A2 user guide and applications description

November 23, 2009 draft

Table of Content

What is A2?

A2 is the name of an an operating system and an integrated software environment developed at ETH in Zürich. It is a single-user, multi-core, multi-tasking system that runs on bare hardware or on top of a host operating system.

The developers aim at producing a reliable, real-time operating system suitable for embedded systems and for industrial and in particular medical applications.

Earlier, A2 was called “Aos” (Active Object System), a nomenclature that is still in use. It is written in the Active Oberon programming language, which evolved from Oberon, a programming language in the Pascal/Modula tradition. The graphical user interface is referred to as “Bluebottle”.

These inter-related web sites provide introductory information:

1. Bluebottle www.bluebottle.ethz.ch the starting point
2. Oberon Community platform www.ocp.inf.ethz.ch with a wiki and a user forum
3. Oberon www.oberon.ethz.ch original site of the ETH Oberon

After having started A2, Tutorial.Text is a concentrated user guide to be used as a quick reference.

Document structure

This document presents some of the A2 applications (further ones will be added as time goes) where each application is summarized in a table, aiming at being sufficiently informative, simple, compact and uniform across all applications.

The production of the A2 deliverables, described later as “builds”, is conducted by a script stored in the Release.Tool file, which serves as input to the central Release.Mod program. Therefore, it seemed appropriate to write this documentation based on two well recognizable sections, called “packages”, of the Release.Tool text, namely “GuiApplicationsMini” and ”GuiApplications” responsible for compiling the applications documented here.

Many passages of this text are either extracted or adapted from Thomas Frey's authoritative Ph.D. thesis

Bluebottle: A Thread-safe Multimedia and GUI Framework for Active Oberon
http://e-collection.ethbib.ethz.ch/eserv/eth:27966/eth-27966-01.pdf - Abstract
http://e-collection.ethbib.ethz.ch/eserv/eth:27966/eth-27966-02.pdf - Full text

A reader should be aware that A2 has still evolved since its inception (2000-2005).

The table documenting an application has the following structure:

Usage	Purpose, concise description and general behavior of the application which manifests itself often as a window inserted on the desktop. A number of applications without GUI are also described, usually supporting a GUI application. These applications are then controlled exclusively by commands. An expert user can dispense with the GUI and commands can be conveniently batched.

<table>
<tr><td>Start</td><td>Lists the command which starts the application and possibly a few more commands providing additional functionality.
Since the mouse is the input device for controlling a GUI, the following conventional abbreviations are defined:<table><tr><td>ML</td><td>Left mouse button</td></tr><tr><td>MM</td><td>Middle mouse button</td></tr><tr><td>MR</td><td>Right mouse button</td></tr></table>These apply to a 3-button mouse which is best suited to control A2. In the following it is shown how to use a 2-button mouse in combination with keyboard keys. A keyboard key is then used as a substitute for MM.

Shortcut: Main menu → .. → .. the path to follow for starting the application.</td></tr>
<tr><td>Stop</td><td>How to stop the application. For a GUI application, it suffices to close the window. None of the object module involved is then unloaded.
Also, how to unload the top level modules involved, thereby closing all similar windows, if any. When more than one module is participating in the application, the modules are listed in the order in which to unload them orderly, that is, the top module comes first. The order in which to compile then is exactly the reverse.
A shorter method for unloading several modules is to use SystemTools.FreeDownTo moduleName ~. This command frees all the indirectly imported modules, although this is somewhat dangerous.</td></tr>
<tr><td>Restorable</td><td>This table entry appears only when the GUI application is restorable to the desktop: the application module was then programmed to be such.
If the user saves the desktop (a button on the Main menu is provided), data on the 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”.</td></tr>
<tr><td>Build</td><td>This table entry appears only when the application is not available/ported to a specified build.

Example: WMUsbInfo is not portable to WinAos since USB devices are controlled by the host.</td></tr>
<tr><td>Data files</td><td>When applicable, data files supporting the application are listed. Some of them contain icons or images allowing some customization of the GUI.</td></tr>
</table>

A number of application descriptions are supplemented with screen shots previewing what a user is expected to see on a live A2 system. Each image is followed by a screenshot command line used to capture it. Most were obtained using a WinAos system.

Run A2 from a live CD

Now is the time to start A2 and to practice with it. A2 need not be installed. Instead, booting from a CD-ROM, A2 will not interfere with the resident operating system and a user will play safe. However, do not mount a FAT file system and play with it while controlling A2, unless it is done purposely. Insert a live CD-ROM created from the ISO CD-image (see section ”Packages, Release builds and Ports) and boot a PC from it, using the boot menu. A2 is customized in such a way that the odds are good for a correct start. The hardware equipment of the machine will be inspected and the available drivers will be installed automatically.

In the case of failure, seek help in the “Installation” text of the Oberon Community Platform:

http://www.ocp.inf.ethz.ch/wiki/Documentation/Installation

Desktop

When A2 starts, the screen presents the Main menu (documented in a follow-up section) and a number of windows depending on how the system is customized and depending on the number of restorable applications that were active when the desktop was stored in an earlier session. In case A2 is started from a live CD, as suggested above, customization is hardly an option and none of the changes will be persistent.

Virtual desktop

The desktop is more than just the screen area. It extends well beyond the physical boundary of the screen which is only the visible part of a conceptually unlimited two-dimensional *display space* in which an arbitrary number *of windows* and other arbitrarily shaped objects can be situated. In the practical implementation, the display space is limited to the range of 32-bit signed integers for the number of pixels. This display space, call it *"virtual desktop"*, can be viewed as:

- either as a conceptual raster of screen-sized areas. The units of the rectangular coordinates system are the screen width and height. The coordinates of screen-sized areas are relative to "standard view port".
  When A2 is started, the top left corner of the screen represents the origin of the coordinates and the "standard view port" is visible.

-1, -1	0, -1	1, -1
-1, 0	0, 0 Standard view port	1, 0

- or a pixels raster. The units of the rectangular coordinates system are pixels. The coordinates of a point in space are relative to the top left corner of the screen.

All desktop areas can be made visible by navigating in the desktop as described below.

View port

A view port is the portion of a rectangular area of the display space made visible on the screen. Such a rectangular area is addressable either pixel-wise, as will be explained below, or by the coordinates of screen-sized rectangles, as used in some applications.

Desktop navigation (generic)

Here, the concept of view port is used without reference to the screen-sized raster. A view port can be moved and resized/scaled by r*epeating the actions* described next.

To move the view port, press the meta key and move the mouse or one of the direction keys as follows:

Meta+Mouse Move (1)	The view port observes the mouse pointer and if the cursor reaches the

	screen border, the visible view port disappears progressively in the opposite direction of the mouse movement and is replaced pixel-wise by the neighboring view port.
Meta+Left	Display the view port at the left.
Meta+Right	Display the view port at the right.
Meta+Up	Display the view port on top.
Meta+Down	Display the view port below.

To resize the view port, press the meta key and turn the mouse wheel up or down or press another key as follows:

Meta+Mouse Wheel (2)	Zoom desktop in & out, by a factor of 0.4, keeping the screen center in position. Remark: not all wheel notches cause resizing. (cannot explain) The alternative to a mouse wheel is to use PageUp or PageDown.
Meta+PageUp	Zoom in 2x, keeping the screen center in position. For a finer grain zooming, use the mouse wheel.
Meta+PageDown	Zoom out 2x, keeping the screen center in position. For a finer grain zooming, use the mouse wheel.
Meta+Home	By an adequate zooming and movement of the view port the used portion of the desktop becomes visible, so as to get an overview of the entire desktop. Useful in case of being lost, to smoothly “motor” zoom into an overview of all windows. A follow-up action could be Meta+ML on a window (see below). Alternatively, use the command WMNavigator.Open ~.
Meta+End	Set the zoom factor to 1, keeping the screen center in position..
Meta+ML on window	Select a window of interest with a ML click. The view port moves so that the window is placed at the top left. If it fits to the screen, the zoom factor is set to 1 so that the window is visible without scaling. Otherwise, the zoom factor is adjusted so that the entire window is visible.

In all situations the Main menu stays at the bottom left of the screen. In case it is hidden by window(s) placed on top of it:

Meta+Esc	Summon the Main menu on top.

Key equivalences:

Meta	**Alt-Shift,** but in WinAos not for (1) and (2)
Meta	**Menu** in WinAos

Windows

[WMScreenShot.SnapShotRange text.bmp 300 60 0 0 ~]

Windows are decorated with four special frame objects, one for the title, two for the sides and one for the bottom. These frame objects offer a meta area that allows the user to manipulate the window in the display space by responding to pointer events. They serve as handles to move, resize and change the z-order (overlapping) of windows. The title includes, at the left, a mini-icon categorizing the window and a descriptive text and, at the right, a hot-spot (bullet) for closing the window with a ML, MM or MR click. Two sets of images and colors are used for the frames: one for the an *active window*, the other for an inactive window. The look of windows, such as the shape of frame objects, the color of buttons, the shape of cursors is implemented in skins (see Skin loader). At any time, there is only one active window, the window in focus with its title and sides that glow brighter.

Even though the A2 metaphor with its large virtual desktop suggests to spread application windows instead of stacking them on top of each other, it is still important for the work-flow to support overlapping. The management of overlapping differs from other window managers.

When the pointer is located on:

1. the border (one of the frame objects) of an inactive window, a MM or ML click activates the window but does not cause it to move to the front (does not change the overlapping). This prevents hiding the information contained in a window placed on top of the clicked window. A double click is needed to move it to the front. A feedback is given by the brighter border.
2. on a window border, pressing the MM or ML key and dragging causes a window resizing. The border is moved and when the pointer is on a corner two borders are moved (the cursor shape reveals what reshaping to expect). Release the key to end reshaping.
3. the window title, pressing the MM or ML key and dragging causes the window to move within the display area. The cursor takes the shape of a cross with 4 arrows. When the cursor reaches a display border, a portion of the window may have disappeared in neighboring view port(s). Release the key to end.

The presence of frame objects is conditioned by the program instantiating a window. Unframed windows are rare but then an artifact, such as providing a “.Close” command, is needed. WMClock and WMCalendar are examples.

These are the basics of the desktop navigation. Applications facilitating navigation are described further on.

Colors and transparency

A2 uses colors extensively either solid or semi-transparent. Colors are stored as RGBA values (red, green, blue and alpha components). The additional alpha channel determines to what level a color is solid or semi-transparent, an alpha channel value of:

- 0 meaning “completely transparent”
- 255 meaning “solid”.

The normal case is that colors are defined in programs but there exist a few applications where the user is responsible for assigning such values, e.g. Menu page generator.

Commands control A2

A user controls A2 with commands. A *command* is, by construction and by convention, an exported procedure in a module written in the programming language *Active Oberon* with the purpose of doing some processing and to change the system state. A *CommandName* is a qualified identifier composed of a *ModuleName* and of the *ProcedureName* of a procedure exported by the named module. The command name is then *valid*. For example, PET.Open denotes the exported procedure Open from the module PET (Programmer's Editing Tool).

Formal definition of a command:

```
Command = CommandName [Parameters] ["~"]
CommandName = ModuleName "." [ ProcedureName ]
Parameters = {Separator} AnyChar
Separator = " " | LineFeed | CarriageReturn | Tab
AnyChar = { 0X..FFX except "~" }
```

The following conditions apply:

```
ModuleName and ProcedureName are case-sensitive
LEN(ModuleName) <= 32 , defined by Modules.Name
LEN(CommandName) <= 256 , defined WMTextView.MaxCommandLength
LEN(Parameters) <= 1024*1024 , defined by WMTextView.MaxCallParameterBuf
```

The parameters are often input or output file names or represent options conditioning the command execution. Parameters are parsed by the command interpreter and options can be processed by Options.Mod designed for the purpose. The command parser, by convention and by construction, expects to find the options at the beginning of the parameter list. Finally, “^” on its own is also a parameter. Each occurrence of “^” will be replaced by the most recent selected text, if any.

Note that the parameters are not the parameters of the procedure represented by CommandName.

It is recommended to terminate a command with a “~” because when omitted, all the characters of *Parameters* up to the end of text (limited by MaxCallParameterBuf) will be copied to the parameter buffer, thereby consuming a lot of memory and execution time.

Example of command:

PET.Open -e PET.Mod Configuration.XML ~

opens two files in a new window external to the window where the command appears. In “-e” the minus signals an option and “e” stands for “external”.

A *command execution* is triggered:

a) either directly by typing a command in an open text editor, such as PET, the Notepad or the Kernel log, then positioning the cursor on the command name and finally clicking MM (or Ctrl+ML for a 2-button mouse). Instead of using the mouse, one can use the keyboard, holding down Ctrl and pressing Enter. Mouse and keyboard *events* are detected and interpreted by a listening program called *command interpreter* in charge of executing the command procedure,

b) or indirectly by a program designed to parse some text, to retrieve a precise command and finally to execute the selected command. This technique is used by the customization described next, in menus and by HotKeys,
c) or using the context sensitive PieMenu opened by pressing MR in text editor.

More details on this subject in: www.ocp.inf.ethz.ch/wiki/Documentation/Miscellaneous and a programming explanation of command in: A2 Programming Quickstart Guide
www.ocp.inf.ethz.ch/wiki/Documentation/Language?action=download&upname=A2QuickStartGuide.pdf

A command designed to open a window on the desktop will usually have “Open” as procedure name, but that is only a good practice convention and is consistent with the intention of opening a GUI application.

When a command fails to execute, an error message is sent to the Kernel log or a suitable context *(to be defined better)*. This is the case when the *CommandName* is not valid . When only *ModuleName* is valid, the module is loaded anyway. Other errors may be detected during the parsing of parameters. It is thus recommended to have the Log open on the desktop and to keep an eye on it to watch how work is progressing.

When a command execution fails completely, without even the possibility to issue an error message, a TRAP window with a red background informs about the location (pc = program counter) of the failing program statement.

The command interface

In A2, commands can be placed within *any text* and command execution is invoked as described above. This paradigm, called TUI (Textual User Interface) has several advantages over Command Language Interfaces (CLIs):

Visibility Commands and their parameters can be placed into *tool texts* prepared for a specific task or set of tasks. The commands are visible in the text and ready to be invoked by the user.

Readability There is no need for commands to be short and cryptic since commands are normally not typed very often. Normally, commands are typed only once for a specific task. If there is a chance to need the command again for the same or a similar task , the respective text can simply be stored as a tool text for reuse.

A good example is given by the scratch text area of the PET.

In CLIs in contrast, commands and parameters must be remembered or looked up before use. Shells are well-known representatives of CLIs, which offer a line editor where the user can type and modify the command line that is then interpreted according to the rules of the command line interpreter when the user presses the Enter key. Three shell applications are available in A2.

Customization of A2

Customization is the task of placing commands at strategic points on the system's evolution path through a cascade of state changes, beginning at the system start, and to

finally establish the best possible initial state from the user's point of view. In contrast, a downloadable A2 release is customized to have the best chance to be successfully installed on a machine of unknown hardware composition. Experience helping, one may assert that, by and large, a freshly installed system is already well customized and needs little adjustments and if so for two major reasons:

1. in the case that an *A2 build* fails to start from its CD-ROM, a customization is needed to circumvent the difficulties experienced by adjusting the “best guess defaults” defined in the build process. The configuration data is then of concern.
2. in the case that an A2 build effectively runs from the start and can thereafter be installed, the customization serves the purpose of adjusting A2 to personal requirements evaluated in terms of ease of use, applications readiness and presentation.

From the start of a session the system state is conditioned by persistent data, essentially some text with embedded commands, located in:

1. the *configuration data* exploited by and controlling the *boot loader*. Detailed information of the subject is found at:
   www.ocp.inf.ethz.ch/wiki/Documentation/Configuration
   WinAos uses configuration data located in aos.ini which is different because the hardware is controlled by the Windows host. A user's preference may also be stored in myaos.ini which is given the priority over aos.ini (the latter may then be left unaltered for safety).
2. the file Configuration.XML with an important Autostart section
3. commands embedded in data files, mostly with names suffixed .XML
4. commands embedded in the Main menu and its sub-menus

and optionally, whenever deemed useful:

5. commands embedded in HotKeys.XML (see section “HotKeys”)
6. commands embedded in the Auto.dsk file (see section “Desktop save/restore”)
7. commands listed as parameters of the system command SystemTools.DoCommands (see section “SystemTools”).

The duty of the commands is manifold such as: inspect the hardware, install drivers, mount/unmount partitions, start and control applications. Quite a number of them are described in what follows.

Items 1. and 2. are imperatively evaluated in that order on the evolution path, while the data conditioning the remaining items will not necessarily be exploited in its entirety. The persistent data must in most cases be maintained by the user with the help of a text editor. On the contrary, item 6. is constructed by the system when the user requests to save the desktop for a subsequent session. Item 7 uses a mechanism which is in fact much more general than suggested here for customizing the system.

From the previous enumeration one can deduce that the strategy for placing commands is not obvious and is a matter to be dealt with by the user. The following explanations should facilitate deciding on where about to insert the commands mentioned in the context of the many application descriptions.

Configuration.XML

An A2 build is conditioned in many ways by the Configuration.XML file. 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 located in the Autostart section, codecs, etc.

The file is part of the A2 delivery. Whenever a modification is desired or needed:

1. edit the configuration text with, for example, Notepad.OpenAscii Configuration.XML ~ (see Text editor).
   Shortcut: Main menu → System → Configuration
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, shadow copy, correct configuration Save.Configuration.XML is used instead. In this way, the next A2 start can succeed. Starting with corrupted specifications might be fatal, meaning that A2 might fail to start altogether.

Autostart section

Among all the sections, the *Autostart section* is of particular interest to the user as it collects system commands that the user may want to let execute automatically when A2 starts. Here is an example Autostart section including three commands:

```
<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>
```

These commands, together a great many other commands, are described in length in the remaining of part this text. The “StartMenu.Open” command is almost always included to let the Main menu appear at start-up.

Desktop description

Main menu

It offers to select and to start an application with simple ML clicks. Behind the scene, an A2 command is executed that a user would have to enter otherwise via the keyboard. These commands are documented in the application descriptions in the remaining of this text. The menu collects some of the most used or representative GUI applications organized in a two-level hierarchy of buttons.

Note: the look of the menu, and all other windows on the desktop, varies with the skin (more on the subject in the section Skin loader). The menu shown here is programmatically generated and uses no skin.

[WMScreenShot.SnapShotRange test.bmp 500 59 0 915 ~]

Each button in the top row represents a group of somewhat related applications.

A ML click on one of the buttons selects the group. On the figure “System” was selected, with a yellow background, and the applications in that group are listed in the two bottom rows. A black and white print shows it in light gray.

A ML click on an application button starts the application and a window is opened on the desktop, with few exceptions. The figure shows “Reboot””and “Shutdown” with a red background stressing their special function and meaning. A black and white print shows them in lighter gray.

Usage	A2 is conditioned by default to show a main menu. Since the main menu is evolving in time, the function associated with each button may vary and a user has the liberty to customize the main menu. If the menu is not visible because some windows are placed on top of it, or after some navigation, press Meta+Esc to let it appear on top. In WinAos, use Meta+Esc or Menu+Esc.
Start	StartMenu.Open ~ Insert the main menu at the bottom left of the desktop. This command is inserted *by default* in the Autostart section of the Configuration.XML file of a build and is thus automatically executed when A2 is started. It is not recommended to remove it, except may be when A2 runs without a display unit, but then A2 must be controlled differently (A2 used as server for example).
Stop	Never (Components are: StartMenu.Mod, MainMenu.Mod)
Data files	MenuPagexy.XML (xy takes the values 00,10, 20, 30, 40, 50, 60, 70, 80,90, all of which are reserved for the release). The delivered XML menu files are described below. The Menu file generator allows generating custom menus. The StartMenu.Open command automatically detects the presence of such menus provided it is correctly named.

Menu page structure

It is useful to know the menu structure for making small changes to existing menu pages. Complete menu pages are best created with the “Menu page generator” described further on.

An XML menu file defines a menu as a hierarchy of Panels with 2 vertically aligned Buttons, each associated with a command. The text in italic is the essential part that is customized.

```
<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

Usage	The desktop with *most* of the task windows and backdrops (a variant of task window) can be saved at any time during a session and restored later on at will. Data on the state of the GUI applications is then (normally) recorded in Auto.dsk. When A2 is started anew, the desktop is restored to the state it had during the previous session with most but not all of the windows at the same location. The user can proceed with work as it stood earlier. Saving the desktop differs from hibernation as known from other systems in that application programs are explicitly asked to store their relevant persistent data. This is much more flexible and robust than loading back an entire system memory image. It is for example possible to update the entire system or even change the hardware setup of the computer and still continue working with a desktop that was saved before the changes. 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. The table documenting an application then includes a **Restorable** entry. An A2 release always includes a model file called Release.Auto.dsk storing the same data as Auto.dsk as delivered with a build.
Start	WMRestorable.Store fileName ~ Store the desktop in the named file. Auto.dsk is the normal repository which is included in a build. WMRestorable.Load fileName ~ Restore the desktop from the named file. **Shortcut**: Main menu → System → SaveDesktop The shortcut saves slightly more data than the Store command does: the current skin is also recorded by executing a command SkinEngine.SetCurrentAsDefault ~ behind the scene. Assuming that the user had loaded a skin during the session, the same look will reappear at the next start.
Stop	Never
Data files	Release.Auto.dsk, Auto.dsk

Editors and font services

The *system-wide* 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 build 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 loader
Oberon	WMOberonFonts.Mod
Bitmap (Chinese, Japanese, Korean)	WMBitmapFont.Mod
CCG	WNCCGFonts.Mod

Open type Unicode TTF (void of copyright)	WMOTFonts.Mod

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
Using embedded font
```

The embedded font is an Oberon font completely defined in a module WMDefaultFont and not loaded from a font file as is otherwise normal.

Usage	Font loader
Start	
Stop	
Data files	

Oberon and Syntax fonts

The Syntax typeface - http://de.wikipedia.org/wiki/Syntax_(Schriftart) - was developed by Hans Eduard Meier - http://de.wikipedia.org/wiki/Hans_Eduard_Meier - who later on, in collaboration with the ETH, developed the Oberon typeface specifically for the ETH Oberon system.

The Oberon typeface combines in a unique manner typical elements of antiqua and modern typefaces. Because of the large number of documents using these fonts, an Oberon font implementation of the abstract A2 font interface has been realized. The font support can load existing Oberon font files and use them in A2 outside of the Oberon environment. For compatibility reasons, the Oberon font file format was not changed but A2 handles the font files differently in two ways:

1. To find an Oberon font, the font manager converts the given font name, size and style into the canonical font file name according to the Oberon font naming convention: the file names are made of the names listed in the tables suffixed with

".Scn.Fnt".Some fonts are available in the styles normal, bold, italic and medium bold.

