A2 applications classification

Draft Feb. 25, 2009 – work in progress
Corrections and comments are encouraged

Table of Content

Central information source on A2

For the release material and the documentation refer to:
  www.bluebottle.ethz.ch which has a link to www.ocp.inf.ethz.ch

Introduction

Release.Tool is the core document for producing the various builds each incorporating various packages. Being so important, it seemed appropriate to prepare this documentation based on the texts of the GuiApplicationsMini and GuiApplications packages.

Each application (not all yet!) is presented in a table, aiming at a compact, simple, informative and uniform description of each of them.

Usage	Purpose, concise description and general behavior of the application which manifests itself, in most cases, in a window (visual component) inserted on the desktop.
Start	How to start the application, usually by executing a command. In some cases, a few more commands provide additional functionalities.
Stop	How to stop the application. In many cases, it suffices to close the application window. None of the modules involved is then unloaded. Also, how to unload the top level modules involved, thereby closing all similar windows. In a few cases, more than one module is participating to the application. The modules are listed in the order in which to unload them orderly, that is, the top module comes first. A shorter method for unloading several modules is to use SystemTools.FreeDownTo modulename ~. This command can be also used to free some of the modules indirectly imported, although this is somewhat dangerous.
Restorable	This table entry appears only when the application is restorable: the application module was then programmed to be such. If the user saves the desktop (a button on the Main menu is provided), enough data on the then current state of the application is recorded for use in the next session. When A2 is started anew, the application window will reappear on the desktop in the state and at the location it had during the previous session. More detail is provided in the section "Desktop save/restore mechanism".
Platform	The aim is to port A2 to the widest possible range of platforms, though some of the applications might not be portable. This table entry appears only when the application is not available in a particular platform. The platforms are: | **Build**	**Object module extension**
A2	.Obx
WinAos	.Obw
Unix	.Obj | Example: WMUsbInfo is not present in WinAos since USB devices are controlled by the host Windows.
Components	The top level modules involved starting from the bottom of the hierarchy, that is in the order in which to compile them, when corrections are needed. A "/" delimits the modules from participating data files, some of them containing icons or images allowing some customization.

Desktop and desktop navigation

When A2 starts, the desktop shows at least the Main menu (see below) and maybe a number of windows depending on how the system is customized (next paragraph) and depending on the number of restorable applications were active when the desktop was stored.

The desktop is more than just the display area. It extends beyond the physical boundary of the screen since A2 is based on a larger virtual desktop of which the visible part is called a view port. The view port can be zoomed in and out, moved several times by one screen in any direction with respect to a screen edge. Windows are normally opened in the view port but as soon as one navigates on the desktop, that is the view port is moved in a direction, the, till then visible, windows disappear in the opposite direction. The desktop navigation is controlled using the keyboard and the mouse cursor and governed by the following rules:

Meta Mouse Wheel (1)	Zoom in / out
Meta Mouse Move (2)	Pan the view port when the cursor hits the screen edge
Meta PageUp	Zoom in 2x
Meta PageDown	Zoom out 2x
Meta Home	All desktop components are made visible by an adequate zooming and movement of the view port.
Meta End	Zoom 1:1. Warning: is not the inverse operation of Meta Home.
Meta Left	Move the view port to the left by one screen in the current zoom factor
Meta Right	Move the view port to the right by one screen in the current zoom factor
Meta Up	Move the view port up by one screen in the current zoom factor
Meta Down	Move the view port down by one screen in the current zoom factor

The Meta key may be substituted by Alt-Shift. Alt-Shift must be used in the case (1) and (2) in WinAos.

Under all circumstances, the Main menu, documented in a follow-up section, remains in the left bottom corner of the view port.

These are the basics of the desktop navigation. More about navigation later on.

A2 customization with Configuration.XML

An A2 distribution is conditioned in many ways by the Configuration.XML file (and its shadow copy Save.Configuration.XML). This file is a structure of nested sections:

```
<?xml version="1.0" encoding="UTF-8" standalone='yes'?>

<!-- Aos configuration -->
    <Config>
        <Section name="A">
            <Section name="B">
                <Section name="C">
```

```
                        <Setting name="D" value="Hello world"/>
                    </Section>
                </Section>
            </Section>
        </Config>
```

with sections and sub-section for specifying such things as: localization, supported file systems, autostart commands, codecs, etc.

Both files are part of the A2 delivery. Whenever a modification is desired or needed:

1. edit the configuration with, for example, Notepad.OpenAscii Configuration.XML ~ or, even easier, use the "Configuration" button in the Main menu,
2. save the configuration, and
3. execute Configuration.Init ~ to finalize the change.

If the modified configuration is syntactically correct, A2 may be restarted and the new configuration applies. If a syntax error is detected in step 3, the faulty configuration is ignored and the stand-by, correct configuration Save.Configuration.XML is used instead. In this way, the next A2 start can succeed. Starting with corrupted specifications might be fatal.

The Autostart section

Among all the sections, the Autostart section is of particular interest to the user as it collects system commands that the user Here is an example content of the section:

```
        <Section name="Autostart">
            …..
            <Setting name="Start the main menu" value="StartMenu.Open"/>
            <Setting name="DefaultSkin" value="SkinEngine.Load stijnbw.skin"/>
            <Setting name="Restore the desktop" value="WMRestorable.Load Auto.dsk"/>
        </Section>
```

may contain a collection of commands that the user may want to let A2 execute automatically when it starts. A great many such commands are described in the remaining of this text and a word is said below concerning three of them.

The only command which must appear in the list is "StartMenu.Open".

It is very convenient to also include PCITools.DetectHardware. In that way, all the drivers that might be needed are activated from the start, and very useful when multimedia is going to be used.

In Windows, to use the Windows command interpreter for starting WinAos the command "CommandLine.Open" may be added in that section.

Installation

Installer

Usage	Install A2 on a partition.
Start	WMInstaller.Open ~
Stop	Close the window. To clean up: SystemTools.Free WMWinstaller ~
Components	PartitionsLib.Mod, FATScavenger.Mod, Installer.Mod, Partitions.Mod, WMPartitionsComponents.Mod, WMPartitionsPlugins.Mod, WMPartitions.Mod WMInstaller.Mod / WMInstaller.tar, InstallerPackages.XML

At the end of Installer.Mod correct Packages.XML to InstallerPackages.XML and delete the creation of Install.Tar

Partitions framework

made of: WMPartitions.Mod, WMPartitionsComponents.Mod, WMPartitionsPlugins.Mod and Partitions.Mod, PartitionsLib.Mod

and the partition editor made of:
PartitionEditor.Mod, PartitionEditorComponents.Mod, PartitionEditorTable.Mod

Desktop and customization

Main menu

Presents a selection of some of the GUI applications in form of a two-level hierarchy of buttons which, when clicked with the left mouse button, execute commands behind the scene. A GUI state change then results such as:
   on the first level, showing the next level in the button hierarchy (defined by an XML file)
   on the 2nd level, starting an application, among those described here, in a GUI window.

Usage	A2 is conditioned by default to show a main menu. Since the main menu is evolving in time, the function associated with the buttons is not described here. The user has the liberty to customize the main menu. If the menu is not visible because some windows are placed on top of it, press Ctrl-Esc to let it appear on top. In WinAos, use Ctrl-Alt-Esc.
Start	The „StartMenu.Open" command, responsible for inserting the menu on the desktop, is placed by default in the Autostart section of Configuration.XML. It is thus not necessary to ever use this command and it is not recommended to attempt to remove it from the Autostart!
Stop	Never
Components	MainMenu.Mod, StartMenu.Mod / MenuPagexy.XML (xy takes the values 00,10, 20, 30, 40, 50, 60, 70, 80,90, all of which are reserved for the release)

Menu page structure

An XML menu file defines a menu as a succession of Panels with 2 vertically aligned Buttons, each associated with a command. The text in italic is the essential part that is customized. The MenuPages utility creates such a menu from a text composed of these essential parts *menuname { buttonCaption command }*.

```
<Panel caption="menuname">
    <Properties>
        <FillColor>0</FillColor>
    </Properties>
    <Panel>
        <Properties>
            <Alignment>1</Alignment>
            <Bounds>
                <Width>120</Width>
            </Bounds>
        </Properties>
        <Button>
            <Properties>
                <Caption>buttoncaption</Caption>
                <Alignment>2</Alignment>
                <OnClickHandler>X Run</OnClickHandler>
            </Properties>

            <SystemCommand>
                <Properties>
                    <ID>X</ID>
                    <CommandString>command</CommandString>
                </Properties>
            </SystemCommand>
        </Button>
        <Button>
            a second button definition
        </Button>
    </Panel>
    as many panels with 2 Buttons each as needed
</Panel>
```

Desktop save/restore mechanism (not on the GUI)

Usage	The desktop with most of its GUI elements can be saved at any time and restored later on at will. Usually, saving the desktop is done just before terminating an A2 session at which time the required relevant information is saved in "Auto.dsk". When A2 is started anew, it is organized to "restore" the desktop from the data deposited in "Auto.dsk" so that the user can proceed with work as it stood in the last session. This default behavior is conditioned by this setting: `<Setting name="Restore the desktop" value="WMRestorable.Load Auto.dsk"/>` in the Autostart section of Configuration.XML. Each program responsible for instantiating a GUI window must be programmed to be "restorable". Technically, a program must include an exported PROCEDURE Restore that will be invoked by WMRestorable.Load
Start	WMRestorable.Store [<filename>] ~ WMRestorable.Load [<filename>] ~

	or use the „SaveDesktop" button in „System" of the Main menu. The default filename is „Auto.dsk".
Stop	Never
Components	WMRestorable.Mod / Release.Auto.dsk (why not call it Build.Auto.dsk ?)

