

# BelA 2.0 Client Platform Specification

**This document is confidential and may not be distributed without permission of Be Incorporated.**

## Revision History

Version #	Date	Comments
2.0Client 1	March 8, 2001	Created
2.0Client 1	March 13, 2001	Updated Overview, Hardware chapters

copyright © 2001 Be Incorporated

# BeIA 2.0 Client Platform Specification

## Table of Contents

<b>Architectural Overview</b>	<b>11</b>
<b>1 Schematic Legend</b>	<b>11</b>
<b>1.1 Internal system service, plug-in, or driver (not related to UI)</b>	<b>12</b>
<b>1.2 Customizable UI component plug-in</b>	<b>12</b>
<b>1.3 Non-customizable UI component plug-in</b>	<b>12</b>
<b>1.4 Shared libraries, providing C/C++ system APIs.</b>	<b>12</b>
<b>1.5 Binder Framework</b>	<b>12</b>
<b>1.6 Team (independent protected memory space).</b>	<b>12</b>
<b>1.7 Core system component</b>	<b>13</b>
<b>1.8 Customizable or optional plug-in, module, or driver</b>	<b>13</b>
<b>1.9 Existing component</b>	<b>13</b>
<b>1.10 Future or custom component</b>	<b>13</b>
<b>2 Kernel Core</b>	<b>13</b>
<b>2.1 CPU Modules</b>	<b>14</b>
<b>2.2 Compression Technology (Uncrusher).</b>	<b>14</b>
<b>3 Kernel Drivers</b>	<b>14</b>
<b>3.1 The Bus Manager</b>	<b>14</b>
<b>3.2 Graphics Drivers</b>	<b>14</b>
<b>3.3 Printer Drivers</b>	<b>15</b>
<b>3.3.1 Imaging Layer</b>	<b>15</b>
<b>3.3.2 Imaging Layer Submodules (Model-specific Plug-ins)</b>	<b>15</b>
<b>3.3.3 Transport Layer</b>	<b>15</b>
<b>3.4 Input Device Drivers</b>	<b>15</b>
<b>3.5 Audio Drivers</b>	<b>16</b>
<b>3.6 Networking Stack</b>	<b>16</b>
<b>3.6.1 Networking Core</b>	<b>16</b>
<b>3.6.2 Networking Protocol</b>	<b>16</b>
<b>3.7 Network drivers</b>	<b>16</b>
<b>3.8 File systems</b>	<b>17</b>
<b>3.9 Storage Drivers</b>	<b>17</b>
<b>4 Graphic and Application Services</b>	<b>17</b>
<b>4.1 Graphic Accelerant</b>	<b>17</b>
<b>4.2 Font Support</b>	<b>17</b>
<b>4.3 LBX: Live Image Compression</b>	<b>18</b>
<b>5 Input Server</b>	<b>18</b>
<b>5.1 Input Devices</b>	<b>19</b>
<b>5.2 Input Filters</b>	<b>19</b>
<b>5.2.1 Custom key mapping</b>	<b>19</b>
<b>5.3 Input Methods</b>	<b>19</b>
<b>6 Media Services</b>	<b>19</b>
<b>6.1 Audio Mixer</b>	<b>19</b>

<b>7</b>	Wagner Core . . . . .	20
<b>7.1</b>	Shared Libraries . . . . .	20
<b>7.1.1</b>	Media plug-ins . . . . .	20
<b>7.2</b>	Resources, Memory, and Cache Manager . . . . .	21
<b>7.3</b>	Cookies Manager . . . . .	21
<b>7.4</b>	Security Manager. . . . .	22
<b>7.5</b>	Network Protocol Handler. . . . .	22
<b>7.6</b>	Content Handler Framework . . . . .	22
<b>8</b>	Wagner Plug-ins. . . . .	23
<b>8.1</b>	Web Browser Engine (Opera) . . . . .	23
<b>8.2</b>	Web Content Handlers . . . . .	23
<b>8.3</b>	Customizable UI Plug-ins. . . . .	23
<b>8.3.1</b>	Email Text Editor . . . . .	23
<b>8.3.2</b>	Customizable List View. . . . .	23
<b>8.3.3</b>	Software Keyboard. . . . .	24
<b>8.3.4</b>	Date & Time . . . . .	24
<b>8.3.5</b>	Future Plug-ins. . . . .	24
<b>8.4</b>	Fully customizable UI skin. . . . .	24
<b>9</b>	Daemon Server . . . . .	24
<b>9.1</b>	Shared Libraries . . . . .	24
<b>9.2</b>	Binder Root . . . . .	24
<b>9.2.1</b>	Driver Parameters. . . . .	25
<b>9.2.2</b>	User Accounts . . . . .	25
<b>9.2.3</b>	Vendor Nodes . . . . .	25
<b>9.2.4</b>	Custom Dynamic Nodes . . . . .	25
<b>9.3</b>	Printer Daemon. . . . .	26
<b>9.4</b>	History and state manager. . . . .	26
<b>9.5</b>	Wagner Watchdog. . . . .	26
<b>9.6</b>	Registrar . . . . .	26
<b>9.7</b>	Simple Custom Daemons. . . . .	26
<b>10</b>	MAP Client Agent. . . . .	27
<b>11</b>	Complex Custom Daemons and Applications . . . . .	27

<b>Hardware Support</b> . . . . .	<b>29</b>
<b>1</b> CPU Architecture. . . . .	29
<b>2</b> Memory Chipsets . . . . .	29
<b>2.1</b> Integrated Chipsets. . . . .	29
<b>2.2</b> Standalone Memory Controllers . . . . .	29
<b>3</b> Busses . . . . .	30
<b>4</b> Graphics Controllers . . . . .	30
<b>4.1</b> Embedded Graphics Controllers . . . . .	30
<b>4.2</b> Independent Graphics Controllers. . . . .	30
<b>4.2.1</b> CRT . . . . .	30
<b>4.3</b> LCD. . . . .	31
<b>5</b> Printers. . . . .	31
<b>5.1</b> Imaging Layer . . . . .	31
<b>5.2</b> Transport Layer . . . . .	31
<b>6</b> Input Devices . . . . .	31
<b>7</b> Audio Devices. . . . .	31
<b>7.1</b> Integrated Audio Controllers. . . . .	31
<b>7.2</b> Non-integrated ISA Audio Controllers . . . . .	32
<b>7.3</b> Non-integrated PCI Audio Controllers . . . . .	32
<b>7.3.1</b> AC97 Codecs. . . . .	32
<b>7.3.2</b> Other Codecs . . . . .	32
<b>8</b> Storage Devices. . . . .	32
<b>8.1</b> IDE . . . . .	32
<b>8.2</b> SCSI Chipsets . . . . .	33
<b>8.3</b> USB. . . . .	33
<b>8.4</b> Flash Storage . . . . .	33
<b>8.4.1</b> CompactFlash . . . . .	33
<b>8.4.2</b> M-Systems DiskOnChip2000. . . . .	33
<b>8.4.3</b> DiskOnModule . . . . .	33
<b>9</b> Networking Chipsets . . . . .	33
<b>9.1</b> External Ethernet Interfaces . . . . .	33
<b>9.2</b> Internal Ethernet Interfaces. . . . .	33
<b>9.3</b> Wireless Home Adapters. . . . .	34
<b>9.4</b> Home Network Adapters . . . . .	34
<b>10</b> Modem Chipsets . . . . .	35
<b>10.1</b> Hardware Modems. . . . .	35
<b>10.2</b> Software Modems. . . . .	35
<b>10.3</b> Host-controlled Modems. . . . .	35

<b>User Interface Specification</b> .....	<b>37</b>
<b>1</b> Screen resolution and layout .....	<b>37</b>
<b>2</b> Control objects. ....	<b>37</b>
<b>2.1</b> Editable Text Field .....	<b>37</b>
<b>2.2</b> Trigger Button. ....	<b>38</b>
<b>2.3</b> Radio Button. ....	<b>38</b>
<b>2.3.1</b> Pop-up Menu. ....	<b>38</b>
<b>2.3.2</b> Scrollbar .....	<b>38</b>
<b>3</b> UI feedback sounds .....	<b>39</b>
<b>3.1</b> Customizing the Sound Effects. ....	<b>39</b>
<b>3.2</b> Sound Data Formats .....	<b>39</b>
<b>3.3</b> Default Sound Effects .....	<b>39</b>
<b>4</b> Toolbar .....	<b>41</b>
<b>4.1</b> URL Display .....	<b>42</b>
<b>4.2</b> URL Editing. ....	<b>43</b>
<b>5</b> Web Content Area .....	<b>43</b>
<b>5.1</b> Page History .....	<b>43</b>
<b>5.2</b> Multiple page display .....	<b>43</b>
<b>6</b> Front page. ....	<b>44</b>
<b>7</b> Favorites panel .....	<b>44</b>
<b>7.1</b> Favorites list .....	<b>45</b>
<b>7.1.1</b> Favorites .....	<b>45</b>
<b>7.1.2</b> Folders .....	<b>45</b>
<b>7.2</b> Buttons. ....	<b>46</b>
<b>7.3</b> Adding Favorites. ....	<b>46</b>
<b>8</b> Settings Panels .....	<b>46</b>
<b>8.1</b> Navigation .....	<b>47</b>
<b>8.2</b> The Panels .....	<b>47</b>
<b>8.2.1</b> General .....	<b>47</b>
<b>8.2.2</b> Date & Time .....	<b>47</b>
<b>8.2.3</b> Connection .....	<b>47</b>
<b>8.2.4</b> Email. ....	<b>49</b>
<b>8.2.5</b> Security .....	<b>49</b>
<b>8.2.6</b> Printing .....	<b>49</b>
<b>8.2.7</b> Languages .....	<b>50</b>
<b>8.2.8</b> Credits. ....	<b>50</b>
<b>9</b> Media Bar .....	<b>51</b>
<b>10</b> Alerts and Dialogs .....	<b>51</b>
<b>11</b> Localization and Multi-language Support. ....	<b>52</b>
<b>12</b> Cursors .....	<b>52</b>
<b>13</b> Email client .....	<b>52</b>
<b>13.1</b> Message Panels. ....	<b>52</b>
<b>13.2</b> Address Book button. ....	<b>52</b>
<b>13.3</b> Check Mail button. ....	<b>52</b>
<b>13.4</b> Mailbox panel. ....	<b>53</b>
<b>13.4.1</b> Mailbox Name and Message Count. ....	<b>53</b>
<b>13.4.2</b> Email list view .....	<b>53</b>

<b>13.4.3</b>	Open Button . . . . .	54
<b>13.4.4</b>	Delete/Undelete Button . . . . .	54
<b>13.5</b>	Email Viewer Panel . . . . .	54
<b>13.5.1</b>	Message Status . . . . .	54
<b>13.5.2</b>	Previous Button. . . . .	54
<b>13.5.3</b>	Next Button. . . . .	54
<b>13.5.4</b>	Delete Button . . . . .	54
<b>13.5.5</b>	Print Button. . . . .	54
<b>13.5.6</b>	Add to Addresses Button . . . . .	54
<b>13.5.7</b>	Close Button . . . . .	55
<b>13.5.8</b>	Reply Button . . . . .	55
<b>13.5.9</b>	Forward Button . . . . .	55
<b>13.5.10</b>	From Field . . . . .	55
<b>13.5.11</b>	Date Field . . . . .	55
<b>13.5.12</b>	Subject Field . . . . .	55
<b>13.5.13</b>	Attachments List . . . . .	55
<b>13.5.14</b>	Email Body Viewer . . . . .	56
<b>13.6</b>	Email Editor Panel . . . . .	56
<b>13.6.1</b>	To Field . . . . .	56
<b>13.6.2</b>	Subject Field . . . . .	56
<b>13.6.3</b>	Send Button . . . . .	56
<b>13.6.4</b>	Save as Draft Button . . . . .	56
<b>13.6.5</b>	Back Button . . . . .	56
<b>13.6.6</b>	Clear Button . . . . .	56
<b>13.6.7</b>	Attach Files Button . . . . .	56
<b>13.6.8</b>	Body Text Editor . . . . .	57
<b>13.7</b>	Address Book Panel . . . . .	57
<b>13.7.1</b>	Address List View . . . . .	57
<b>13.7.2</b>	Use Button . . . . .	58
<b>13.7.3</b>	Edit Info Button. . . . .	58
<b>13.7.4</b>	New Button. . . . .	58
<b>13.7.5</b>	Delete Button . . . . .	58
<b>13.7.6</b>	Close Button . . . . .	58
<b>14</b>	File Browser Panel . . . . .	59
<b>14.1</b>	Storage Device Selection Panel . . . . .	59
<b>14.2</b>	File Browser List View Panel. . . . .	59
<b>14.2.1</b>	File Type Filtering Selector. . . . .	59
<b>14.2.2</b>	Title Bar . . . . .	59
<b>14.2.3</b>	Open Button. . . . .	60
<b>14.2.4</b>	Edit Info Button. . . . .	60
<b>14.2.5</b>	Copy To Button. . . . .	60
<b>14.2.6</b>	Attach Button . . . . .	61
<b>14.2.7</b>	Delete Button . . . . .	61
<b>14.2.8</b>	Close Button . . . . .	61
<b>15</b>	Error Messages . . . . .	61