2. While loading an Oberon font file, the contained metric data is converted to the Bluebottle glyph metric.

Greek	Math	Oberon			Syntax			
		Oberon8	Oberon8b	Oberon8i	Syntax8	Syntax8b	Syntax8i	Syntax8m
Greek10	Math10	Oberon10	Oberon10b	Oberon10i	Syntax10	Syntax10b	Syntax10i	Syntax10m
Greek12	Math12	Oberon12	Oberon12b	Oberon12i	Syntax12	Syntax12b	Syntax12i	Syntax12m
Greek14	Math14	Oberon14	Oberon14b	Oberon14i	Syntax14	Syntax14b	Syntax14i	Syntax14m
Greek16	Math16	Oberon16	Oberon16b	Oberon16i	Syntax16	Syntax16b	Syntax16i	Syntax16m
Greek20	Math20	Oberon20	Oberon20b	Oberon20i	Syntax20	Syntax20b	Syntax20i	Syntax20m
Greek24	Math24	Oberon24	Oberon24b	Oberon24i	Syntax24	Syntax24b	Syntax24i	Syntax24m

Courier	Philus		Shanghai
Courier8			Shanghai
courier10	Philus10	Philus10b	
Courier12	Philus12	Philus12b	

Bibliography: Max Caflisch. Die Entstehung der Syntax-Antiqua
OFFICINA, Mitteilungen des Hauses Schwabe & Co., Basel, 1996.

Note: International Typeface Corp. markets another OpenType typeface called "Oberon" but quite different.

OpenType fonts

OpenType is a font format defined by Microsoft and Adobe that can contain glyph outlines in the TrueType or Type I format. OpenType fonts support Unicode and can contain information for more than 65'000 characters. The names of the installed OpenType fonts can be obtained by executing: WMOTFonts.MultiTest ~ described elsewhere.

Many thousand OpenType fonts are by now available, which made the integration of the format in A2 as a font plug-in attractive. The implementation is based on the off-line TTF to Oberon font converter built by Erich Oswald at the ETH Zürich under the name OType "a package for loading and rendering TrueType fonts within the ETH Oberon system.

Bitmap fonts

Their purpose is not understood. This holds also true for the exported procedure WMBitmapFont.Import which uses cjkfont.xml found nowhere, also not on the web.

***CCG fonts** - Chinese Composite Glyphs*

A software for drawing and viewing Chinese glyphs. Citing Thomas Frey's thesis:

Traditional Open Type fonts are not very well suited to store glyphs of the group of Chinese, Japanese and Korean (CJK) languages in a space efficient way. The 60'000 most commonly used CJK glyphs require about 40MB of storage. While this size is

acceptable for current desktop computers, it is by far too large for smaller devices such as PDAs or wearable computers. Making use of the highly structured composition of Chinese glyphs, it is possible to store the same number of glyphs in a file as small as about 1MB. Each Chinese glyph is either a radical or contains one or more radical-like elements. There are 214 unique radicals in traditional writing, 189 in the simplified form. Most parts of a glyph can be drawn re-using radicals in different sizes and positions. Complex glyphs can be constructed by repeated re-use of radical elements or by the recursive use of other complex glyphs. To produce aesthetic complex glyphs, several different variations of the radical are needed.
The Taiwan-based company eForth developed a font format and database, called CCG font, based on recursive composition of radicals and radical-like elements. The font format can store the glyphs of more than 82'000 CGK characters in a file of about 2MB. This is a sufficiently small size for PDA-like computer systems. eForth donated a single line stroke font with about 82'000 glyphs (Single.ccg file) and an outline font with about 27'000 glyphs (Song.ccg file) to the A2 project based on a free license.

Example of character composition:

The radical 日 rî which means “sun” composed with
the radical 月 yuè which means ”moon” produces
the composite glyph 明 ming which means “bright”, a radical-like glyph which itself
composed with the radical 皿 min which means “basin” produces
the composite glyph 盟 méng which means “ally”.

A2 includes a Chinese tutorial text cn.PrgInOberon.txt translation of:
“Programming in Oberon” by H. Mössenböck and N. Wirth, ETH Zürich.
Shortcut: Main menu → Docu → Chinese Tutorial

Also the A2 compiler errors are available in their Chinese translation in the file cnErrors.XML. These translations were kindly provided by Qingyong Chen
Chongqing University
Chongqing 400044
P. R. China

The following short collection of commented references will help to clarify what the aim is:

Justification for implementing Dynamic Glyph Generation for Chinese (Hanzi or Kanji or Cang Jie, as you like) glyphs in A2:
“Dynamic Glyph Generation Based on variable length encoding schema”
by Yap Cheah Shen – Kyoto University 21st Century COE Program 2003
coe21.zinbun.kyoto-u.ac.jp/papers/ws-type-2003/093-yap.pdf
coe21.zinbun.kyoto-u.ac.jp/papers/ws-type-2003/yap.ppt
Yap is working at eForth Technology, Inc. and kindly helped to implement CCG in A2.
The expression CCG = Chinese Composite Glyph was casted there.

Based on the work of:
Prof. Hsieh Ching-Chun - www.iis.sinica.edu.tw/~hsieh/
Adjunct Faculty Research Fellow at the Academia Sinica in Taipei

www.sinica.edu.tw/main_e.shtml

Used by eForth Technology, Inc. in Taiwan for their OS on embedded CPU
www.eforth.com.tw/
www.eforth.com.tw/academy.htm
Some details in: www.eforth.com.tw/CT/Efeditor/index1.html

Papers about Chinese glyphs:

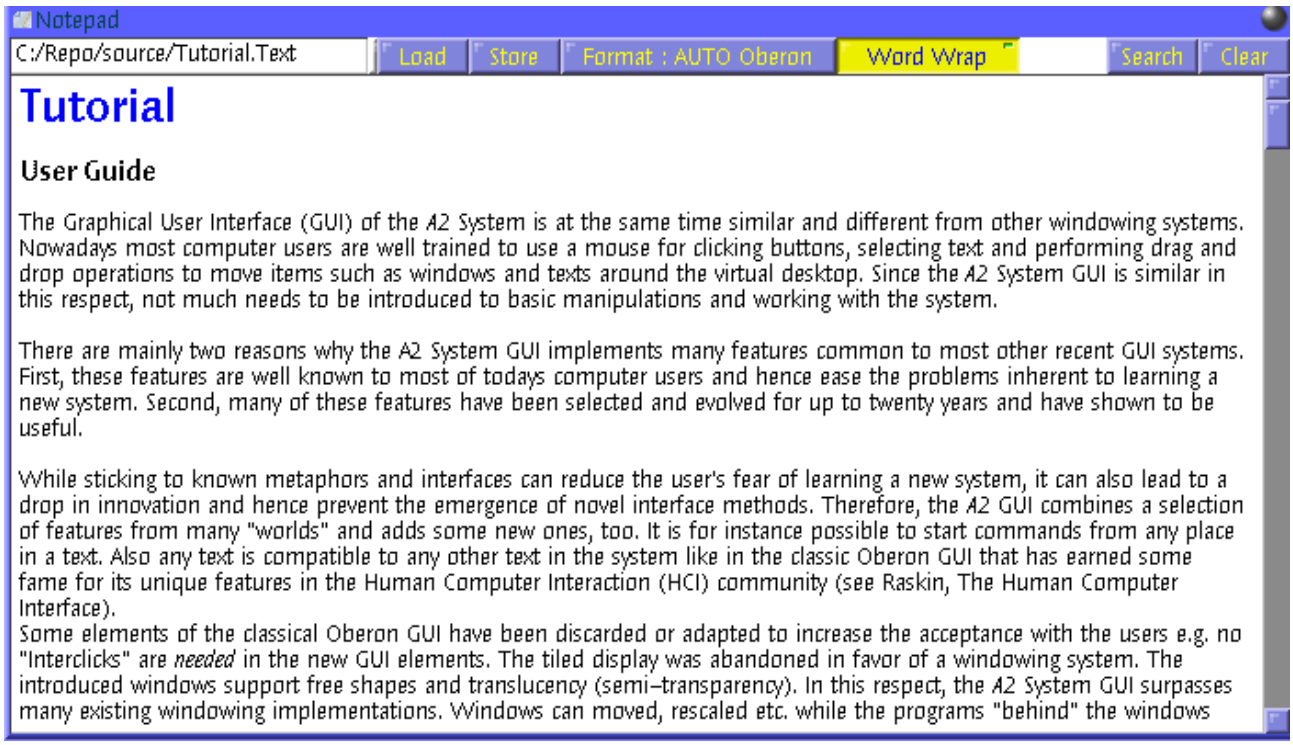
"A Heuristic Search Approach to Chinese Glyph Generation Using Hierarchical Character Composition"
by P.K. Lai, D.Y. Yeung, M.C. Pong, Computer Processing of Oriental Languages, Vol. 10, No. 3, 1996

"New Ideographs in Unicode 3.0 and Beyond"
by J.H. Jenkins, 15th International Unicode Conference, San Jose, CA, 1999.

For those interested in Chinese literature, culture and in particular CJK:
Robert Oestling's web site: www.robos.org/chinese/
Dylan's Place! : www.sungwh.freeserve.co.uk/index.html

Chinese glyphs are inputted using the Pinyin IME described elsewhere.

Text editor *– GUI*



[WMScreenShot.SnapShotRange text.bmp 700 400 0 0 ~]

<table>
<tr><td>Usage</td><td>Edit a new text or a text 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><th>Format : pop-up</th><th>Codec</th></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>Pure ASCII</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></td></tr>
<tr><td>Start</td><td>Notepad.Open [-f] [fileName] ~

When “fileName” is omitted, an empty text appears.
Several editors may be running in parallel.

Shortcut: Main menu → Edit → Text (with empty text)</td></tr>
<tr><td>Stop</td><td>Close the window(s).
To clean up: SystemTools.Free Notepad ~</td></tr>
<tr><td>Restorable</td><td>Yes</td></tr>
<tr><td>Data files</td><td>DefaultTextStyles.XML</td></tr>
</table>

Keyboard layout for A2 (on bare hardware)

A2 is set up to operate with a standard US keyboard. To customize A2 for a specific language the value assigned to “Keyboard=” in the configuration data must be adjusted to use the corresponding layout file. Details are found in:
http://www.ocp.inf.ethz.ch/wiki/Documentation/Configuration#toc11

Mouse layout for A2 (on bare hardware)

A freshly installed A2 system, using the A2 deliverable, assumes that the machine is equipped with a 3-button mouse, either with a MM button or a wheel as equivalent, for ease of use. 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 the Ctrl key, either left or right, after having set the configuration string “MB=2”,
See: http://www.ocp.inf.ethz.ch/wiki/Documentation/Configuration#toc12

Text markers and text navigation

The A2 text editor supports several kinds of text *markers*. The following are used everywhere:

- the *point position marker*
  - a small vertical red bar of the height of a line, called “caret”
- the *text range marker*
  - a translucent blue text overlay to highlight a text selection

- a straight underline to highlight a command just about to be executed

PET uses in addition:

- a special *point position marker*, a translucent image positioned with an offset relative to a text position at the base line. It is used to mark compiler detected errors in a program text
- a special *text range marker*, an undulated underline to mark errors or possible problem places

For the navigation in a text, the caret, the mouse pointer, the mouse scroll wheel and other special keys are used, most often in combination. Selection of text is performed either with the mouse or with the keyboard. Supporting both is dictated by the time overhead of moving the hands from the keyboard to the mouse and back.

Editing with the mouse

Point position in text	
MLclick	**Set the caret**: Move the mouse pointer to the intended point in the text then click ML.

Select a text stretch and operate on this text range	
ML press & drag	**Select stretch**: Move the mouse pointer to the beginning of the intended selection, press ML to set the caret and drag the mouse in any direction, progressively highlighting the text stretch with a blue text range marker. Release ML. The scroll wheel can be used during the operation to navigate in the text.
ML click double	**Select word**: double click ML selects the entire surrounding word.
ML press & drag + MR	**Cut**: Select stretch and instead of releasing ML, press MR. This cuts the text stretch.
ML press & drag, ML press & drag	**Cut and drop**: Select stretch, then place the mouse cursor amidst the selection, press ML and drag the selection to the insertion point with the help of the moving caret.
ML press & drag, Ctrl press + ML press & drag	**Copy and drop**: Select stretch, hold Ctrl down, then set the caret amidst the selection and drag the selection to the insertion point with the help of the moving caret.

Operate on a command name or a document name appearing in a text. When the mouse pointer is positioned on a word and MM is pressed, the word is underlined in red.	
MM	**Execute** a command when the mouse pointer is positioned on a valid command name.
MM + MR	**Open a document** when the mouse pointer is positioned on a document name.

Editing with the keyboard

	Move the caret
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 caret**	**and**
Arrow left	one character left	select a character, a line, a page OR de-select depending in which direction the caret 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 name or a document name appearing in a text. The same control as can be exercised with MM on a 3-button mouse. With a 2-button mouse, Ctrl is the substitute for MM
Ctrl+Enter	**Execute** a command when the caret is positioned on a valid command name.
Ctrl+Shift+Enter	**Open the document** when the caret is positioned on a document name.

	Check text healthiness; control IME
Ctrl+T	Check the healthiness of the text piece list. Errors are listed in the Kernel Log.

Ctrl+Space	Enable/disable an IME.

Editing with a combination of mouse and keyboard

	Operate on a command name appearing in a text in PET
Shift+MM	Open the document in a new window and not in the existing PET window.

	Operate on a document name appearing in a text in PET
Shift+MM+MR	Open the document in a new window and not in the existing PET window.

Editing with the PieMenu

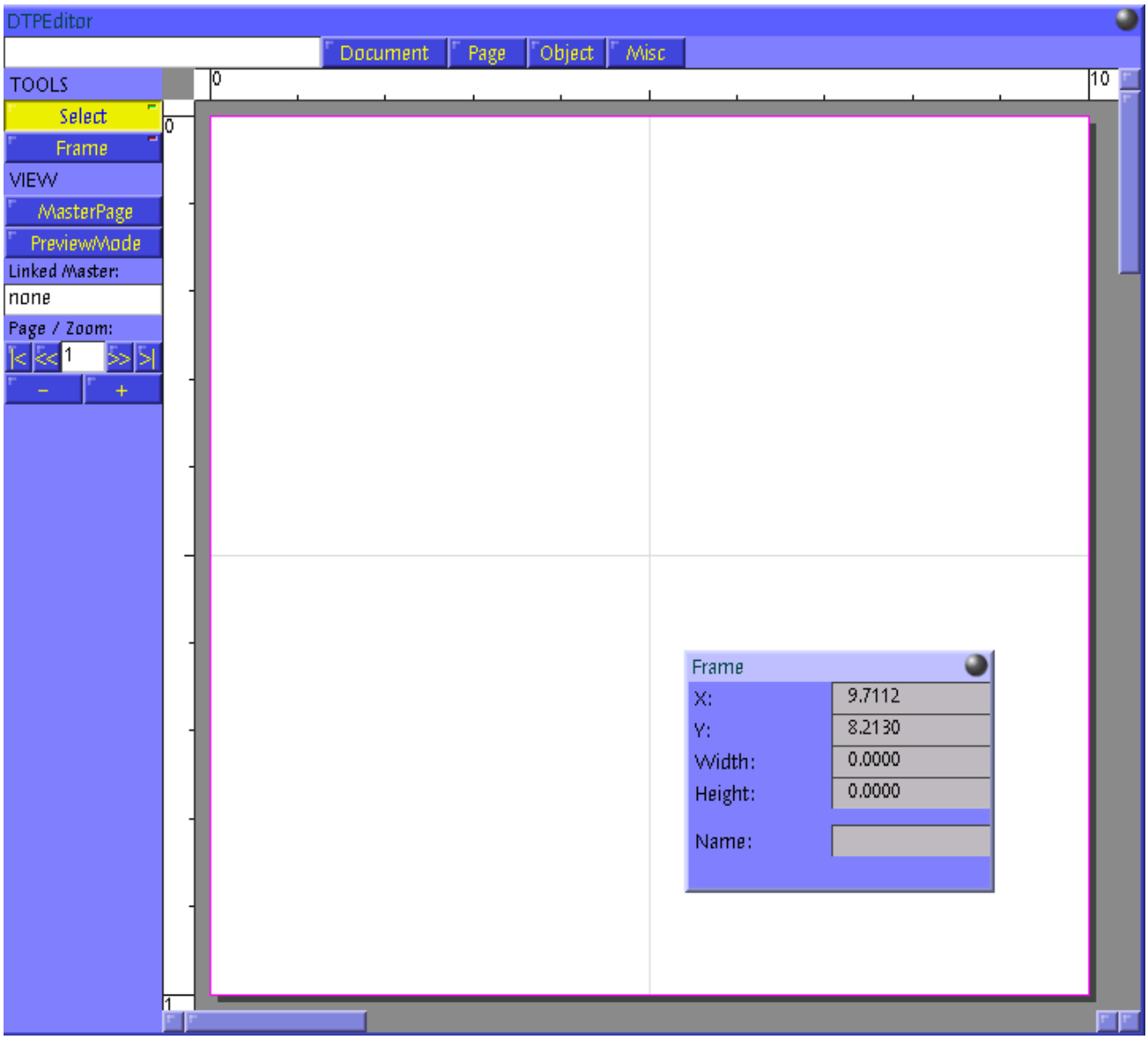
An alternative method to using mouse and keyboard for a few often used operations.

PieMenu *- GUI*

Usage	A context sensitive menu for editing text, opening a text or executing a command in a text editor. Use it as a substitute for mouse and keyboard actions described above in four frequently used situations.
Start	In any text editor, move the mouse pointer to an intended point in text, then press MR and hold it down to open a pie with four labeled sectors. Hovering the mouse pointer over the sectors causes the color of the pointed sector to darken. Releasing MR launches one of the following action, depending on the sector:
Stop	The pie disappears automatically when MR is released. (Component: WMPieMenu.Mod)

Sector	**Action on releasing MR**
Copy	Copy a selected text to the clipboard.
Paste	Paste the clipboard text to the mouse pointer's position.
Open	Open the text corresponding to the name pointed to.
Start	Start the command pointed.

Desktop publishing editor *– GUI*



[WMScreenShot.SnapShotRange test.bmp 720 650 0 0 ~]

Usage	A desktop publishing editor.
Start	DTPEditor.Open [fileName] ~ Several editors may be running in parallel. **Shortcut**: Main menu → Edit → DTP
Stop	Close the window(s). To clean up: SystemTools.Free DTPEditor DTPView DTPUtilities DTPData ~
Restorable	Yes
Data files	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 editor - *GUI*

Usage	A tool for changing the look of a text stretch (font, font size, text color) and for marking text stretches according to predefined conventions used by programmers.
Start	WMTextTool.Open ~ Open a text style panel  [WMScreenShot.SnapShotRange test.bmp 125 240 0 0 125 240 ~] An MM-click on one of the several buttons allows changing the look of a text stretch selected in a document opened in an editor (Notepad or PET). The top 8 buttons are for use by *Active Oberon* programmers and enforce conventions which have progressively adopted as a means to make programs more readable. These are not compulsory but simply common sense. The Get button allows gathering font, font color and size, background color information for the selected text stretch. The Apply button sets these values. **Shortcut**: Main menu → Edit → Styles The following commands are at a user's disposal, Some of them are implicitly and more conveniently executed using the text style panel. These could also be appropriately used in HotKeys.XML. WMTextTool.SetFontSize ["Absolute" | "IncrementBy" |"DecrementBy"] [value] ~ When no value is given for increment or decrement the default value 1 is used. WMTextTool.SetFontStyle [“normal” | “bold” | “italic”] ~ normal is the default value. WMTextTool.SetFontName [fontName] ~ Oberon the the default fontName. WMTextTool.SetFontColor [foreroundColor [backgroundColor]] ~ These commands operate on a selected text stretch without altering it and the result appears in the Kernel Log: WMTextTool.CountLines ^ ~ WMTextTool.CountWords ^ ~ WMTextTool.CountCharacters ^ ~ WMTextTool.CountAll ^ ~ count characters, words and lines Several tools may be running in parallel.
Stop	Close the window(s).

	To clean up: SystemTools.Free WMTextTool ~
Restorable	Yes

OpenType True Type Fonts (TTF) detector

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 ~
Data files	OpenTypeInt.Mod, OpenTypeScan.mod, OpenType.Mod, OpenTypeFonts.Mod, WMOTFonts.Mod

Bitstream Vera

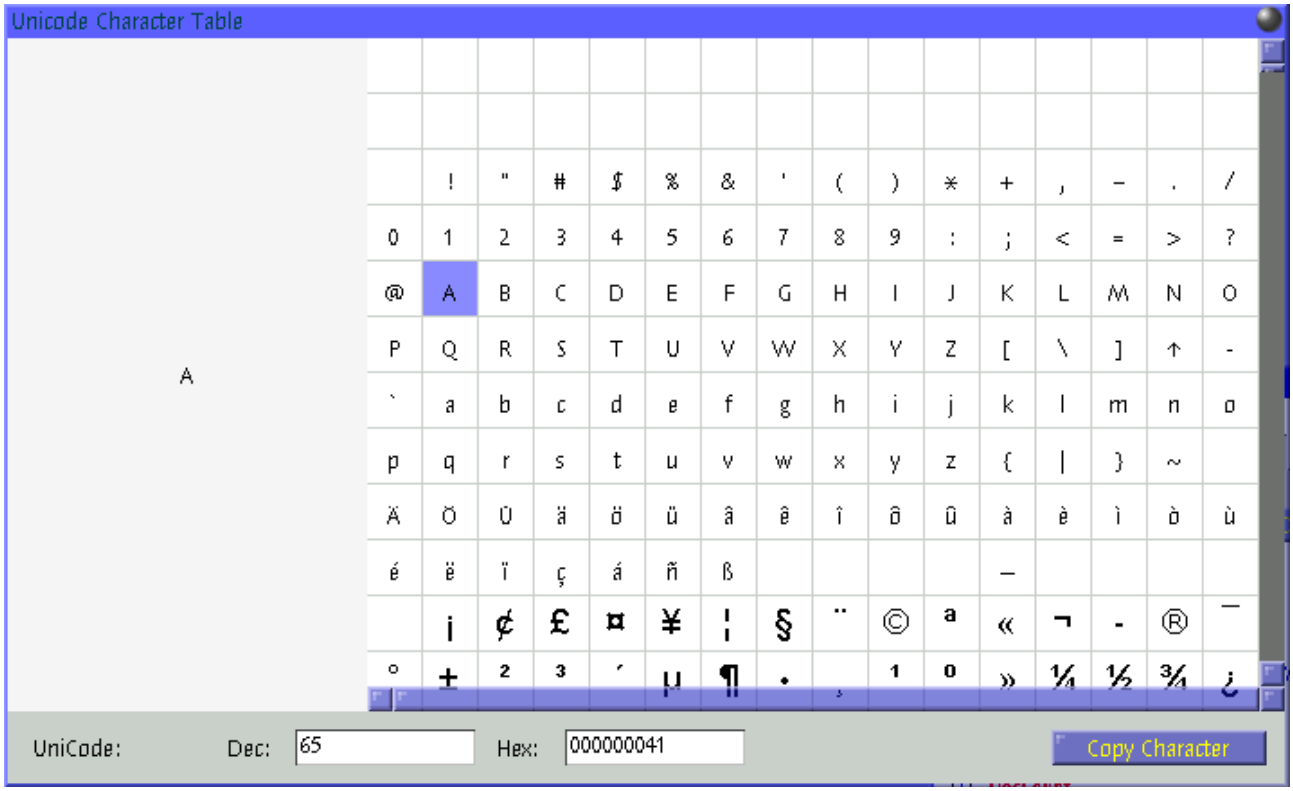
Bitstream Vera from Bitstream – http://new.myfonts.com/foundry/bitstream/ is a True Type font with full hinting instructions, which improve its rendering quality on low-resolution devices such as computer monitors. It consists of serif, sans-serif and monospace fonts and covers only common punctuation 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.