Virtual desktop navigate commands

Each window inserted on the virtual desktop, which extends well beyond the screen boundary, can be considered a task and many such tasks may exist, making it difficult to keep an eye on them. To localize the many windows up to 10 summaries, or task lists, can be inserted on the desktop. A summary consists of a juxtaposition of icons each of which is a symbolic representation of one of the windows.

Usage	

Start	WMNavigate.Open [options] id x y componentFile ~
	options: f = surround task list with a frame (having a „Close" button)
	        s = stay on top (of other windows)
	        v = view port
	id : task list identifier (starting with 1)
	x, y : position of the task list's upper left corner
	componentFile : an XML file describing the task list (WMNavigateTaskList.XML)

	Command setting the visible view port: WMNavigate.SetViewportRange [options] [x y [w h]] ~ options: s = show the transition to another view port progressing d = interpret x, y, w and h as multiple of the display width and height x, y : position of (upper left corner of) view port w, h : width and height of view port in pixels (effective display width and height are used if omitted). Examples of parameters for some view ports: | View port above at the left -sd -1 -1 1 1	View port above -sd -1 0 1 1
View port at the left -sd -1 0 1 1	Standard desktop -sd 0 0 1 1 | The rectangle's center is considered to have the coordinates 0, 0 and corresponds to the upper left screen corner when the zoom level is 1:1. All four view ports together: -sd -1 -1 2 2 ~ which may be considered as an implicit zoom level 0.5 **Commands acting on all task lists:** WMNavigate.HideNavigation ~ WMNavigate.RestoreNavigation ~ WMNavigate.ToggleNavigation ~ **Command acting on a single task list:** WMNavigate.ToggleVisibility id ~ **Commands acting on all windows, not on task lists:** WMNavigate.HideAll ~ WMNavigate.RestoreAll ~ WMNavigate.ToggleAll ~

	Commands acting on a single window, the one owning the focus: WMNavigate.MoveWindow [-d] [x [y]] ~ move relative to its current position WMNavigate.ToggleFullscreen ~ WMNavigate.CloseWindow ~
Stop	WMNavigate.Close id ~ To clean up: SystemTools.Free WMNavigate ~
Components	WMNavigate.Mod / WMNavigateIcons.tar WMNavigateOverview.XML

Hot key definitions for controlling the GUI

Hot key definitions bind program functions to a single keystroke or a few keystrokes saving the user time. It is the user's prerogative to establish such bindings in an XML file according to the following abstract syntax:

```
<!DOCTYPE HotKeys [
    <!ELEMENT HotKey ANY>
    <!ATTLIST HotKey
        name NMTOKEN #REQUIRED
        keys NMTOKEN #REQUIRED
        command CDATA #REQUIRED
    >
]>
```

Here follows a sample of concrete syntax:
```
<HotKeys>

    <!-- Standard hot keys (don't change) -->
    <HotKey name="Show Hotkeys" keys="F11" command="HotKeys.Show"/>
    <HotKey name="Disable Hotkeys" keys="F12" command="SystemTools.Free HotKeys"/>
...
</HotKeys>
```

Warning: this tool allows definitions that can render A2 unusable. Have always a hot key bound to SystemTools.Free HotKeys.

Usage	Install a number of useful keyboard shortcuts (executing some command in the background), as for example WMNavigate commands (described above) setting the view port: <table><tr><td>F1 :View port above at the left</td><td>F2 :View port above</td></tr><tr><td>F3 :View port at the left</td><td>F4 : Standard desktop/view port</td></tr><tr><td>F5 : All these four view ports together</td><td></td></tr></table>
Start	HotKeys.Open [<filename>] ~ Load the hot keys definition from an XML file. The default filename is HotKeys.XML.
	HotKeys.Show ~ Report all currently loaded hot keys in the Kernel Log, in the reverse order of the XML definition file.
Stop	To clean up: SystemTools.Free HotKeys ~
Components	HotKeys.Mod / HotKeys.XML

The HotKeys.XML file included proposes the following bindings:

Key stroke	Command	XML name
Standard hot keys		
F11	SystemTools.Free HotKeys	Disable Hotkeys
F12	HotKeys.Show	Show Hotkeys
Screenshot		
F10	WMOSD.Close;WMScreenShot.SnapShotView test.bmp;WMOSD.Open 'Screenshot saved in test.bmp'	Screenshot
Start an application		
Meta+N	Notepad.OpenEmpty	Notepad
Meta+O	LogWindow.Open (not for WinAos)	LogWindow
Meta+P	PET.Open	PET
Meta+S	WMShell.Open	Shell
Virtual desktop control		
F1	WMOSD.Close;WMNavigate.SetViewportRange -d -1 -1 1 1;WMOSD.Open 'Desktop 1'	Desktop 1
Meta+F1	WMOSD.Close;WMNavigate.SetViewportRange -d -1 -1 1 1;WMOSD.Open 'Desktop 1'	Desktop 1 (WinAos)
F2	WMOSD.Close;WMNavigate.SetViewportRange -d 0 -1 1 1;WMOSD.Open 'Desktop 2'	Desktop 2
F3	WMOSD.Close;WMNavigate.SetViewportRange -d -1 0 1 1;WMOSD.Open 'Desktop 3'	Desktop 3
F4	WMOSD.Close;WMNavigate.SetViewportRange -d 0 0 1 1;WMOSD.Open 'Desktop 4'	Desktop 4
F5	WMOSD.Close;WMNavigate.SetViewportRange -d -1 -1 2 2	Desktops Overview
F6	WMNavigate.HideAll	Desktops Overview
F7	WMNavigate.RestoreAll	Desktops Overview
F8	WMNavigate.ToggleAll	Desktops Overview
Move the window owning the focus relative to its current position		
Alt+Right	WMNavigate.MoveWindow 1280 0	MoveRight
Alt+Left	WMNavigate.MoveWindow -1280 0	MoveLeft
Alt+Up	WMNavigate.MoveWindow 0 -1024	MoveUp
Alt+Down	WMNavigate.MoveWindow 0 1024	MoveDown
Switch the focus to previous/next window		
Ctrl+Alt+Up	WMNavigate.Open -vs 6 0 0 WMNavigateOverview.XML	Open Windows Navigation
Ctrl+Alt+Down	WMNavigate.Close 6	Close Windows Navigation
Ctrl+Alt+Right	WMNavigate.FocusToNext	FocusToNext
Ctrl+Alt+Left	WMNavigate.FocusToPrevious	FocusToPrevious
Open a window for entering and executing a command		
Ctrl+R	HotKeysCommands.EnterCommand	Run
Apply text style to selected text stretch		

Shift+Delete	WMTextStyleTool.SetStyleByName Bold	Bold
Misc		
Meta+Ctrl-O	HotKeysCommands.ClearLog	Clear LogWindow
Control window owning focus		
Alt+Return	WMNavigate.ToggleFullScrenn	Toggle Fullscreen
Alt+F4	WMNavigate.CloseWindow	Close Window
Simulate mouse wheel		
Ctrl+Up	HotKeysCommands.Simulate MouseWheel -3	WHEEL UP
Ctrl+Down	HotKeysCommands.Simulate MouseWheel 3	WHEEL DOWN
Key remapping		
Ctrl+I	REMAP UP	UP
Ctrl+J	REMAP LEFT	LEFT
Ctrl+K	REMAP DOWN	DOWN
Ctrl+L	REMAP RIGHT	RIGHT

Navigator

The Navigator presents, in a single small window (sized 400*200 pixels), an overview of approximately 2*3 view ports around the standard view port scaled down like this:
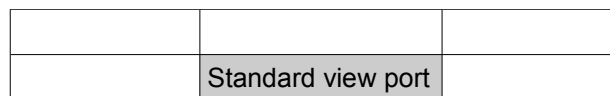
	Standard view port	

The standard view port is easily localized by the thin red line surrounding it. All the windows currently existing in these 6 view ports are scaled down, just to the limit of being recognizable. Each of them can be clicked and dragged to another position as one would do on the visible view port with the added possibility to drag windows not seen on the currently visible view port. The rest of the virtual desktop, beyond this surface is ignored by the Navigator but can, of course, be made visible by navigating using meta key combinations as described at the beginning of this text. When doing so, observe how the red line surrounding the standard view port moves along and can occasionally be reduced to a single straight line on the window border. The Navigator always stays at the same position on the screen independently of which view port is visible on the screen.

Usage	Allows moving around windows visible only in the Navigator and not on the visible view port.
Start	WMNavigator.Open ~ At that time the Navigator's upper left corner it at coordinate 20, 20, but it can be moved around and resized like any other window. When resized, the virtual desktop area represented is adjusted accordingly (elarged or reduced).
Stop	WMNavigator.Close ~ Close the window. To clean up: SystemTools.Free WMNavigator ~
Components	WMNavigator.Mod

Skin editor

A skin is the look and feel of an application's graphical user interface (GUI), conditioned by data stored in a skin file (suffix .skin) and customizable. A2 comes with a selection of skins, some of them are look-alikes of third-party skins. Destined to GUI designers, one should refrain from modifying skins without good knowledge.

Usage	Skins are stored in files, with the extension .skin, containing a description text, entitled skin.bsl, written in BSL (Bluebottle Skin Language), and images or bitmaps for use in the composition of visual components. SkinTutorial.Text gives some more details.
Start	SkinEditor.Open ~ SkinEditor.OpenSkin <filename> ~ or use the „SkinEditor" button in „Looks" of the Main menu. Several skin editors may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free SkinEditor SkinEngine FNHistories SkinLanguage ~
Restorable	
Components	SkinLanguage.Mod FNHistories.Mod SkinEngine.Mod SkinEditor.Mod (should be named WMSkin... and the tutorial mentions "Bluebottle")

Skin editor window representation

Skin Editor	
Skin filename input area	Control buttons area (see next table)
Resource index Tree-structure of the skin	Skin description (in BSL) editing area
Entry Info Currently selected resource info	
Colors Color palette with large square showing selected color. From there, drag and drop color to the edit area.	System reporting area appearing on clicking „Check". In the end, when no more error is detected, „No errors" is reported.