<b>The Browser</b> .....	<b>63</b>
<b>1</b> Overview .....	63
<b>2</b> Hypertext Markup Language (HTML) 4.0. ....	63
<b>2.1</b> HTML elements .....	63
<b>2.2</b> HTML attributes .....	63
<b>3</b> Cascading Style Sheets (CSS) .....	64
<b>3.1</b> Level 1 (CSS1) .....	64
<b>3.2</b> Level 2 (CSS2) .....	64
<b>4</b> Javascript and ECMAScript .....	64
<b>4.1</b> ECMA-262 Version 2 and Javascript 1.3 .....	64
<b>4.2</b> ECMA-262 Version 3 and Javascript 1.4 .....	64
<b>5</b> DHTML .....	65
<b>5.1</b> DOM Objects .....	65
<b>5.2</b> No layer Extension .....	65
<b>Browser Plug-ins</b> .....	<b>67</b>
<b>1</b> RealPlayerG2 .....	67
<b>1.1</b> Size .....	67
<b>1.2</b> Delivery and Markup Formats .....	67
<b>1.3</b> Codecs .....	67
<b>1.4</b> Plug-ins (and associated MIME types) .....	67
<b>2</b> Macromedia Flash 4.0. ....	68
<b>3</b> PersonalJava 3.0.1 .....	69
<b>3.1</b> Size .....	69
<b>3.2</b> Validation .....	69
<b>3.3</b> Optional components .....	69
<b>4</b> MediaPlayer .....	69
<b>4.1</b> Supported Media Formats, Codecs, and File Formats .....	69
<b>5</b> Beatnik .....	69

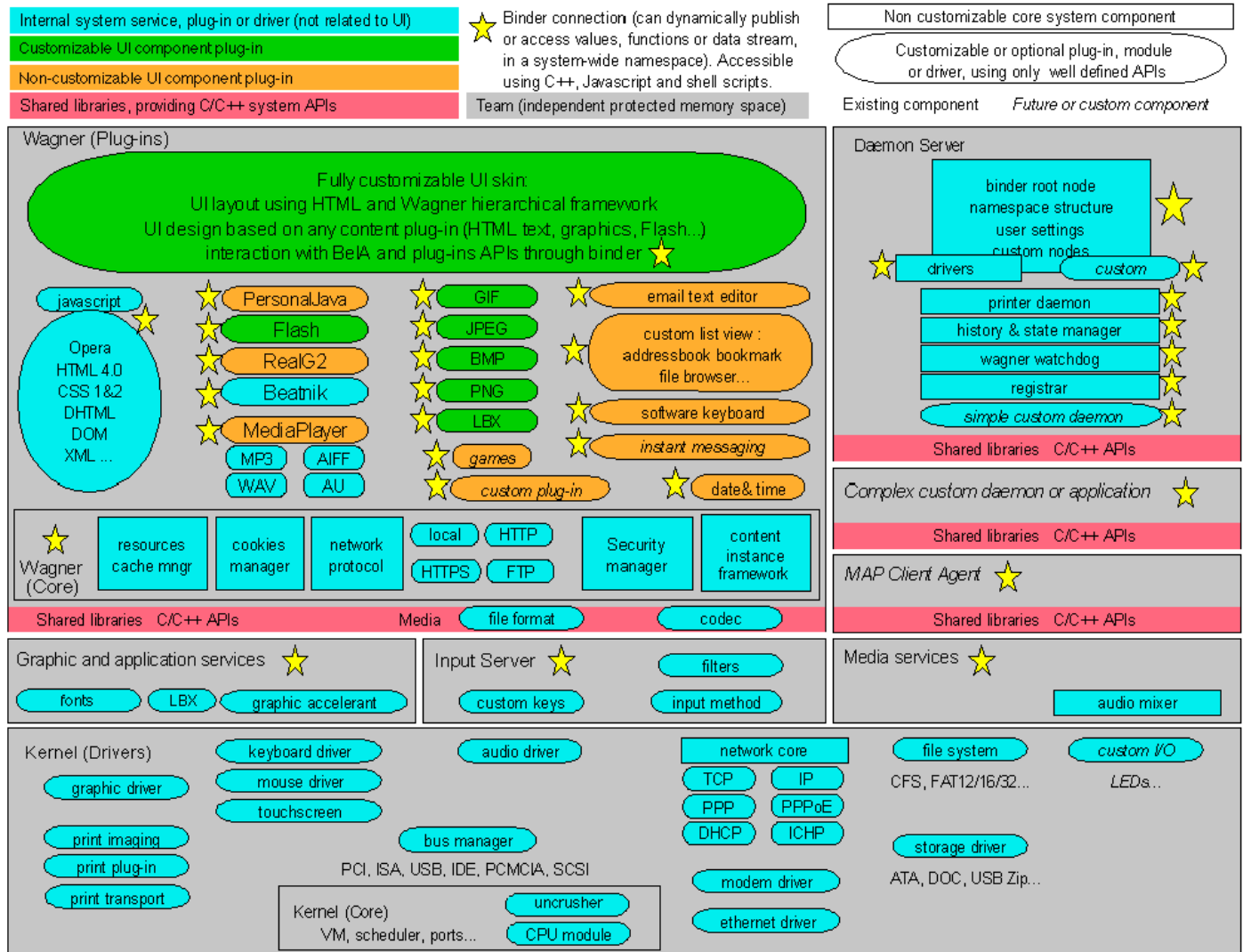


<b>The Development Environment</b> .....	<b>71</b>
<b>1</b> Develop Using BeOS 5 .....	<b>71</b>
<b>1.1</b> Modern C++ Architecture Framework .....	<b>71</b>
<b>1.1.1</b> Posix API Compatibility .....	<b>71</b>
<b>2</b> BeIDE - the Integrated Development Environment .....	<b>71</b>
<b>2.1</b> Easy to Learn and Use .....	<b>71</b>
<b>2.2</b> Easy Application Building .....	<b>72</b>
<b>2.3</b> Syntax Highlighting Editor .....	<b>72</b>
<b>2.4</b> Multi-File Find Capability .....	<b>72</b>
<b>2.5</b> Navigation .....	<b>72</b>
<b>3</b> bdb - the Be Source Code Debugger .....	<b>72</b>
<b>3.1</b> Graphical User Interface .....	<b>73</b>
<b>3.2</b> Program Control .....	<b>73</b>
<b>3.3</b> Breakpoints .....	<b>73</b>
<b>3.4</b> Assembly Code Support .....	<b>73</b>
<b>3.5</b> Remote Debugging .....	<b>73</b>
<b>4</b> gcc Compiler and Linker .....	<b>73</b>
<b>5</b> Shell Commands and Tools .....	<b>74</b>
<b>6</b> Remote Program/Device Monitor (Spy) .....	<b>74</b>
<b>7</b> Profiling .....	<b>74</b>
<b>8</b> Java Development Tools .....	<b>74</b>



# Bela Technical Specification

## Architectural Overview



### 1 Schematic Legend

The subsections below provide explanations of the color-coded elements and groupings in the diagram above. The following major sections look at each of the components in greater detail, with an emphasis on the sort of features or data each component supports, and whether third-parties can modify or augment the component.

### 1.1 Internal system service, plug-in, or driver (not related to UI)

Any component of the system that doesn't influence the user interface. Concomitantly, these components are heedless of changes to the UI.

### 1.2 Customizable UI component plug-in

Any component of the system that has an influence on the UI, and that can be modified without having to recompile C/C++ source code. Such components are typically defined in HTML files, as JavaScript parameters, or in media data files (images or sounds).

### 1.3 Non-customizable UI component plug-in

Any component of the system that has an influence on the UI, but which includes some UI elements that can't be customized without modification and recompilation of C/C++ code. (Note that these components typically include some customizable UI elements as well.)

### 1.4 Shared libraries, providing C/C++ system APIs

BeIA provides a two-level application framework:

- The lower level comprises a set of C and C++ APIs that are implemented as a set of shared libraries. These libraries provide protected access to the resources that are managed by system-wide services (the kernel, graphics system, network server, media server, and so on). The API to the shared libraries are fully documented with plenty of sample code.
- The higher layer comprises the Wagner framework and the Binder APIs. API documentation for Wagner and Binder is generally complete, and is constantly updated.

### 1.5 Binder Framework

The Binder is a namespace tree that lets programs expose data and functionality to the rest of the system. The Binder's most important role is to expose system and user settings to programs, modules, and (local) HTML pages.

Each Binder "node" contains a set of name/value pairs (or "properties"), where the name is unique within the node, and the value can be a string, a number, a function, another node, or even raw data. Functional properties can take arguments and return data of any type. For example, the `printing` node publishes a functional `Print` property that causes the printer to print a page.

In general, Binder means you means an application or server may never have to read an environment variable or open a file to get initialization data. The Binder "tree" can be defined in the factory through an XML definition file, and can be accessed dynamically on deployed devices through well-documented JavaScript or C++ functions.

Although the Binder framework itself can only be modified by Be, new Binder nodes can be written as C++ plug-ins by third parties.

### 1.6 Team (independent protected memory space)

A team comprises the threads and associated data that share a common, protected memory space. The code that a team loads through an add-on becomes part of that team. Memory space is protected in that one team can't write in (or corrupt) another team's memory. Furthermore, if one team dies, all other teams continue to function.

The division of memory space into teams provides efficient protection of the system against complex code that's difficult to bring to a unassailable level of stability. For example, the Web browser team is constantly faced with resource limitations, and has to load third-party code that may not be perfectly stable. Consequently, it's critical that this team be allowed to malfunction without bringing down the entire system. (The Web browser team, in particular, is monitored by the system to detect failure, and to quickly clean-up and relaunch the browser to its last known stable state. As described in the Smoooved section, this transition is as quick and smooth as possible.)

### 1.7 Core system component

Any component that is an integral part of the BeIA architecture and that cannot be removed or replaced by a third party. Some very low-level system components (like the bus manager) have been abstracted into clean, well-defined modules, but the corresponding API has never been documented (as no third party ever had any reason to use it), and so these components are also considered core system components.

### 1.8 Customizable or optional plug-in, module, or driver

Any component that is optional, or that can be easily replaced by an equivalent component developed by a third party using available documented APIs.

### 1.9 Existing component

Any component that is available in the BeIA client platform 2.0 release.

### 1.10 Future or custom component

This is a placeholder for components that will be available in a future BeIA release, or that can be written by a third party.

---

## 2 Kernel Core

**Customization:** The core of the kernel can only be modified by Be.

**Documentation:** The resources provided by the kernel (ports, threads, semaphores, etc) are fully documented in the Kernel Kit, with many examples.

The kernel is the heart of the operating system. It knows how to “talk to” the CPU and memory, and is responsible for coordinating the execution of compiled code. Kernel resources includes memory management (virtual memory, protected memory), the core interrupt handler, ports, the binary loader, and all services related to threads and teams (such as the thread scheduler).

The BeIA kernel supports a number of CPUs, memory controller chipsets, busses, storage devices, and so on. For the full list, see the *Hardware Support* document.