[WMScreenShot.SnapShotRange test.bmp 715 435 0 0 ~]

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, with a MM click, to let the decimal and hexadecimal character value appear at the bottom. Conversely, enter a decimal or hexadecimal value in an input field to see the corresponding glyph. To insert a character in an editor, select the character, click „CopyCharacter“, set the cursor at the desired location in a text 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 a preview of a selected glyph appears. Several maps may be running in parallel. **Shortcut**: Main menu → Edit → UnicodeMap
Stop	Close the window(s). To clean up: SystemTools.Free WMCharMap ~

Unicode marker tool

Usage	??? Purpose ?
Start	WMUnicodeMarkerTool.Open ~ Several marker tools may be running in parallel. **Shortcut**: Main menu → Edit → Unicode Markers
Stop	Close the window(s). To clean up: SystemTools.Free WMUnicodeMarkerTool ~

Restorable	Yes

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 Cyberbit is available free to the Unicode consortium members for non-commercial use.
Start	CyberbitNetInstall.Start ~
Stop	To clean up: SystemTools.Free CyberbitNetInstall FTPClient ~

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/

CJK refers to the languages of Chinese, Japanese, and Korean, though occasionally they will mean the countries China, Japan and Korea respectively, and therefore refer to the 'locale' of the script in question.
These countries use non-alphabetic characters based on the traditional Chinese script "hanzi", otherwise known in Japan as "kanji", and by Koreans as "hanja". For simplicity, we shall call them "characters". We can further differentiate the forms of the same characters as 'glyphs'. A 'glyph', therefore, is a variant of a character and varies depending on locale.

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://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 (often with multiple words or meanings) 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 ~
Data files	Unihan.txt

Input Method Editors (IME)

WMInputMethods defines an abstract IME and a plug-in mechanism for specific implementations. A text editor can call an instance of an IME as a layer between its keyboard message handler and its character insertion routine. The editor discerns between “navigation key” events and “modification key” events. In case of a “modification key” event, it checks if an IME is active on the editor. If so, it forwards that event to the active IME and the character is processed:

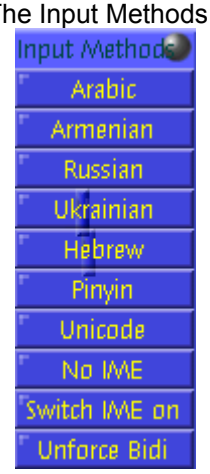
either with simple mapping to another Unicode character (thus acting like a keyboard driver), as is for example the case for Russian and Ukrainian.

or with complex processing, as is the case for Pinyin.

When no IME is active, the character represented by the key is directly inserted into the text model.

After having installed an IME, enabling and disabling the IME makes it possible to quickly change between input methods for different character sets. For example, disabling the IME is used to type email addresses, URLs or commands. Experienced writers often enable, disable and change IME in rapid successions to write words in different languages or scripts. Enabling and disabling an IME is done using the IME tool described next or with the key shortcut Ctrl-Space (see Text Editor section). It is not required to unload the IME.

IME tool *- GUI*

Usage	A graphical tool for activating one of the IMEs.
Start	WMInputMethodTool.Open ~ The Input Methods panel opens:  [WMScreenShot.SnapShotRange test.bmp 90 220 0 0 ~] A mouse click on one of the 7 supported IMEs installs it, by executing, in the background, one of the Install commands described in the seven following sections. For Pinyin and Unicode, the way to proceed with inputting data to the editor is described. It is necessary to activate the IME and that is done: 1. with a mouse click on the “Switch IME on” button 2. by pressing Ctrl-Space on the keyboard (toggles IME on/off) 3. automatically when switching to another IME and one is already active The button of an active IME turns green. Inactivating the current IME is done: 1. with a mouse click on the “Switch IME off” button 2. by pressing Ctrl-Space on the keyboard (toggles IME on/off) Returning to operating without IME is done with a mouse click on “No IME”. Arabic and Hebrew, two languages written from right to left, require an editor with that capability. Though code was specially developed for that purpose, it is not operational. Arabic also wants an editor capable of changing the letter shape depending on whether they appear at the beginning, middle or end of a word, or on their own. That was not taken into account. Note: Bi-directional input is something different and is also called “boustrophedon”.

	Shortcut: Main menu → Edit → Input Methods WMInputMethodTool ChangeAppearance ~ ??? Several tools may be running in parallel. (What does it mean and imply?)
Stop	Close the window(s). To clean up: SystemTools.Free WMInputMethodTool WMUnicodeIME WMPinyinIME WMHebrewIME WMArmenianIME WMUkrainianIME WMRussianIME WMArabicIME ~
Restorable	Yes

Arabic

Usage	Not usable as is. Must be improved. The editor must handle right to left input and this efficiently. It does not.
Start	WMArabicME.Install ~

Armenian

Usage	Simple one-to-one mapping to Armenian.
Start	WMArmenianIME.Install ~ After having installed the IME, activate/deactivate it with the hot key Ctrl-Space.

Russian

Usage	Simple one-to-one mapping to Russian Cyrillic.
Start	WMRussianIME.Install ~ After having installed the IME, activate/deactivate it with the hot key Ctrl-Space.

Ukrainian

Usage	Simple one-to-one mapping to Ukrainian Cyrillic.
Start	WMUkrainianIME.Install ~ After having installed the IME, activate/deactivate it with the hot key Ctrl-Space.

Hebrew

Usage	Not usable as is. Must be improved. The editor must handle right to left input and this efficiently. It does not.
Start	WMHebrewIME.Install ~

Chinese - Pinyin for Hanzi glyphs

Pinyin is the most commonly used romanization system for standard Mandarin and *pinyin* means "phonetics", or more literally, "spelling sound" or "spelled sound" in the Latin alphabet of a Hanzi glyph.

Usage	WMPinyinIME implements a complex IME.
Start	WMPinyinIME.Install ~ When the IME gets the first key event from the editor, it opens a small window next to the cursor position in the main editor and tells the display space manager to forward keyboard events to the new window. Inside that IME window, key strokes are sent to an embedded editor. While typing, all possible glyphs or compounds that match the pronunciations typed so far are displayed. The list is sorted by usage frequency as taken from the Unihan database, and each line shows at the left the Hanzi glyph and at the right the Pinyin value with a tone number appended. Chinese uses 4 tones given here in the example “ma” with an assumed loudness scale from 0 to 5 (see the “Tone symbol” in the next figure).

STANDARD CHINESE **ma**

Chinese Character	Tone symbol	Tone description	English gloss
媽	55	high level	‘mother’
麻	35	high rising	‘hemp’
馬	214	low falling	‘horse’
罵	51	high falling	‘scold’

The appended tone number can be mapped as follows:

Tone number	Tone description
1	High level
2	High rising
3	Low falling
4	High falling
5	neutral

Use the cursor up or down to select the desired glyph and pres Enter to finish.

Data files	PinyinIMEPhrases.txt, PinyinIMETable.txt Single.ccg: Single line stroke font with about 82'000 glyphs Song.ccg: Outline font with about 27'000 glyphs

The simple selection strategies of WMPinyinIME and its limited dictionary of characters and phrases leave room for improvement. To match the usability level of the best commercial Pinyin IMEs, support for automatic learning of new character usages, a better prediction method as well as support for certain pronunciations tolerance should be added to it.

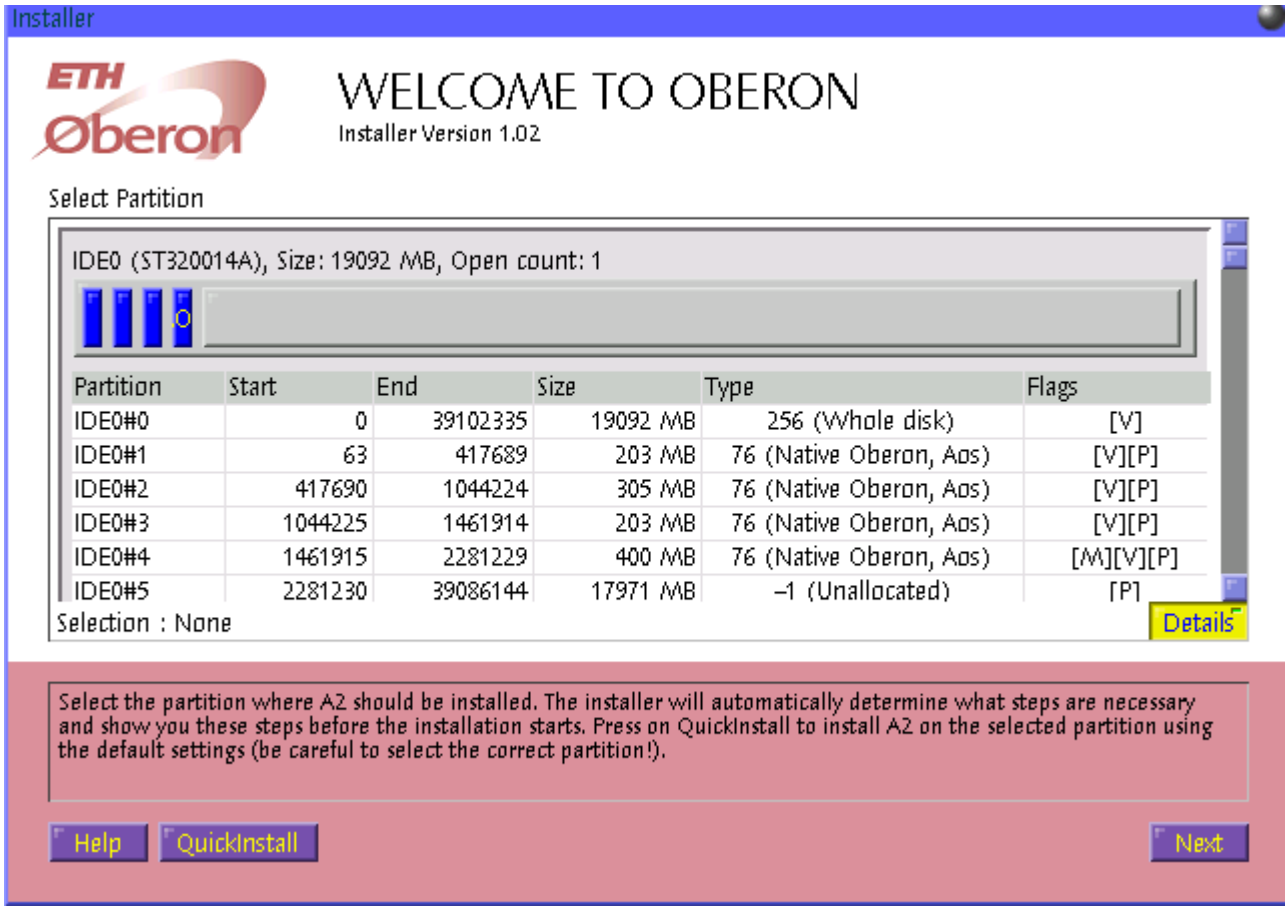
Unicode

Usage	A very primitive way to insert a Unicode character in an editor.

Start	WMUnicodeIME.Install ~ After having installed the IME, activate/deactivate it with the hot key Ctrl-Space. Position the cursor at the desired position in an editor. Then, knowing the decimal position in the Unicode table of the character to insert, type its value and finish with a carriage return. As soon as the first digit is pressed, a small window opens next to the cursor position. The top line displays the digits as input progresses, the lower part displays the corresponding Unicode, which may exist or not. Use Backspace to correct the input.

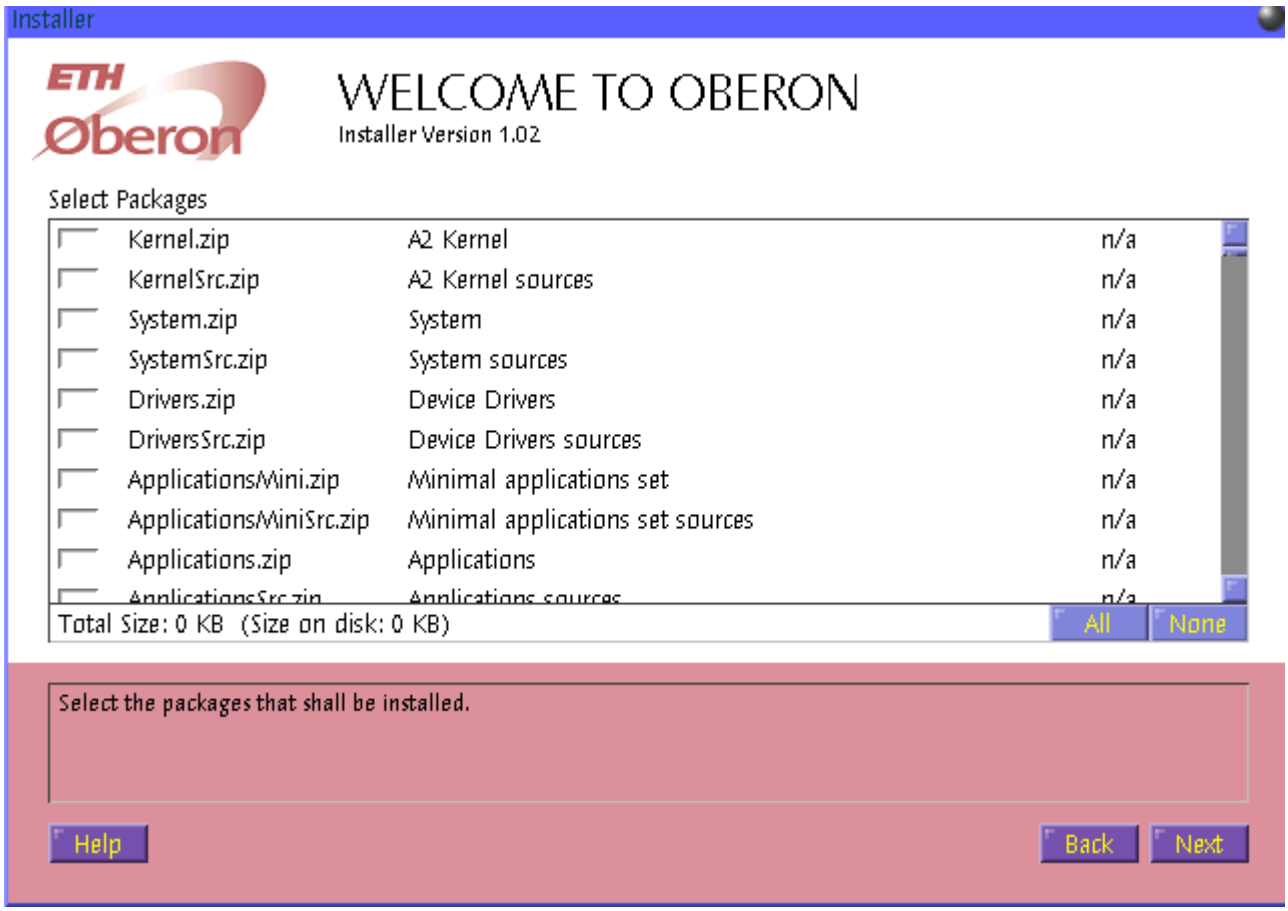
Installation including partitioning and formatting

Installer – *GUI*



Usage	Install A2 on a partition and optionally, according to the situation, create a partition, format it and perform various advanced operations, facilitating the installation of A2. This is the first application that is tackled when undertaking the installation of A2 from an A2 live CD.
Start	WMInstaller.Open ~ Opens a “Welcome to Oberon” window, as shown, with a list of all the **detected mass storage devices for which a driver is installed.** For each device, a text line indicates the device name assigned by A2, the volume size and the number of mounted partitions. Below, a horizontal bar summarizes the position and the size of partitions and free space. Clicking on [Details] exposes the details needed to proceed with the installation. A volume is logically divided in 512-byte blocks

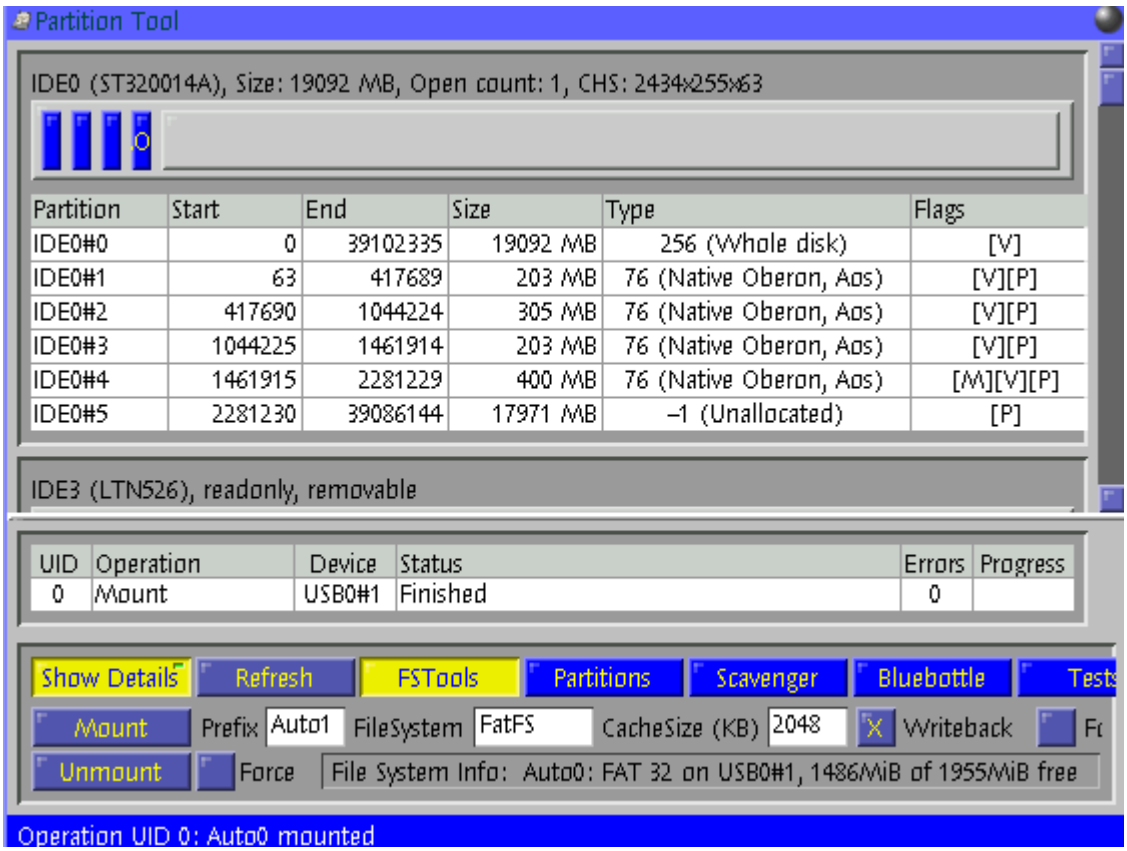
	addressable with their LBA (Logical Block Address) starting from 0 up to the volume capacity. The columns “Start” and “End” contain LBA values. The first detail line exposes the (Whole disk) named device#0. Partitions and unallocated space are listed and numbered sequentially in the order in which they are recognized. A2 partitions have the partition type 72. Partitions can be flagged as: [B] bootable = active (only one may be active) [M] mounted = accessible to A2 [P] primary = defined by a partition entry of the MBR [V] valid = volume contains a valid partition table Read the instructions at the bottom and choose between “QuickInstall” to conduct the installation without further ado or “Next” to start a customized installation through a series of windows. The second screen shot below gives some idea of what packages are (see section “Packages”). While selecting packages, the amount of required space is displayed. Use “Next” to progress though the installation which should not take longer than a few minutes. Note that, en route, a button “Advanced” appears that allows to perform various actions selected with check boxes. A few of then are already selected because they are anyhow required for installing A2, that is, they will be performed even when “Advanced” is not invoked. Hint: a fresh complete A2 installation will not take more than 5 minutes. **Shortcut**: Main menu → System → Installer WinAos: “No Disks” will normally appear in the first window, but virtual disks may be created (refer to that section) and A2 can then be installed in the same manner with the Installer.
Stop	Close the window. To clean up: SystemTools.Free WMWInstaller ~
Data files	WMInstaller.tar, InstallerPackages.XML



Partitions framework *– GUI*

A great many elementary operations that can be commanded in this application can be routinely executed in a simplified manner with well chosen preset parameters and file names by the Installer application described before.

Usage	Volume partitioning and partition management tool with multiple capabilities. This application will play a greater role in the management an the maintenance of an installed A2 system.
Start	WMPartitions.Open ~ The window shows by and large the same information as explained in the section Installer followed by a row of buttons [Show Details] [Refresh] [FSTools] … A mouse click on one of them reveals detail capabilities on a couple lines underneath. The tables below explain how to proceed. WinAos: No partition will normally be shown, but virtual disks may be created and partitioned. **Shortcut**: Main menu → Files → Partitions
Stop	Close the window. To clean up: SystemTools.Free WMPartitions WMPartitionsComponents WMPartitionsPlugins ~
Restorable	Yes



Show Details

Toggle between showing only a list of the accessible volumes or a a list showing all the partitions existing on each volume.

Refresh

Refresh the information after having attached/detached volumes.

FSTools

Pre-condition

Mount requires information on how to mount a specific FS, information which is extracted from the “Files” section of Configuration.XML.

Mount	Mount the selected partition under the condition that it hosts a supported file system (see below). A2 applications have access to the hosted files. The Flags column shows the partition state: [M] for Mounted.
Unmount	Unmount a mounted partition. A2 takes care of updating all open files. Make sure to unmount removable devices, e.g. USB flash drives, before removing them.
Force	Set the check box prior to unmount to force the unmount.

Supported file systems FS	
AosFS	The central FS of A2 accepting 128-byte file names. AosFS limits: Block size: 512 bytes Sector size: 4096 bytes Max. file size: 2.0 GB Max. partition size: 282.48 GB

	These values can be retrieved by executing: PartitionsLib.ShowAosFSLimits ~
OldAosFS	A former A2 FS accepting 32-byte file names. The file name length was increased to 128 bytes for AosFS.
NatFS	FS of the earlier ETH Native Oberon a predecessor of A2. Not likely to be hosted by any contemporary A2 system.
RamFS	A2 internal FS residing in main storage which which must not be formatted. Use as temporary file storage that will be deallocated when A2 is terminated. This technique is used by the Installer.
FatFS	Early Windows and DOS file systems which can be mounted and accessed: FAT12 - partition type 1 - DOS 12-bit FAT16 - partition type 4, 6, 14 – DOS 3.0 16-bit FAT32 - partition type 11, 12 – Win 95 It is thus possible to exchange files between A2 and a mounted Windows partition and even to access and edit windows files in A2. Long names are supported. Enclose long names containing spaces or characters that are invalid in A2 in quotes.
IsoFS	ISO 9660 FS found on CD-ROMs. Always shown as (Whole disk) not partitioned. A FS used on an A2 live CD and mounted by the Installer. Can be neither mounted nor written. The only meaningful operations on such a volume are Eject and ShowBlocks.
ZipFS	Existed in Native Oberon: FS that can mount a zipped file, allowing to boot from a compressed A2 in a file. This FS must be mounted by executing the command FSTools.Mount alias ZipFS fileName.zip ~ Now not operational: zip file appears as empty.
RelativeFileSystem	In WinAos, a subdirectory can be mounted by executing the command FSTools.Mount alias RelativeFileSystem ./ ~ resulting in mounting the A2 subdirectory Work as ”alias” which then appears as FS in the File Manager.
SmbFS	Samba FS.