Load	Open the skin file for editing
Store	Save the edited data in a skin file
Search	Search and replace text string
Split	Open a second text view
Check	Check that the skin description is syntactically correct
Apply	Check the correctness and install if correct
SetAsDefault	Behind the scene execute SkinEngine.SetCurrentAsDefault ~ (see next section)
Back	Undo
Forward	Redo
Tutorial	Open the tutorial

Config	Open the SkinConfig.XML file for editing

Bluebottle Skin Language (BSL) definition

The information stored in a .XML file describes the values to assign to properties of the visual components used in the composition of the GUI and has 4 parts, starting with meta data:

```
skin{
    meta{
        name : "<skin name>";
        description : "<skin description>";
        author : "<author name>";
        date: "<creation date>";
    }
window{ … window frame definition
}
cursor{ … cursor definition
}
component{ … GUI components definition: button, scrollbar, etc.
}
}
```

Skin engine (not on the GUI)

Usage	Install a skin either temporarily, or permanently in Configuration.XML Instead of using one of the commands described under „Start": - use the skin loader (below) offering a choice of skins to install temporarily - edit the Autostart section of Configuration.XML so that the desired skin is automatically installed: \<Setting name="DefaultSkin" value="SkinEngine.Load xyz.skin"/> \<Setting name="DefaultSkin" value="SkinEngine.Unload"/>
Start	SkinEngine.Load <filename> ~ Install the specified skin SkinEngine.Unload ~ Set the system to an initial state using Zeroskin.skin SkinEngine.SetCurrentAsDefault ~ Register the current skin in the Autostart section of Configuration.XML
Stop	To clean up: SystemTools.Free SkinEngine ~
Components	SkinEngine.Mod

Skin loader

Usage	A list of skins of various designs to choose from is presented. The information is extracted from the SkinList.XML file. The available skins can be classified in two design groups: | **Name**	**Skin filename** | |---|---| | ZeroSkin | ZeroSkin.zip (the default skin – name it Zero) | | Reptile | reptile.skin | | Kramer | pklook.skin |

Traditional	traditional.skin
XP	winxp.skin (look-alike of Win XP)
OSX	aqua.skin (look-alike of Mac OSX)
Glass	glass.skin
StijnBW	stijnbw.skin

Name	Skin filename (using FancyStartMenu)
Blue	BluePlastic.skin (name it blueplastic)
Red	redlook.skin
Stijn	stijn.skin
Christmas	xmas.skin

The "Glass" and "Stijn" skins were designed by Stijn Ossevoort, a Dutch freelance designer http://www.linkedin.com/in/sostechnologydesign
Pronounce "stijn" as "stine" (english) or "stein" (german but with "s", not "sch").
The "StijnBW" skin is derived from "Stijn" by Sven Stauber.

Start	WMSkinLoader.Open SkinList.XML ~ (name it WMSkinList.XML) or use the „SkinLoader" button in „Looks" of the Main menu.
Stop	Close the window. To clean up: SystemTools.Free WMSkinLoader Looks ~
Components	Looks.Mod WMSkinLoader.Mod

Soren Renner, an OCP participant, developed srskin.skin, adapted from Thomas Frey's traditional.skin.

Backdrop installer

Usage	Install a backdrop image
Start	WMBackdrop.AddBackdropImage <filename> x y w h ~ x : abscissa upper left image corner wrt upper left view port corner y : ordinate upper left image corner wrt upper left view port corner w : image width h : image height
Stop	Close the window. To clean up: SystemTools.Free WMBackdrop ~
Restorable	
Components	WMBackdrop.Mod

Backdrop loader

Usage	A list of desktop backdrops of various designs to choose from is presented. The information is extracted from the BackdropList.XML file. The currently available backdrops are:

Name	Filename

	Mars	mars.png
	Bluebottle	BluebottlePic0.png
	Saas Fee	SaasFee.jpg
Start	WMBackdropLoader.Open BackdropList.XML ~ (name it WM...)	
Stop	Close the window. To clean up: SystemTools.Free BackdropLoader Looks ~	
Components	Looks.Mod WMBackdropLoader.Mod	

Editors and font services

The font name, size (in points) and style (normal, bold, italic) of text appearing in GUI components is determined by the section "WindowManager" in Configuration.XML:

```
<Section name="WindowManager">
        <Section name="FontManager">
            < Section name="DefaultFont">
                    <Setting name="Name" value="VeraBd"/>
                    <Setting name="Size" value="12"/>
            </Section>
        …
</Section>
```

An A2 distribution is conditioned to use the "Oberon" font of size 12. Whenever a different font, size or style is preferred, edit Configuration.XML as described earlier. This lets customize the appearance of text in window titles, button captions, dialog boxes, list entries, editors, etc.

The font name, size and style must be chosen among the available font files which can be internalized by the font loaders included in the release. A2 supports Unicode TTF fonts: simply copy a TTF file in a suitable directory to install it. Picking a font size larger than 12 will cause bits of text to be clipped in many places as A2 lacks the native ability to adjust component layouts to accommodate unusually large fonts. The current release includes font loader programs for:

Font type	Font file names & extension	Font loader
Oberon	Courierxy.Scn.Fnt Greekxy.Scn.Fnt Mathxy.Scn.Fnt Oberonxy.Scn.Fnt Philusxy.Scn.Fnt Shangai.Scn.Fnt	WMOberonFonts.Mod
Bitmap (Chinese, Japanese, Korean)	cjkfont.bfnt	WMBitmapFont.Mod
CCG	Single.ccg, Song.ccg	WNCCGFonts.Mod
Open type Unicode TTF (void of copyright)	Vera….ttf benevento.ttf Cyberbit.ttf (see Cyberbit TTF downloader)	WMOTFonts.Mod

	The list of installed .ttf files can be detected by executing WMOTFonts.MultiTest ~ described below.	

The "FontLoaders" sub-section of the Configuration.XML file dictates which font loaders must be made available to the running A2 system.

```
<Section name="FontLoaders">
      <Section name="OberonFonts">
            <Setting name="Exact" value="WMOberonFonts.LoadExactFont"/>
            <Setting name="Approximate" value="WMOberonFonts.LoadApproximateFont"/>
      </Section>
      <Section name="BitmapFonts">
            <Setting name="Exact" value="WMBitmapFont.LoadExactFont"/>
            <Setting name="Approximate" value="WMBitmapFont.LoadApproximateFont"/>
      </Section>
      <Section name="CCGFonts">
            <Setting name="Exact" value="WMCCGFonts.LoadExactFont"/>
            <Setting name="Approximate" value="WMCCGFonts.LoadApproximateFont"/>
      </Section>
      <Section name="OTFonts">
            <Setting name="Exact" value="WMOTFonts.LoadExactFont"/>
            <Setting name="Approximate" value="WMOTFonts.LoadApproximateFont"/>
      </Section>
   </Section>
```

When that section is missing in Configuration.XML, the system reports in the Kernel Log:
"WindowManager.FontManager subsection missing in Configuration. Running on defaults" and
"Using embedded font"

Simple text editor

Usage	Edit new data or data stored in a file while offering to encode, respectively decode it, with one of the available Codecs defined in the Configuration.XML file. The currently available Codecs are (in the order in which they appear in the pop-up menu):
	<table><tr><td>**Format : popup**</td><td>**Codec**</td></tr><tr><td>Oberon</td><td>The ETH Oberon format, which is used for source code</td></tr><tr><td>BBT</td><td>The Bluebottle format</td></tr><tr><td>UTF-8</td><td>8-bit Unicode Transformation Format</td></tr><tr><td>ISO8859-1</td><td></td></tr><tr><td>HEX</td><td>Hexadecimal with 16 bytes per lines separated by blanks</td></tr><tr><td>AUTO</td><td>Automatically using the decoder matching the encoder used when the data was stored. This information is recorded.</td></tr></table>
Start	Notepad.Open [-f] [<filename>] ~ When the filename is missing, an empty text appears. Several editors may be running in parallel.
Stop	Close the window(s).

	To clean up: SystemTools.Free Notepad ~
Restorable	
Components	Notepad.Mod

Editing with mouse commands

A freshly installed A2 system, like the A2 deliverable, assumes that the machine is equipped with a 3-button mouse, or with a mouse with a wheel, in place of the middle button, for ease of use. In this description, the three button are named from left to right ML, MM, and MR. ML and ML are most essential for editing and navigating in a text.
For a 2-button mouse, considered to have only ML and MR, MM can be simulated by a Ctrl+MR combination, after having modified the configuration string MB –
see http://www.ocp.inf.ethz.ch/wiki/Documentation/Configuration#toc12

	Select a text stretch and operate on the selection
ML press & drag	Select: Place the mouse cursor at the beginning of the intended selection, press ML and drag the mouse to the end of the intended selection, release ML
ML press & drag + MR	Cut: Select a text stretch and cut it
ML press & drag followed by ML press & drag	Cut and drop: Select a text stretch, then place the cursor (ML) amidst the selection and drag the selection to the intended position
ML press & drag followed by Ctrl press + ML press & drag	Select and drop a copy: Select a text stretch, hold Ctrl down, then place the cursor (ML) amidst the selection and drag the selection to the intended position

	Operate on a command or a document name appearing in a text
MM pressed	When the cursor is positioned in a text and MM is pressed the surrounding word is underlined in red. If that is recognized as a command, it is executed. Make sure that the command, possibly followed by parameters, is delimited with a „~". Example: Notepad.Open Tutorial.Text ~ Although a command works even without delimiter, it is a waste of memory. Since many commands can run in parallel, the parameters need to be copied before execution.
MM + MR = interclick	When the cursor is positioned on a document name, the document is opened.