## 2.1 CPU Modules

**Customization:** Currently developed and modified by Be only. Third-party access may be considered.

**Documentation:** N/A.

A CPU module defines performance-enhancing CPU-specific memory attribute (MTRR) optimizations. CPU modules are optional. Modules are available for the supported **Intel**, **AMD** and **National Semiconductor** CPU families listed in "**1** CPU Architecture" in the *Hardware Support* document.

## 2.2 Compression Technology (Uncrusher)

**Customization:** Be only.

**Documentation:** N/A.

BeIA's **Crusher** technology increases the compression ratio on executable files—it tries to squeeze the last bit of fat out of an executable as it's stored on disk (or on Flash). Crusher concentrates on executables (and ignores data files) because that's where most of the storage is.

The BeIA kernel contains an **Uncrusher** component that knows how to inflate crushed executables. The uncrushing process is

---

## 3 Kernel Drivers

**Customization:** New kernel drivers can be created by third-parties through C programming.

**Documentation:** The general principles of driver creation is documented in the Driver Kit.

An important way in which the kernel can be accessed is through device drivers. Every driver, whether written by Be or by a third-party, contains code that's loaded by the kernel.

### 3.1 The Bus Manager

**Customization:** The bus manager framework is developed and modified by Be only. Third-party access may be considered.

**Documentation:** The API for accessing the bus manager, the default bus manager modules, and for creating new modules is documented with examples as part of the Driver Kit.

The bus manager is a Be-defined kernel resource that provides access to the machine's busses. Each bus is represented by a bus manager module that a driver can open in order to list the underlying hardware that's attached to that bus. The bus manager abstracts that device-listing API, making it easy for a driver to find and choose the device it wants to drive.

Bus manager modules are available for a number of busses (ISA, PCI, USB, PCMCIA, etc). For the complete list, go to "**3** Busses" in the *Hardware Support* document.

### 3.2 Graphics Drivers

**Customization:** New graphics drivers can be created by third-parties through C programming.

**Documentation:** In addition to the Driver Kit, graphics driver creation is also described in the Game Kit documentation.

The kernel-loaded part of a graphics driver is responsible for finding the graphics card, mapping IO and memory, setting the display mode (resolution, depth, refresh rate) and options (e.g. back light on/off), power management, and interrupt handling. (As with all drivers, a graphics driver also contains some user-space code.)

BeIA supports both embedded and independent graphics controllers. For the complete list, see "4 Graphics Controllers" in the *Hardware Support* document.

### 3.3 Printer Drivers

The BeIA printing architecture comprises an *imaging layer* and a *transport layer*:

- The imaging layer is responsible for creating printable page data. BeIA provides a C++ framework that does most of the imaging work; individual printer driver plug-ins "inherit" this framework, and complete the imaging work that needs to be done for a particular printer. In some cases, a driver plug-in supports a particular printer manufacturer, and individual models (from that manufacturer) are implemented as separate submodules (that are loaded by the driver plug-in).
- The transport layer implements the "path" (a flavor of port, typically) that page data takes to get to the printer. Transport layer modules are shared by all printer drivers.

#### 3.3.1 Imaging Layer

**Customization:** New imaging layer plug-ins can be created by third-parties through C and C++ programming.

**Documentation:** Imaging layer driver creation is fully documented in the Print Kit.

A list of supported imaging layer drivers is given in "5.1 Imaging Layer" in the *Hardware Support* document.

#### 3.3.2 Imaging Layer Submodules (Model-specific Plug-ins)

**Customization:** Usually Be only. Third-party development requires negotiation with the printer manufacturer

**Documentation:** Be doesn't provide documentation for printer-specific private API.

A list of supported model-specific submodules is given in the "5.1 Imaging Layer" list.

#### 3.3.3 Transport Layer

**Customization:** New transport layer plug-ins can be created by third-parties through C and C++ programming.

**Documentation:** Transport layer driver creation is fully documented with examples in the Print Kit.

A list of supported transport layer modules is given in "5.2 Transport Layer" in the *Hardware Support* document.

### 3.4 Input Device Drivers

**Customization:** New input device plug-ins can be created by third-parties through C programming.

**Documentation:** Documented as part of the Driver Kit.

An input device driver plug-in handles input events that are generated by a specific brand or model of hardware (mouse, keyboard, touch screen, etc.). A single plug-in can handle more than one device driver. A list of supported input device drivers is given in "6 Input Devices" in the *Hardware Support* document.

### 3.5 Audio Drivers

**Customization:** New audio drivers can be created by third-parties through C programming.

**Documentation:** Documented as part of the Driver Kit.

Audio drivers bring sound into and send out of an audio device. A list of supported audio device drivers is given in "7 Audio Devices" in the *Hardware Support* document.

### 3.6 Networking Stack

#### 3.6.1 Networking Core

**Customization:** The core networking framework (BONE) is created and modified only by Be.

**Documentation:** The interface to BONE is fully documented in the Network Kit.

BONE, the BeOS Network Environment, is the backbone of BeIA's high-speed networking interface. BONE features include:

- **Internet standard compliance.** This makes it easy to import a variety of servers to the BeIA platform—including Apache, the most commonly used server.
- **Server security.** Bone rejects malformed IP packets and has better error checking than most networking infrastructures.
- **Performance.** BONE runs at 99% of the capacity of its server wire; its speed is competitive with Linux and FreeBSD. As IA hardware scales up in speed, BONE can keep up without code modification.
- **Modular architecture.** Programmers can easily add new protocols and modules to the BONE framework.
- **Multiple network interface support.** For example, BONE can support dialup and Ethernet connections at the same time.

#### 3.6.2 Networking Protocol

**Customization:** New networking protocols can be developed by third-parties through C programming.

**Documentation:** Protocol creation documentation is being developed.

In theory, any protocol layered on top of IP can be supported by BONE. Specifically, BONE supports TCP and UDP. AppleTalk and IPX protocols could be developed upon request.

In compliance with BSD, sockets are represented as file descriptors in user space. Any user-level program could, therefore, implement a "user level" protocol by capturing and interpreting incoming network data.

### 3.7 Network drivers

**Customization:** New network drivers, both modem and ethernet, can be created by third-parties through C programming.

**Documentation:** Documented as part of the Driver Kit, and supported by documentation in the Network Kit.

BeIA supports a number of ethernet and modem drivers. Third parties can develop drivers for both modem and ethernet devices.

For a list of internal and external ethernet chipsets, wireless home adaptors, and home network adaptors, see "9 Networking Chipsets" in the *Hardware Support* document.



For lists of supported hardware, software, and host-controlled modems, see "**10** Modem Chipsets" in the *Hardware Support* document.

### 3.8 File systems

**Customization:** New file systems can only be written by Be, or with Be's help.

**Documentation:** The API for the "file system independent layer" (FSIL) is well-defined and stable, but is currently not public.

File system access is translated to a common file system independent layer (FSIL) through plug-ins that are designed to read and write specific storage formats. FSIL plug-ins for the following file systems are supported:

- **CFS 2.0:** CFS is a Be proprietary compressed file system designed to provide optimal performance on limited-size Flash based devices.
- **BFS 1.0:** BFS is a Be proprietary file system designed to provide optimal performance on traditional hard disk storage, and is particularly fit for intense media streaming.
- **FAT 12 / 16 / 32:** Used by DOS, Windows 3.x, Windows 95/98.
- **CDDA:** Used by most audio CDs.
- **iso9660:** Used by most digital CDs.
- **NTFS** (read-only): Used by Windows NT and Windows 2000.

### 3.9 Storage Drivers

**Customization:** New storage device drivers can be created by third-parties through C programming.

**Documentation:** The general principles of driver creation is documented in the Driver Kit.

BeIA supports drivers for Flash, IDE, SCSI, and USB storage devices. For details, see "**8** Storage Devices" in the *Hardware Support* document.

---

## 4 Graphic and Application Services

### 4.1 Graphic Accelerant

**Customization:** Third-parties can create their own graphic accelerants through C programming.

**Documentation:** Fully documented as part of the Game Kit.

A graphic accelerant is a user-space module that can be used to implement "accelerated" functions and configurations that are recognized by a graphics driver. Typically, each accelerant corresponds to a specific graphics driver.

### 4.2 Font Support

**Customization:** Be only.

**Documentation:** N/A

BeIA uses the **Bitstream** TrueDoc font engine, using Scalable True Type fonts. The standard font package from **Bitstream** includes:

- **Monospace:** Courier10 Roman, Bold, Italic, Bold italic
- **Sans serif:** Swiss721 Roman, Bold, Italic, Bold italic, Extra compressed

- **Serif:** Dutch801 Roman, Bold, Italic, Bold italic

BeIA provides native international support through **Unicode** using **UTF8** encoding. UTF8 is a multi-length encoding that's equivalent to Unicode, and fully compatible with standard ASCII 7-bit encoding. This allows transparent support of international languages without requiring special tools or API to manipulate the strings.

### 4.3 LBX: Live Image Compression

**Customization:** Be only.

**Documentation:** N/A

BeIA provides a proprietary image compression technology, LBX. This technology greatly reduces bitmap storage size, memory footprint, and load time, without imposing any visible picture degradation or drawing performance. Some of the features of LBX are:

- LBX can compress multiple images used in the same context into one “image collection” file; the more similar the pictures are, the greater the compression ratio
- LBX uses an adaptative colormap reduction, with up to 1024 different colors for one picture.
- It's easy to use: The adaptative colormap reduction is controlled by a single parameter, and the matching between similar pictures is fully automated.
- Image collections are kept compressed in memory at all times; individual images can be directly plucked from the collection using any clipping, zoom or drawing mode on the fly—there's no messy temporary buffering.
- Alpha channel (transparency) data is analysed at compression time for fast redraw. In some cases, an alpha image will draw much faster after it's been LBX-compressed.
- The file system knows how to decompress an LBX image, thus keeping load time to a minimum (since no additional format conversion is necessary).
- Load time is further reduced by the smaller size of the compressed data; images typically compress to 20 to 30 times smaller than their uncompressed size, and 2 to 3 times smaller than the industry-standard PNG format.

---

## 5 Input Server

The Input Server is a system service that accepts user events (generated by the mouse and keyboard, for example), runs the events through a series of *input methods* and *input filters* that can reject or modify them, and then dispatches the surviving events to the rest of the system.

The generation and filtering of events is performed by the plug-ins that the Input Server loads; the Server itself provides the plumbing.

- Event-generating modules (*input device* plug-ins), which typically correspond to one or more device drivers, translate user actions into input server events.
- Input method plug-ins convert keyboard events into character sets that can't be easily represented on a standard keyboard, such as Kanji.
- Input filter plug-ins provide further modification or rejection of the events that are fed to it.

## 5.1 Input Devices

**Customization:** Plug-ins can be developed by third parties through C++ programming.

**Documentation:** Fully documented in the Input Server documentation.

The list of supported input drivers is given in "6 Input Devices" in the *Hardware Support* document.

## 5.2 Input Filters

**Customization:** Plug-ins can be developed by third parties through C++ programming.

**Documentation:** Fully documented with source code examples in the Input Server documentation.

Input filter plug-ins are by far the easiest Input Server modules to develop. BeIA provides a filter that captures and maps events generated by special hardware keys and buttons, as described below.

### 5.2.1 Custom key mapping

**Customization:** Easy customization by modifying a textual key-to-script mapping file.

**Documentation:** Fully documented in the BeIA UI Customization guide.

Every hardware key and button on a device generates a unique key code. BeIA provides an input filter that can map key codes to pre-defined scripts and system commands. The input filter is configured through a text file that the vendor can easily modify and augment. The file can include entries that, for example, map a "volume up" button to the (BeIA-defined) volumeUp command, a "sleep" button to the sleep command, and so on. A key (or button) can also be mapped to a Web site, or to launch a vendor defined script.

## 5.3 Input Methods

**Customization:** Plug-ins can be developed by third parties through C++ programming.

**Documentation:** Fully documented in the Input Server documentation.

BeIA can provide plug-ins for:

- **Soft keyboard**
- **Kanji**
- **Simplified Chinese**
- **Traditional Chinese**

---

## 6 Media Services

**Customization:** Be only.

**Documentation:** N/A.

Currently, BeIA's stable of home-grown media services contains a single service: an audio mixer that can handle playback of a number of audio data formats. The mixer is used in MediaPlayer, BeIA's media-playback Web browser plug-in.