Partitions

Create	Create a *primary partition* in the unallocated space which must be selected first. A maximum of 4 primary partitions may exist (in the MBR, the only partition table A2 can modify), and only 3 when an extended partition already exists. A2 may be installed in an extended partition but an extended partition cannot not be create. Use a commercial application, Partition Magic for example, for that purpose or use an existing free partition and change its type before formatting it. The Flags column shows the partition state: [P] for Primary. Unallocated space is also flagged [P].
Delete	Delete a selected partition. The space becomes unallocated space.
Format	Format the selected partition, installing an AosFS and an A2 BootLoader making the partition bootable. The kernel and further A2 software may then be installed.
ChangeType	Change the partition type of the selected partition from the current type (e.g. "6" for a DOS partition) to the new type (e.g. "76" for an AosFS) that is entered in a dialog box. After that, a file system must be established with “Format”. Use after creating a partition with another OS, or for reusing an existing free partition. Changing the partition type can make it unaccessible to other OSes! An accidental type change can be reverted as long as the partition is left unchanged.
FromFile	Extract data from a named file and write the data into a number of blocks of the selected partition starting at a a specified offset (dialog box).
ToFile	Write a number of blocks of the selected partition to a named file starting at a a

	specified offset (dialog box).
Activate / Deactivate	Set/reset the activate flag for the selected partition. The Flags column shows the partition state: [B] for Bootable when active.
Eject	Eject a removable device. Select the #0 partition which is marked (Whole disk).
ShowBlocks	Display, in hexadecimal and in characters 16 bytes per line, the contents of a number of 512-byte blocks starting at a given block number (dialog box). Note: An IsoFS inspection reveals that a block is 4 x 512 bytes long.
Check	Verify the readability of the sectors of a selected partition (of any type) and show the progress.
WriteMBR	Write a simple MBR into block with LBA=0 of a volume by selecting the line suffixed #0 and with partition type 256 (Whole disk). This is a special A2 MBR behaving exactly like the simplest Windows MBR used when no boot manager is installed. !!! Be extremely cautious when undertaking to overwrite the MBR of the boot volume especially when a boot manager such as that of Windows, GRUB or else is already installed. A working system may become totally unaccessible. It is recommended to save the existing MBR by first executing ToFile (see above) to save the MBR in a file outside of the boot volume, for example on a removable USB flash drive.
Benchmark	

Scavenger

Scavenger and format a FatFS. Similar to Windows' Scandisk.

Bluebottle

Config	Display the configuration data of the selected A2 partition in a special editor where the configuration strings can be edited and written back to disk, ready for the next A2 start. See http://www.ocp.inf.ethz.ch/wiki/Documentation/Configuration
Update BootLoader	Install the A2 BootLoader at the beginning of the selected A2 partition. Useful after upgrading an existing A2 system for which one does not want to format the partition. Implicitly performed in a new install.
Update BootFile	Install the A2 BootFile in the selected A2 partition, that is the driver corresponding to the disk controller to which the boot device is attached.
Install BootManager	Install the A2 BootManager in the reserved volume space below LBA=63. Select the volume line suffixed #0 and with partition type 256 (Whole disk). Two files names are offered in a dialog box: the first file contains the MBR that will be replace the current MBR, the second one contains the tail and longer part of the BootManager. See http://www.ocp.inf.ethz.ch/wiki/Documentation/BootManager

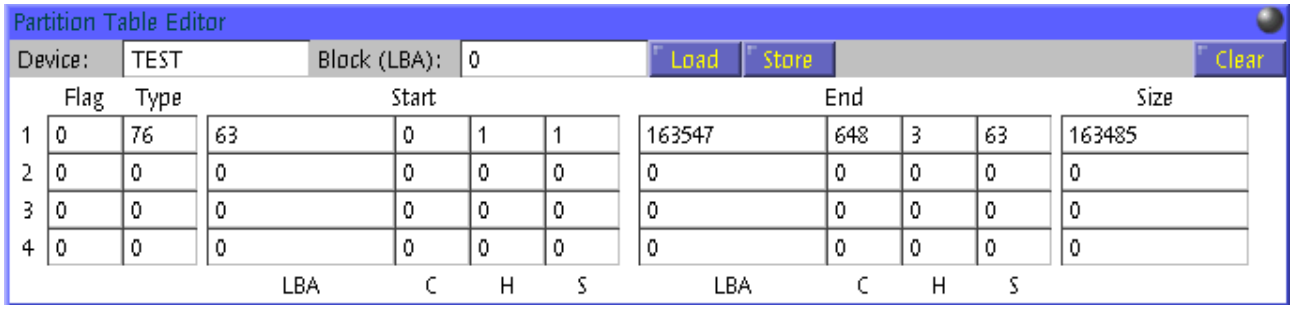
Tests

Write Test Data	
Verify Test Data	
Test Partition	
Write Zeros	Write zeros in all blocks of the selected partition. Specify the number of blocks per transfer (dialog box).

Operation

Obtain details of the execution of the selected operation.

Partition editor *– GUI*



Usage	Inspect and verify, and in rare cases edit, partition table entries.
Start	PartitionEditor.Open ~ opens a window as shown above, displaying the contents of the four partitions entries in the partition table in a block, under the condition that the specified block number is well chosen. Use the Partition tool information to locate the 1st block of a partition. Device: enter the device name, for example IDE1, then Block (LBA): 0 - is for sure the MBR of a volume Block (LBA): xzy choose it well as 1st block of an extended partition or of a chained partition (has two occupied entries out of 4) and press “Load”. Unless you know what you are doing, use neither “Clear” nor “Store” to avoid a catastrophe. The command may also be used with a virtual disk created in WinAos. Multiple windows may be opened.
Stop	Close the window(s). To clean up: SystemTools.Free PartitionEditor PartitionEditorComponents ~
Restorable	Yes

Desktop navigation

The concept of view port is used with reference to the conceptual raster (see Virtual desktop):

1. To enhance navigation:
   - in the WMNavigate.SetViewportRange command to precisely pinpoint individual view ports of any size. These commands are used by HotKeys.
   - in the WMNavigator.Open command to pinpoint 6 individual view ports around a view port as a group.
   - In the WMBackdrop.AddBackdropImage --fullscreen … ~ command to set a backdrop image for a view port.
2. In the screenshot utility, a portion of the virtual desktop, defined by a view port, is captured in a file.

Multiple backdrops for navigation

An interesting navigation technique is obtained by installing multiple backdrop images on the virtual desktop as described in the section “Backdrop installer”.

Virtual desktop navigation - GUI

Each window inserted on the virtual desktop can be considered a task and many such task windows may exist, making it difficult to keep an eye on all of them. To localize the many windows up to 10 window overviews or compact task lists, can be inserted on the desktop. An overview consists of a juxtaposition of icons each of which is an iconic representation of one of the windows.

<table>
<tr><td>Usage</td><td>A task bar showing the application windows open in the specified view port.</td></tr>
<tr><td rowspan="2">Start</td><td>WMNavigate.Open [options] id x y componentFile ~
options:
-f = frame the window to include a “Close” button. By default the window is not framed and the window can be hidden, restored or removed by executing one of the commands listed below.
-s = stay on top of other windows
-v = view port
id : unique identifier 0 <= id <= 10, declared in MaxNavigationWindows
x, y : position of the window's upper left corner
componentFile : the window content depends on this file which is either WMNavigateOverview.XML or WMNavigateTaskList.XML.</td></tr>
<tr><td>Command setting the view port size:
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 multiples of the display width and height
x, y : coordinates 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:
<table><tr><td>View port above at the left
-sd -1 -1 1 1</td><td>View port above
-sd 0 -1 1 1</td></tr><tr><td>View port at the left
-sd -1 0 1 1</td><td>Standard view port
-sd 0 0 1 1</td></tr></table>
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.

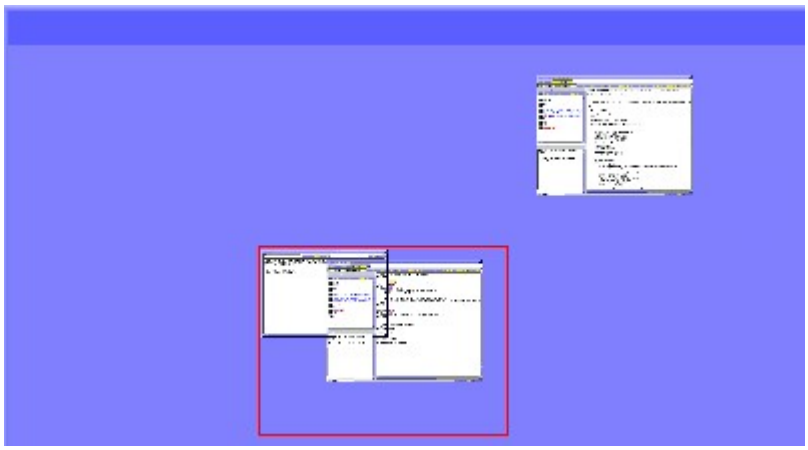
All four view ports together: -sd -1 -1 2 2 ~ equivalent to zoom factor 0.5

Hints:
1. To bring back the standard view port on the screen, use WMNavigate.SetViewportRange -sd 0 0 1 1 ~
2. To restore the desktop to the state it had after A2 started, use: WMRestorable.Load Auto.dsk ~

Commands acting on all task lists:
WMNavigate.HideNavigation ~
WMNavigate.RestoreNavigation ~
WMNavigate.ToggleNavigation ~</td></tr>
</table>

	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 ~ Close window with specified id number. To clean up: SystemTools.Free WMNavigate ~
Data files	WMNavigateIcons.tar WMNavigateOverview.XML, WMNavigateTaskList.XML

Navigator *- GUI*

Usage	Move around windows in an area larger than the screen.
Start	WMNavigator.Open ~  [WMScreenShot.SnapShotRange test.bmp 400 220 0 0 ~] The Navigator presents in its window, sized 400*200 pixels, an overview of approximately 2*3 view ports in which the current view port is easily localized by the thin red line surrounding it. The contained windows are scaled down, just to the limit of being recognizable for manipulating them as follows:

	Pressing ML on a mini-window and dragging moves it and its real counterpart simultaneously. The z-order of windows is not altered. A move can cause to loose sight on the counterpart as it crosses the screen border and moves beyond it. Conventional desktop navigation would require several elementary moves to reshuffle windows as easily as here. Note that backdrop images can likewise be moved around. Conversely, moving and resizing real windows is immediately reflected in the Navigator. Likewise, desktop navigation operations are reflected with a possible displacement of the thin red line locating the standard view port. A Meta+ML click on the Navigator brings the standard view port back on the screen (useful when lost in the blue). The Navigator remains anchored at the same position on the screen during the manipulations. However, it can be moved and resized. Note: unframed windows are not shown. The Navigator is placed and remains on top of all windows. It is framed but has no hot-spot in the title so that it cannot be closed except with a Close command or by unloading the module.
Stop	WMNavigator.Close ~ Closes the Navigator. To clean up: SystemTools.Free WMNavigator ~

Screen shot

Usage	Take a screen shot of a named view port or of a view port defined by position and size, scale it and save it in a file. The saved screenshot 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 screen shot of the named view, scales it to width and height, and stores it in the named file. fileName: extension can be .bmp or gif. viewName: has the structure “View#i” where 0 <= i <= 9. “View#0” is the standard view port and may be omitted. Other view ports correspond to VNC sessions (see section “VNC client”). height (or width and height): pixels. When omitted, no scaling is done and the view size is used. A list of all existing view ports is obtained by executing SystemTools.ListPlugins ~ 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 screen shot of a virtual desktop, 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 ~

Desktop customization

Menu page generator

Usage	Create an XML menu file from entries in the parameter list.
Start	MenuPages.Generate menuFilename menuName { entry } ~ entry = buttonCaption command [backgroundColor [hoverColor]] Defines the properties of a button: buttonCaption : choose an intuitive name command : enclose in quotes “ backgroundColor, hoverColor : hexadecimal value of RGBA for the background color and for the color while hovering with the mouse pointer. The default values are ?? See Alpha blending. menuFilename : is structured as “MenuPagexy.XML” and 0 <= xy <= 99 keeping in mind that: - in A2, as delivered, 00,10, 20, 30, 40, 50, 60, 70, 80, 90 are used - a maximum of 16 menus can appear in the Main menu - a menu has at least 4 columns, 120 pixels wide, with 2 buttons each. Example: MenuPages.Generate MenuPage15.xml TestMenu Hello "SystemTools.Show Hello" World "SystemTools.Show World" News "SystemTools.Show News" "" "" Reboot "SystemTools.Reboot" 0FF000060H 0FF0000FFH ~
Stop	To clean up: SystemTools.Free MenuPages ~

Skin loader – GUI

A skin is the look and feel of an application's graphical user interface (GUI). It is conditioned by data stored in a skin file (suffix .skin) and is customizable. The skin file describes the frame objects that decorates a window, that is the frame shape and color(s), the color of buttons and a set of predefined mouse pointers. A2 comes with a selection of skins, some of them are look-a-likes of third-party skins.

Usage	Offer to the user a list of skins to choose from and to install the selected one.
Start	WMSkinLoader.Open fileName ~ fileName : the name of an XML file containing a list of skin file names and the associated name that will appear in the GUI. SkinList.XML is the only file included in an A2 release. 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: **Group 1**

	<table><tr><th>GUI name</th><th>Skin file name</th></tr><tr><td>ZeroSkin</td><td>ZeroSkin.zip This was the original A2 look and is generated by program without skin file.</td></tr><tr><td>Reptile</td><td>reptile.skin</td></tr><tr><td>Kramer</td><td>pklook.skin</td></tr><tr><td>Traditional</td><td>traditional.skin</td></tr><tr><td>XP</td><td>winxp.skin (look-alike of Win XP)</td></tr><tr><td>OSX</td><td>aqua.skin (look-alike of Mac OSX)</td></tr><tr><td>Glass</td><td>glass.skin</td></tr><tr><td>StijnBW</td><td>stijnbw.skin</td></tr></table> **Group2 using a so-called FancyStartMenu.** <table><tr><th>GUI name</th><th>Skin file name</th></tr><tr><td>Blue</td><td>BluePlastic.skin</td></tr><tr><td>Red</td><td>redlook.skin</td></tr><tr><td>Stijn</td><td>stijn.skin</td></tr><tr><td>Christmas</td><td>xmas.skin</td></tr></table> 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. The list is limited to only 11 files, plus ZeroSkin, but it is customizable. A ML, MM or MR mouse click installs the skin. Behind the scene a SkinEngine.Load command is executed that installs the selected skin (see the next section “Skin engine”). **Shortcut**: Main menu → Looks → SkinLoader Alternatively, a skin can be installed using the “File manager” (refer to that section).
Stop	Close the window. To clean up: SystemTools.Free WMSkinLoader Looks ~
Data files	SkinList.XML and all of the files appearing in the table above.

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

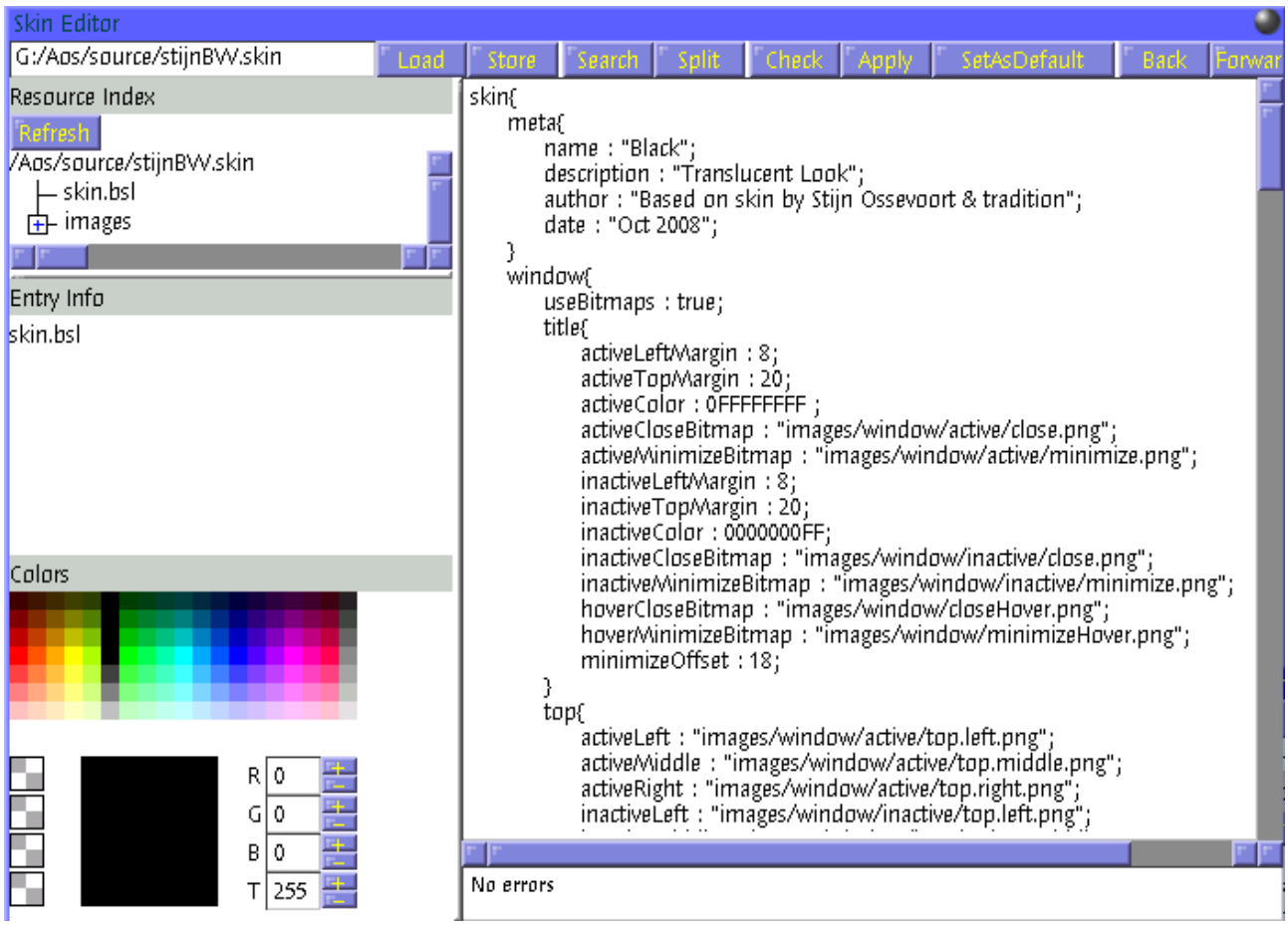
Skin engine

Usage	Install a skin either *temporarily*, or *permanently* in Configuration.XML. Instead of using one of the commands described in „Start, one can: 1. use the Skin loader offering a choice of skins to install temporarily 2. or edit the Autostart section in Configuration.XML: **either** install the desired skin automatically when A2 starts with: <Setting name="DefaultSkin" value="SkinEngine.Load xyz.skin"/> **or** install the programmatic ZeroSkin

	<Setting name="DefaultSkin" value="SkinEngine.Unload"/> **or** comment out such a line (equivalent to Ünload") <!-- Setting name="DefaultSkin" value="Skin …. "/ –>
Start	SkinEngine.Load fileName ~ Install the specified skin *temporarily*. Alternatively, a skin can be installed using the "File manager" (refer to that section). SkinEngine.Unload ~ Install the original ZeroSkin.zip which uses no skin file. SkinEngine.SetCurrentAsDefault ~ Register *permanently* the current skin in the Autostart section of Configuration.XML. This skin will be used the next time A2 starts.
Stop	To clean up: SystemTools.Free SkinEngine ~

Skin editor *- GUI*

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 is only a short introduction to the subject. This application is destined to experienced designers and one should refrain from modifying skins without good knowledge.
Start	SkinEditor.Open [fileName] ~ Open the named skin file for editing. Whenever a good skin is finalized, add it to the SkinList.XML file in order to let it appear the next time the "Skin loader" application is started. **Shortcut**: Main menu → Looks → SkinEditor Several skin editors may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free SkinEditor SkinEngine FNHistories SkinLanguage ~
Restorable	Yes



[WMScreenShot.SnapShotRange test.bmp 700 500 0 0 ~]