Editing with keyboard commands

	Move the cursor
Arrow left	one character left
Arrow right	one character right
Arrow down	one line down
Arrow up	one line up

End	to the end of the line
Home	to the beginning of the line
Ctrl+End	to the end of the text
Ctrl+Home	to the beginning of the text
PageDown	one visible page down
PageUp	one visible page up

Shift+	**Move the cursor**	**and**
Arrow left	one character left	select a character, a line, a page OR deselect depending in which direction the cursor is moving
Arrow right	one character right	
Arrow down	one line down	
Arrow up	one line up	
End	to the end of the line	
Home	to the beginning of the line	
Ctrl+End	to the end of the text	
Ctrl+Home	to the beginning of the text	
PageDown	one visible page down	
PageUp	one visible page up	

	Operate a selection or on a selection
Ctrl+A	Select entire text
Ctrl+C or **Ctrl+Insert**	Copy to clipboard
Ctrl+V or **Shift+Insert**	Paste from clipboard
Ctrl+W	Paste clipboard of host operating system
Ctrl+X or **Shift+Delete**	Copy to clipboard and delete
Ctrl+Y	Redo – Can occur step by step if the last operation was complex
Ctrl+Z	Undo – Can occur step by step if the last operation was complex
Tab	Indent selection by one tab length to the right
Shift+Tab	Indent selection by one tab length to the left

	Operate on a command or a document name appearing in any text. **The same control can be exercised with the mouse**
Ctrl+Enter	When the cursor is positioned on a command, the command is executed
Ctrl+Shift+Enter	When the cursor is positioned on a document name, the document is opened

	Check text healthiness, control IME
Ctrl+T	Check the healthiness of the text piece list. Errors are listed in the Kernel Log.
Ctrl+Space	Switch IME on/off and active the next IME

Program editing and compilation tool

made of: PET.Mod and PETIcons.tar
capable of displaying a source module tree and an XML tree.

Desktop publishing editor

Usage	A desktop publishing editor.
Start	DTPEditor.Open filename ~ Several editors may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free DTPEditor DTPView DTPUtilities DTPData ~
Components	DTPData.Mod, DTPUtilities.Mod, DTPView.Mod, DTPEditor.Mod DTPText.Mod, DTPRect.Mod, DTPImage.Mod / Demo.Style.XML, Demo.Layout.XML, Demo.Content.XML

Text style tool (font, font size, color, etc.)

Usage	
Start	WMTextTool.Open ~ Several file managers may be running in parallel. WMTextTool.SetFontSize "Absolute"|"IncrementBy" |"DecrementBy") [value] ~ WMTextTool.SetFontSize Absolute 20 ~ WMTextTool.SetFontStyle normal ~ WMTextTool.SetFontStyle bold ~ WMTextTool.SetFontName Courier ~ WMTextTool.SetFontName Oberon ~ WMTextTool.SetFontColor 0FF0000FFH ~ WMTextTool.SetFontColor 0FFH ~ WMTextTool.CountLines ^ ~ WMTextTool.CountWords ^ ~ WMTextTool.CountCharacters ^ ~ WMTextTool.CountAll ^ ~
Stop	Close the window(s). To clean up: SystemTools.Free WMTextTool ~
Restorable	
Components	WMTextTool.Mod

Useful hot keys commands *(not on the GUI)*

Usage	Three useful commands.
Start	HotKeysCommands.ClearLog ~ HotKeysCommands.EnterCommand ~ HotKeysCommands.SimulateMouse MouseWheel -3
Stop	To clean up: SystemTools.Free HotKeysCommands ~
Components	HotKeysCommands.Mod

OpenType True Type Fonts (TTF) detector *(not on the GUI)*

Usage	Searches through all the mounted file systems for True Type Fonts, listing them in the Kernel log. Example of output: *** TrueType MultiTester v0.1 *** Testing File: G:/Aos/source/VeraSeBd.ttf all ok Testing File: G:/Aos/source/Vera.ttf all ok Testing File: G:/Aos/source/benevento.ttf all ok Testing File: G:/Aos/source/VeraMoI.ttf all ok Testing File: G:/Aos/source/VeraI.ttf all ok Testing File: G:/Aos/source/VeraBd.ttf all ok Testing File: G:/Aos/source/VeraBI.ttf all ok Testing File: G:/Aos/source/VeraMoBd.ttf all ok Testing File: G:/Aos/source/VeraSe.ttf all ok Testing File: G:/Aos/source/VeraMo.ttf all ok Testing File: G:/Aos/source/VeraMoBI.ttf all ok ..... *** all done *** 10 Bitstream Vera TTFs and benevento.ttf are freely available without copyright offence (FreeType).
Start	WMOTFonts.MultiTest ~
Stop	To clean up: SystemTools.Free WMOTFonts ~
Components	OpenTypeInt.Mod, OpenTypeScan.mod, OpenType.Mod, OpenTypeFonts.Mod, WMOTFonts.Mod

Bitstream Vera

Bitstream Vera from Bitstream – http://new.myfonts.com/foundry/bitstream/ is a True Type font with full hinting instructions, which improve its rendering quality on low-resolution devices such as computer monitors. It consists of serif, sans-serif and monospace fonts and covers only common punctuation and the Latin alphabet with some diacritics. Its liberal license allows others to make and distribute derivative works with some restrictions, and the DejaVu fonts (also Open Type) project is expanding it with additional glyphs.

Scriptorium Benevento

Benevento from Scriptorium - new.myfonts.com/foundry/scriptorium/ is a True Type font family with 3 fonts based on late Roman period Lombardic calligraphy.

Complete Unicode table

A large number of Unicode fonts are available from the Web. Alan Wood's Unicode Resources at www.alanwood.net/unicode/fonts.html is a good starting point for exploring the subject.

Usage	A table with all the Unicode characters: latin, cyrillic, arabic, hebraic, chinese, korean, japanese, etc. The program attempts to use the Bitstream Cyberbit font when installed (see Cyberbit TTF downloader) or else it uses the system defined Default font. Select a character to let the decimal and hexadecimal character value appear at the bottom. Conversely, enter a value (decimal or hexadecimal) to see to which character that corresponds. To insert a character in an editor, select the character, click „CopyCharacter", set the cursor and press Ctrl+V.
Start	WMCharMap.Open ~ At the right the entire array of 65'535 glyphs appears in lines of 16 glyphs/line. At the left an enlarged preview of a selected glyph appears (under certain conditions only). Several maps may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMCharMap ~
Components	WMCharMap.Mod

Unicode marker tool

Usage	??? Purpose ?
Start	WMUnicodeMarkerTool.Open ~ Several marker tools may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMUnicodeMarkerTool ~
Restorable	
Components	WMUnicodeMarkerTool.Mod

Cyberbit TTF downloader

This does not qualify as application, only as a demonstration on how a manual download using FTPClient, described in the Communication section, can be automated. It can also be downloaded using WMFTPClient (use then anonymous@the.net as user@password).

A2 supports Unicode effectively: PET and Notepad are Unicode-savvy applications.

Usage	The font file Cyberbit.ZIP is downloaded and stored locally. Only this larger file is downloaded. Unzip it to obtain Cyberbit.ttf (Size is 13.4 MB) The complete Unicode table can be viewed by executing WMCharMap.Open ~ Documentation in: ftp://ftp.netscape.com/pub/communicator/extras/fonts/windows/ReadMe.htm

	A professionally designed serif Open type font for Arabic, Cyrillic, Greek, Hebrew, Japanese, Korean, Western European and English, Eastern and Central European, Turkish, Baltic Rim, Simplified Chinese, Traditional Chinese, Thai and Vietnamese. It's free for non-commercial use.
Start	CyberbitNetInstall.Start ~
Stop	To clean up: SystemTools.Free CyberbitNetInstall FTPClient ~
Components	FTPClient.Mod, CyberbitNetInstall.Mod

Bitstream Cyberbit

Bitstream Cyberbit from Bitstream – http://new.myfonts.com/foundry/bitstream is a font family with 1 style of the Times Roman family. A font family encompasses different foundries' versions, or a foundry's different cut, of basically the same typeface design. It was developed by Bitstream to provide Unicode Consortium members with a large Unicode font for testing and development purposes and is free-ware for non-commercial uses.

It is a serif font with small finishing strokes at the end of the main stems, arms, and tails of characters (a sans-serif font does not have).