### 6.1 Audio Mixer

**Customization:** Be only.

**Documentation:** N/A.

The audio data formats that the audio mixer can handle are listed in "7.1.1 Media plug-ins", below.

## 7 Wagner Core

Wagner is the BeIA "master application." Wagner is responsible for presenting the BeIA UI and for loading and controlling the Web browser. This section describes the services that Wagner provides. Third-parties can augment and modify particular parts of Wagner by implementing Wagner plug-ins, as described in the next section, "**8** Wagner Plug-ins".

### 7.1 Shared Libraries

**Customization:** Be only.

**Documentation:** N/A.

Like all BeIA applications, Wagner links against the shared libraries that are part of BeIA. The following sections describe some of the services and functionality that are available through the shared libraries.

*More thorough descriptions of the shared libraries are being developed.*

#### 7.1.1 Media plug-ins

##### 7.1.1.1 Audio Codecs

**Customization:** Audio codec plug-ins can be developed by third-parties through C and C++ programming.

**Documentation:** Documented; source code example available upon request.

The following codecs are usually included in BeIA. Note that these codecs aren't necessarily supported in all file formats (see the File Format Handlers section below for details):

- CCITT Audio Compression
- IMA-4 Audio Compression
- MPEG I Audio, Layers 1, 2, 2.5, 3 Xing (MMX optimized)
- MS-ADPCM Audio Compression
- Raw audio
- uLaw Audio Compression

The following codecs are available upon demand:

- DV Audio
- MPEG I Audio, Layers 1, 2, 2.5, 3 Fraunhofer

##### 7.1.1.2 Video Codecs

**Customization:** Video codec plug-ins can be developed by third-parties through C and C++ programming.

**Documentation:** Documented; source code example available upon request.

The following codecs are usually included in BeIA:

- MPEG1 video (MMX optimized)

The following codecs are available upon demand:

- AppleVideo Compression
- Cinepak Compression
- DV Compression by Canopus Inc.
- Indeo-5 Compression

- MS RLE Compression
- MS-Video Compression
- Photo-JPEG Compression
- Raw RGB Codec
- Raw YUV Codec

#### 7.1.1.3 File Format Handlers

**Customization:** File format handler plug-ins can be developed by third-parties through C and C++ programming.

**Documentation:** Documented; source code example available upon request.

The following file format handlers, listed with the codecs they support, are usually included with BeIA:

- **AIFF:** raw audio, Ima-4 audio
- **AU:** CCITT audio, u-law audio
- **MPEG1:** MPEG1 audio layer 1, 2, 2.5, 3 (MMX optimized) Xing; MPEG1 video (MMX optimized)
- **WAV:** raw audio, MS-adpcm, Ima-4 audio

The following format handlers are available on demand:

- **AVI (Video):** AppleVideo Compression, Cinepak Compression, DV Compression by Canopus Inc., Indeo-5 Compression, MS RLE Compression, MS-Video Compression, Photo-JPEG Compression, Raw RGB Codec, Raw YUV Codec
- **AVI (Audio):** MS-ADPCM Audio Compression, Raw audio
- **AVR:** Raw Audio
- **DV:** DV Compression by Canopus Inc., DV Audio
- **MPEG1:** MPEG I Audio, Layers 1, 2, 2.5, 3 Fraunhofer
- **QuickTime (video):** AppleVideo Compression, Cinepak Compression, DV Compression by Canopus Inc., Indeo-5 Compression, Photo-JPEG Compression, Raw RGB Codec, Raw YUV Codec
- **QuickTime (audio):** IMA-4 Audio Compression, Raw audio

### 7.2 Resources, Memory, and Cache Manager

**Customization:** The manager itself is Be only. Certain parameters can be modified through the Binder.

**Documentation:** N/A.

Wagner aggressively caches network content. Page-directed cache control is supported, and the cache size and parameters are easily configurable through the Binder.

### 7.3 Cookies Manager

**Customization:** The manager itself is Be only. Certain parameters can be modified through the Binder.

**Documentation:** N/A.

Downloaded cookies are managed and stored by Wagner.

## 7.4 Security Manager

**Customization:** The manager itself is Be only. Certain parameters can be modified through the Binder.

**Documentation:** N/A.

The Security Manager is responsible for approving programmatic requests to load local resources. For example, the Security Manager can disallow access to local content from a downloaded Web page. The Security Manager is configured in the Binder's XML definition file. The default configuration is scrupulously conservative. In the interest of security, extreme care must be taken when modifying the default configuration.

## 7.5 Network Protocol Handler

**Customization:** The framework is Be only. The protocol handlers themselves are plug-ins, and can be developed by third parties through C++ programming.

**Documentation:** Documentation of the network protocol plug-in API is under development.

The following network protocol handlers are included in BeIA 2.0.

- **file:** Responsible for loading content from local files. Contains special code that maps requests for images contained in LBX archives to the corresponding archive.
- **http:** Responsible for loading content from remote files. Some features of the http implementation are:
  - Persistent connections
  - Basic password authentication
  - **http** proxies
- **https:** This is the same as **http**, but with **SSL 2/3** support.
  - **Cryptography:** *Details pending.*
- **ftp:** Displays directory listings and downloads files using the FTP protocol as defined in the appropriate IETF RFC documents (RFC 959, et. al.). Support for the FTP RESTART command is not yet included, but will likely be included in a future version.
- **beos:** Special protocol used to launch Wagner plug-ins.

## 7.6 Content Handler Framework

**Customization:** The framework is Be only. The content handlers themselves are plug-ins, and can be developed by third parties through C++ programming.

**Documentation:** Documentation of the content handler plug-in API is under development.

Content handlers are responsible for displaying specific types of data, based on the data's MIME type. The handlers are implemented as plug-ins that are loaded by the browser as needed. Handlers for other data (particularly other graphic data) can be developed by third parties though fairly complex C++ programming.

See "8.2 Web Content Handlers" for a list of content handlers provided by BeIA.

---

## 8 Wagner Plug-ins

### 8.1 Web Browser Engine (Opera)

**Customization:** Be and Opera Software only.

**Documentation:** N/A

For a complete technical description of the browser engine (Opera 5.0), see *The Browser* document.

### 8.2 Web Content Handlers

**Customization:** The content handlers provided by BeIA can only be modified by Be or the specific partners listed below. Vendors can choose to forego the partner-provided handlers (because of size constraints, presumably). Additional handlers can be created by third parties through C++ programming (see "7.6 Content Handler Framework", above).

**Documentation:** N/A

BeIA provides a number of Web content handlers, implemented as plug-ins. The plug-ins are listed below, and described in greater detail in the *Browser Plug-ins* document:

- PersonalJava (Sun Inc.)
- Flash 4.0 (Sun Inc.)
- RealG2 (Real Networks Inc.)
- Beatnik RMF (Beatnik Inc.)
- MediaPlayer (Be Inc.)
- Graphics: BMP, GIF, JPEG, PNG, LBX (Be's proprietary image-compression format)
- Hypertext: HTML

For more detail on these plug-ins, see the *Browser Plug-ins* document.

### 8.3 Customizable UI Plug-ins

#### 8.3.1 Email Text Editor

**Customization:** A third-party can partially customize the editor through the Binder.

**Documentation:** Documented as part of the Binder namespace.

The email text editor supports text coloring and sizing of any selected text. Custom UI for text coloring and sizing can be developed by using the appropriate Javascript/Binder functions.

#### 8.3.2 Customizable List View

**Customization:** A third-party can partially customize the list view through the Binder.

**Documentation:** Documented as part of the Binder namespace.

BeIA provides a customizable "list view" object that's used to displays lists of data. Within BeIA, the list view is used by:

- Address book
- Bookmark editor
- File browser

### 8.3.3 Software Keyboard

**Customization:** Typically Be only; third-party access may be considered.

**Documentation:** N/A

BeIA may provide a software keyboard upon request.

### 8.3.4 Date & Time

**Customization:** A third-party can set the date & time display format through simple JavaScript programming.

**Documentation:** Documented in the UI Configuration guide

The browser toolbar provides a date & time display that can be configured, through simple JavaScript programming, to display only the date, or only the time, with or without seconds, and so on. The Binder also provides settable date, time, and timezone properties.

### 8.3.5 Future Plug-ins

Plug-ins developed by Be or third-parties such as games, instant messaging, vendor-specific custom plug-ins, and so on.

## 8.4 Fully customizable UI skin

**Customization:** Easily customizable using (primarily) HTML and JavaScript, include the JavaScript Layout Library

**Documentation:** Documented with example in the UI Configuration guide

The default BeIA UI is fully customizable through simple modification (or replacement) of HTML/JavaScript and image files. For complete details of the default UI, see the *User Interface Specification* document.

---

## 9 Daemon Server

**Customization:** The daemon server itself is Be only. New simple daemons can be added through C/C++ programming.

**Documentation:** N/A

The daemon server is a low-level service that launches daemons.

### 9.1 Shared Libraries

See comments in "7.1 Shared Libraries".

### 9.2 Binder Root

**Customization:** Easily customizable through modification of an XML file.

**Documentation:** The Binder namespace hierarchy is fully documented.

The Binder namespace is defined in a vendor-editable XML file that's read at startup. New nodes can be added dynamically by Binder C++ add-ons or through JavaScript code. The Binder can be examined and monitored through C++ and JavaScript code. Access to the Binder is also brought out through a command line program that can be used during the development cycle.



The “root” of the Binder tree contains three nodes: **service**, **user**, and **application**.

- The **service** node brings out most of the important system information pertaining to the device, information such as the device ID, ISP number, security configuration, system beep volume levels, and so on.
- The **user** node is used to store user account information as described further in the next section.
- The **application** node is used by applications and plug-ins to store application-specific data. The **application** node is meant to be used by third party plug-ins: Currently, Be doesn't store any data in the **application** node.

### 9.2.1 Driver Parameters

**Customization:** A driver developer can publish the driver's parameters by creating a new Binder node.

**Documentation:** Binder node creation through C++ is fully documented with examples.

One of the important features of the Binder is that it gives a place for drivers to publish their parameters. For example, the network driver (BONE) uses the Binder to publish connection status information that can be displayed in the user interface.

### 9.2.2 User Accounts

**Customization:** New user accounts can be added dynamically through JavaScript programming.

**Documentation:** The user account branch of Binder is fully documented.

Binder's **user** subroot contains a node for every user account, each of which contains user-specific settings and data—language, locale, bookmarks, email info, and so on. New user account nodes can be created dynamically through simple JavaScript code. The template that's used to populate a new user account node is defined in a text file that can be modified by the vendor during development.

User account data is stored persistently in a special **user** directory.

### 9.2.3 Vendor Nodes

**Customization:** The pre-defined vendor nodes are meant to be customized by the vendor.

**Documentation:** The vendor nodes are fully documented.

As a convenience, Binder defines two prominent nodes that are reserved for the vendor. One of the nodes in the **service** node, the other is copied into every user account node. The vendor can define the contents of these nodes through XML definition during development.

### 9.2.4 Custom Dynamic Nodes

**Customization:** New Binder nodes can be created dynamically through JavaScript or C++ programming.

**Documentation:** Binder node creation is fully documented with examples..

While the outline of the Binder tree is defined in the factory by the vendor, it can be changed dynamically through C++ and JavaScript programming. Nodes and properties can be modified, added, and removed as the device is running. This means, for example, that new plug-ins that are downloaded to a device during an update can create their own branches of the Binder and populate them with initialization data without having to bring down the system.

### 9.3 Printer Daemon

**Customization:** Be only.

**Documentation:** Indirect access to the printer daemon through the Binder is fully documented.

The printer daemon updates the Binder as printers are added and removed from the device's USB ports. The Binder representation of a printer provides lists of the printer's capabilities (resolutions, paper types, etc.) that can be presented for selection by the user in the UI. The Binder also publishes functional properties that tell the printer to start and stop printing.

### 9.4 History and state manager

**Customization:** Be only.

**Documentation:** N/A

The browser's history is currently stored in Wagner, and backed up in a special driver so that the history of the current browsing session can be recovered after a Wagner relaunch.

The Binder uses a non-persistent node to store the state of the UI. For example, the node keeps track of whether the UI is displaying the browser or a settings panel; if Wagner needs to be relaunched, the node can be queried to restore the display state. The state node is cleared when the device is shut down.