Control buttons at the top	
Load	Open the skin file for editing. The tree structure of skin appear at the left.
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. Errors are listed in the reporting area at the bottom right. In the absence of errors, „No errors“ is reported.
Apply	Check the correctness and install if correct
SetAsDefault	Behind the scene execute SkinEngine.SetCurrentAsDefault ~
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.
}
}
```

Backdrop loader - GUI

Usage	Offer to the user a list of desktop backdrops to choose from and to install the selected one.
Start	WMBackdropLoader.Open fileName ~ fileName : the name of an XML file listing the names of files containing images that are suitable for placing on the desktop and the associated name that will appear in the GUI. BackdropLoader.XML is the only file included in an A2 release. A list of desktop backdrops of various designs to choose from is presented. The information stored in the current BackdropList.XML file is: **GUI name** / **File name** Mars / mars.png Bluebottle / BluebottlePic0.png Saas Fee / SaasFee.jpg Clouds (JP2) / Clouds.jp2 Evening (JP2) / Evening.jp2 Rainbow (JP2) / Rainbow.jp2 The list is limited to 6 files only but it is customizable. A ML, MM or MR mouse click installs the backdrop. Behind the scene a WMBackdrop.AddBackdropImage filename ? ? ? ? ~ command is executed that installs the selected backdrop. Refer to the next section “Backdrop installer” which also explains which other image files are suitable. **Shortcut**: Main menu → Looks → Backdrops Alternatively, a backdrop can be installed using the “File manager” (refer to that section).
Stop	Close the window. To clean up: SystemTools.Free BackdropLoader Looks ~
Data files	BackdropList.XML and the listed file names

Backdrop installer - GUI

Usage	Install a backdrop image at a specified location on the virtual desktop. For better

	visual orientation and organization it is possible to place different backdrops on the desktop that can serve to group task windows into projects. Apart from serving as decoration and visual clues, the backdrops (a backdrop is a variant of a task window) can be navigated to with a ML click, whenever they are visible. Remember that he entire setup of the desktop can be saved persistently on demand (see section “Desktop save/restore”).
Start	**Install a backdrop relative to the *current* view port in the conceptual raster:** WMBackdrop.AddBackdropImage [--fullscreen | -f] fileName x y ~ -f : install the backdrop relative to the current view port fileName : see below x : abscissa upper left y : ordinate upper left image corner w.r.t. upper left view port corner An interesting navigation technique can be derived from this, according to this example, in concordance with the view port schema in “Virtual desktop navigation”: WMBackdrop.AddBackdropImage --fullscreen BluebottlePic0.png ~ WMBackdrop.AddBackdropImage --fullscreen mars.png 0 -1 ~ WMBackdrop.AddBackdropImage --fullscreen SaasFee.jpg -1 -1 ~ WMBackdrop.AddBackdropImage --fullscreen Clouds.jp2 -1 0 ~ Note how the images are installed first in the standard view port and from there 3 further ones counter-clockwise. Press Meta+Home to see the four view ports and an ML click on one of the backdrop images navigates back to it. **Install a backdrop at the specified position using the original image size:** WMBackdrop.AddBackdropImage fileName [x y] ~ fileName : see below x : abscissa upper left image corner relative to view port y : ordinate upper left image corner relative to view port **Install a backdrop at current view port and scale the image to the port size:** WMBackdrop.AddBackdropImage fileName ? ? ? ? ~ fileName : see below fileName may have the file name extensions: .bmp, .gif, .jp2, .jpeg, .jpg, .png
Stop	Close the window. To clean up: SystemTools.Free WMBackdrop ~
Restorable	Yes

Utilities

SystemTools

Usage	A collection of commands giving access to system information and functions. One cannot classify them easily, since each acts on very distinct system components either the hardware or the software.
Start	The number of system commands is very large and well documented in the “SystemTools” table of www.ocp.inf.ethz.ch/wiki/Documentation/Miscellaneous Only a few are documented here. SystemTools.DoCommands is special in that it accepts a list of commands as parameters where each command is delimited by a “~”. A final “~” is of course also required. Example:

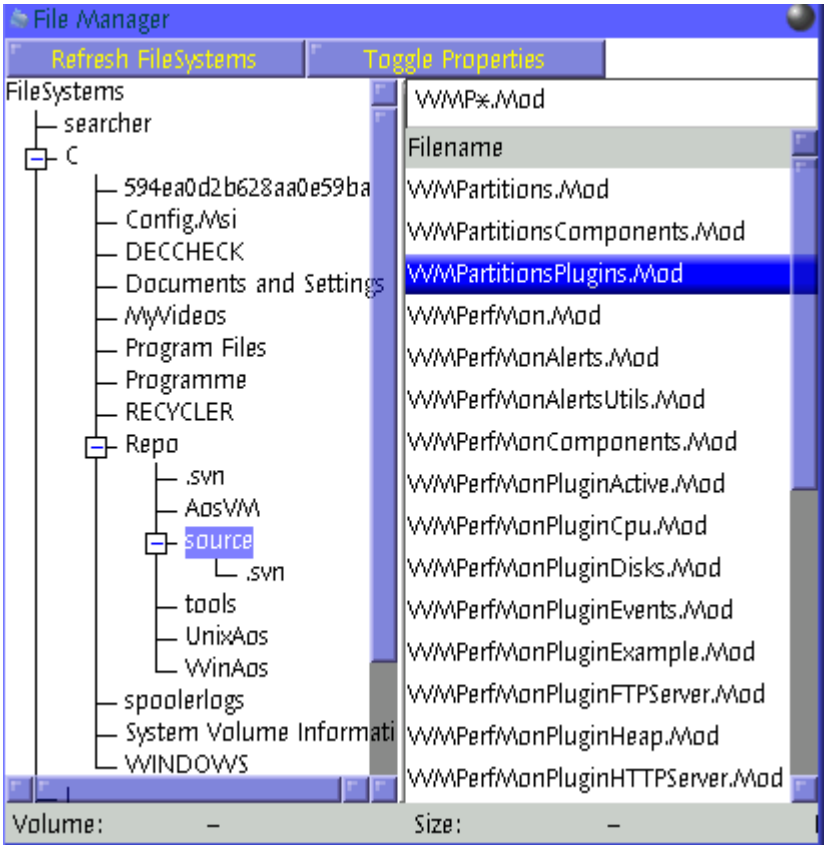
	SystemTools.DoCommands SystemTools.Version ~ SystemTools.Time ~ ~ Useful to batch commands either occasionally or for recurrent applications such as compiling the whole system. The example is edited with indents making it well readable however long the list of commands. SystemTools.Reboot ~ Reboot the system. **Shortcut**: Main menu → System → Reboot SystemTools.PowerDown ~ Shutdown the system. The user must turn power off. A2 does not support power management. **Shortcut**: Main menu → System → Shutdown
Stop	Module SystemTools cannot be unloaded since it contains Free.

Virtual keyboard *- GUI*

Usage	An application susceptible to find a use with a touch screen. Unfinished but operational.
Start	WMKBD.Open ~ Open a virtual keyboard that always stays on top. Set the cursor in a text editor and click with any mouse key to edit the text. The virtual keyboard keys CAPS, SHIFT, CTRL, META and ALT can all be toggled on and off. The keyboard description is stored in kbd.xml and can be customized.
Stop	To clean up: SystemTools.Free WMKBD VirtualKBD ~
Data file	kdb.xml

File manager *- GUI*

Usage	A file manager or file browser for operating on selections of files: edit, rename, duplicate, delete, compress and when the file contains source text, compile it.
Start	WMFileManager.Open ~ Opens a two-part panel with at the left the hierarchical structure of volumes, directories and sub-directories, and at the right a list of file names matching the wild card value edited in the input field at the top. The snapshot below is a WinAos sample.



[WMScreenShot.SnapShotRange text.bmp 410 420 0 0 ~]

A ML click on a file name selects the file. A subsequent MR click open a pop-up menu with a selection of buttons captioned as shown in the left column:

Open	the selected file with the application associated with the file name extension, as defined in Configuration.XML in the section “Filehandlers”. For example, when the extension is: “mod” or “asm” the PET editor is opened, “text” or “txt” Notepad is opened.
Rename	the selected file
Duplicate	the selected file
Tar	compress the selected file(s)
Delete	the selected file(s)
Compile	the selected file, when the suffix is “.Mod”
Set as skin	the selected file, when the suffix is “.skin”
Set as background	The selected file, when the suffix is: .bmp, .gif, .jp2, .jpeg, .jpg, .png

The 3 last entries will only show up when the file name suffix is as declared.

To select several file names, press MM on a file name, then hold and drag the mouse upwards or downwards. When several files are selected, only “Tar” and “Delete” will show up.

Refresh FileSystems: collapse the file hierarchy adjusting the list after a mount

	or unmount operation. **Toggle Properties**: insert or remove file size and date at the right of the file name list. **Shortcut**: Main menu → Files → Files Several file managers may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMFileManager ~
Restorable	Yes

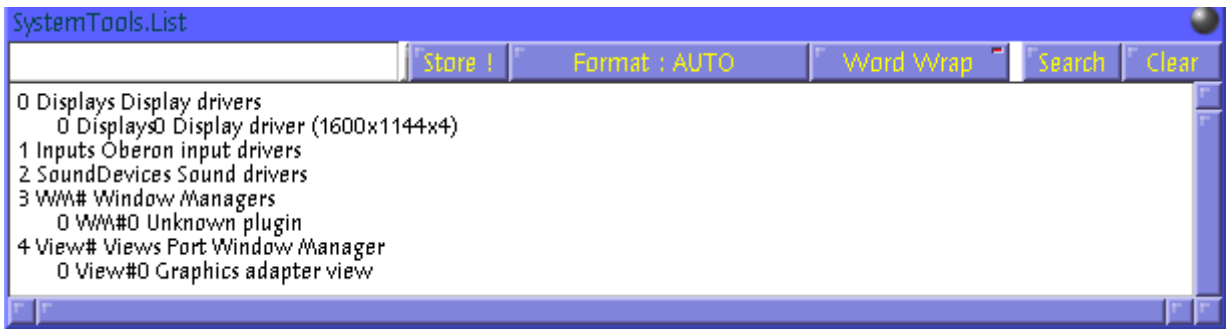
Hot key commands

Usage	Three commands developed for use in hot key definitions.
Start	HotKeysCommands.ClearLog ~ Clear the Kernel Log, even when the window is not on the desktop. HotKeysCommands.EnterCommand ~ Opens a blue window where a command can be entered and executed with a final ENTER. The window cannot be moved and is not persistent, i.e. disappears unless a command is entered. HotKeysCommands.SimulateMouse MouseX | MouseY | MouseButtons | MouseWheel value ~
Stop	To clean up: SystemTools.Free HotKeysCommands ~

Text converter

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 ~

WMUtilities *- GUI*

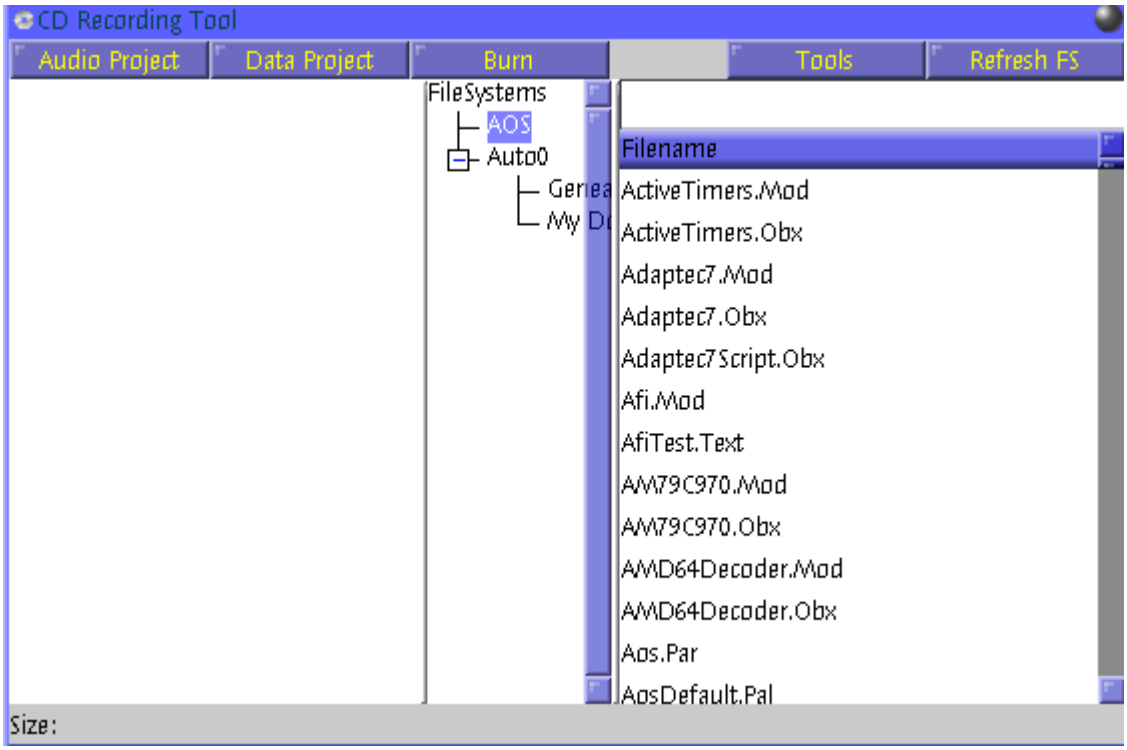


Usage	A utility command for executing a command and to redirect its output to a dedicated window instead of sending it to the Kernel log..
Start	WMUtilities.Call [--blocking | -b] [commandName [parameterList]] ~ Execute a command appearing as parameter. The data produced is redirected to a new window. The option -b forces to wait the end of the output. The text editor in the window has the functionality as a Notepad, except for the missing “Close” button. Example: WMUtilities.Call SystemTools.List plugins ~ see picture Several WMUtilities may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMUtilities ~

Task scheduler *– GUI*

Usage	
Start	WMTaskScheduler.Open ~ Several task schedulers may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMTaskScheduler ~
Restorable	

CD recorder *- GUI*



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 an ISO-9660 image file as produced with IsoImages for instance. Copy Data CD Blank CDRW Disc Information
Start	WMCDRecorder.Open ~ Several CD recorders may be running in parallel. **Shortcut**: Main menu → Files → CDRecorder
Stop	Close the window(s). To clean up: SystemTools.Free WMCDRecorder ~
Build	Native only: A2, A2Mini, AMD64
Data files	CDRecordUtils.Mod, CDRecordLib.Mod, MakeIsoImages.Mod, CDRecord.Mod, WMCDRecorder.Mod

Iso image file creator

Usage	Create a bootable ISO-9660 image file, following the El Torito specifications. The volume id is “BLUEBOTTLE” and the publisher id is ETH-ZURICH”. This tool is used for producing an A2 build (see Build.Tool).
Start	IsoImages.Make isofileName imageName ~ The file so produced can be used for burning a CD with the WMCDRecorder or any other utility such as Nero.
Stop	SystemTools.Free IsoImages ~
Build	Native only: A2, A2Mini, AMD64

***Kernel log** - GUI*

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 ~ **Shortcut**: Main menu → System → Log
Stop	Close the window. To clean up: SystemTools.Free WMKernelLog ~
Restorable	Yes
Data files	KernelLogger.Mod WMKernelLog.Mod

Clock *- GUI*

<table>
<tr><td>Usage</td><td>A transparent digital or analog clock, presented in an unframed window.</td></tr>
<tr><td>Start</td><td>WMClock.Open ~
At first, a digital clock (hh:mm:ss) is presented. A ML 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>Alternate the color black and white</td></tr>
</table>
The analog clock face can be resized by holding MM on it, then pressing and dragging MR.
Several clocks may be opened.

Shortcut: Main menu → System → Clock</td></tr>
<tr><td>Stop</td><td>Let the pop-up window appear and click „Close“.
To clean up: SystemTools.Free WMClock ~</td></tr>
<tr><td>Restorable</td><td>Yes</td></tr>
</table>

Calendar *- GUI*

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

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

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. Tells 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 ~

Desktop icon singleton *- GUI*

<table>
<tr><td>Usage</td><td>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.</td></tr>
<tr><td>Start</td><td>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.</td></tr>
<tr><td>Stop</td><td>Close the window(s) via the pop-up menu.
To clean up: SystemTools.Free WMDesktopIcons ~</td></tr>
<tr><td>Restorable</td><td>Yes</td></tr>
</table>

Display or convert a picture in a file *- GUI*

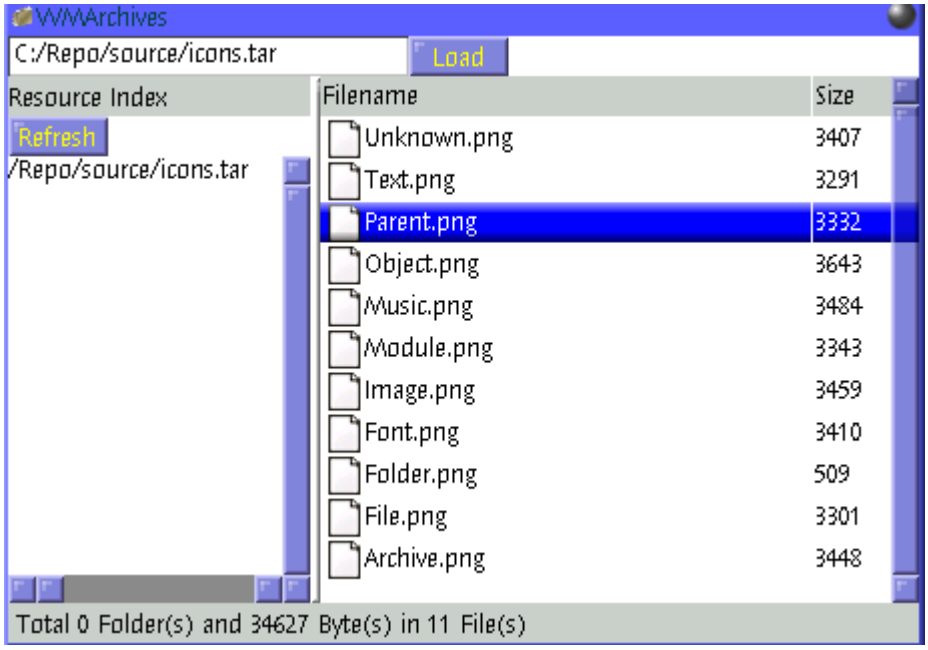
<table>
<tr><td>Usage</td><td>Display a picture stored in a file or convert it to another format, provided the appropriate Codec(s) is found. The available image Codecs defined in Configuration.XML are:

Decoder for: bmp, gif, png, jpeg, jpg, jp2, svg
Encoder for: bmp, gif</td></tr>
<tr><td>Start</td><td>WMPicView.Open fileName.extension ~

WMPicView.Convert sourcefile destinationfile ~</td></tr>
<tr><td>Stop</td><td>To clean up: SystemTools.Free WMPicView ~</td></tr>
</table>

Archives *- GUI*

<table>
<tr><td>Usage</td><td>A file archiving tool.</td></tr>
<tr><td>Start</td><td>WMArchives.Open [fileName] ~ Opens a window with at the left the entire path of the file. A MM click on the name inserts at the right a list of the archived files with their size. The command is useful for files with file name extension .tar and .skin.</td></tr>
</table>

	 A ML click on a file name selects it. To select several contiguous names, press ML on the first name, then drag on ML downwards or upwards and release ML on the last one. A ML click on the header line de-selects all. Selected file name(s) may be copied to an open file manager by drag and drop with MM. Several file archives may be running in parallel. **Shortcut**: Main menu → Tools → Archiver
Stop	Close the window(s). To clean up: SystemTools.Free WMArchives ~
Restorable	Yes
Data files	Archives.Mod WMArchives.Mod

Tar

Tar (derived from “tape archive”), is both a file format and the name of the program handling a file of that format. A .tar file is the concatenation of one or more files and is uncompressed. A component file can be accessed directly with this syntax:

tarFileName://componentName

Example:

Usage	Pack several files into a single one. Using these commands is an alternative to using Archives.
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 ~
Data files	Archives.Mod, Tar.Mod

Oberon

Usage	The complete Oberon system can run as one active object within a window in the display space. A special window simulates an Oberon-compatible display driver and forwards mouse and keyboard events to the Oberon loop. This allows continued use of a wealth of application programs that have been developed for Native Oberon and Plugin Oberon. In the early days of the A2 system, Oberon was used as a cross-development environment for A2 applications.
Start native	OberonDisplay.Install [name] width [character] height [x y] ~ name : window name. Default is Oberon width : window width in pixels character : a single character, conveniently “x”, as for 1280x1024 height : window height in pixels x, y : top left window corner coordinates, Default is 0 0 as is needed to start Native Oberon in full screen. Install a special window able to host a Native Oberon system. followed by: Oberon.Call System.Init Oberon ~ to start the Native Oberon in the window. **Shortcut**: Main menu → Tools → Oberon or Main menu → Develop → Oberon FullScreen OberonDisplay.Remove ~ ? OberonDisplay.SwitchToWM ~ ?
Stop native	Close the window or use the Main menu shortcuts to reboot or to shutdown A2. Refer to the section “SystemTools”.
Start WinAos	OberonDisplay Install ~ compatible with the above and sufficient for starting Oberon. In full screen. The window is entitled “ETH Oberon”.
Stop WinAos	Close the window or use the Main menu shortcuts to reboot or to shutdown A2. Refer to the section “SystemTools”. Reboot and shutdown are identical.

System drivers

Serial ports set-up

Usage	Set the serial port operating parameters ahead of starting the driver.
Start	V24.Install ~ Set the operating parameters (port base address and IRQ) for up to 8 serial ports. For the two first COM ports default values are ready for use. Only in exceptional cases the data must be extracted from the configuration data, and that is necessarily so for the remaining 6 ports. The default values for the two first ports are: **Port** / **Port base address** / **IRQ** COM1 / 3F8H / 4 COM2 / 2F8H / 3 Execution in WinAos has no effect. V24.Scan ~ Scan the installed serial ports and determine their chip type. Execution in WinAos lists all ports from COM1 to COM8, telling which exist.
Stop	SystemTools.Free V24 ~

Serial port generic driver

Usage	Install a generic serial communication driver.
Start	As soon as it is loaded, the module offers all the needed driving capabilities. The following commands help to verify the readiness. Serials.Show ~ Lists the available COM Ports in the Kernel log. Serials.Test ~ Test COM1 and if present COM2 with the generic driver. Serials.CloseAllPorts ~
Stop	SystemTools.Free Serials V24 ~

Diskette driver

Usage	Install a legacy diskette driver.
Start	Diskettes.Install ~ Install a driver for a diskette device supporting 1.44MB floppies. The Installer and the Partition framework will recognize up to 2 such devices.
Stop	SystemTools.Free Diskettes ~

Virtual serial port driver

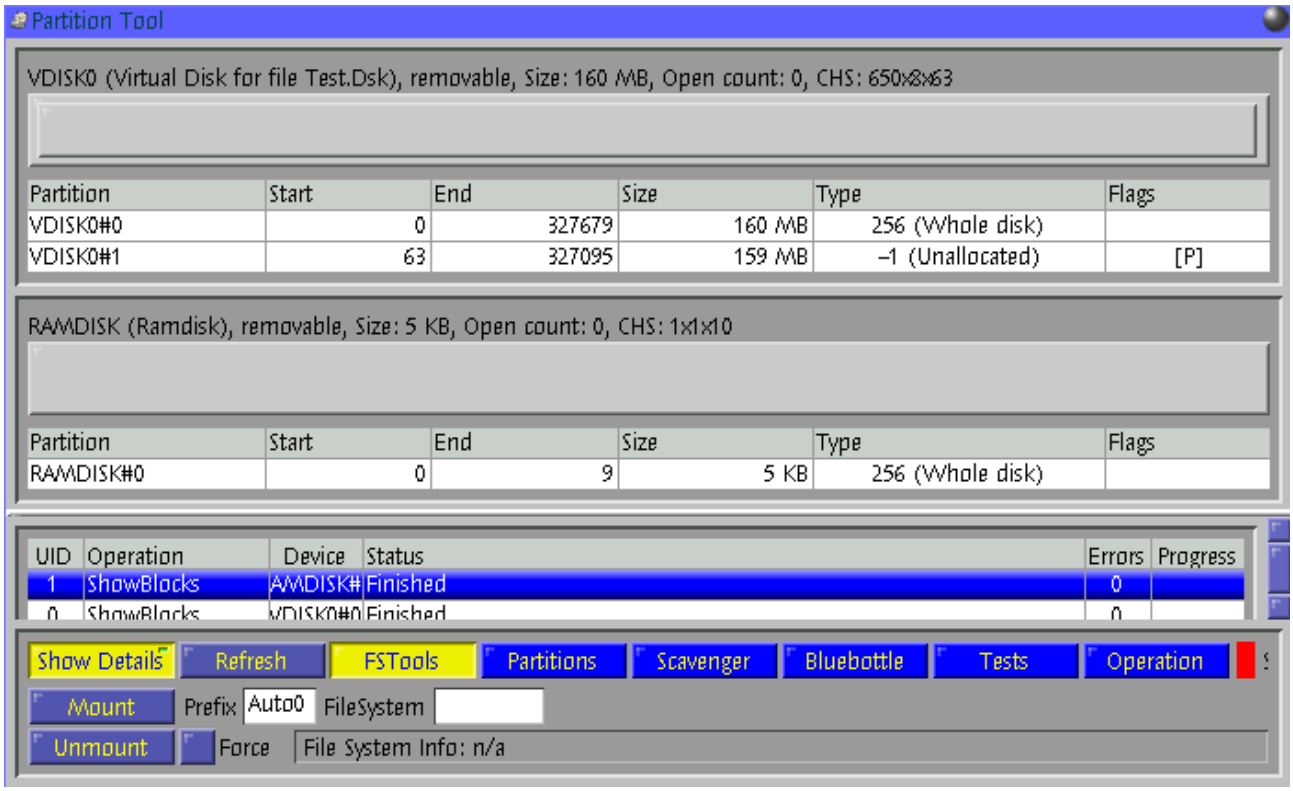
Usage	Simulate a data stream produced by a serial port driving an application under development which should receive data from a real serial port.
Start	SerialsVirtual.Install ~ Install two virtual serial ports linked to each other, as would two real serial ports be linked by a null-modem cable. Data sent by one port is received by the other and vice-versa. Observe the defined port names in the Kernel Log. SerialsVirtual.SendFile portNbr fileName [Loop] ~ portNbr : decimal value extracted from the COMxy string appearing in the Kernel Log. fileName : Loop : send the named file endlessly. Sending can be stopped with a StopSendFile command. Send the content of the specified file to the specified serial port, which receives it. SerialsVirtual.StopSendFile ~ Stop sending. SerialsVirtula.InstallSniffer [portNbr] ~ Install a virtual sniffer port as proxy for the specified serial port.
Stop	SystemTools.Free SerialsVirtual ~

Virtual disk

Usage	Create a virtual disk in a file or in RAM. For use when building A2, when A2 is hosted by Windows or Unix, or in a virtual machine environment.
Start	VirtualDisks.Create [-b=blocksize] fileName nbrOfBlocks ~ -b=blocksize - Is 512 by default Create an empty file fileName for use as virtual disk. The virtual disk is then created by executing the next command. VirtualDisks.Install [options] diskName fileName ~ Install fileName as file disk diskName. The disk will appear in a partition list shown

	by WMPartitions.Open ~ . “options” define the disk size and its geometry: [-b=blocksize] [-c=cylinders] [-h=heads] [-s=sectors] blocksize is 512 by default. The 3 remaining options: a) may be omitted and are then assigned values in agreement with the file size. b) specific values are only needed when the geometry is relevant. The virtual disk has all the characteristics of a partitioned volume with an unallocated partition, and the user may proceed with partitioning and formatting. VirtualDisks.InstallRamDisk [options] diskName size ~ Install diskName of the specified size, in 512 bytes blocks, in RAM. The disk will appear in a partition list shown by WMPartitions.Open ~ . The volume is totally empty. An MBR must be installed, the disk must be partitioned or not as needed and the partition(s) formatted. “options” define the disk size and its geometry [-b=blocksize] [-c=cylinders] [-h=heads] [-s=sectors] blocksize is 512 by default. The 3 remaining options: a) may be omitted and are then assigned the values -c=1 -h=1 -s= size. b) specific values are only needed when the geometry is relevant. VirtualDisks.Uninstall diskName ~ Uninstall the virtual disk. If it was hosted by a file, the file is not deleted, and must be deleted manually.
Stop	To clean up: SystemTools.Free VirtualDisks ~

This is the partition layout after having executed:
VirtualDisks.Create Test.Dsk 163840 ~
VirtualDisks.Install VDISK0 Test.Dsk~
VirtualDisks.InstallRamdisk RAMDISK 10 ~



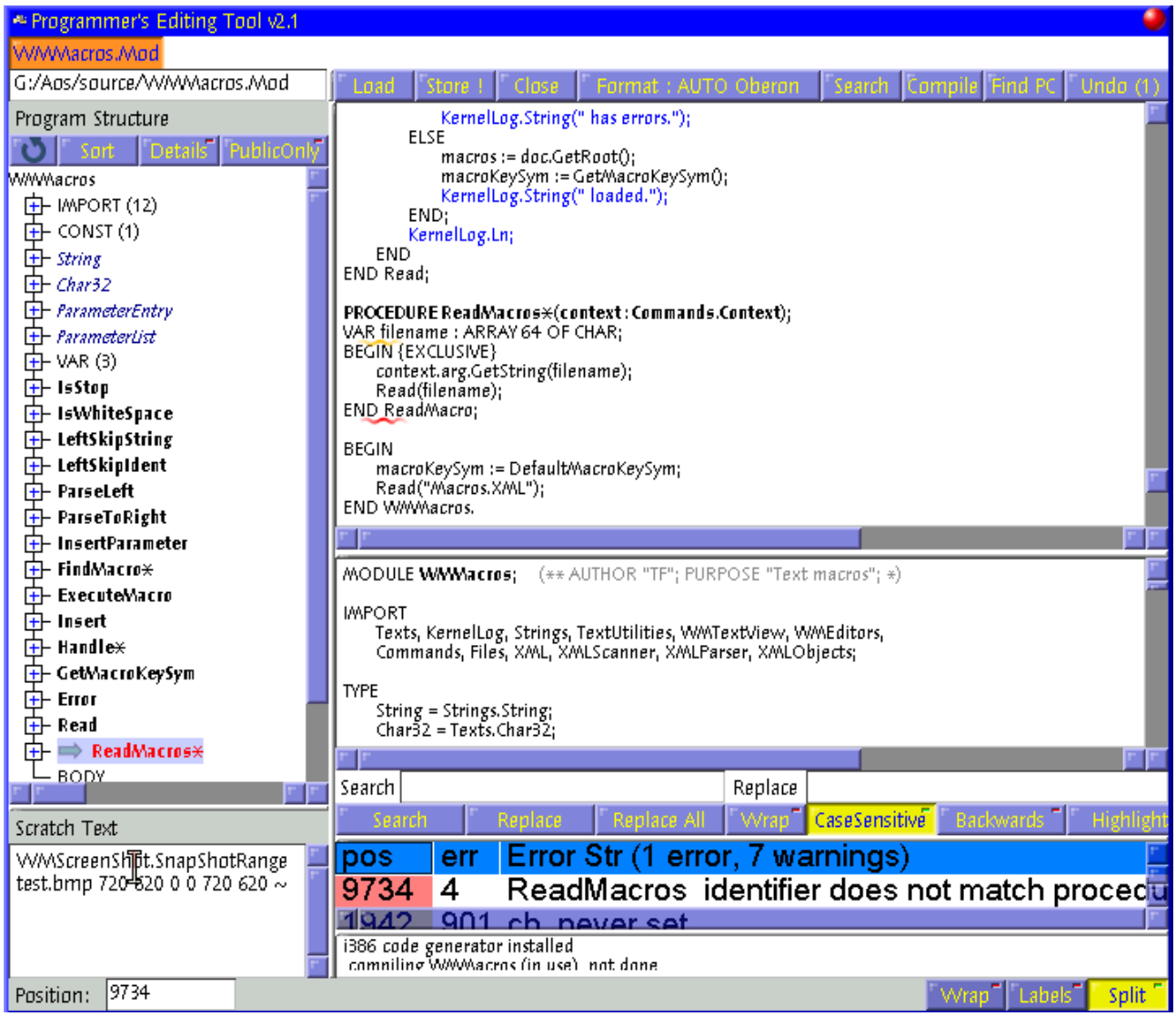
[WMScreenShot.SnapShotRange test.bmp 700 425 0 0 ~]

Hauppauge TV driver

Usage	A driver for Hauppauge television cards.
Start	BT848.Install ~
Stop	To clean up: SystemTools.Free BT848 TVDriver ~
Build	Native only: A2, A2Mini, AMD64

Developer tools

Programmer's Editing Tool or PET - GUI



<table>
<tr><td>Usage</td><td>The central Programmer's Editing Tool consisting of a number of standard GUI components and some glue code to integrate the compiler or the XML parser.
The snapshot above is described in this table.</td></tr>
<tr><td>Start</td><td>PET.Open [fileName] ~

<table>
<tr><th colspan="2">GUI widgets on top from left to right</th></tr>
<tr><td>File name input field</td><td>Contains the file name of the program text to load or to store.</td></tr>
<tr><td>Load</td><td>Load the text from the file specified in the File name field</td></tr>
<tr><td>Store</td><td>Store the text in the file specified in the File name field.
The button is marked with an “!” if the text has been changed since it was last stored.</td></tr>
<tr><td>Close</td><td>Close the currently loaded text and warns with a dialog box when the text has been changed since it was last stored.</td></tr>
<tr><td>Format</td><td>Open a menu for selecting the text format to load or store.</td></tr>
<tr><td>Search</td><td>Open a search panel for searching a string within the text.</td></tr>
<tr><td>Compile (*)</td><td>Compile the text using the compiler options specified</td></tr>
<tr><td>FindPC (*)</td><td>Search for a selected program counter position in the text. The counter value is taken from the last selection, normally in a TRAP window. If no number is selected, a query input dialog opens asking to enter a value.</td></tr>
<tr><td>Undo (i)</td><td>Undo the last edit action. “i” denotes the total number of successive text alterations.</td></tr>
<tr><td>Redo (i)</td><td>Redo what was last undone.</td></tr>
<tr><td>Compiler options input field (*)</td><td>Contains the options destined to the compiler.</td></tr>
<tr><td colspan="2">(*) These disappear and are replaced by “Parse” when an XML document is loaded instead of a program text.</td></tr>
<tr><td>Parse</td><td>Parse the text to check the XML text correctness.</td></tr>
</table>

<table>
<tr><th colspan="2">GUI widgets at the left from top to bottom</th></tr>
<tr><td>Program structure</td><td>Displays the structure of a loaded Active Oberon source text. Clicking into the structural overview positions the text cursor at the respective position in the source text at the right.

If an XML document is loaded instead of a program text, the the document's structure is represented.</td></tr>
<tr><td>Scratch text</td><td>A text tool area that is synchronized between all instances of PET. It is used to note frequently used commands. It is a good example of the model view architecture of the text system: a single text model is displayed by a possibly large number of text views.</td></tr>
</table>

Several editors may be running in parallel.

Shortcut: Main menu → Develop → IDE (with empty text)</td></tr>
<tr><td>Stop</td><td>Close the window(s).</td></tr>
</table>

	To clean up: SystemTools.Free PET ~
Restorable	Yes
Data files	PETIcons.tar

Macros

Usage	A tool destined mainly for *Active Oberon* programmers.
Start	WMMacros.ReadMacro [fileName] ~ Readies the macro plug-in. The default fileName is Macros.XML included in a build and based on long time experience. A personal macro file may be derived from it. Now, position the cursor at the desired location in an editor and type a macro string according to the rules. When the macro plug-in recognizes the *Insert* key, it reads the the text at the cursor position backwards to either a white space character (space, line break or colon). The text between this white space character and the cursor position is interpreted as a macro procedure name. Hence, the number of different macros that can be invoked by the *Insert* key is not limited. The macro names and macro functions are specified in Macros.XML. They replace the macro name in the text with a larger piece of text. The larger text can either be a fixed string or a parameterized text. The macro parameters are searched in front of the macro name, each separated by a colon. The macro parameters can be inserted between fixed string elements in any order and repetition.
Stop	To clean up: SystemTools.Free WMMacros ~
Data files	Macros.XML organized in four macro groups, as suggested in these examples:

Group	Macro	Evaluation
Mail	jg	gutknecht@inf.ethz.ch
XML	key:t	<key></key>
	macro:T	<macro> </macro>
Active Oberon A large number of macros supplied	test:P	PROCEDURE Test; BEGIN END Test.
	IME:WMInputMethods.IME:o	TYPE IME= OBJECT(WMInputMethods.IME) VAR END IME;
	Mydoc:doc	(** AUTHOR "Mydoc"; PURPOSE ""; *)
Greek alphabet	alpha	

The middle column specifies the unprocessed macro string with optional parameters. The right column shows text that results when the macro at he left is invoked with the *Insert* key.

The Insert key is defined as the one that triggers the macro processing in Macros.XML as a hexadecimal value:

```
<MacroKey keysym="FF63"/>
```

	The specified key cannot be used for other purpose. Use WMKeyCode.Open ~ to find out the keysym value of keys.

HotKeys

Hot key definitions bind program functions to a single keystroke or a few keystrokes for efficiency users. 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>
```

The “command=” token may list multiple commands separated by a semi-column “;”.

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 commands in the background), as for example WMNavigate commands (described above) setting the view port:
	F1 :View port above at the left / F2 :View port above
	F3 :View port at the left / F4 : Standard desktop/view port
	F5 : All these four view ports together
Start	HotKeys.Open [fileName] ~ Load the hot keys definition from an XML file. The default file name 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 ~
Data files	HotKeys.XML

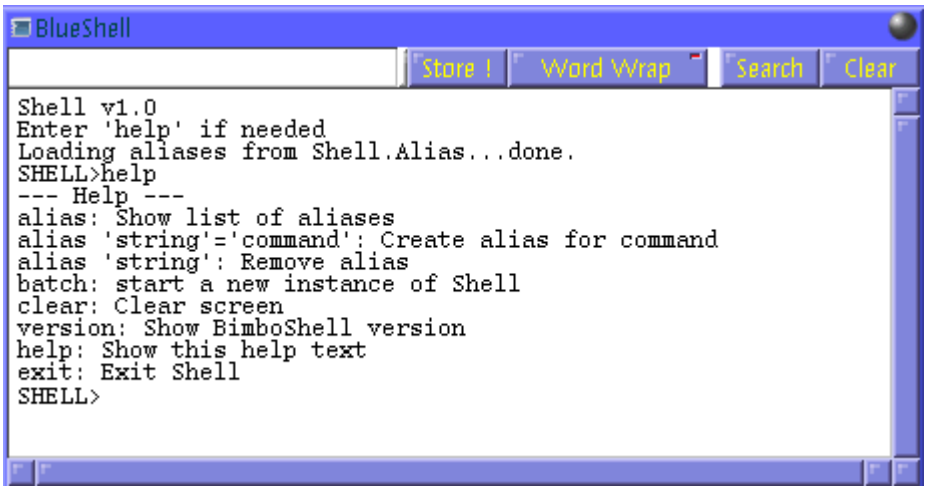
The HotKeys.XML file included proposes the following bindings:

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
Clear the Log window		
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 scrolling text up or down		
Ctrl+Up	HotKeysCommands.Simulate MouseWheel -3 = scroll 3 lines up	WHEEL UP
Ctrl+Down	HotKeysCommands.Simulate MouseWheel 3 = scroll 3 lines down	WHEEL DOWN
Key remapping		
Ctrl+I	REMAP UP	UP
Ctrl+J	REMAP LEFT	LEFT
Ctrl+K	REMAP DOWN	DOWN
Ctrl+L	REMAP RIGHT	RIGHT

Shell



[WMScreenShot.SnapShotRange test.bmp 460 240 0 0 ~]

Usage	The main task of a shell is to offer a command line editor where the user can type and modify a command that is interpreted according to the rules of the command interpreter when the user presses the *Enter* key. In the simplest case, the command line contains a command that is optionally followed by parameters. Recall that in the TUI of A2 a command can be placed within *any text*, and the likelihood of having at least an open text editor on the desktop is great, but the real power of the shell resides in possibility to glue commands together to: Redirect 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 „|“ (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. Note: the same Shell module is used in Serial shell.
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, facilitating the user's task, 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“ **.** A long lasting command execution can be observed until it terminates. For commands executed in a TUI environment, 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 the output can also be redirected to a file. One observes that more aliases can be added and removed at will to a user's preference and on-the-fly, 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. Use CursorUp to retrieve earlier commands in reverse order and CursorDown to retrieve then in oder. The list is organized as a round-robin. Several shells may be running in parallel. **Shortcut**: Main menu → Tools → Console
Stop	At the prompt, enter “exit” followed by a carriage return. Close the window(s). To clean up: SystemTools.Free WMShell Shell ~
Restorable	Yes
Data files	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. **Shortcut**: Main menu → Tools → Search
Stop	Close the window(s). To clean up: SystemTools.Free WMSearchTool ~
Restorable	Yes

Search/replace tool for a text stretch in a selection of files

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 [option] 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 ~

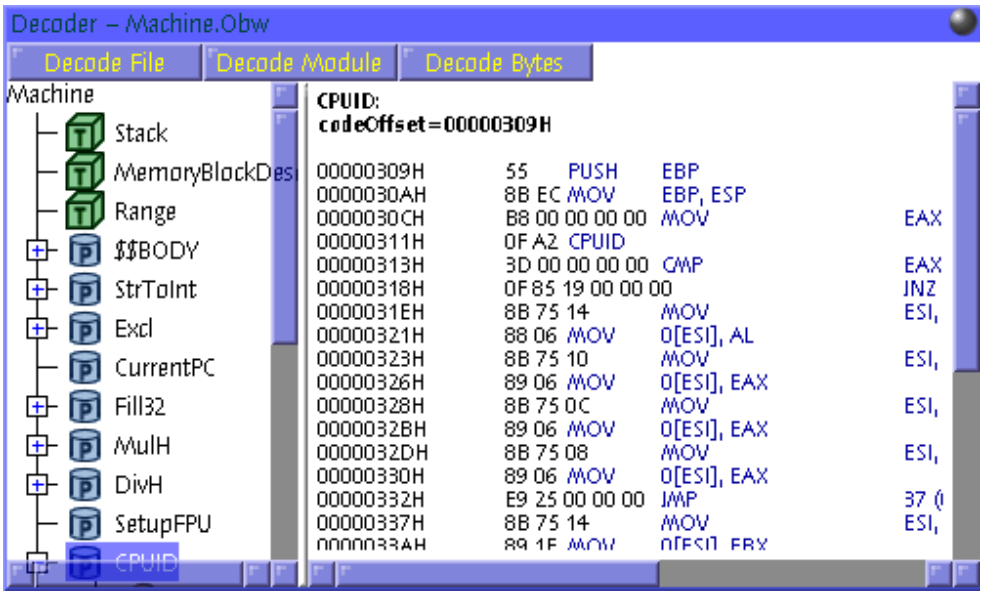
Text comparison tool for comparing two texts in parallel – GUI

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 file names 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 file name 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. **Shortcut**: Main menu → Tools → Diff
Stop	Close the window(s). To clean up: SystemTools.Free WMDiff DiffLib ~
Data files	DiffLib.Mod WMDiff.Mod

Text comparison tool

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 file name at the left or at the right and with (linenumber:position). When no difference is detected nothing appears in the log, except the file names.
Start	DiffLib.Compare fileNameL fileNameR ~
Stop	To clean up: SystemTools.Free DiffLib ~

Decoder of binary executable code - GUI

Usage	Decode a binary executable code for one of the supported CPUs in the corresponding assembly language.
Start	Decoder.OpenEmpty ~ Decoder.Open fileName (procName | progCounter) ~ fileName: name of the program to decode procName: name of the procedure to decode progCounter: integer value for the program counter Open a decoder window showing the structure and the assembly language text of a binary executable. The decoder is directed by the module name suffix which can be: .Obx : native i386 .Obw : WinAos uses the same decoder as .Obx .Oba : ARM .Abx : AMD64  [WMScreenShot.SnapShotRange test.bmp 500 300 0 0 ~] Three possibilities are offered (a dialog window helps the user): Decode File: decode a file specified by the full name Decode Module: decode a *loaded* module, specified without suffix Decode Bytes: decode a series of bytes in hexadecimal format, separated by spaces. Several decoders may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free AMD64Decoder ARMDecoder I386Decoder Decoder MemoryReader ~
Data files	DecoderRes.zip

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 ~

Communication

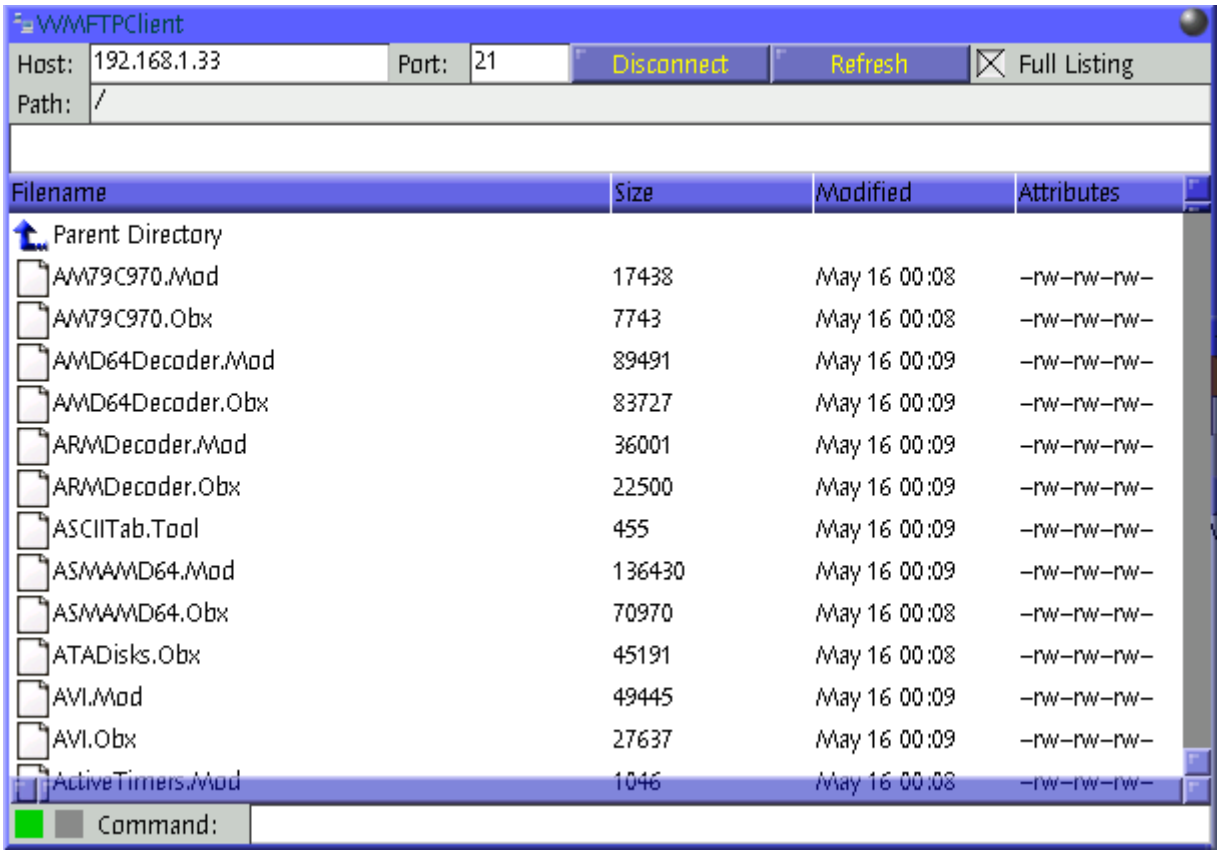
FTP server

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. Data transfer activity is automatically logged.
Start	WebFTPServer.Start [„\l:“ logfileName] ~ Start the FTP server. The optional parameter prepares the logging of FTP clients activity. The default is named FTP.log. A confirmation message is sent to the Kernel log. The server is listening on port 21. Thereafter, any number of potential FTP client users may be added by executing: WebFTPServer.AddUser userName password maxLogins permissions [root] ~ userName : the name with which an FTP client user must sign in password : the password to enter by the client maxLogins : maximum number of concurrent logins. A negative value allows an unlimited number of logins. permissions : assigned to an FTP client r : permission to read from server w : permission to write to server p : a password must be used m : must use an e-mail address as password root: the root directory that is accessible to a client Add a user with its specifications. Users data is stored in WebFTPUsers.dat and is persistent from session to session. Here 2 examples: WebFTPServer.AddUser userName password -1 rwp FAT:~ WebFTPServer.AddUser anonymous none 3 rwpm FAT:/ftproot/ ~ WebFTPServer.RemoveUser userName ~ Remove the named user. WebFTPServer.ListUsers ~ List all users with their details.
Stop	WebFTPServer.Stop ~
Data files	Log file, WebFTPUsers.dat

A selection of FTP clients other than the A2 server

Connect to server → ftp:// ...	Mac OS X	Files can be downloaded from the server but not uploaded.
Cyberduck	Mac OS X	http://cyberduck.ch