Based on Bitstream's Dutch 801 BT font family, it is a Unicode True Type Font including many of the typographic characters for most of the world's languages according to Unicode 2.0 standards:

Basic Latin/English letters (Latin)
West European diacritics (Latin 1)
Ligatures
Central Europe (Latin 2)
Baltic Rim (Latin 6)
Turkish (Latin 5)
Romanian
Vietnamese
Phonetic
Cyrillic
Greek Modern
Arabic
Hebrew
Thai
Hanzi/Kanji (Chinese/Japanese/Korean)
Other Open Type
Dingbats/Symbols

Three separate files: Cyberbit.ttf (complete font), Cyberbase.ttf (Cyberbit without the CJK) and CyberbitCJK,ttf (CJK only) can be downloaded from Netscape by FTP:
ftp://ftp.netscape.com/pub/communicator/extras/fonts/windows/

Hobbes' Chinese tool - Chinese character identifier

Requires to install the file Unihan.txt which is not included in the release. A complete copy of the Unihan database is available as a (very large) zipped text file on the Unicode Consortium's official ftp site unicode.org. This file includes all the data of the on-line database plus additional information. Information on how to parse the file is included in the file itself. For an overview, see the description of Unihan fields in the accompanying

Unihan.html file. Unihan.zip is sized 6.24 MB, the extracted Unihan.txt is 29'206 MB. Use WMFTPClient.Open ~ and connect to: ftp.unicode.org signing-in as "anonymous" and password <e-mail address> then follow the path: /Public/zipped/5.1.0/Unihan.zip

Usage	A tool to identify one out of 83201 chinese glyphs by interpreting it, with the help of the Unihan.txt file, into: pinyin, mandarin, cantonese, korean, to translate it approximately into english and finally to give its Unicode position. Conversely, given the code, the glyph is obtained. Observe the parallel with the Unicode character table described earlier.
Start	CharacterLineup.Open ~ The lower window lists all the available glyphs. When a glyph on the line above is selected, the lower part then shows the filtered-out glyphs containing the selected one. For each additional character selected the filtering is repeated. A selected glyph can be de-selected. This is best demonstrated by operating on the 3 first glyphs at the left in different combinations. Follow the Kernel Log output to see how many glyphs are still composed with the selected glyphs while the selection progresses.
Stop	To clean up: SystemTools.Free CharacterLineup UnihanParser ~
Components	UnihanParser.Mod, CharacterLineup.Mod

Chinese Composite Glyphs - CCG

Created by eForth Technology in Taiwan http://www.eforth.com.tw/
Details in: http://www.eforth.com.tw/CT/Efeditor/index1.html

Utilities

File manager

Usage	A file manager with at the left a tree structure of all the mounted files systems and on the right a list of the files of the currently selected file system. The list is empty otherwise.
Start	WMFileManager.Open ~ Several file managers may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMFileManager ~
Restorable	
Components	WMFileManager.Mod

Text converter *(not on the GUI)*

Usage	Convert a text to the Oberon text format. The Oberon text format is used for source text and has the characteristic of collecting formatting information in a header. The compiler can thus skip over the header and can therefore parse the remaining plain ASCII text faster.
Start	TextConverter.Oberon filename ~ The converted text takes over the name of the input file. An acknowledgment of

	the conversion appears in the KernelLog, possibly with a diagnostic of what could not be converted.
Stop	To clean up: SystemTools.Free TextConverter ~
Components	TextConverter.Mod

Menu page constructor *(not on the GUI)*

Usage	Construct an XML menu page as used by the Main menu.
Start	MenuPages.Generate pagename menuname { buttonCaption command [bgColor [hoverColor]] } ~ pagename is the name of the XML file bgColor is the color of the button hoberColor is the color of the button when hovering on it with the mouse cursor
Stop	To clean up: SystemTools.Free MenuPages ~
Components	MenuPages.Mod

CD recorder

Usage	Multi-functional CD burner for two different project types: audio and data. A project must be created first. Audio Project: list of songs ... ?? Data Project: with these session types No Multisession StartMultisession Continue Multisession Finish Multisession Boot Burn a project The following tools are integrated: Burn Iso Image Copy Data CD Blank CDRW Disc Information
Start	WMCDRecorder.Open ~ Several CD recorders may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMCDRecorder ~
Components	CDRecordUtils.Mod, CDRecordLib.Mod, MakeIsoImages.Mod, CDRecord.Mod, WMCDRecorder.Mod

Activity log

Usage	Used by A2 to record system activity. The data is in fact produced by program statements introduced by programmers. The amount of data can be in some programs controlled by some kind of Debug constant set to TRUE or FALSE. Since the log is a simple editor, it can be used as a scratch pad for entering commands.

Start	WMKernelLog.Open ~
Stop	Close the window. To clean up: SystemTools.Free WMKernelLog ~
Components	KernelLogger.Mod WMKernelLog.Mod

Clock

Usage	A transparent digital or analog clock.
Start	WMClock.Open ~ At first, a digital clock (hh:mm:ss) is presented. A left mouse click opens a pop-up list offering 6 choices:
<table><tr><td>Close</td><td></td></tr><tr><td>Time</td><td>As hh:mm:ss</td></tr><tr><td>Date</td><td>As dd.mm.yy</td></tr><tr><td>Day of Week</td><td>As letters and digits "lll dd"</td></tr><tr><td>Analog</td><td>A clock with a face and fingers</td></tr><tr><td>Toggle Color</td><td>alternates the digital clock black and white</td></tr></table> The clock can be resized by holding the middle mouse button on it, then pressing and dragging the right mouse button. Several clocks may be running in parallel.
Stop	Let the pop-up window appear and click „Close". To clean up: SystemTools.Free WMClock ~
Restorable	Yes
Components	WMClock.Mod

Calendar

Usage	A transparent calendar of a month presented in a raster of 7 x 7 cells with a title.
Start	WMCalendar.Open ~ At first, the current month is presented. The current day appears in red. A left mouse click on one of the "<" or ">" buttons framing the month name allow moving from month to month forward or backward. A right mouse click on the calendar opens a pop-up list offering to close the calendar or to return to the current month. Several calendars may be opened.
Stop	Let the pop-pup window appear and click „Close". To clean up: SystemTools.Free WMCalendar ~
Restorable	Yes
Components	WMCalendar.Mod

Singleton overlay window for short-lived On-Screen message Display (OSD)

Usage	Alert the user by inserting a message during a specified number of ms inside of a blue overlay window at the top of, and as wide as, the screen. Used to tell which view port is visible in conjunction with hot keys F1 ... F4 (WMNavigate.SetViewportRange) and also used by WMPerfMonAlerts.
Start	WMOSD.Open message [ms duration] ~ WMOSD.Test ~ Perform endless self-test. To stop, unload the module. Several messages may be inserted in parallel.
Stop	WMOSD.Close ~ Close an open window. To clean up: SystemTools.Free WMOSD ~
Components	WMOSD.Mod

Desktop icon singleton

Usage	An expeditive tool for inserting a programmable icon with limited functionality on the desktop without programming. When the desktop is not stored, the icon disappears without trace.
Start	WMDesktopIcons.Open ~ The icon which appears on the desktop is empty and has no apparent function. Mouse commands allow the following: <table><tr><td>ML</td><td>Drag and move the icon</td></tr><tr><td>MR</td><td>let a pop-up menu appear offering the following: Close: Set Command: a dialog box ask to enter a command Set Image: a dialog box asks to enter an image name, from the icons.tar collection for instance Set Caption: a dialog box asks to enter a short text Toggle StayOnTop:</td></tr><tr><td>MM</td><td>execute a command if it was defined earlier via the pop-up menu</td></tr><tr><td>ML+MR</td><td>resize the icon</td></tr></table> Several icons may be inserted.
Stop	Close the window(s) via the pop-up menu. To clean up: SystemTools.Free WMDesktopIcons ~
Restorable	
Components	WMDesktopIcons.Mod

Screen shot utility for taking screen shots of the desktop or part of it

Usage	Take a snapshot a specific desktop area, scale it as specified and save it in a file. The saved snapshot can be viewed with: WMPicView.Open filename ~ a command that is anyway automatically placed at the end of the Kernel log.
Start	WMScreenShot.SnapShotView filename [viewname] [width [height]] ~ takes a snap shot of a specified view, scales it to width and height, and stores it in the named file. When height (or width and height) are missing, no scaling is done and the the view size is used by default. When viewname is omitted the default view, that is the desktop, is View#0, is used. The filename extension can be .bmp

	or gif. Examples: WMScreenShot.SnapShotView xyz ~ WMScreenShot.SnapShotView xyz 100 100 ~ WMScreenShot.SnapShotView xyz View#0 200 ~ WMScreenShot.SnapShotRange filename width height [(left top) | (left top width height)] ~ takes a snapshot of a display area, scales it to width and height, and stores it in a file with the specified filename – Example: WMScreenShot.SnapShotRange test.bmp 300 300 -100 -100 300 300 ~
Stop	To clean up: SystemTools.Free WMScreenShot ~
Components	WMScreenShot.Mod

Display a picture stored in a file or convert it to another format

Usage	Display a picture stored in a file or convert it to another format, provided the appropriate Codec(s) is found. The available image Codecs defined in Configuration.XML are: Decoder for: bmp, gif, png, jpeg, jpg, jp2, svg Encoder for: bmp, gif
Start	WMPicView.Open filename.extension ~ WMPicView.Convert sourcefile destinationfile ~
Stop	To clean up: SystemTools.Free WMPicView ~
Components	WMPicView.Mod

Input method editors for various language and keyboards

File archiving application

Usage	A file archiving with at the left a tree structure of the resources and on the right a list of the archived files.
Start	WMArchives.Open ~ Several file archiving GUIs may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMArchives ~
Restorable	
Components	Archives.Mod WMArchives.Mod

Tar processor (not on the GUI)

Usage	Pack several files into a single one.

Start	Tar.Create <filename.tar> {<filename>} ~ Tar.List <filename.tar> ~ Tar.Extract <filename.tar> ~ For the 2 last commands, the names of the component files are enumerated in the Kernel Log.
Stop	To clean up: SystemTools.Free Tar ~
Components	Archives.Mod, Tar.Mod

Developer tools

The program editing tool listed above with an integrated compiler

Shell

Usage	A graphical command interpreter, as another way to issue commands instead of typing commands wherever an editor is available on the desktop, as is intrinsically possible in A2, for these reasons: When no suitable editor is seen, one would have to open a Notepad for issuing a short-lived command and leave an extra window on the desktop. After having typed a command, the mouse is needed to launch its execution. In the shell instead, a carriage return is enough. A long lasting command execution can be observed until it terminates. For many commands the standard output is the Kernel Log which is not always present on the desktop. In the shell, the standard output becomes the BlueShell window itself and that can also be redirected to a file. Predefined command aliases facillitate the user's task.