### 9.5 Wagner Watchdog

**Customization:** Be only.

**Documentation:** N/A

The Wagner Watchdog is a special thread that monitors Wagner. If Wagner crashes, the watchdog automatically restarts it (and restores its display state, as described above in History and State Manager).

### 9.6 Registrar

**Customization:** Be only.

**Documentation:** N/A

The registrar is a system service that keeps track of all running applications. Its role as an inter-application communication facilitator has mostly been usurped by the Binder.

### 9.7 Simple Custom Daemons

**Customization:** Third-parties are invited to create and add their own daemons.

**Documentation:** The documentation for adding a new daemon is being developed

The daemon server is a convenient framework in which to install a simple custom daemon, implemented as a daemon server plug-in. More complex daemons—where complexity is measured in the resources and processing attention the daemon demands—can be launched as stand-alone applications, as described in "**11** Complex Custom Daemons and Applications".

---

## 10 MAP Client Agent

**Customization:** Be only.

**Documentation:** The Map Client Agent documentation is under development.

The MAP Client Agent (MCA) is a client-side application that handles communication with the MAP backend server. The MCA itself is implemented by Be; access to the MCA is provided to the vendor through simple JavaScript functions.

---

## 11 Complex Custom Daemons and Applications

**Customization:** Vendor or third-party.

**Documentation:** The BeOS Book.

BeIA provides an entire development environment for developing a custom daemon or application that can be launched as its own team. The environment is supported by complete documentation of the APIs and conventions brought out by the Be shared libraries.

For more information on the development environment, see *The Development Environment* document.



# BeIA Technical Specification

## Hardware Support

This document lists the CPUs, chipsets, storage devices, bus types, printers, and input devices that are supported by BeIA.

**Important:** Because of size and support considerations, the exact components that are included in a specific build is determined on a case-by-case basis. The components listed here are supported by BeIA, but only those components that are specifically negotiated to be present are actually installed in a build for a specific hardware platform.

**Important:** The status of individual components with regard to availability, testing, and reliability will be published in a later version of this document.

---

## 1 CPU Architecture

BeIA runs on Intel x86-based CPUs from a variety of vendors.

- **Intel:** Celeron, Pentium MMX / Pro / II / II Xeon / III / III Xeon / III Coppermine / IV
- **AMD:** K6, K6-2, K6-III, Athlon
- **National Semiconductor:** Geode family (SCx200)
- **Rise:** mP6 x86
- **Transmeta:** Crusoe TM 3120

---

## 2 Memory Chipsets

### 2.1 Integrated Chipsets

- **National Semiconductor:** MediaGX family, and compatible
- **Via:** MVP4
- **Intel:** i810
- **SIS:** 5597, 5598, 620, and 6326
- **Geode:** GX (CPU plus integrated chipset)

### 2.2 Standalone Memory Controllers

- **Intel:** 430 HX/VX/FX/TX, 440 FX/LX/BX, 450 KX/GX/NX, and compatibles
- **Via:** MVP3
- **ALi:** Aladdin 5

---

### 3 Busses

BeIA supports bus manager modules for the busses listed below.

- **ISA**
- **PCI**
- **PCMCIA:** Almost all PCMCIA adaptors are supported (list available upon request).
- **USB:** UHCI, OHCI. Isochronous transfers are not yet supported.
- **IDE**
  - **IDE**
  - **EIDE**
  - **ATAPI**
  - **UltraDMA 33 / 66**
  - **Zip, SyQuest, Jaz,** and other removable media.
- **SCSI**
  - **Adaptec:** AIC-78xx
  - **Symbios:** 53C8xx
  - **BusLogic:** BT948, BT958, BT958D
- **CardBus:** An experimental module for cards based on the Dec 21145 is available. This includes:
  - **Farallon** FastEther TX-10/100
  - **TDK** Network Flyer 100

---

### 4 Graphics Controllers

#### 4.1 Embedded Graphics Controllers

In an embedded graphics controller, the chipset is integrated into the memory controller, allowing part of main memory to be used as video memory. Embedded controllers provide good performance at a very low price, and so are ideal for internet appliances. BeIA supports the following embedded graphics controllers:

- **National Semiconductor:** Geode GXM and GX1 families
- **Via:** MPV4 (Trident core)
- **Intel:** i810
- **SIS:** 5597, 5598, 620 and 6326

#### 4.2 Independent Graphics Controllers

An independent controller provides higher performance but uses separate packaging and has its own memory buffer (including various flavors of VRAM). BeIA supports the CRT and LCD independent graphics controllers listed below

##### 4.2.1 CRT

- **3dfx:** Voodoo III, Banshee
- **ATI:** Rage 64 and Rage 128 families (most of them)
- **Matrox:** Millennium, Millennium2, Mystique, Mystique 200, G100, G200, and G400
- **Nvidia:** Riva 128, TNT, TNT2, Vanta

### 4.3 LCD

- **ATI:** Rage 64 LT, Mobility
- **NeoMagic:** 128, 128XD
- **NeoMagic:** 256AV (experimental)

## 5 Printers

### 5.1 Imaging Layer

- **Epson:**
  - Stylus Color 600 / 670 / 740 / 760 / 777 / 800 / 850 / 880 / 900 / 980 / 1160
  - Stylus Photo, Photo 700 / 750 / 870 / EX / 1270
  - Old Stylus Color Pro / Pro XL
- **Hewlett-Packard:**
  - **T930C** has been fully tested.
  - **Deskjet** family is supported, but all models need to be tested and qualified. The SDK supports the 6xx, 8xx and 9xx families.

### 5.2 Transport Layer

Transport layer modules typically correspond to hardware ports, or write their data to a file. The availability of the expected transport modules is given below:

- **USB:** Available.
- **Parallel port:** Available upon request.
- **Serial port:** The serial port requires custom work; available upon request.
- **lpr:** Available upon request.

## 6 Input Devices

The availability and types of drivers for the expected input devices is given below:

- **Keyboard:** USB and PS2
- **Mouse:** USB, PS2, and serial
- **Touch screen:** Touch screen driver development is by request only.
- **Tablet:** Tablet driver development is by request only.

## 7 Audio Devices

Drivers for the following audio chips and chipsets are supported (with qualifications as noted). 16-bit 44.1/48 kHz stereo in/out, MIDI UART in/out, gameport functionality is generally supported on these controllers.

### 7.1 Integrated Audio Controllers

- **Intel:** i810, i810e, i820
- **VIA:** MVP-4

- **National Semiconductor:** Geode GXM and GX1

## 7.2 Non-integrated ISA Audio Controllers

- **Cirrus/Crystal:** cs4235, cs4236, cs4237
- **Yamaha:** YMF715 (OPL3-SA)
- **OPTi931**
- **Creative:** SB16 Vibra PnP, AWE32, AWE64
- **ESS:** ESS1868, ESS1878, ESS1879 (AudioDrive)

## 7.3 Non-integrated PCI Audio Controllers

- **Aureal:** 8820, 8830 (possibly 8810 and 8808)
- **Yamaha:** DS-1 (ymf724, ymf740, ymf744, and others)
- **Crystal:** SoundFusion (cs4280 and compatibles)
- **ESS:** ESS1938, ESS1942 (Solo-1)
- **S3:** SonicVibes
- **Ensoniq/Creative:** es1370, es1371 (48kHz only), es1373
- **E-mu:** 10k1 (only with Node support)
- **IC Ensemble:** Envy24 (can go to 96 kHz, 24-bit, 10 channels)

### 7.3.1 AC97 Codecs

Many of the non-integrated audio controllers need an **AC97** codec chip to work. BeIA works well with any compatible **AC97** codec. The following chips have been explicitly tested:

- **SigmaTel:** 9701, 9704, 9708, 9744
- **Cirrus/Crystal:** 4297, 4299
- **Wolfson:** WM9702, WM9703
- **IC Ensemble:** ICE1232

### 7.3.2 Other Codecs

- **AKM:** AK4393

---

## 8 Storage Devices

### 8.1 IDE

BeIA supports the following IDE protocols and devices:

- **IDE**
- **EIDE**
- **ATAPI**
- **UltraDMA/33** and **UltraDMA/66**
- Removable media (**Zip**, **SyQuest**, **Jaz**, etc.).
- **SIIG** UltraIDE Pro PCI cards.
- **Promise** Ultra33 and Ultra66 PCI IDE cards.



## 8.2 SCSI Chipsets

- **Adaptec:** AIC-7850, AIC-7870, AIC-7880, and AIC-7895
- **Symbios:** 53c8xx
- **BusLogic:** BT948, BT958, and BT958D

## 8.3 USB

UHCI, OHCI. Isochronous transfers are not yet supported.

## 8.4 Flash Storage

There are three types of Flash storage: **CompactFlash**, **DiskOnChip**, and **DiskOnModule**.

### 8.4.1 CompactFlash

**CompactFlash** allows easy removal and transport of the storage unit. Used for still picture digital cameras, it's an ideal format when easy expandability or data exchange is required.

Supported **CompactFlash** disks:

- **Lexar**
- **SanDisk**
- **Simple**

### 8.4.2 M-Systems DiskOnChip2000

M-Systems **DiskOnChip2000** modules are more cost effective than CompactFlash. They're ideal for systems that don't require easy expandability or removable/exchangeable storage units. For more information, see

<http://www.m-sys.com>

### 8.4.3 DiskOnModule

A **DiskOnModule** is a removable memory module with an IDE interface. It's easier to install than DiskOnChip, but not as transportable as CompactFlash.

---

## 9 Networking Chipsets

### 9.1 External Ethernet Interfaces

The following external ethernet modems are supported:

- **ADSL:** Connected via ethernet using 802.1d ethernet/DSL bridging on the ISP side.
- **ISDN:** Terminal adaptors connected via ethernet or serial cable.
- **Cable:** Connected via ethernet; must not require proprietary software (such as RoadRunner).

### 9.2 Internal Ethernet Interfaces

BeIA supports the following chipsets, and all chipsets that are compatible with these:

- **PCI:**
  - **Novell:** NE2000
  - **DEC:** Tulip (610xx)
  - **TI:** ThunderLan
  - **Intel:** EtherExpress
  - **3Com:** 3c905
  - **RealTek:** RTL8139
  - **AMD:** HomeLan
  - **Via:** VT86C100
- **ISA:** A limited number, primarily those based on the NE2000 chipset. List available upon request.
- **PCMCIA:** Most NE2000 compatible ethernet cards are supported, including:
  - **D-Link:** DE-660 Ethernet Card
  - **Kingston:** KNE-PC2
  - **Linksys:** NetworkEverywhere 10-base-T Card
  - **Linksys:** Combo PCMCIA EthernetCard (EC2T)
  - **NETGEAR:** FA410TX Fast Ethernet
  - **SohoWare Socket Communications:** Ethernet EA Lan Adapter

Also these 3Com cards:

- **3Com:** 3c589 Twisted pair (10BaseT) and Coax.
- **3Com:** 3c575 is under development.

### 9.3 Wireless Home Adapters

The following chipsets are fully supported:

- **Proxim:** 6330 and compatible chipsets.

Support for the following chipset is under-development

- **Lucent** 802.11b wireless PCMCIA cards
  - Lucent Technologies 11Mb WaveLAN (a.k.a.) Orinoco
- **Prism2/Intersil** 802.11b wireless PCMCIA cards
  - **W-Lan:** 11Mbs Wireless PC card
  - **Z-Com:** Lanescape XI-300
  - **Farallon:** SkyLINE 11Mb Wireless PC Card

### 9.4 Home Network Adapters

The following chipsets are fully supported:

- **AMD:** HomeLan and compatible chipsets
- **Intel (DEC):** 21145 Chipset Cards including
  - **Farallon:** HomeLINE PCI card

Drivers are being developed for the following chipsets:

- **HPNA:** 2.0
- **BroadCom:** BCM42xx chipset cards including
  - **Linksys:** Phoneline 10M HPN200SK

---

## 10 Modem Chipsets

Modems come in three flavors:

**Hardware modems** are devices that provide all modem functionality on-board and look (to the system) like a modem connected via a serial (COM) device. Hardware modems offer the best and most efficient performance (and lowest CPU load).