FTP client – *GUI*



[WMScreenShot.SnapShotRange test.bmp 606 422 0 0 ~]

Usage	
Start	WMFTPClient.Open ~ The field „Host“ contains “bluebottle.ethz.ch”. Replace that with either the domain name or the IP address of the FTP server to access. The field “Port” contains “21” which is the number of the listening port on the server. A two-step dialog asks to enter a userName and a password. The green/red square in the lower left corner of the window indicates the state of the connection. When the connection is well established, the files contained in the root made accessible to the client are listed. Example of connection with the Bluebottle FTP server: Host : bluebottle.ethz.ch Port : 21 (leave the default) Username : ocp (for Oberon Community Platform) Password : download giving access to several A2 build images ready for download, as documented in http://www.bluebottle.ethz.ch/download.html Files can be transferred from one PC to the other by selecting and dragging with ML. Several connections may be running in parallel. **Shortcut**: Main menu → Tools → FTP

Stop	Hit the „Disconnect“ button. Close the window(s). To clean up: SystemTools.Free WMFTPClient FTPClient ~

A selection of FTP servers other than the A2 server

Sharing → File sharing	Mac OS X	

FTP

Usage	Command line controlled FTP.
Start	FTPClient.Open hostName userName password ~ hostName : domain name or IP address of the FTP server userName : name assigned by the FTP server password : password assigned by the FTP server FTP.Directory ~ FTP.ChangeDir directoryName ~ FTP.MakeDir directoryName ~ FTP.GetFiles {fileName} ~ FTP.PutFiles {fileName} ~ FTP.PutTexts {fileName} ~
Stop	FTP.Close ~ To clean up: SystemTools.Free FTP FTPClient TCP DNS IP Network WSock32 ~

TFTP server

Usage	Command line controlled TFTP server.
Start	TFTP.Start ~
Stop	TFTP.Stop ~ To clean up: SystemTools.Free TFTP IP UDP ~
Build	Native only: A2, A2Mini, AMD64

Ping

Usage	Determine whether it is possible to reach an address. Handles Ipv4 and Ipv6.
Start	Ping.Ping hostName [pingSize] [timeout] ~ hostName : domain name or IP address pingSize : ping packet size in bytes. Default is CONST PingSize = 32. timeout : echo reply time-out in ms. Default is CONST Timeout = 1000. The information appears in the Kernel log.
Stop	To clean up: SystemTools.Free Ping ~
Build	Native only: A2, A2Mini, AMD64

Trace route

Usage	Determine the route taken by packets across an IP network. Handles Ipv4 and Ipv6.
Start	TraceRoute.TraceRoute hostName [pingSize] [timeout] ~ hostName : domain name or IP address pingSize : ping packet size in bytes. Default is CONST PingSize = 32. timeout : echo reply time-out in ms. Default is CONST Timeout = 1000. The information appears in the Kernel log.
Stop	To clean up: SystemTools.Free TraceRoute ~
Build	Native only: A2, A2Mini, AMD64

TCP port logger

Usage	Log TCP port activity.
Start	TCPPortLog.Install ~ Trace data is directed to the Kernel log. TCPPortLog.Remove ~
Stop	To clean up: SystemTools.Free TCPPortLog ~
Build	Native only: A2, A2Mini, AMD64

Quote server

Usage	Quote of the day server.
Start	QuoteServer.Open [fileName] ~ fileName : name of the text (ASCII) file from where the quotes are extracted. Default is CONST DefaultQuoteFile = “Quotes.txt”. The maximum length of a quote is CONST MaxQuoteLen = 511. A quote may contain line breaks. Quotes are separated by at least one comment line, beginning with a hash “#”. QuoteServer.ReadQuotes fileName ~ fileName : name of the text file from where the quotes are extracted.
Stop	To clean up: SystemTools.Free QuoteServer ~
Data files	Quotes.txt

Quote reader

Usage	Get a quote from an RFC865-compliant quote server.
Start	RFC865Client.GetQuote hostName ~ hostName : domain name or IP address. The quote appears in the Kernel log.
Stop	To clean up: SystemTools.Free RFC865Client ~

Raw printing using port 9100

Usage	Raw printing to a printer supporting what is commonly referred to as any of the

	following: port 9100, port printing, standard IP or raw sockets. www.office.xerox.com/support/dctips/dc00cc0104.pdf
Start	TCPTools.SendFile hostName port fileName ~
Stop	To clean up: SystemTools.Free TCPTools ~

V24 Tracer

Usage	Man in the middle attack for serial ports, with the purpose of analyzing the data stream flowing in and out of an unknown serial device. Using the tracer entails setting up two computers communicating via serial ports: A PC running A2 installed between a PC driving a serial device and the device itself. The PC running A2 must have at least 2 serial ports.
Start	V24Tracer.Enable 0 1 bps ~ Start the tracer. bps : transmission speed in bps, e.g. 9600. Proceed as follows: 1. Connect the device to be traced to port 0. 2. Connect the machine that knows the device to port 1. 3. Start the tracer, guessing the connection settings (speed, etc.). 4. Start using the device from the machine that knows the device. 5. Analyze the trace data in the kernel log. 6. If the traced data looks strange, disable the tracer, re-guess the connection settings and goto 3. V24Tracer.Disable ~ V24Tracer.SetMode [mode] ~ mode : the default value is “fine” which means display the origin of each character, its hex value and printable value. Produces a long trace, better avoid it by specifying “coarse” for example.
Stop	To clean up: SystemTools.Free V24Tracer ~

HTTP server

Usage	
Start	WebHTTPServerTools.Start ["\r:" default root directory] ["\l:" log file] ~ WebHTTPServerTools.AddHost hostName ["\r:" root directory] ["\d:" default file] ["\e:" error file] ~ WebHTTPServerTools.RemoveHost hostName ~ WebHTTPServerTools.ListHosts ~
Stop	WebHTTPServerTools.Stop ~ To clean up: SystemTools.Free WebHTTPServerTools WebHTTPServer TFClasses TCPServices TLS CryptoRSA CryptoSHA1 CryptoMD5 CryptoHMAC CryptoHashes CryptoUtils CryptoBigNumbers Random CryptoCiphers BIT TCP WebHTTP TFLog IP Network WSock32 ~

VNC server

Usage	Implements the Remote Frame Buffer protocol version RFB 3.3 of January 1998

	which was closed in 2002. http://en.wikipedia.org/wiki/RFB_protocol This A2 application does not support file transfer, compression or encryption and not HTTP. Install a Virtual Network Computing server which allows VNC clients to remotely control of the server machine. Server and client use a simple VNC protocol, Remote Frame Buffer (RFB), based on: 1. a graphic primitive from server to client, put a rectangle of pixel data at the specified X,Y position and 2. event messages from client to server. **Security** The server supports a simple challenge-response protocol to verify a password of up to 32 characters (other servers have usually a limit of 8), supplied by the connecting user. The rest of the session is unencrypted and so anything typed into the viewer passes “in the clear” to the server. The server is suitable for use within a local network or secure VPN, but not for general use over untrusted networks, such as the Internet.
Start	WMVNCView.Install name password [port [x [y [width [height]]]]] ~ name: string, optionally enclosed in " ". This name will be observable by the VNC client. password: string, optionally enclosed in " ". Use "" for no password. port: listening port of the server: 5900, 5901, 5902, … Here, the default is “5901”. Note: A VNC client may use another default port number, not readily noticed. It is advisable to find out that number in advance, and to use it. Each observable view is associated with a different port. x, y, width, height: define a view port, area of the virtual desktop observable by VNC clients connected to the listening port. The default values are respectively 0, 0, 1024 and 768. Define a view port observable by VNC clients. A view port is given an internal name in the form of “View#i” where 1 <= i is incremented by 1 after each execution of an Install command. Note: View#0 is the name of the first and standard view port and is described in the section “Screen shot”. A list of all existing view ports is obtained by executing SystemTools.ListPlugins ~ The VNC server is started as soon as the first Install command is launched. Example: WMVNCView.Install "Bluebottle VNC ViewA" "guessit" ~ acknowledged in the Kernel log with: VNC server started. Listening on port : 5901 Position (x, y)): 0, 0 Size (w, h): 1024, 768 Multiple Install commands may be executed allowing to observe different view ports of different sizes and with different access rights. After having installed a view, a potential VNC client user must be handed over this set of data: server address, port number, and password. The port number may have been chosen in advance to fit the default used by the VNC client. Example in concordance with the view port schema in in “Virtual desktop navigation”: WMVNCView.Install “VNC ViewA” “guessA” 0 0 ~ WMVNCView.Install “VNC ViewB” “guessB” -1024 0 ~

	WMVNCView.Install “VNC ViewC” “guessC” 0 -768 ~ WMVNCView.Install “VNC ViewD” “guessD” -1024 -768 ~
Stop	WMVNCView.Uninstall ~ Stop all listening ports and remove the view ports accessible to VNC clients. Do not execute before all VNC clients have closed their window(s). To clean up: SystemTools.Free WMVNCView VNCServer ~

A selection of VNC clients other than the A2 VNC clients

Windows	RealVNC VNC Free Edition 4.1	www.realvnc.com/products/free/4.1/index.html
Windows	TightVNC (*)	www.tightvnc.com
Mac OS X	1) Sharing → Screen sharing 2) Connect to Server vnc://host:port	
Mac OS X	Chicken of the VNC	sourceforge.net/projects/cotvnc

(*) To establish a connection from the client to the server, specify the VNC server to connect to, in one of the following formats:

host
host:display
host::port

host can be either an IP address or a domain name.
display is an offset from the TCP port 5900. 0 will be used if not specified.

VNC client – GUI

Usage	Install a client that watches, controls and interacts with a VNC server. The client observes a view port defined by the server.
Start	VNC.Open serverName [password] port ~ serverName : the server domain name or IP address. Password : supplied by the VNC server administrator. Use “” when no password is required. A wrong or omitted password is acknowledged with: „Error 1“. port : number of the listening port on the VNC server. The port number defines which server view is to be accessed. Open a session with the VNC server. Multiple sessions may exist. Window s are explicitly entitled to discriminate them. **Shortcut**: Main menu → Tools → VNC VNC.Show ~ List the existing connections with number, pixel format (16 or 32 bits) and window size in the KernelLog. Example: VNC connections 0 32-bit 1 32-bit 1280x1024 VNC.Paste connection textString ~ Use for ?
Stop	Close the window(s). To clean up: SystemTools.Free VNC ~

A selection of VNC servers other than the A2 one

Windows	RealVNC	www.realvnc.com/products/free/4.1/index.html

	VNC Free Edition 4.1	In the Options (Properties : Legacy), Import VNC 3.3 Settings and make sure to set Authentication with a password. When a client is connected, an icon is shown in inverted color.
Windows	TightVNC	www.tightvnc.com When a client is connected, an icon is shown in inverted color.
Mac OS X	Sharing → Screen sharing	Client does not work with secure OS X.

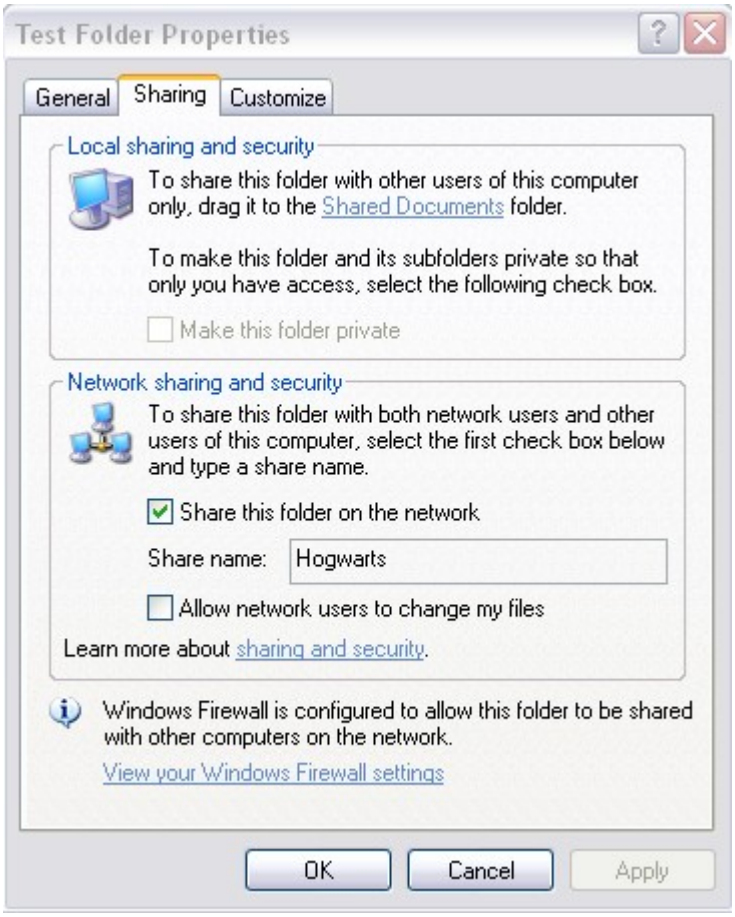
Samba server

Usage	Provide file services in cooperation with Samba clients of various makes. Folders and files of a computer may be accessed by others users on a network.
Start	SambaServer.StartServer ~ Start the server. SambaServer.AddShare folderName sharePath ~ folderName: the name by which a client can access a folder on the server. sharePath: the access path on the server. Example: SambaServer.AddShare client AOS: SambaServer.ListShares ~ List the names of all shared folders, acknowledged with (for example): SambaServer share list: folderName AOS:
Stop	SambaServer.StopServer ~ SystemTools.Free SambaServer ~

Samba client

Usage	Provide file services in cooperation with Samba servers of various makes.
Start	FSTools.Mount prefix “SmbFS” ipAddress folderName [user [password]] ~ Mount the shared file system made available by the Samba server. prefix : the name of the volume mounted by the client. IpAddress : the IP address or the domain of the server. FolderName : the name assigned on the server. User : use “guest” password : use :guest” FSTools.Unmount prefix [\f] ~ Unmount the specified volume. Use \f to force unmounting.
Stop	SystemTools.Free SambaClient ~

- A2 client to A2 server (not tested)
- Windows XP client to A2 server (not tested)
- iMac client to A2 server (Test fails: signals that user name and password are invalid (using “guest guest”)
- A2 client to Windows XP server



On Windows, operate as Administrator to grant access to the target folders. The folders to which access will be granted must be be given the sharing property “Network sharing and Security” Share this folder on the network.

- A2 client to iMac server (test fails: “CONNECTION ERROR!”)

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 Synergy client. The computers must be connected to 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 is a KM solution for sharing equipment, not KVM (keyboard, video, mouse): each machine must have a display attached. A display with dual-input cannot be used to control two machines. For this A2 Synergy client, it is not the case that Synergy merges the clipboards of all the machines into one, allowing cut-and-paste between them, as do commercial products. Security Synergy provides no built-in encryption or authentication. Given that, Synergy should not be used on or over any untrusted network, especially the Internet. It's generally fine for home networks. SSH (secure shell) is not available in A2 for strong encryption and authentication.
Start	SynergyClient.Connect serverName screenName ~ serverName : an IP address or the machine name (Windows or iMac) screenName : name of the client screen assigned in the Synergy server configuration. In case the Synergy server is stopped, the client is also stopped but the connection is not closed. It is required to close the client before resuming operation.
Stop	SynergyClient.Close To clean up: SystemTools.Free SynergyClient ~

BlueTerminal for V24 communication *- GUI*

Warning: it will not last long until this technology completely disappears from modern machines. An alternative solution, say via USB port, is needed

Usage	Have machines A and B, each equipped with a serial port, connected via a “null modem” cable: machine A runs this A2 application, machine B will be running the same or an equivalent terminal application, such as Windows' Hyperterminal.
Start	On machine A, install the serial port driver first by executing V24.Install ~ Verify how many ports are installed and ready by executing Serials.Show ~ Pick the port number of an available COM port, which will be used. Start the terminal application on machine B and on machine A execute: WMV24Component.Open ~ The two machines are now in communication. I**mportant application:** Low-level tracing of the kernel activity of a starting A2 which then fails (the machine reboots for example). That machine can be configured to behave as a terminal application as is described in the “Enable Low-Level Tracing” section of http://www.ocp.inf.ethz.ch/wiki/Documentation/Installation#toc13 . The trace will show up on machine B and analyzing the trace might hint at the problem. Several BlueTerminals may be running in parallel. **Shortcut**: Main menu → Tools → Terminal
Stop	Close the window(s). To clean up: SystemTools.Free WMV24Component ~
Restorable	Yes

Serial shell

Usage	Have machines A and B, each equipped with a serial port, connected via a “null modem” cable: machine A runs this A2 application, machine B will be running the BlueTerminal application or an equivalent terminal application, such as Windows' Hyperterminal. Once both applications are running, machine B can take control of the A2 operating system.
Start	On machine A, install the serial port driver first by executing V24.Install ~ Verify how many ports are installed and ready by executing Serials.Show ~ Pick the port number of an available COM port, which will be used. Start the terminal application on machine B and on machine A execute: ShellSerial.Open [portNbr bps parity stopBits] ~ portNbr : COM port number. Default is DefaultPort = 1 bps : transmission speed. Default is DefaultBPS = 115200 parity : odd, even, mark, space, no. Default is DefaultParity = no stopBits : stop bit length 1, 1.5, 2. Default is DefaultStop = 1 Starts the listening application on the specified COM port. The terminal application is then prompted as is described in the “Shell” section, which describes how a user can take control of the remote A2 operating system.
Stop	To clean up: SystemTools.Free ShellSerial Shell~
Data files	Shell.Alias

Telnet shell

Usage	Have machines A and B connected via TCP/IP: machine A runs this A2 application, machine B will be running a Windows hyper-terminal application or an equivalentWMVTWMVT terminal application. Once both applications are running, machine B can take control of the A2 operating system using a set of predefined shell commands.
Start	Telnet.OpenService ~ Start a telnet server listening on port 23. Pick the domain name or the IP address of the host machine running A2. Windows XP: Define a Hyperterminal connection via TCP/IP to the host and open the connection. Windows Vista: Activate the accessory Telnet and in the search field enter “telnet hostName 23” OS X: Start the Terminal application, then execute “telnet hostName 23” to open the connection. To exit the connection, hold CTRL, press “]” and then enter “quit”. When the connection is established, “>” prompt appears in the window. From now on the host can be controlled with a set of shell commands. The shell commands are defined in ShellCommands.Mod and not in Shell.Mod as is the case for other shells.
Stop	To clean up: SystemTools.Free Telnet ShellCommands ~

VT100 terminal emulator - GUI

Usage	This application is not working. VT100 terminal emulation supporting color and cursor control through a system of

	ESC sequences.
Start	WMVT100.Open hostName [port] ~ hostName : IP address port : The default port number is 23.
Stop	Close the window. To clean up: SystemTools.Free WMVT100 ~

IMAP/SMTP mail client - GUI

Usage	Handle mail.
Start	IMAPGUI.Open ~ **Shortcut**: Main menu → Apps → MailClient
Stop	Close the window. To clean up: SystemTools.Free IMAPGUI ~
Restorable	Yes
Data files	IMAPUtilities.Mod, IMAP.Mod, IMAPClient.Mod, RMSMTP.Mod, IMAPGUI.Mod (should be named WMMail.Mod) / IMAPIcons.tar

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 YMF754	0004 000D 000A 000C 0010 0012	YMF754.Mod YMF754Util.Mod YMF754.Bin	YMF754.Install ~

(*) 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 ~

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 viewer – GUI

Usage	A television viewer for a specific Hauppauge card with a BT848 chip or equivalent. This chip is antiquated and difficult to acquire. A Hauppauge TV driver must be installed first.
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	Yes
Build	Native only: A2, A2Mini, AMD64
Data files	TVChannels.XML

Teletext viewer – GUI

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	Yes
Build	Native only: A2, A2Mini, AMD64
Data files	teletext.bfnt, next.png, prev.png, refresh.png

HTTP server for teletext access through a Web browser

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 ~
Build	Native only: A2, A2Mini, AMD64.
Data files	SystemTools.Free DynamicWebpagePlugin WebHTTPServerTools WebHTTPServer ~

System inspection and performance measurement

CPU inspector

Usage	Inspect the CPU characteristics, optionally in detail, routing the information to the Kernel Log. Applies to Intel CPU >= i486 and to AMD processors.
Start	CPUID.Show [-d | --details] ~ Sample output: Prozessor: Intel(R) Pentium(R) 4 CPU 3.00GHz Vendor: GenuineIntel, Family: 0FH, Model: 04H, Stepping: 03H Logical processor count: 2 Features: MMX: Yes, SSE: Yes, SSE2: Yes, SSE3: Yes, Supplemental SSE3: No Extended 3DNow!: No, 3DNow!: No, AMD MMX Extensions: No 64bit instructions: Yes
Stop	To clean up: SystemTools.Free CPUID ~
Build	Native only: A2, A2Mini, AMD64.

Memory inspector

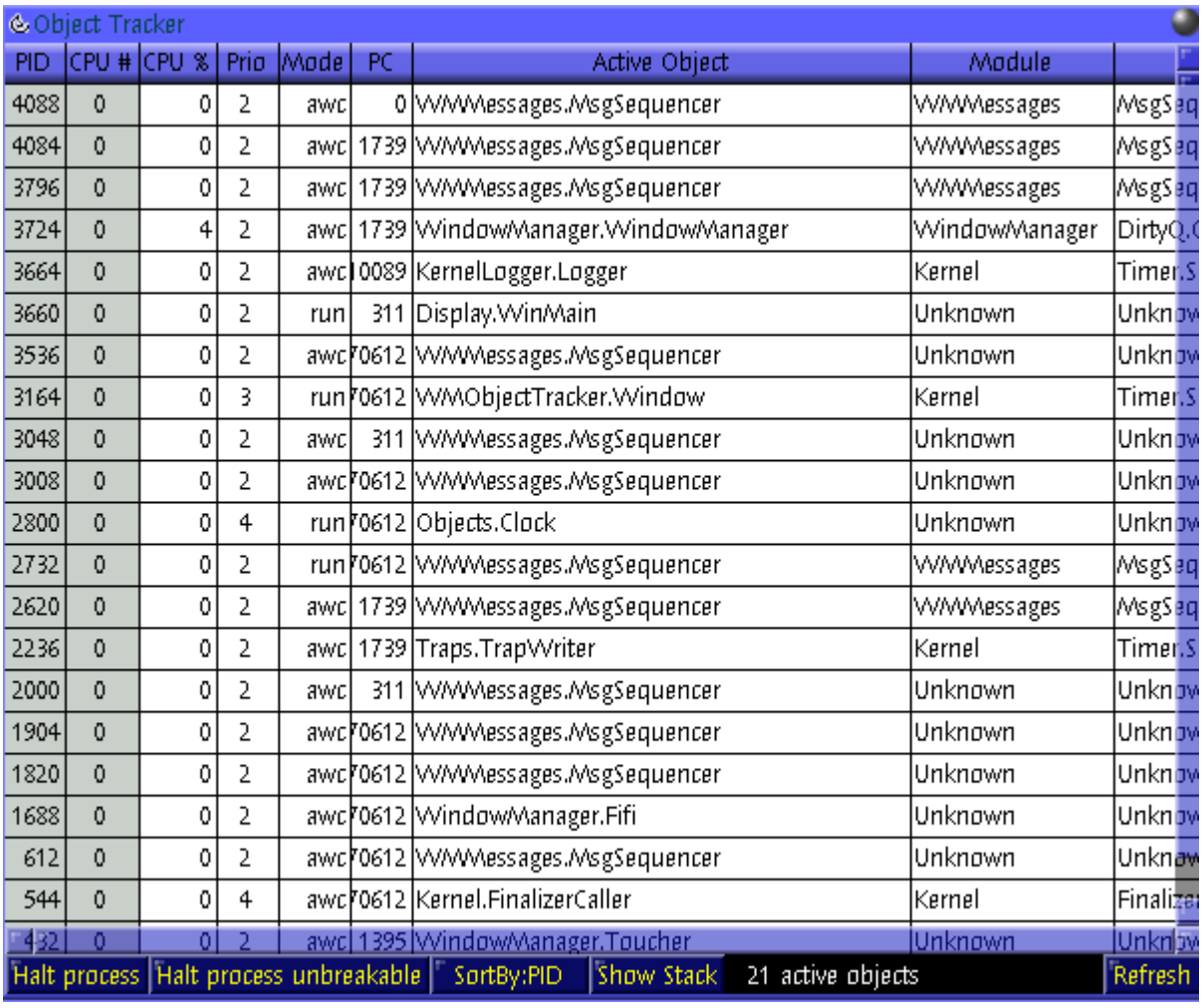
Usage	Inspect the memory mapping, routing the information to the Kernel Log. Applies to Intel CPU >= i486 and to AMD processors.
Start	MemInfo.DisplayMap ~ Display the memory mapping for each processor.