Start	WMShell.Open ~ The user is prompted to enter a command after „SHELL>". At the same time as the BlueShell window opens a number of short, easy to remember command aliases are extracted and loaded from Shell.Alias. Some of the abbreviated commands require valid command parameters as is described elsewhere.
	 To get some assistance type „help" to see this:
	 <pre>--- Help ---
	alias: Show list of aliases
	alias 'string'='command': Create alias for command
	alias 'string': Remove alias
	batch: start a new instance of Shell
	clear: Clear screen
	version: Show BimboShell version
	help: Show this help text
	exit: Exit Shell
	SHELL></pre>
	 One observes that loaded aliases can be added and removed at will to a user's preference, but such changes do not alter the Shell.Alias file. The latter must be maintained with an editor. No space is allowed around the „=" in the definition of a new alias.
	 Several shells may be running in parallel.
	
	 The real power of the shell resides in possibilty to glue commands together to:
	 <u>Redirect</u> a command output to a file, with „>"
	 Example: Tar.List present.tar > anyname.Text

	Redirect and append a command output to a file, with „>>" Example: Tar.List PETIcons.tar >> anyname.Text Pipe a command output to another command, with „|" This does not work. Background command execution, with „&" When a command is entered at the prompt, the shell will wait until that command has completed before prompting again. This is called foreground processing. Only one command at a time can be running in the foreground in any shell window. Alternatively, by typing an ampersand „&" at the end of a command, it will run in background and even when the command takes some time to complete, the prompt will reappear immediately. Use CursorUp to retrieve earlier commands in reverse order and CursorDown to retrieve then in oder. The list is organized as a round-robin.
Stop	At the prompt, enter exit + CR Close the window(s). To clean up: SystemTools.Free WMShell Shell ~
Restorable	
Components	Shell.Mod WMShell.Mod / Shell.Alias

Search tool for searching a text stretch in a selection of files

Usage	Prompts to enter a search path, a selection of files specified with wild cards, and a text stretch (Content). Click „GO". The result will appear more or less rapidly depending on the number of files to explore and the text stretch length. Each file listed can be viewed by clicking on it. It is displayed using PET. The search can be prematurely halted by clicking „Stop".
Start	WMSearchTool.Open ~ Several search tools may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMSearchTool ~
Restorable	
Components	WMSearchTool.Mod

Search/replace tool for a text stretch in a selection of files *(not on the GUI)*

Usage	Search all occurrences of a text stretch in all files with a name matching a given pattern, or replace these occurrences by another text stretch. The text stretch may not contain a carrier return. The names of the files where the text stretch is found are enumerated in the Kernel log.

Start	SearchTools.Find [<Options>] <filePattern> <searchString> ~
	    Options = ["-" Option [{WhiteSpace "-" Option}]]
	    Option = „v" for verbose or „f" for formatted.
	Example:
	  SearchTools.Find -v -f E:/Repository/winaos/src/*.Mod Commands ~
	
	SearchTools.Replace [<Options>] <filePattern> <searchString> <replaceString> ~
	    Option = „v" for verbose

	Example:
	    SearchTools.Replace E:/Repository/source/*.Mod AosCommands Commands ~
	or this recently used case after having renamed Utilities.Mod to Strings.Mod:
	    SearchTools.Replace *.Mod Utilities. Strings. ~

Stop	To clean up: SystemTools.Free SearchTools ~
Components	SearchTools.Mod

Text comparison tool for comparing two texts in parallel (Left and Right)

Usage	Two texts are presented one at the left, the other at the right of a window with a scrollbar in the middle. Each line of both texts is numbered and where a discrepancy is detected the lines are offset and highlighted in red at the left and in blue on the right. The relative position of the differences is indicated by a red and respectively blue horizontal line at the left and at the right of the scrollbar. The total number of discrepancies is displayed at the bottom. The filenames appear in green when no difference is found and red otherwise. The two texts cannot be edited but, after having edited any of them, its filename can be re-entered and the comparison done by hitting „Diff". This application is a GUI extension of DiffLib.Mod described below.
Start	WMDiff.Open [<filenameL> <filenameR>] ~ If one of the files does not exist, the window remains empty. Several comparisons may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMDiff DiffLib ~
Components	DiffLib.Mod WMDiff.Mod

Text comparison tool (not on the GUI)

Usage	Two texts are compared line by line and the result appears in the Kernel log. Each line of both texts is prefixed with „<" or „>" depending if it belongs to the filename at the left or at the right and with (linenumber:position). When no difference is detected nothing appears in the log, except the filenames.
Start	DiffLib.Compare <filenameL> <filenameR> ~
Stop	To clean up: SystemTools.Free DiffLib ~
Components	DiffLib.Mod

Decoder of binary executable code

Usage	Uses DecoderRes.zip
Start	Decoder.OpenEmpty ~ Decoder.Open (<filename> | <progCounter>) ~ with filename suffix: Obw for i386 Oba for ARM Abx for AMD64 Several decoders may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free Decoder ~

Components	MemoryReader.Mod, Decoder.Mod, I386Decoder.Mod, ARMDecoder.Mod, AMD64Decoder.Mod

Component viewer

Usage	A tool for visualizing a visual component defined by an XML description. Example: Used by the FractalDemo application.
Start	ComponentViewer.Open filename.XML ~
Stop	Close the window. To clean up: SystemTools.Free ComponentViewer ~
Components	ComponentViewer.Mod

Windows command line interpreter activator

Usage	Allows to start A2 from a Windows command line and at the same time to open any number of text files in A2. In Windows, use one of these: | Open the command interpreter C:\WINDOWS\system32\cmd.exe	and execute aos {filename} With "aos" only, A2 is started and no more.
Select any number of text files and drag and drop them	onto Aos(.exe) in the WinAos directory	
Start	CommandLine.Open ~ It is useful to add the command to the Autostart section of Configuration.XML.
Stop	To clean up: SystemTools.Free CommandLine ~
Platform	WinAos only
Components	Win32.CommandLine.Mod

Communication

FTP client GUI, extension of FTPClient.Mod

Usage	The files accessible from the current path are listed. The connection state is indicated by a green/red square at the lower left.
Start	WMFTPClient.Open ~ Enter a host name in the field „Host" and a port number in „Port:" then hit „Connect". By default the host name is „bluebottle.ethz.ch" and the port is „21" . A two-step dialog asks to enter a Username and a Password. For the defaults the following apply: Username: ocp Password: download giving access to several release build images ready for download,

	as documented in http://www.bluebottle.ethz.ch/download.html Several connections may be running in parallel.
Stop	Hit the „Disconnect" button. Close the window(s). To clean up: SystemTools.Free WMFTPClient FTPClient ~
Components	FTPClient.Mod, WMFTPClient.Mod

FTP server

Usage	The FTP server implements most of RFC-765 and supports user accounts with different permissions and virtual root directories. Start the server first, then add and remove user with their authorizations at will. Users data is stored in WebFTPUsers.dat. FTP.log stores the data transfer activity log. The state of the Bluebottle FTP server where the deliverable material is available can be observed in: http://www.bluebottle.ethz.ch/serverstate.ssmp
Start	WebFTPServer.Start [„\l:" <logfilename>] ~ This confirmation message appears in Kernel Log: {P cpuid= 0, pid= 3628 Service 21 open 0} {P cpuid= 0, pid= 3628 WebFTPServer started} WebFTPServer started WebFTPServer.AddUser <username> <password> <max concurrent logins> <permissions> [<root>] ~ Here 2 examples: WebFTPServer.AddUser <username> <password> -1 rwp FAT:~ WebFTPServer.AddUser anonymous none 3 rwpm FAT:/ftproot/ ~ WebFTPServer.RemoveUser <user> ~ WebFTPServer.ListUsers ~
Stop	WebFTPServer.Stop ~
Components	WebFTPServer.Mod / FTP.Log, WebFTPUsers.dat

VNC client

Usage	Each window is entitled: „servername Port nnn – VNC i" where „i" is the session number.
Start	VNC.Open <servername> [<password>] „5901" ~ A wrong or missing password is acknowledged with: „Error 1". VNC.Show ~ The number of connections is reported in Kernel Log.
Stop	Close the window(s). To clean up: SystemTools.Free VNC ~
Components	VNC.Mod

VNC server viewer, extension of VNCServer

Usage	Each window is entitled: „servername – VNC i" where „i" is the session number.

Start	WMVNCView.Install <name> <password> <port> x y w h ~ name and password are strings optionally in " " use "" for no password Example: "Bluebottle VNC View1" "" 5903 0 0 1024 768~
Stop	Close the window(s). WMVNCView.Uninstall ~ To clean up: SystemTools.Free WMVNCView VNCServer ~
Components	VNCServer.Mod, WMVNCView.Mod

Synergy client

Synergy – http://synergy2.sourceforge.net - lets you share a single mouse and keyboard between multiple computers with different operating systems, each with its own display, without special hardware. It's intended for users with multiple computers on their desk since each system uses its own monitor(s). Redirecting the mouse and keyboard is as simple as moving the mouse off the edge of the screen.
Synergy consists of a server, which provides a keyboard and a mouse so to speak, and a client per "second" machine which can then be operated without input devices. The server and the clients may run different OSes (Windows, Mac OS X and Unix) in any combination. A2 offers only a client software. The computers must be connected to each via TCP/IP.

After having installed the Synergy server, it must be configured and started. The server and each client should be given a (screen) name for identification during the configuration. The name is irrelevant and an alias such as the machine's IP address is good enough.

