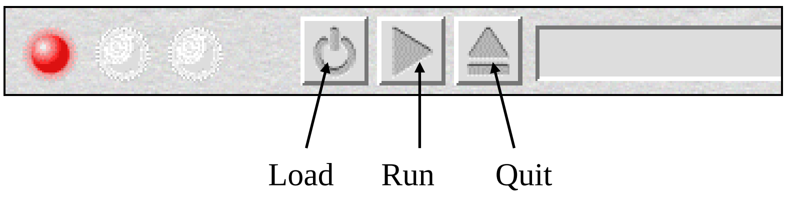
The output of a component linking to an input of the same component

Changing Existing Links

Any link that has been created can also be easily changed. To alter an existing link, simply drag on the arrow head of the link (this will be fixed to a component's input). The link will be released from the input of the component. You can then drag the link to another input to re-fix it in place. If the arrow head of the link is released somewhere other than over the input of a component, the link will be deleted entirely.

The Load-Run-Quit Cycle

When the components are set up with a suitable configuration for a program, the next step is to execute (also described as running) the program. To run a program you must first load it by clicking on the load button in the toolbar (see the diagram below). This loads the components that make up the program from disk. Each component is actually a wimp application. Loading a component is therefore equivalent to running each of these wimp applications so that they are resident in memory. The lights on the left hand side of the toolbar will change colour to indicate the status of the components. These lights are explained in greater detail below in the section on 'The Status Lights'.



The Load, Run and Quit buttons of the toolbar

Once loaded, components can be configured (see the section on 'Configuring a Component' below). Having the component loaded is different from actually running the program. To run the program, click on the 'Run' button. This will initialise the components and they will start to undertake whatever task they have been created to perform. Often this may involve sending some data across a link, or opening a window on screen.

Once the program has been run in this way, and the program's task has been completed, the component must be stopped and removed from memory. To achieve this, click on the 'Quit' button. This will quit all of the components.

The Status Lights

The three lights at the top left of the toolbar indicate the status of the current program. The three lights from left to right are red, amber and green. At any one time, only one of these will ever be 'lit'.

When the red light is showing, this means that none of the components are loaded and nothing is executing.

The amber light glows when some, but not all, of the components are loaded. This state might occur for a number of reasons. For example, components are loaded sequentially, so the amber light should show briefly whenever your program is constructed using more than one component and you click on the ‘Load’ button. You may also see an amber light if one of the components fails unexpectedly and has to be quit for whatever reason. Also, if you drag an additional component onto the Compose canvas whilst a program is already loaded, you are likely to find the amber light becomes ‘lit’.

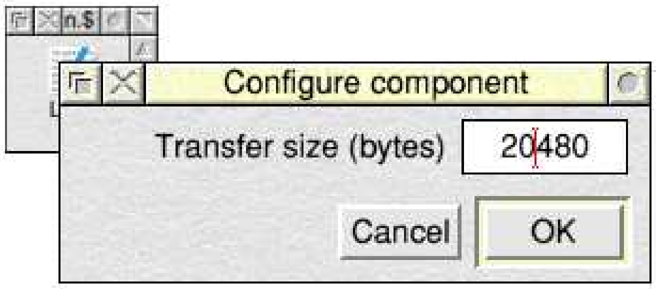
The green light shows when all of the components of the program are loaded. It does not necessarily mean that the program is actually running, although in this case the green light will also be shown. A Component can only be configured when it is loaded (see the next section on ‘Configuring a Component’), hence this should only be attempted if the green light is shown.

Similarly, you cannot execute a program that has not first been fully loaded. Hence the program will only run when the green light is showing. In essence, this means that the ‘Run’ button will have no effect until the ‘Load’ button has already been pressed.

Configuring a Component

Components can be configured at any time when the green light is shown (*i.e.* after the component has been loaded). However, changing a component’s configuration whilst a program is running may have unexpected consequences. In general, then, configuration should be undertaken between the component being loaded and the program being run.