**Software modems** rely upon software running on the host CPU to provide modem functionality. Software modems are typically cheaper than hardware modems, but require specialized drivers to be of use.

**Host-Controlled modems** are a mix of the two. Host-Controlled modems are typically inexpensive and also require specialized drivers.

### 10.1 Hardware Modems

BeIA supports almost any hardware modem that uses generic or preconfigured modem initialization strings, and that includes specific modem initialization strings for one or more modems from each of the following vendors:

**Acer, Actiontec, AIWA, Apex, Archtek, AT&T, Avtek, Banksia, Best\_Data, Boca, BT, Cardinal, Compaq, Creative\_Labs, Digicom, Digitan, Dynalink, Elsa, Gateway, Global\_Village, GVC, Hayes, IBM, Intel, Lucent, Mastro, MaxTech, Megahertz, Microcom, Motorola, MultiTech, Pctel, Penril, Pine, Practical\_Peripherals, Supra, USRobotics, Viva, Vobis, ZOOM, ZyXEL**

In addition, UART-style PCMCIA serial modems are supported.

### 10.2 Software Modems

BeIA supports the following software modems:

- **PCTel:** PCT789 and compatible chipsets
- **AltoCom:** TigerJet (experimental, limited support)

### 10.3 Host-controlled Modems

BeIA supports the following host-controlled modems:

- **Lucent:** 1645, 1646, and compatible chipsets
- **Conexant/Rockwell:** PCI DeviceID 1002, 1036, 1085



# BeIA 2.0 Client Platform Specification

## User Interface Specification

This document is a complete functional description of the default BeIA 2.0 user interface. Many of the elements of this reference UI can be customized by the vendor; unless otherwise noted, the customization is performed through standard JavaScript programming. Relative estimates of the amount of work involved for vendor customization are given.

---

### 1 Screen resolution and layout

"The" reference UI is designed for an 800x600 screen. At higher resolutions (such as 1024x768 or 800x1024) the UI should look and perform quite well. Fitting the UI into a lower resolution (such as 640x480) requires customization, both in laying out the UI, and in resizing individual elements.

---

### 2 Control objects

There are five types of objects that let the user respond to (or control) the UI: Editable text fields, trigger buttons, radio buttons, pop-up menus, and scrollbars.

Some general properties of all five object types are:

- **Keyboard Navigation:** All user-response objects except scrollbars can be navigated (or "focused") using the keyboard. **Tab** switches focus to the "next" object. **Shift+Tab** switches to the previous object.
- **Behaviour Customization:** None of the user-response objects is easily customizable to change the object's behaviour. Behavioural changes require C++ programming work by Be.
- **Look Customization:** Except for editable text fields, all of the user-response objects will be skinnable in a later release. Currently, changes to the look of a user-response object may require some C++ programming work by Be.

#### 2.1 Editable Text Field

The user provides text input to an editable text field by clicking or focusing on the field and then typing. The typed characters are inserted at the text field's cursor; the cursor is only visible while the text field is selected. The text field responds to a number of mouse and keyboard actions:

- A single mouse click moves the cursor so it's under the mouse.
- A double mouse click selects the word under the mouse.
- A triple mouse click selects the entire text displayed in the text field.
- Selected text can be cut (**Alt+x**), copied (**Alt+c**), pasted (**Alt+v**), deleted (**Delete**), or replaced by typing new text. Cutting, copying and pasting works system-wide across any text edit field.

Certain text fields, such as the URL field displayed in the toolbar, may alter this behaviour slightly. Such alterations are noted in the descriptions below.

## 2.2 Trigger Button

A trigger button is a stateless object that's used to initiate an action. The user activates a trigger button by clicking it with the mouse, or by focusing on the object and then pressing the spacebar.

## 2.3 Radio Button

Radio buttons represent a group of mutually exclusive options. Radio buttons are always presented in groups of two or more; at all times, one of the buttons in the group is selected.

The user selects one of a group of radio buttons by clicking it with the mouse, or by focusing on the object and pressing the spacebar. When a radio button is selected, the previously selected radio button (in that group) is automatically deselected.

Selecting a radio button doesn't typically initiate an action; instead, the button simply holds a state that's confirmed or applied by clicking an associated trigger button.

### 2.3.1 Pop-up Menu

A pop-up menu is a list of mutually exclusive items. In its "unselected" state, the pop-up menu is represented by a single-item (trigger) button. To display the menu's contents, the user must "pop open" the menu; there are three ways to do this:

- **Mouse click:** The user clicks the button and the menu is displayed. The user then activates an item in the menu by clicking on the item. To abort the selection--to close the pop-up menu without making a choice--the user clicks anywhere else in the user interface.
- **Mouse drag:** The user presses the button (without releasing the mouse) and the menu is displayed. The user then selects an item in the menu by dragging the mouse to that item and releasing the mouse. To abort the selection the user drags outside the area of the pop-up menu and releases the mouse.
- **Keyboard navigation:** The user presses Tab until the menu's button is focused. Pressing the spacebar pops open the menu. The user then selects an item in the menu by using the up and down arrows (on the keyboard) to navigate to the item, or by pressing the key that's underlined in the item's name. Pressing Enter or the spacebar activates the selection. To abort, the user presses Escape, Cancel, or Dele~~t~~e.

In all cases, the pop-up menu is closed after the selection is activated or aborted.

A pop-up menu's button displays the name of the most recently selected item, or a default if no item has yet been selected.

### 2.3.2 Scrollbar

A scrollbar is a control that's used to scroll the position of a document. A scrollbar consists of a continuously adjustable slider that sits in a slider track, and two arrowhead buttons, one on either end of the slider track:

- **The slider.** The slider is a knob whose length is inversely proportional to the size of the document: If the length of the entire slider track represents the entire document, the knob's length represents the currently visible portion of the document, both in size and position. Dragging the knob causes the document to scroll smoothly and continuously in the direction of the drag. Clicking in the slider track

(i.e. not on the knob itself) causes the document to jump one "screenful" in the direction of the click. Pressing and holding in the slider track produces a series of jumps.

- **Arrowheads.** Clicking an arrowhead button scrolls the document by one "scroll unit" (typically slightly less than one line of text) in the direction of the arrowhead. Pressing and holding an arrowhead repeats the scroll unit to create a smooth even scroll.

A scrollbar can be oriented horizontally or vertically. A horizontal scrollbar controls its document's horizontal position; a vertical scrollbar controls the document's vertical position.

---

### 3 UI feedback sounds

Specific UI actions or events can trigger specific sound effects. The reference UI supplies a number of default effects for specific actions, as listed in a section below. The data for a sound effect is stored as a soundfile in a specific "sound effects" directory. Each effect file is identified by name, where the names indicate the default action to which the effect corresponds. At boot time, the files are read into memory (and the data is decompressed, if necessary) for fast access.

#### 3.1 Customizing the Sound Effects

- The vendor can swap, reuse, or delete existing default effects by simply renaming, copying, or deleting the appropriate sound effect files.
- The vendor can replace the default effects with custom effects by loading the new effects into the sound effects directory and naming the new files appropriately (i.e. overwriting the default files).
- The vendor can adjust the volume of each effect by editing the "sound effect initialization" file that's stored in the sound effects directory.
- A BeIA JavaScript function lets the vendor add sound effects to any HTML page based action.
- Adding an effect to a system-level action, such as a mouse-moved event or a "just plugged in a printer" event, requires C++ programming by Be.

**Note:** BeIA 2.0 does *not* provide sound tools for creating new effects, or for modifying the sound of the default effects.

#### 3.2 Sound Data Formats

Sound effect soundfiles can use any sound data format that's supported by BeIA, but we recommend 22 kHz, 16-bit, 1 channel sounds in ADPCM WAVE format.

#### 3.3 Default Sound Effects

The following sound effects (listed by name) are included in BeIA 2.0

##### Beep

Generic system beep provided for use by plug-ins.

##### PowerUp

Played when the sound system is initialized at boot time, and whenever the device is brought out of sleep mode.

PowerDown

Played when the device is put into sleep mode. (The sound is not played when the device is literally powered down).

VolumeUp

Played when the user presses the hardware volume up button.

VolumeDown

Played when the user presses the hardware volume down button.

MuteToggle

Played when the user unmutes audio by pressing the hardware audio mute button. The effect is *not* played when the user mutes the audio.

MouseDown

Played when the user presses a mouse button, except within a scrollbar or a Java applet window title bar.

MouseUp

Played when the user releases a mouse button, except within a scrollbar or a Java applet window title bar.

MouseEnter

Played when the user moves the mouse into an area that contains an HTML link.

MouseLeave

Played when the user moves the mouse out of an area that contains an HTML link.

MouseEnterTool

Played when the user moves the mouse over a button in the toolbar.

KeyDown

Played when the user presses a character-generating key on the keyboard.

KeyUp

Played when the user releases a character-generating key on the keyboard.

PageLoaded

Played when an HTML page finishes loading. Note that the sound doesn't wait for referenced pages (images files in particular) to finish loading.

WindowActivated

Played when a Java applet window is activated.

WindowOpen

Played when a new Java applet window is opened.

WindowClose

Played when a Java applet window is closed.

Connected

Played when the modem completes the connection process.

Disconnected

Played when the modem hangs up.

Login

Played when the user logs in (on multi-user system).

Logout

Played when the current user logs out (on multi-user system).



**NewMail**

Played each time the system detects one or more new incoming mail messages.

---

## 4 Toolbar

The toolbar is the horizontal, rectangular area that runs along the length of the <<top>> of the screen. It contains two row of icons and control objects that provide information about the state of the device, control the browser, let the user open settings panels, and so on.

- The vendor can move the toolbar (to the top of the screen, presumably) through simple JavaScript programming.
- The vendor can easily reorder the images and buttons in the toolbar.
- Adding and removing elements, or reapportioning the widths of the existing elements may require a moderate amount of layout work.
- Resizing the toolbar's width is easy, but it may not provide a satisfactory appearance unless the toolbar's contents are resized as well.
- Resizing the toolbar's height is also easy but may require a moderate amount of layout and image tweaking for the pages that appear in the "balance" of the screen (the "start" page, the user's home page, preference panels, etc).
- In general, reorienting the toolbar is strongly discouraged: Because of the presence of the URL address text field, a vertical toolbar that runs along the side of the screen (for example) is unworkable. However, if the URL field is removed (a possibility for specialized or highly-constrained browsing experiences), the toolbar can be reoriented with a moderate amount of JavaScript layout work.
- BeIA provides a set of "button state" images (normal, mouse-over-button, mouse-down-on-button) for each of the buttons in the toolbar.

The elements in the toolbar are listed and described below in the order they appear from left to right and top to bottom.

**Branding logo**

The branding logo is a device or vendor-identifying image that can act as a link to a specific page, typically the device's "home page" or the vendor's Website. The default image is the Be logo; it's expected that the vendor will replace this image with an image of their own.

**Back button**

Loads the previous page from the URL history tree.

**Forward button**

Loads the next page from the URL history tree.

**Reload button**

Reloads the page that's currently being displayed.

**Home button**

Loads the device's "home page". The home page identifies the device, welcomes the user, and typically provides links to preference panels and recommended Websites. The page can be stored locally on the device, or it can be an external URL.

**Date & Time**

The date and time are displayed as:

Month Day, Year - Hour:Minute AM|PM

For example:

```
June 3, 2001 - 9:15 AM
```

The date conversion is performed using a standard (and well-known) `strftime()` style format string. Customizing the format is easy.

#### **Zoom In button**

Increases the default font size used to display HTML text in the content area. The default UI supports four sizes. This button has no effect if the largest font size is being used.

#### **Zoom Out button**

Decreases the default font size used to display HTML text in the content area. The default UI supports four sizes. This button has no effect if the smallest font size is being used.

#### **Favorites button**

Toggles the display state of the panel that lists the user's favorite Web sites. If the favorites panel isn't being displayed, clicking the Favorites button displays it; if the panel is being displayed, clicking the Favorites button hides it.

#### **URL field**

This text field displays an image that contains the word Location: followed by a text field that contains the URL of the page currently being downloaded or displayed. The text field provides a place for the user to enter a new URL. See "**4.1 URL Display**" and "**4.2 URL Editing**" below for more information.