Usage	Pick which keyboard and mouse will be shared. The computer with that keyboard and mouse is called the „primary screen" and will run the Synergy server. All of the other computers are „secondary screens" and run a Synergy client. Synergy also merges the clipboards of all the machines into one, allowing cut-and-paste between them. And it does it all in Unicode so any text can be copied.
Start	SynergyClient.Connect <serverName> <screenName> ~ serverName: the Windows name for instance or an IP address. screenName: name of the client screen as declared to the Synergy server when it was configured. Since A2 has no name, use an IP address.
Stop	SynergyClient.Close To clean up: SystemTools.Free SynergyClient ~
Components	SynergyClient.Mod

BlueTerminal for V24 communication

An indispensable testing tool for communicating between two machines during the start phase. Warning: it will not last long until this technology completely disappears from modern machines. We are almost there anyway, hence there is a need for an alternative solution, say via USB port.

Usage	A serial port must be available on the machine and a serial port driver must be installed by executing V24.Install ~ . Verify that it is ready by executing: Serials.Show ~

Start	WMV24Component.Open ~ Several BlueTerminals may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMV24Component ~
Restorable	
Components	WMV24Component.Mod

Telnet VT100 terminal

Usage	A network attachment must be available on the machine.
Start	WMVT100.Open <address> [<port>] ~ The address may be an IP address between hyphens e.g. „192.168.1.13" The default portnumber is 23.
Stop	Close the window. To clean up: SystemTools.Free WMVT100 ~
Components	WMVT100.Mod

IMAP/SMTP mail client

Usage	Uses IMAPIcons.tar
Start	IMAPGUI.Open ~
Stop	Close the window. To clean up: SystemTools.Free IMAPGUI ~
Restorable	
Components	IMAPUtilities.Mod, IMAP.Mod, IMAPClient.Mod, RMSMTP.Mod, IMAPGUI.Mod (should be named WMMail.Mod)

Multimedia

Sound cards/chips for which a driver exists

Vendor / ID	Chip	DeviceID	Driver	Install command (*)
Intel / 8086	i810 chipset (has integrated AC'97 controller)		i810Sound.Mod	i810.Install ~
Ensoniq / 1274	ES1371 ES1373 5880 Audio PCI ES1371 on Creative SoundBlaster PCI 128	1371 1373 5880 ???	EnsoniqSound.Mod	Ensoniq.Install ~
Yamaha / 1073	YMF724 YMF724F YMF740 YMF740C YMF744	0004 000D 000A 000C 0010	YMF754.Mod YMF754Util.Mod YMF754.Bin	YMF754.Install ~

	YMF754	0012		

(*) These commands can be launched automatically either:
  by placing them in the Autostart section of Configuration.XML or
  by inserting "Boot=PCITools.DetectHardware" in the Configuration strings

Yamaha: The driver requires the microcode YMF754.Bin for the controller and the DSP.
The microcode can be retrieved from the Alsa server at:
ftp://ftp.alsa-project.org/pub/manuals/yamaha/pci/724hwmcode.c
and converted to YMF754.Bin by YMF754Util.Mod

Sound mixer

Usage	A sound device driver must be installed already.
Start	WMMixer.Open ~ Several sound mixers may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMMixer ~
Components	WMMixer.Mod

Ogg Vorbis player

made of: WMOGGPlayer.Mod, OGGUtilities.Mod and OGGRadios.Text. The latter lists
  - URLs (pre-requisite: communication driver installed)
  - or local file names having .ogg as extension

Sound stream Ogg Vorbis Codec: http://www.vorbis.org and http://vorbis.audiohq.de
The stream server accessed are mostly based on icecast - http://www.icecast.org

Television driver (not on the GUI)

Usage	A driver for Hauppauge television cards.
Start	BT848.Install ~
Stop	To clean up: SystemTools.Free TVDriver BT848 ~
Platform	A2 only.
Components	BT848.Mod, TVDriver.Mod

Television viewer

Usage	A television viewer for a specific Hauppauge card with a BT848 chip. This chip is antiquated and difficult to acquire. A TV device driver must be installed already.
Start	TV.Open [[cardNumber] TXT] ~

	The optional parameters are: cardNumber: to use when more than one card is installed TXT: provided a teletext decoder is built-in, start teletext capturing at open time. After that, the TV channel is switched every 5 minutes. This can be used for automatic Teletext caching, e.g. on a web server. TV.BuildChannelTable ~ start a full scan and build the channel table TVChannels.XML automatically.
Stop	The window cannot be closed. To clean up: SystemTools.Free TV TVChannels ~
Restorable	
Platform	A2 only
Components	TVChannels.Mod, TV.Mod / TVChannels.XML

Teletext viewer

Usage	A teletext viewer for a specific Hauppauge card with a teletext decoder. A TV device driver must be installed already.
Start	TeletextViewer.Open ~
Stop	The window cannot be closed. To clean up: SystemTools.Free TeletextViewer TeletextBrowser TeletextFont ~
Restorable	
Platform	A2 only
Components	TeletextDecoder, TeletextFont.Mod, TeletextBrowser.Mod, TeletextViewer.Mod / teletext.bfnt, next.png, prev.png, refresh.png

HTTP server for teletext access through a Web browser *(not on the GUI)*

Usage	To enable Server Side Includes and servicing dynamic the captured teletext pages as web pages, insert two new sections in Configuration.XML:
	    <Section name="DynamicWebpages">
	        <Section name="ActiveElementModules">
	            <Setting value="WebStd"/>
	            <Setting value="WebTeletextViewer"/>
	        </Section>
	    </Section>
	    <Section name="PrevalenceSystem">
	        <Section name="PersistentObjectModules">
	            <Setting value="WebStd"/>
	        </Section>
	    </Section>
	
	To view the teletext pages with a Web browser, enter the host name or its IP address followed by /teletext.dxp
	Example: http://192.168.1.33/teletext.dxp

Start	WebHTTPServerTools.Start \r:AOS \l:AOS:/HTTP.Log ~ followed by: DynamicWebpagePlugin.Install ~
Stop	DynamicWebpagePlugin.Uninstall ~ followed by: WebHTTPServerTools.Stop ~

Platform	A2 only.
Components	SystemTools.Free DynamicWebpagePlugin WebHTTPServerTools WebHTTPServer ~

System inspection and performance measurement

TCP tracker

Usage	A table listing of the recognized connections appears on the desktop. A number of informations is given for each connection: Remote address, Local Port, State (Opened, Closed, ...), etc After having selected one of those connections, it can be either closed or discarded.
Start	WMTCPTracker.Open ~
Stop	Close the window. WMTCPTracker.Close ~ To clean up: SystemTools.Free WMObjectTracker ~
Restorable	
Components	WMTCPTracker.Mod

Objects tracker

Usage	A table listing all the currently active objects appears on the desktop. In the absence of a module name a dialog window asking for it appears.
Start	WMObjectTracker.Open ~
Stop	Close the window. WMObjectTracker.Close ~ To clean up: SystemTools.Free WMObjectTracker ~
Restorable	
Components	WMObjectTracker.Mod

Module state inspector

Usage	The state of the named module appears in a desktop window.
Start	WMModuleState.Open modulename ~
Stop	Close the window. To clean up: SystemTools.Free WMModuleState ~
Components	WMModuleState.Mod

USB hardware inspector

Usage	A tree listing the detected USB devices appears in a desktop window. Three different data views are available: * Standard * Detailed * Drivers : lists the USB devices registered and shows which devices are bound to drivers. When external USB devices are inserted or removed, the views can be refreshed to reflect the new situation. Uses the data file WMUsbInfo.tar containing a number of mini-icons and a list of USB devices vendor Ids (that data is available at www.usb.org).
Start	WMUsbInfo.Open ~ Several inspectors may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMUsbInfo ~
Restorable	
Platform	Not available in WinAos
Components	WMUsbInfo.Mod / WMUsbInfo.tar

System performance monitor

made of: WMPerfMon.Mod ++

Event log

made of: WMEventLog.Mod

Keyboard key code inspector

Usage	The key code corresponding to a key stroke (key pressed or key released) is displayed. Key combinations with Ctrl, Alt or Meta-key included. Instead of reading the evanescent display information, one may direct the information to the KernelLog. Example with the 'm' key: Key: UCS=0000006D, KeySym=0000006D (No Keysym), Key: 'm', Flags= Key: UCS=00000000, KeySym=00FFFFFF (No Key), Key: '', Flags=[RELEASE]
Start	WMKeyCode.Open ~ WMKeyCode.StartLog ~ Direct the information to the KernelLog. WMKeyCode.StopLog ~ Stop sending information to the KernelLog.
Stop	Close the window. To clean up: SystemTools.Free WMKeyCode ~
Restorable	
Components	WMKeyCode.Mod

Example programs

These examples demonstrate how simple it is to start programming in Active Oberon using

the already available infrastructure.

Text writer

Usage	An example text editor, though not full-fledged since the text cannot be saved as a document. Uses WMEditors.
Start	ExampleTextWriter.Open ~
Stop	Close the window. To clean up: SystemTools.Free ExampleTextWriter ~
Components	ExampleTextWriter.Mod

Drawing pad

Usage	An example drawing pad using the mouse as a pen. Uses WMGraphics and WMDialogs
Start	WMScribble.Open ~ Press "s" to store the scribble. A dialog window then appears, asking for a filename. The default "scribble.bmp"is offered. Use PicView.Open filename ~ to visualize.
Stop	Close the window. To clean up: SystemTools.Free WMScribble ~
Components	WMScribble.Mod

Graphic animation

Usage	An example graphical application using a picture of the ETHZ. Uses WMGraphics.
Start	WMGraphicsDemo.Open ~
Stop	Close the window. To clean up: SystemTools.Free WMGraphicsDemo ~
Components	WMGraphicsDemo.Mod / BluebottlePic0.png

Visual components test bed

Usage	A test bed for some of the visual components, similar to the "Component viewer" described above, but less general since the components used are hard-coded.
Start	TestComponents.TestStandardComponents ~ Presents a window containing a Panel with a Label, and a toolbar (in fact another Panel) with two captioned Buttons. When clicked, the Buttons send a short message to the KernelLog. TestComponents.TestStringGrids ~ Presents a window containing a Panel with a Label, and a toolbar (in fact another Panel) with a captioned Button. When the Button is clicked, a 4 by 10 grid appears, the top 20 cells being numbered from 0 to 19. TestComponents.Test name ~ Presents a window containing a visual component. "name" must refer to a valid visual component.