	MemInfo.DisplayMTTR ~
Stop	To clean up: SystemTools.Free MemInfo ~
Build	Native only: A2, A2Mini, AMD64.

TCP tracker - GUI

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 ~ **Shortcut**: Main menu → Inspect → TCPTraffic
Stop	Close the window. WMTCPTracker.Close ~ To clean up: SystemTools.Free WMTCPTracker ~
Restorable	Yes

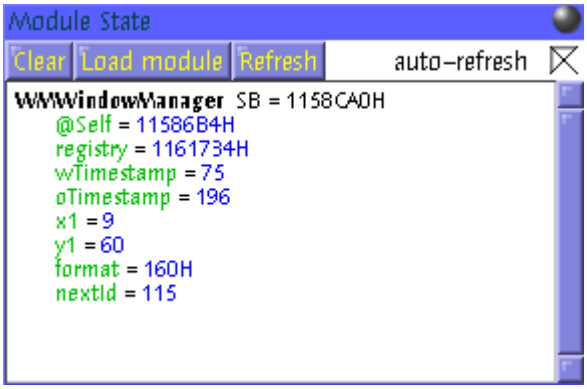
Object tracker – GUI



[WMScreenShot.SnapShotRange text.bmp 600 500 0 0 ~]

Usage	A table listing of the active objects on the stack appears on the desktop.
Start	WMObjectTracker.Open [interval] [interleave] ~ interval = interval between updates in ms interleave = number of updates until refresh When interval < 1 then the default is = 100 When interleave < 1 the the default is = 10 A ML click on an object selects it. To select several contiguous objects, press ML on the first object, then drag on ML downwards or upwards and release ML on the last one. A ML click on the header line de-selects all. **Shortcut**: Main menu → Inspect → Objects
Stop	Close the window. WMObjectTracker.Close ~ To clean up: SystemTools.Free WMObjectTracker ~
Restorable	Yes. When restored, the default interval and interleave values are used.

Module state inspector – GUI

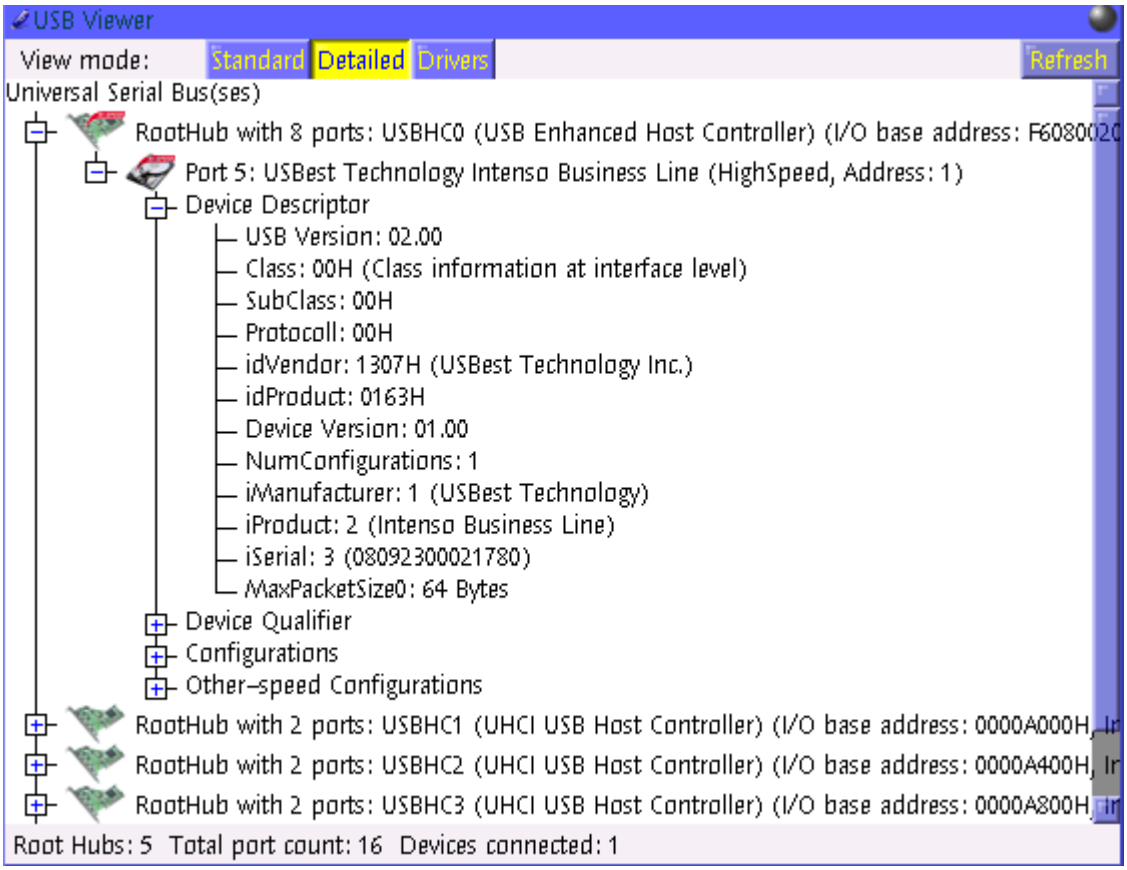
Usage	Inspect the state of a module, that is the current value of the global variables.
Start	WMModuleState.Open [moduleName [interval]] ~ The state of the named module is displayed if it is loaded, else signals “moduleName not loaded”. When moduleName is omitted, a dialog box asks to enter a name. interval = interval between refreshes in ms. Default is 2000.  Four controls are provided: Clear: the window Load module: a dialog box asks for a name (auto-refresh must be off) Refresh: refresh immediately, irrespective of interval auto-refresh checkbox: toggle automatic refresh
Stop	Close the window. To clean up: SystemTools.Free WMModuleState ~

PCITools

Usage	Provide PCI-related services: explore hardware and install drivers. Uses pci.ids, a file containing PCI vendor ID to vendor string mapping, and DriverDatabase.XML a file listing the currently supported devices for which a driver exists.

Start	PCITools.Scan [-d | --details] ~ Perform a bus enumeration and display information about found PCI buses/devices. PCITools.DetectHardware ~ Perform a bus enumeration and install the appropriate device drivers when declared available in the DriverDatabase.XML file. It is very convenient to include PCITools.DetectHardware either: in the Configuration data, or in the Configuration.XML file. In that way, all the drivers that might be needed are activated from the start. Very useful when multimedia is going to be used.
Stop	To clean up: SystemTools.Free PCITools DriverDatabase ~
Build	Native only: A2, A2Mini, AMD64
Data files	pci.ids, DriverDatabase.XML

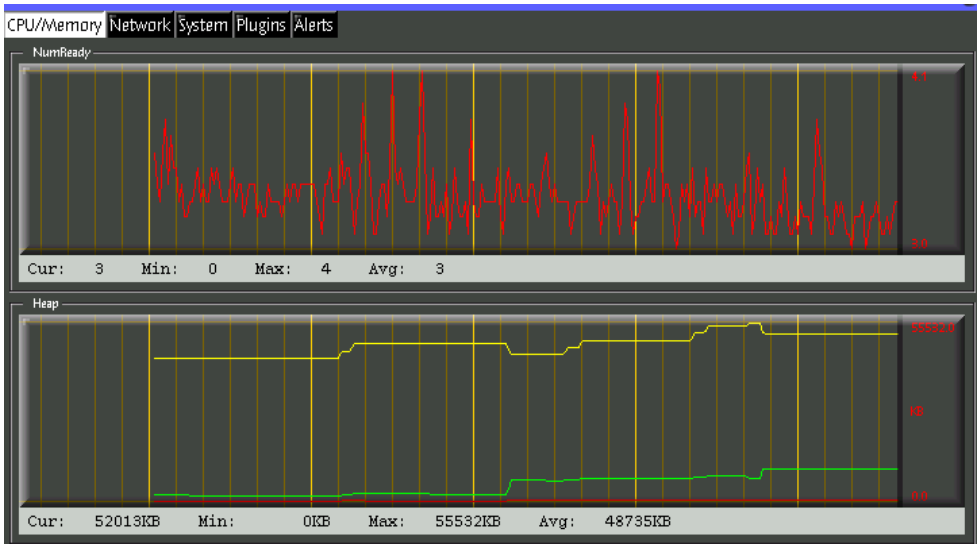
USB hardware inspector *– GUI*



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. **Shortcut**: Main menu → Inspect → USBViewer
Stop	Close the window(s). To clean up: SystemTools.Free WMUsbInfo ~
Restorable	Yes
Build	Native only: A2, A2Mini, AMD64
Data files	WMUsbInfo.tar

System performance monitor – GUI

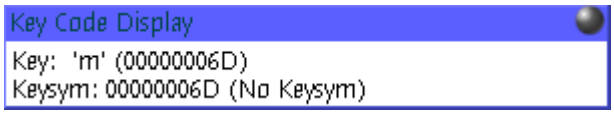
Usage	
Start	WMPerfMon.Open ~  [WMScreenShot.SnapShotRange test.bmp 720 400 0 0 ~] **Shortcut**: Main menu → Inspect → Performance Several inspectors may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMPerfMon WMPerfMonTabSystem WMPerfMonTabAlerts WMPerfMonAlerts WMPerfMonComponents WMPerfMonPluginDisks Disks WMPerfMonPluginNetwork Network WSock32 WMPerfMonPluginEvents WMPerfMonPluginMessages WMPerfMonPluginPerfMon WMPerfMonPluginMemory WMPerfMonPluginCpu WMPerfMonPlugins WMDiagramComponents ~
Restorable	Yes

Event log – GUI

Usage	
Start	WMEventLog.Open ~

	WMEventLog.OpenFile [fileName] ~ **Shortcut**: Main menu → Inspect → Events Several logs may be running in parallel.
Stop	Close the window(s). To clean up: SystemTools.Free WMEventLog EventsMemoryLog EventsUtils Events ~
Restorable	Yes

Keyboard key code inspector - *GUI*

Usage	The key code corresponding to a key stroke (key pressed or key released) is displayed. Key combinations with Ctrl, Alt or Meta-key can also be inspected.
Start	WMKeyCode.Open ~ The Key Code Display panel opens and pressing and holding the “m” key shows this:  [WMScreenShot.SnapShotRange test.bmp 303 52 0 0 ~] WMKeyCode.StartLog ~ Direct the information to the KernelLog to retain the evanescent display information. Pressing “m” writes this to the KernelLog: Key: UCS=0000006D, KeySym=0000006D (No Keysym), Key: 'm', Flags= Key: UCS=00000000, KeySym=00FFFFFF (No Key), Key: '', Flags=[RELEASE] WMKeyCode.StopLog ~ Stop sending information to the KernelLog.
Stop	Close the window. To clean up: SystemTools.Free WMKeyCode ~
Restorable	Yes

Test and benchmark programs

Programs aiming at helping developers. Some of these are questionable and might be dropped in future.

Drag and drop test program - *GUI*

Usage	Of questionable value. Have the Kernel Log open to follow the happening.
Start	TestComponenDragDrop.Open ~ Position the cursor in the blue or green area, press ML, drag the cursor within the container panel and observe the information appearing in the log. A MM or ML click opens a dialog, then close it.
Stop	To clean up: SystemTools.Free TestComponenDragDrop ~

Visual components test bed - *GUI*

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 tool bar (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 tool bar (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 ~

Example programs

These examples demonstrate how simple it is to start programming in *Active Oberon* using the already available infrastructure.

Text writer *- GUI*

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 ~

Drawing pad *- GUI*

Usage	An example drawing pad using the mouse as a pen. Uses WMGraphics and WMDialogs
Start	WMScribble.Open ~ Press and hold down ML, then move the mouse to draw a blue scribble until ML is released. Repeat indefinitely. To erase everything, press MR. Press “s” to store the scribble. A dialog window then appears, asking for a file name. The default “scribble.bmp” is offered. Use PicView.Open fileName ~ to visualize.
Stop	Close the window. To clean up: SystemTools.Free WMScribble ~

Graphic animation *- GUI*

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 ~

Data files	BluebottlePic0.png

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

Pie menu test *- GUI*

Usage	Only the central dark green panel reacts to pressing MR: 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 ~

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

Menu editor *- GUI*

Usage	An editor of a fantasy menu of the kind of XmasMenu.XML used in PresentViewer.
Start	MenuEdit.Open ~ The tool bar 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 MM 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 ~
Data files	present.tar

Mail reader *- GUI*

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

Demonstration

Vectorized/rasterized 3D menu - *GUI*

Usage	A 3D menu opens on the desktop. This application does not work.
Start	W3dMenu.Open fileName ~ fileName: one of 4 XML files: W3dFun.XML, W3dMenu.XML, W3dNetTools.XML, W3dPersonal.XML using images extracted from .tar files. The XML file describes the structure of the 3D menu, in particular the number of menu items represented by boxes. A 3D world opens on the desktop. Press and hold ML and move the mouse to rotate the axises. Additionally, press MR to zoom. **Shortcut**: Main menu → Demos → 3d.Menu , uses data file W3dMenu.XML Fails to work and traps.
Stop	Close the window. To clean up: SystemTools.Free W3dMenu W3dObjectGenerator Random W3dWorld W3dGeometry W3dRasterizer W3dAbstractWorld W3dMatrix W3dVectors TFClasses MathL~
Data files	W3dFun.XML, W3dMenu.XML, W3dPersonal.XML, W3dNetTools.XML, W3dMenuIcons.tar, Cluster.XML (used by W3ClusterWatch.Mod and TestServer.Mod)

3D viewer - *GUI*

Usage	3D object rendering and manipulation. An image is placed in the axises.
Start	W3dExplorer.Open ~ A 3D world opens on the desktop. Press and hold ML and move the mouse to rotate the axises. Additionally, press MR to zoom. Key strokes condition the image aspect: „0“ the image grain varies (“0” is the starting value). „1“ idem „2“ idem „3“ wire frame view of the axises. No image. „s“ speed check – the time it takes to rotate the image by 360 degrees is displayed in the Kernel log in ms. Press “escape” to stop. The rotation stops only after a complete revolution.
Stop	Close the window. To clean up: SystemTools.Free W3dExplorer W3dObjectGenerator Random W3dWorld W3dGeometry W3dRasterizer TFClasses W3dAbstractWorld W3dMatrix W3dVectors MathL ~ Error: fails to free and traps!
Data files	BluebottlePic0.png

Slide show - *GUI*

Usage	A simple slides how presentation tool with transitions effects.
Start	WMSlideshow.Open [fileName] ~

	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" **Shortcut**: Main menu → Demos → SlideShow
Stop	Close the window. To clean up: SystemTools.Free WMSlideshow WMTransitions ~
Data files	RetoSlideshow.XML, SlideShowData.tar

Fractal *- GUI*

Usage	A demonstration of fractals.
Start	The demo requires to execute: FractalDemo.Register ~ followed by ComponentViewer.Open FractalDemo.XML ~ **Shortcut**: Main menu → Demos → Fractal
Stop	Close the window. To clean up: SystemTools.Free ComponentViewer FractalDemo ~
Data files	FractalDemo.XML

Turing *- GUI*

Usage	
Start	TuringCoatWnd.Open ~ TuringCoatWnd.OpenAlpha ~ Several Turing animations may be running in parallel. **Shortcut**: Main menu → Demos → Fractal
Stop	Close the window(s). To clean up: SystemTools.Free TuringCoatWnd ~
Data files	WMIcons.tar

Fractal voxel ray tracer *- GUI*

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 *- GUI*

Can serve as example of WMGraphics use.

Rules	A familiar Tetris game. 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: Cursor 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. **Shortcut**: Main menu → Demos → Tetris
Stop	Close the window(s). To clean up: SystemTools.Free WMTetris ~

Tetris server (extension of VNCServer)

Rules	The Tetris game server which can be accessed by any VNC client connecting to port 5999.
Start	VNCTetrisServer.Run ~ acknowledged in the KernelLog with: „VNC Tetris server started“ Whenever a Tetris game terminates a status report line with the score is reported in the log. VNCTetrisServer.StopNew ~ Stop the server. After restart, the game fails to run.
Stop	VNCTetrisServer.Uninstall ~ To clean up: SystemTools.Free VNCTetrisServer VNCServer ~
Data files	VNCTetris.dat

VNC client of a Tetris server *- GUI*

Rules	The same Tetris game as described previously except that a VNC client is used, the game starts immediately and the window is not transparent.
Start	VNC.Open serverName password 5999 ~ the password is necessary but any is valid. serverName: the server domain name or IP address. Open a session and start the game immediately. Controlling it is as explained earlier. The state of the game appears at the bottom of the window: Score:

	Games active: Max concurrent: Served total: High score: Press p to toggle pause Multiple sessions may exist. Each window is entitled: „serverName Port 5999 – VNC i“ where „i“ is the session number for discriminating multiple sessions. Any other VNC client may also be used but recall to specify port 5999.
Stop	Close the window(s). To clean up: SystemTools.Free VNC ~

Bimso *- GUI*

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 series 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 ~

Color (or Colored) lines *- GUI*

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.

Fun

Animated images *- GUI*

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 imgName [nofFrames step] ~ imgName: WMBunnyImages.tar://image – there are 9 images to chose from.

	nofFrames: the number of image components for an animation. Default is 8. step: . Default value is 32. Several images may be inserted. **Shortcut**: Main menu → Demos → Bones **Shortcut**: Main menu → Demos → Bunny
Stop	Removing an image requires some dexterity: While moving, try a ML+MR inter-click in the approximative image area. With a bit of luck it is deleted. WMBunny.Free ~ is the better way to kill all images.
Data files	WMBunnyImages.tar (9 images).

Fantasy menu - GUI

Usage	A test bed for a fantasy menu representing a heap of presents to be opened. When clicked with MM, 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 ~
Data files	XmasMenu.XML, present.tar

Christmas snow - GUI

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] ~ nofFlakes: . Default value is DefaultNofFlakes = 20
Stop	SystemTools.Free Snow ~
Data files	xmas04.tar

WinAos notes

The web site serving as starting point is www.ocp.inf.ethz.ch/wiki/OCP/WinAos . “Some notes on WinAos” written by Felix Friedrich introduces the reader to the matter:
http://www.ocp.inf.ethz.ch/wiki/OCP/WinAos?action=download&upname=WinAos.pdf

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	onto Aos(.exe) in the WinAos directory

	drag and drop them
Start	CommandLine.Open ~ Parse the Windows command line when WinAos is started and execute the commands. It is useful to add the command to the Autostart section of Configuration.XML.
Stop	To clean up: SystemTools.Free CommandLine ~
Build	WinAos, eWinAos

Packages, Release builds and Ports

The developers' objective is to port A2 to the widest possible range of platforms (target CPUs and operating systems), though some applications developed for the native system might not be portable in case A2 is hosted by a third party operating system. Also, for practical reasons, the entire system is divided into “packages” or collections of applications of a given type. A user can thus choose to deploy all of the packages or only some of them so as not to overload an installation. Packages are defined, that is described, in the Release.Tool file. In the same file, a “Build” section describes the components of a specific build. All builds include the relevant source code. The file License.Text, included in a build, contains the copyright information.

Whenever an original A2 source code module must be adapted for porting it, the ported module name is prefixed so as to assign a unique name. Also a number of additional modules needed by the ports have prefixed source module names.
These tables summarize the various builds which can be produced for a given A2 release:

Native, running on bare hardware

Build	Target CPU	Source module prefix	Object module extension	Excluded packages
A2	i386	---	.Obx	---
A2Mini	i386	---	.Obx	Applications, GuiApplications, Fun, Testing, Education, Contributions, Oberon, OberonGadgets, OberonApplications, OberonDocumentation, OberonVoyager, OberonAnts
AMD64	AMD64	AMD64.	.Abx	Contributions, Oberon, OberonGadgets, OberonApplications, OberonDocumentation, OberonVoyager, OberonAnts
---	ARM	PC*ARM.	.Oba	all

The latest builds of A2 and A2Mini, as bootable ISO CD-images, can be downloaded from bluebottle.ethz.ch/download.html for burning a live CD.
AMD64 is in development.
The ARM Xscale processor support is not up-to-date.

Other builds can be downloaded from www.ocp.inf.ethz.ch/wiki/OCP/Downloads :

Windows is host

Build	Target CPU	Source module prefix	Object module extension	Excluded packages
WinAos	i386	Win32.	.Obw	---
eWinAos				

Virus warning

At the time of this writing a Windows system protected by Avira AntiVir Personal will report the detection of a Trojan horse TR/CryptXPack.Gen in the A2 start up module Aos.exe. One may safely ignore it. One can hope that this warning will disappear in a later build.

Unix is host

Build	Target CPU	Source module prefix	Object module extension	Excluded packages
LinuxAos	i386	Linux. & Unix.	.Obj	OberonVoyager, OberonAnts
DarwinAos		Darwin. & Unix.		
SolarisAos		Solaris. & Unix.		

Qemu is host

A2 / A2Mini	Use a bootable A2 or A2Mini CD. Not tested

VirtualBox is host

A2 / A2Mini	Use a bootable A2 or A2Mini CD. Only A2 can be installed on a virtual disk. Experience shows that the system is very sluggish.

VirtualPC is host

A2 / A2Mini	Use a bootable A2 or A2Mini CD. Not tested.

VMWare is host

A2	Raw image of A2 for use with VMWare Player, VMWare Workstation or, on iMac, WMWare Fusion.

When required, a user can trim the Release.Tool text to customize a build at will.

Developers

A2 is being developed by the Native Systems Research Group headed by Prof. Dr. Jürg Gutknecht in the Department of Computer Science at ETH, the Swiss Federal Technical University in Zürich.

The current developers are, at this date in 2009:

Dr. Felix Friedrich
Ulrike Glavitsch

Thomas Kägi-Trachsel
Florian Negele
Sven Stauber

Index