#### **Go button**

Loads the URL currently displayed in the URL field.

#### **Print button**

Prints the content page that's currently being displayed. The entire page is printed, even if some of it is scrolled off-screen. See the printing UI specification for more details.

#### **Add to favorites button**

Silently adds, to the favorite pages list, the URL of the page currently being loaded or displayed. The URL is added as the first item at the top level of the favorites hierarchy.

- If the **Favorites** panel is being displayed, the new URL will appear at the top of the favorites list without deselecting the currently selected favorite (if there is one), and without changing the range of favorites that's being displayed (in other words, if the favorites list is scrolled down, the new URL will be added "invisibly" to the top of the list).
- If the **Favorites** panel is hidden, it remains hidden.

#### **Connection status icon**

This three-image icon represents different connection states; it's mostly useful for modem connections. The three states, for which distinct images are provided, are: off-line, connecting, and on-line. More details are provided in the connection settings panel.

The vendor can easily replace the images for the three connections states. The images can be animated (using animated GIFs). Defining and detecting other connection states requires C++ programming.

## **4.1 URL Display**

The URL text field presents remote pages as:

```
Protocol://Domain/Path/File
```

For example:

```
http://www.be.com/products/beappliance/BeAppliance.html
```

- `index.html` and `index.htm` filenames are not shown.

- The URLs accessing local pages are kept hidden. This allows the user to open a settings panel that's defined as a local HTML page (for example) without losing the current remote URL.

## 4.2 URL Editing

The URL field can be used to enter a new URL, by either modifying the currently displayed URL, or by erasing it and entering a new one. The edited URL doesn't any effect until the Enter key is pressed or the Go button is clicked.

- A single click in the URL field selects the entire line of text, thus making it easy for the user to replace the entire address. A second click deselects the line and inserts a cursor at the position of the click. In all other particulars, the URL field acts like a normal text field (including keyboard navigation).
- If no protocol is supplied, "http://" is prepended.
- If the domain name doesn't contain any periods, "www." is prepended (to the domain) and ".com" is appended.
- If the domain name contains a single period, "www." is prepended.

---

## 5 Web Content Area

This Web content area is where remote and local pages are displayed. It takes up the portion of the screen that's not used by the toolbar.

### 5.1 Page History

Each page that's displayed in the content area is added to the "page history list." New pages are always added to the end of the list. If the user clicks the Back button, the content area displays the previous page in the list. The forward button causes the next page in the list to be loaded. If the user has backed up into the list and then goes to a new URL (possibly by clicking on a link), the subsequent pages (i.e. the "forward" pages) are removed from the list and the new page is added.

### 5.2 Multiple page display

The Web content area can display, at most, two pages at a time. By default, it only displays a single page; if the currently displayed page tries to open a page in a second window, the content area is split in two, and the two pages are displayed next to each other.

- The orientation and proportions of the two pages depends on the preferred sizes of the pages. By default (if no preference is stated), the content area is split vertically and equally: The "old" page is displayed in the top half, and the "new" page is in the bottom half.
- A separator bar separates the two pages. The bar includes buttons that lets the user dismiss either page. A dismissed page disappears (as does the separator bar), and the remaining page takes up the entire content area.

The new page is added to the end of the page history list. Dismissing a new page removes it from the list.

---

## 6 Front page

The front page is displayed just after the device has finished booting (and after the user has logged on, if required). It contains a set of graphically unique buttons that, when activated, display local content such as the settings panels, and that take the user to predefined Web locations.

The default Front Page is not meant to be used as is. It's anticipated that the vendor will customize the Front Page to direct the buttons to load vendor-approved Web pages, to add new buttons for specific applications, and to give the page a custom look. Redirecting the default buttons requires a very small amount of JavaScript programming; adding buttons and customizing the look can require considerably more work.

The following list describes the buttons in the default Front Page; keep in mind that they're provided as an example for the vendor.

### Web Search

Goes to the search form page of a Web search engine; the default button goes to <http://www.google.com>.

### Settings

Opens the locally-defined Settings Panels (described in a major section in this document).

### Tutorial

Typically loads a local flash animation that presents a guide to using the BeIA device. The default UI includes a placeholder.

### Email

Goes to the Yahoo email page, <http://mail.yahoo.com>.

### News

Goes to the Yahoo news page, <http://dailynews.yahoo.com>.

### Finance

Goes to the finance page, <http://www.schwabwelcome.com>.

### Entertainment

Goes to the entertainment page, <http://movies.go.com>.

### Shopping

Goes to the shopping page, <http://www.us.buy.com>.

---

## 7 Favorites panel

The **Favorites** panel displays a one-folder deep hierarchy of the user's favorite Web locations. The user can go to a favorite Web page by double-clicking on an item that's displayed in the panel. Clicking on an entry in the list commands the browser to go to that location.

The **Favorites** panel is displayed when the user clicks the **Favorites** button in the toolbar.

Because of its complexity, the **Favorites** panel is implemented as a C++ plug-in. The behaviour of the list view, therefore, can only be modified through C++ programming (by Be). However, the look of the panel -- its colors, fonts, font size, text, and icons -- can be easily modified by the vendor through simple JavaScript programming.

The UI for the favorites panel comprises the favorites list and a row of buttons at the bottom of the panel.

## 7.1 Favorites list

The favorites list displays its items in a vertical list. An item in the list can be a favorite or a folder.

- Clicking on a favorite loads the URL for the item in the browser (and the favorites list is dismissed).
- Clicking on a folder causes the folder's contents to be displayed indented in the list, just below the folder itself. More than one folder can be open at a time. A folder can only contain favorites -- it can't contain another folder.

If the list contains more items than can be displayed on the screen, it "grows" a vertical scrollbar.

Favorites and folders as they appear in the list are described below

### 7.1.1 Favorites

A favorite is displayed in one of two modes:

#### Normal mode

In this mode, a favorite item displays, from left to right: the favorite icon (the same icon is used for all favorites), a name (initially, the page's HTML title), and the URL to the page. As mentioned above, the favorite is indented beneath its folder, if applicable. Actions in normal mode are:

- **Single click.** Selects the favorite and puts it in editing mode.
- **Single click.** Tells the browser to load the favorite's page, and dismisses the favorites panel.
- **Mouse drag.** The user can put a favorite into a folder by dragging it and dropping it onto the folder's line.

#### Editing mode

In editing mode, the name and URL become text fields that the user can modify. Two buttons, apply and cancel, are added to the right of the URL. The apply button confirms changes that the user has made to the name or URL; cancel aborts the changes. Both buttons switch the favorite back to normal mode.

### 7.1.2 Folders

A folder is displayed in one of three modes:

#### Normal (closed) mode

In this mode, a folder item displays, from left to right: the closed folder icon, the name of the folder, and a count of the items in the folder.

- Double click on the icon. Puts the folder into open mode.

#### Open mode

In this mode, a folder item displays, from left to right: the open folder icon, the name of the folder, and a count of the items in the folder. The items (favorites) in the folder are displayed in normal mode on subsequent lines immediately below the folder, and indented to the right.

- Double click on the icon. Puts the folder into closed mode

#### Editing mode

A folder can be put into editing mode whether its open or closed. In editing mode, the folder's name becomes an editable text field that lets the user modify the folder's name. Two buttons, apply and cancel, are added to the right of the item count. The apply button confirms changes that the user has made to the name; cancel aborts the changes. Both buttons switch the folder back to its previous mode.

## 7.2 Buttons

A row of buttons are provided at the bottom of the **Favorites** panel:

### **Go to Favorite**

Sends the browser to the currently selected favorite, or opens or closes the currently selected folder.

### **Edit item**

This will switch whatever item is selected to edition mode.

### **Delete item**

This will delete the currently selected item.

### **New Folder**

This will create a new folder, named New Folder, and insert it before the currently selected item. The favorites panel only support one level of folders.

### **Close**

This button will hide the favorite panel.

## 7.3 Adding Favorites

The only way to create a new favorite is to click on the **Add to Favorites** button in the toolbar. See "4 Toolbar" for details.

---

## 8 Settings Panels

The sample UI includes two parts: a large panel on the right, that will display the currently active setting panel, and a column of buttons on the left indicating what the currently active setting panel is, and allowing to switch to another one.

### **General button**

Makes the General settings panel the active panel.

### **Date & Time button**

Makes the Date & Time settings panel the active panel.

### **Connection button**

Makes the Connection settings panel the active panel.

### **Email button**

Makes the Email settings panel the active panel.

### **Security button**

Makes the Security settings panel the active panel.

### **Printing button**

Makes the Printing settings panel the active panel.

### **Languages button**

Makes the Language settings panel the active panel.

### **Credits button**

Makes the Credits panel the active panel.

## 8.1 Navigation

The settings panel are just a set of local Web pages, and they obey the standard rules as far as the toolbar is concerned. The back and forward buttons can be used to navigate between the different settings pages that were visited, and the user can quit the settings panel the same way it would quit any other Web page, by going to another one using the toolbar navigation, special keys

## 8.2 The Panels

### 8.2.1 General

#### **System volume slider**

Set the volume for all audio output (UI widgets sound effects, MP3 player, RealPlayer, Flash, MIDI,)

#### **System volume mute switch**

Turn on/off all audio output

#### **Feedback sound switch**

Turn on/off all UI widgets sound effects.

#### **Pointer speed slider**

The equivalent of the mouse speed on a desktop system.

### 8.2.2 Date & Time

#### **Date display**

Display the date in numeric form, in the month/day/year form. The complete data, starting with the day of the week, is also printed.

#### **Time display**

Display the time in hours and minutes, followed by the appropriate am/pm option.

#### **Setting arrows**

Each meaningful numeric value (month, day, year, hour and minute) are surrounded top and bottom by a pair of arrows to increase or decrease their value.

#### **Apply changes button**

The real date and time are not modified until this button is pressed. If any modification is made and another settings panel is selected before clicking on the **Apply Changes** button, the modifications will be ignored.

### 8.2.3 Connection

#### **Connection mode selection**

This is radio button selection to switch between the four different connection modes:

- Ethernet (DHCP)
- Ethernet (Static IP)
- Modem (PPP)
- PPP-over-Ethernet (PPPoE)

#### **Connection status**

This includes additional information on the exact configuration and state of the network interface.

### Status

Two states for the Ethernet connection, six states for the modem connection, and one temporary state when switching between interfaces.

- **Online, Offline.** Ethernet only.
- **Disconnected, Dialing, Chatting, Negotiating, Waiting, Connected.** Modem only.
- **Switching to new interface.** Temporary state while switching from one interface to another (like Ethernet to modem).

### IP address

IP address, if any.

### Gateway address

Gateway IP address, if any.

### Advanced settings

Clicking this button will open an additional panel used to configure Proxy settings.

### Static IP settings

This will appear only when the connection mode Ethernet (Static IP) is selected.

- **IP Address text field**
- **Net Mask text field**
- **Gateway text field**
- **Local domain text field**
- **Primary DNS text field**
- **Secondary DNS field**
- **Apply changes button**

### Modem (PPP) settings

This will appear only when the connection mode Modem (PPP) is selected.

- **Phone number text field**
- **User ID text field**
- **Password text field**
- **Modem country code pop-up menu**
- **Init String text field**
- **Apply changes button**

### PPP-over-Ethernet settings

This will appear only when the connection mode PPP-over-Ethernet is selected.

- **User ID text field**
- **Password text field**
- **Connect (dial up) button**
- **Apply changes button**

### Proxy settings panel

- **Proxy server text field**
- **Proxy server port text field**
- **Real server text field**
- **Real server port text field**
- **Apply changes button**

### Network settings button



### 8.2.4 Email

This panel allows the user to configure his email parameters. It might be turned into a generic user setting panel. All these information can be easily removed from the user accessible panels as they can all be configured by the server using the binder APIs. The default UI makes them accessible to the user as follow:

**First name field**

Text edit field to enter the first name of the user (to be used in the extended email address form)

**Last name field.** Text edit field to enter the last name of the user (to be used in the extended email address form)

**Email address field**

Text edit field to enter the user email address to be used for the from field.

**Email address for reply field**

Text edit field to enter the user email address to be used for the reply field.

**Mail server login field**

Text edit field to enter the user login on the mail server.