	Several component testers may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free TestComponents ~
Components	TestComponents.Mod

Drag and drop test program

Usage	Open the Kernel Log first. Position the cursor in the blue or green area, press the left mouse button, drag the cursor within the container panel and observe the information appearing in the log. A middle or left mouse click opens a dialog. Close it.
Start	TestComponenDragDrop.Open ~
Stop	Close the window. To clean up: SystemTools.Free TestComponenDragDrop ~
Components	TestComponenDragDrop.Mod

An example program for testing WMtree components as are used in PET, made of: TestTrees.Mod

Pie Menu test

Usage	Only the central dark green panel reacts to a right mouse button pressing: a pie selector appears. Uses WMPieMenu.
Start	PieTest.Open ~ Several tests may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free PieTest ~
Components	Pietest.Mod

An example program for testing WM menu components made of: TestMenu.Mod, WMMenus.Mod using Menu.XML

Component test bed PresentViewer

Usage	A test bed for a fantasy menu representing a heap of presents to be opened. When clicked with the middle mouse button, each present opens an application chosen from the ones described in this paper. The fantasy menu is described by XmasMenu.XML.
Start	PresentViewer.Open XmasMenu.XML ~
Stop	Close the window. To clean up: SystemTools.Free PresentViewer ~
Components	PresentViewer.Mod / XmasMenu.XML, present.tar

Menu editor

Usage	An editor of a fantasy menu of the kind of XmasMenu.XML used in PresentViewer.

Start	MenuEdit.Open ~
	The toolbar at the top of the window allows the following:
	
	- Add : a present to the menu in construction. The present can be:
	    dragged in the drawing area or
	    selected with a middle mouse click
	- Delete: delete the selected present
	- To Front: move the selected present to the front
	- Edit: add a caption to the selected present
	- GetXML: display the XML text elaborated to this point
	
	There remains to copy/paste the final XML text to an editor and to store the document in a file. This file can be exploited by the PresentViewer.
	
	Several menu editors may be running in parallel.

Stop	Close the window(s). To clean up: SystemTools.Free MenuEdit ~
Components	MenuEdit.Mod / present.tar

Mail reader

Usage	??
Start	BimboMail.Open ~ Several mail readers may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free BimboMail ~
Components	MailStorage.Mod, BimboMail.Mod

Demonstration

Vectorized/rasterized 3D menu

Usage	A 3D menu opens on the desktop. Uses the data files W3dFun.XML, W3dMenu.XML, W3dNetTools.XML, W3dPersonal.XML All of these XML files extract images from W3dMenuIcons.tar W3dClusterWatch.Mod extracts images from W3dClusterWatchIcons.tar
Start	W3dMenu.Open filename ~
Stop	Close the window. To clean up: SystemTools.Free W3dMenu ~
Components	W3dVectors.Mod, W3dMatrix.Mod, W3dGeometry.Mod, W3dAbstractWorld.Mod, W3dObjectGenerator.Mod, W3dRasterizer.Mod, W3dWorld.Mod, W3dExplorer, W3dClusterWatch, W3dMenu

3D viewer

Usage	A 3D world opens on the desktop. Press the left mouse button and move the mouse to rotate the axis. Additionally, press the right button mouse to zoom.

	These key strokes condition the image: (and the meaning is?) „0" affine „1" subdivision „2" perspective „3" wire frame „s" speedy rotation Uses the data file BluebottlePic0.png .
Start	W3dExplorer.Open ~
Stop	Close the window. To clean up: SystemTools.Free W3dExplorer ~
Components	W3dVectors.Mod, W3dMatrix.Mod, W3dGeometry.Mod, W3dAbstractWorld.Mod, W3dObjectGenerator.Mod, W3dRasterizer.Mod, W3dWorld.Mod, W3dExplorer.Mod

Slide show

Usage	A simple slides how presentation tool with transitions effects. Keyboard and mouse controls: Next Spacebar/LeftMouseButton/PageDown/RightArrow Previous PageUp/LeftArrow First Home/UpArrow Last End/DownArrow Exit ESC (Re)Open navigation panel "n" (Re)Open slide window "w" Show/Dump internal file list "l"
Start	WMSlideshow.Open [filename] ~
Stop	Close the window. To clean up: SystemTools.Free WMSlideshow WMTransitions ~
Components	I386.WMTransitions, WMSlideshow.Mod / RetoSlideshow.XML, SlideShowData.tar

Fractal

Usage	A demonstration of
Start	The demo requires to execute: FractalDemo.Register ~ followed by ComponentViewer.Open FractalDemo.XML ~
Stop	Close the window. To clean up: SystemTools.Free ComponentViewer FractalDemo ~
Components	FractalDemo.Mod / FractalDemo.XML

Turing

Usage	
Start	TuringCoatWnd.Open ~ TuringCoatWnd.OpenAlpha ~ Several Turing animations may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free TuringCoatWnd ~
Components	TuringCoatWnd.Mod / WMIcons.tar

Fractal voxel ray tracer

A contribution of Soren Renner. Modules prefixed with "sr".

Up to 38 ray tracer videos can be seen on:
http://www.youtube.com/profile?user=xenopusRTRT&view=videos
where Soren Renner signs as XenopusRTRT. The video "filtor 2.2" demonstrates that the videos were developed with A2.

The files required to run the tracer are collected in tracer.Zip.

Games

These are developed as additional program examples and not for using an A2 system as a gaming console. Games are of course welcome distractions and more would be welcome.

Tetris

Can serve as example of WMGraphics use.

Rules	A familiar tetris game window appears. The peculiarity of this GUI component is that it is transparent. Press the „Space" bar to start. The shape of the next block to fall from the top is announced at the top left. The cursor positioning keys have the following functions: Cusor right: move the block to the right Cursor left: move the block to the left Cursor up: rotate the block 90 degrees clockwise Cursor down: drop the block immediately Press „p" to Pause and resume. The score is given in number of lines filled, number of blocks used, level ?, points scored. The game is over when blocks are piled up to the top.
Start	WMTetris.Open ~ Several Tetris games may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMTetris ~
Components	WMTetris.Mod

Tetris client of a Tetris server

Rules	The same Tetris game as described previously except that the game starts immediately. Each window is entitled: „servername – VNC i" where „i" is the session number. The state of the game appears at the bottom: Score: Games active: Mac concurrent: Served total: High score: Press p to toggle pause
Start	VNC.Open servername [password] 5999 ~ the password is necassary but any is valid.
Stop	Close the window(s). To clean up: SystemTools.Free VNC ~
Components	VNC.Mod

Tetris server (extension of VNCServer.Mod)

Rules	The Tetris game server which can be accessed by any VNC client connecting to port 5999.
Start	VNCTetrisServer.Run ~ This confirmation message appears in Kernel Log: „VNC Tetris server started" Whenever a Tetris game terminates a status report line with the score is reported in the log. VNCTetrisServer.StopNew ~
Stop	VNCTetrisServer.Uninstall ~ To clean up: SystemTools.Free VNCTetrisServer VNCServer ~
Components	VNCServer.Mod, VNCTetrisServer.Mod

Bimso

Can serve as example of WMGraphics and WMDialogs use.

Rules	A memory test game. On pressing Start, a succession of flashing signals are emitted by four colored fields. Try to memorize their order. When no more signal is perceived, try to mimic the signal serie by clicking the colored fields. At the first mistake, the games stops and the level of dexterity is reported in a dialog window.
Start	Bimso.Open ~
Stop	Close the window(s). To clean up: SystemTools.Free Bimso ~
Components	Bimso.Mod

Color (or Colored) lines

Can serve as example of WMGraphics and WMDialogs use.

Rules	On a 19x19 board, 3 colored balls appear. Try to arrange balls of the same color in vertical, horizontal or diagonal lines. To move a ball, click on it to select, then click on a destination square. A ball can only move vertically and horizontally along free paths. Once a line has 4 or more balls of the same color, the line is removed from the board and the next move is allowed. When the move does not lead to a line removal, three balls of random color are randomly added to the board which becomes congested. The game is over when the board is filled up.
Start	WMColorLines.Open ~
Stop	SystemTools.Free WMColorLines ~ Remark: Cannot restart after simply closing the window.
Components	WMColorLines.Mod

Fun

Animated images

Purpose	Insert animated images on the desktop. Images are extracted from WMBunnyImages.tar are moving on the desktop from left to right.
Start	WMBunny.Insert <filename> [<nofFrames> <step>] ~ Default values for nofFrames = 8 and for step = 32 Several images may be inserted.
Stop	Removing an image requires some dexterity: While moving, try a ML-MR interclick in the approximative image area. With a bit of luck it is deleted. WMBunny.Free ~ is the better way to kill all images.
Components	WMBunny.Mod

Christmas snow

Purpose	Let it snow two small gifts icons Flake1.png and Flake2.png from the top of the screen in random order. The icons are extracted from xmas04.tar created on the occasion of Christmas 2004.
Start	Snow.Snow [<nofFlakes>] ~ Default value is DefaultNofFlakes = 20
Stop	SystemTools.Free Snow ~
Components	Snow.Mod / xmas04.tar