When loaded, you can change the configuration of a component by double clicking on its icon in the main canvas window. This will generally bring up the configuration window for the component. The configuration window will depend on which component is being configured. You should therefore consult a component’s help file to establish the details of a component’s configuration window. Below is an example of a configuration window, taken from the ‘Load’ component.


Configure component

Transfer size (bytes) 20480

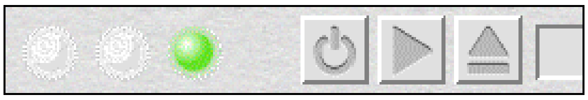
The ‘Load’ component’s configuration window

Saving Programs

To save a program, use the ‘Save’ menu item from the main canvas window’s menu (see the section on ‘The Main Canvas Menu’) in the usual way, by dragging the icon to a filer display. This will save the current component layout, including the links between the components.

There are some important points to note when you do this. Saving a program in this way does not save the actual component themselves in the save file; only the component layout is saved. Therefore, in order to load the layout back into Compose you will also need a copy of each of the components used.

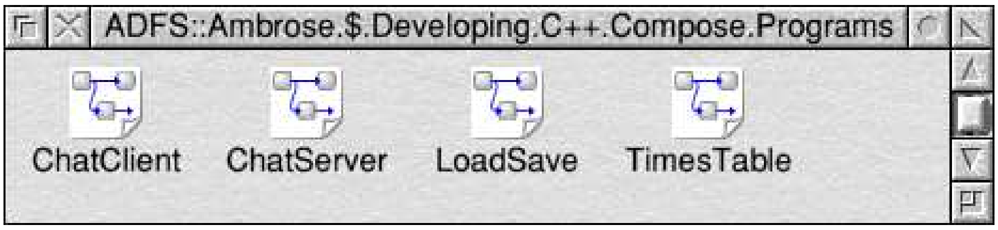
Perhaps more crucially, each component has its own configuration, accessed by double clicking on a component's icon (see the section above on 'Configuring a Component'). The configuration of each individual component will *only* be saved if the component itself is currently loaded. In other words, the configuration will *only* be saved if the green status light is showing.


The green status light

You can still save the program when the green light is not showing, but this will *not* save the individual configurations, only the general layout will be saved and components will revert to their default configurations when the program is reloaded. If you've just loaded a configuration, it is therefore important to ensure that the green light is showing by clicking on the 'Load' toolbar button before re-saving the program. Otherwise there is a good chance that the configuration data held in the file will be lost. Hopefully this behaviour will be changed in future versions of the program.

Loading Programs

A program file looks like a file with some components and links on it, as shown below.



Program files in a Filer window

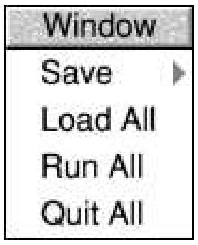
A Compose program file will have filetype of &1bf, or 'ComProg'.

Double clicking the file will load it into the Compose application. Note however that only the structure is actually held in the program file, not the components themselves. Before a file can be successfully loaded, all of the components needed to make up the program must therefore have been seen by the Filer. You can also drag a file to the Compose iconbar icon or to a Compose canvas window to load it.

If a canvas window already has components shown in it, loading a new program file will *merge* the two sets of components. If you want to load a file without merging like this, make sure that you start with a blank canvas. This behaviour may change in future versions of the Compose program.

The Main Canvas Menu

Clicking menu on a Compose canvas window will bring up the following menu.



The main canvas menu

The ‘Save’ menu item allows you to save the current configuration in a file. This is described in the section on ‘Saving Programs’. The ‘Load All’, ‘Run All’ and ‘Quit All’ menu items replicate the three buttons found on the toolbar to load, run and quit the current Compose component program respectively. For details about this, see the section on ‘The Load-Run-Quit Cycle’.

Contact

If you wish to contact me about Compose for any reason, please feel free to do so. Compose can be found on the web at

http://www.flypig.co.uk/compose.htm

and I can be emailed at: david@flypig.co.uk