**Mail server password field**

Text edit field to enter the user password on the mail server.

**Mail server (IMAP) field**

Text edit field to enter the IMAP server access path.

**Mail server (SMTP) field**

Text edit field to enter the SMTP server access path.

### 8.2.5 Security

This allows the user to change the password and hint, by entering the old password, then the new one twice, and the hint.

**Current password text field**

This is only for input; for security, the characters in the password field are displayed as asterisks.

**New password text field**

This is only for input; for security, the characters in the password field are displayed as asterisks.

**Confirm password text field**

This is only for input; for security, the characters in the password field are displayed as asterisks.

**Hint for password text field**

**Change password button**

The password wont be modified until this button is pressed.

### 8.2.6 Printing

**Printer selection pop-up**

This page only appears when at least one printer is detected. The first option allows you to select a printer among all the ones available (physically connected to the BeIA device), using the printer selection pop-up. The rest of the UI will vary depending the capabilities of the selected printer. The following is an example using the Epson Stylus Color 760.

: **Paper Size**

- Letter

- Legal
- A4
- A5
- A6 index
- Executive
- Half letter
- Index card 5in x 8in
- Index card 8in x 10in
- #10 envelope
- DL envelope
- C6 envelope
- Epson photo paper
- Panoramic photo paper

: **Page orientation**

- Portrait
- Landscape

: **Media & Quality**

- Plain paper
- Normal (360 dpi)
- Fine (720 dpi)
- Photo (720 dpi)
- Inkjet 360 DPI paper
- Normal (360 dpi)
- Photo Quality Inkjet paper
- Fine (720 dpi)
- Photo (720 dpi)
- Photo paper
- Fine (720 dpi)
- Photo (720 dpi)
- Photo Quality glossy paper
- Photo (720 dpi)
- Transparency
- Normal (360 dpi)

### 8.2.7 Languages

This page is dynamically created based on the language currently installed on the device. The sample UI only contains English.

### 8.2.8 Credits

There are two panels that display trademarks, copyrights, and other system information: The Be panel displays Be trademark and system information; the Partners panel displays the same for Be's partners.

#### 8.2.8.1 Be

The Be panel provides the following information:

**BeIA revision**

This string identifies the version of the operating system; it comprises a revision name, a build date, and (optionally) a unique build number.

**Machine ID**

This is the unique ID for the device. If the device has an Ethernet chipset, the MAC address from the chipset is used; otherwise the BIOS device ID is used.

**BIOS revision**

The BIOS revision string as provided by the BIOS itself.

**8.2.8.2 Partners**

The following partners provide trademark and copyright information. The exact definition of the partners credits panel depends on the features and components that are included in a device. Note that Be's partners may have other branding requirements in addition to the information displayed in the partners credits panel

- Opera software
- Macromedia
- Fraunhofer
- RSA
- RealG2
- Intel corporation

---

**9 Media Bar**

When a movie or sound is playing, the main window displays a **media bar** that contains a timeline, pause/resume buttons, and a volume control slider. When the media data stops playing, the media bar goes away.

There's only one media bar, and only one media playback at a time: A new media download will interrupt and dismiss currently-playing data. If the user launches a sound playback and then navigates to a new page, the playback continues in the background.

*Detailed UI description pending.*

---

**10 Alerts and Dialogs**

All alerts and dialogs (for error messages, primarily) are based on simple HTML templates. There's one template for each dialog configuration; in other words, there are separate templates for alerts that display a line of text and one button, a line of text and two buttons, a line of text and three buttons, and so on.

The text strings that will be used in the alerts are stored as an index list in an editable file.

The dialog templates and text string files are stored in a pre-defined directory. When an error occurs (or any time an alert needs to be presented to the user), an error code is sent to the browser. On the basis of this code, the browser selects the proper template, creates an alert and populates it with the proper text

This system is configurable: The vendor can modify the existing alert templates and text strings, create new templates and strings, create new error codes, and remap the existing codes (to different templates and/or strings). The vendor can also induce an alert by sending a message (in **JavaScript**) to the browser.

*Detailed UI description pending.*

---

## 11 Localization and Multi-language Support

All text strings that are presented in the UI are stored in HTML files or other editable text files. Localization is as simple as translating the text. Note that BeIA natively supports UTF8 encoding.

BeIA can also support multiple, parallel sets of HTML files (and associated text files) that share the same graphic art, making it possible to select between multiple installed languages in real-time, at only a small footprint cost.

*Detailed UI description pending.*

---

## 12 Cursors

The cursor can change its appearance as it's positioned over a link, as the user clicks the link, and as the new page is being loaded. Each of these cursors is defined as a customizable art file.

Animated cursors are allowed. To create an animated cursor, you simply provide a sequence of art files for that cursor.

*Detailed UI description pending.*

---

## 13 Email client

### 13.1 Message Panels

The top level UI lets you choose from among the four email message panels listed below. Only one panel can be visible at a time.

#### **Inbox**

Displays a list of email messages that have been received by the user.

#### **Drafts**

Displays a list of messages that have been composed by the user but not yet sent.

#### **Outbox**

Displays a list of email messages that the user has sent. Sent messages are automatically archived in the Outbox.

#### **Compose**

Displays the email composition panel.

### 13.2 Address Book button

Brings up the address book panel.

### 13.3 Check Mail button

Connect to the email server to check for new email, and send email waiting to be sent.

## 13.4 Mailbox panel

### 13.4.1 Mailbox Name and Message Count.

All three mailboxes display the name of the mailbox with the total count of emails stored in each one.

### 13.4.2 Email list view

This is implemented using a custom C++ plug-in. All texts, fonts, font size, font style and colors are customizable. The total width of the list view can be customized to fit any UI design. Each column can be resized or reorder independently. These options can be modified only during UI design. For the final product, all these parameters are fixed to specific values.

#### **Title bar**

The title bar is composed of a list of column headers. By clicking on any of them, you can make the list view present its items sorted out in ascending or descending order based on the value of that email parameter. You can sort based on status (sorted by group of same status), from or to field (sorted in alphabetical or reverse alphabetical order), subject (sorted in alphabetical or reverse alphabetical order), date (sorted from most recent to oldest, or reverse), attachment (sorted based on the fact there is an attachment or not) or size (sorted from largest to smallest, or reverse).

#### **Status column**

The column displays an icon to illustrate the status of each email. Existing status are New and Already read. The icons are fully customizable. All strings displayed in any other column can have different font style based on the status of the email (typically standard and bold).

#### **From column/To column**

The column displays the email address of the person who sent you the email (from field for the Inbox). It displays the email address of the person who you are sending the email to (to field for Draft and Outbox).

#### **Subject column**

The column displays the subject of the email.

#### **Date column**

The column displays the date and time the email was received (Inbox) or sent (Draft or Outbox). The format used to display this information can be configured using the same strftime standard time format configuration string.

#### **Attachment column**

The column displays an icon (fully customizable) next to the email that include one or more attachments.

#### **Size column**

The column displays the total size of the email (attachment included) in the following units: in bytes if less than 1KB, in kilobytes with two decimals if less than 1MB, in megabytes with two decimal if less than 1GB, and in GB with two decimals above.

#### **Email line**

Clicking on any of the 6 columns of a specific email line will switch to the email viewer panel for the Inbox and the Outbox, and will switch to the email editor panel for the Draft box. If the email has been marked to be deleted, the whole line will be grayed out and all column will be crossed by an horizontal line.

### 13.4.3 Open Button

Clicking this button will switch to the email viewer panel for the currently selected email, if using the Inbox or the Outbox. Clicking this button will switch to the email editor panel for the currently selected email, when using the Draft mailbox. This button is disabled if no email is selected.

### 13.4.4 Delete/Undelete Button

Clicking the delete button will delete a currently selected email that has not been marked deleted yet. Clicking the undelete button will restore an email that was marked to be deleted, but has not been deleted by the server yet (and so is still displayed as marked for deletion in the email list). This button is disabled if no email is selected.

## 13.5 Email Viewer Panel

### 13.5.1 Message Status

Indicates the index of the email displayed, out of the total count of email in the current mailbox.

### 13.5.2 Previous Button

This will switch to the previous email in the current mailbox, if any. For example, if the third email is currently being viewed, it will switch to the second email. Emails marked for deletion are excluded from this scan.

### 13.5.3 Next Button

This will switch to the next email in the current mailbox, if any. Emails marked for deletion are excluded from this scan. For example, if the third email is currently being viewed, and the fourth is marked for deletion, it will switch to the fifth email.

### 13.5.4 Delete Button

Delete the currently viewed email, and automatically switch the view to the next one (if any). Email can be undeleted only from the mailbox panel.

### 13.5.5 Print Button

Print the current email message, including basic header fields (from, to, subject, date and time), and body text.

### 13.5.6 Add to Addresses Button

Create a new entry in the address book, populated with information extracted from the from field of the current message (first name and last name if available, email address, nickname set to first name by default). In information panel will confirm that the operation has been completed. Further editing these values requires opening the address book panel.

### 13.5.7 Close Button

Close the email viewer panel and switch back to whatever mailbox panel the user was coming from.

### 13.5.8 Reply Button

Switch to the email editor panel, and create a reply email addressed to the from field of the original email. The subject is prefixed by Re:. Each line of the body is prefixed by ">" , and a header is inserted at the beginning (customizable text).

### 13.5.9 Forward Button

Switch to the email editor panel, and create a forward email. The to field is left blank. The subject is prefixed by Fwd:. A header is inserted at the beginning (customizable text).

### 13.5.10 From Field

This field displays the complete from field of the email header (including email address and embedded name informations).

### 13.5.11 Date Field

This field displays the date and time at which the current email has been received (for Inbox) or sent (for Outbox). The format is the same than the one used in the mailbox list view.

### 13.5.12 Subject Field

This field displays the subject of the email.

### 13.5.13 Attachments List

This optional subpanel will appear when the currently email message includes attachment. It will include one line per item, with the following information and options:

**Icon**

Fully customizable icons, illustrating the different file type families currently identified.

**Name link**

The name of the file is underlined and clickable, like any HTML text link. Clicking on that link will open the document viewer corresponding to a specific graphic type, or will open a media content handler corresponding to a specific media type, similarly to what would happen on a regular web page.

**Type**

A text string describing the document type.

**Size**

The size of the document, following the same unit format convention than the mailbox list view.

**Save button**

This will open the storage device selection panel. This panel include the following options:

- **Storage device selection list**

This is a list of the currently available storage device. The user can only select one of them.

- **Cancel button**

This will abort the operation and close the storage device selection panel

- **Save button**

This will save the attachment document at the top level of the selected storage device, then close the panel.

#### 13.5.14 Email Body Viewer

This can display both regular text and complex HTML email.

### 13.6 Email Editor Panel

#### 13.6.1 To Field

Text edit field used to enter the email address of the person you are sending the email to. It can be automatically populated by the reply button of the email viewer or the use button of the address book panel.

#### 13.6.2 Subject Field

Text edit field used to enter the subject of the email. It can be automatically populated by the reply or forward button of the email viewer panel.

#### 13.6.3 Send Button

Send this email as currently displayed. After that, the email will be only visible in the Outbox mailbox, and can no longer be modified. Also, it will be sent to the server at the first opportunity.

#### 13.6.4 Save as Draft Button

Save this email into the draft mailbox, to be completed at a later time.

#### 13.6.5 Back Button

Abort whatever has been done in the email editor panel and go back to the previous panel the user was coming from.

#### 13.6.6 Clear Button

Reset all fields and the body to their default empty value.

#### 13.6.7 Attach Files Button

Open the file browser panel, and display an alert explaining how to attach a file using the file browser UI.



### 13.6.8 Body Text Editor

Can be used to send both regular text and rich text email messages. The email will always be send in both format to be readable by as many email client as possible. When using the rich text format, the following attributes can be configured for each character (each attribute being applied to either the current selection or the default settings for new text):

#### **Color picker**

The default UI select a color among 6. All UI and color selection is completely done in Javascript and can be completely customized.

#### **Style selector**

Offer two different font styles: normal and bold.

#### **Font size selector**

The default UI provides two arrows (up and down) to increase or decrease the default font size, among 6 different HTML sizes.

#### **Text editor frame**

This is a custom C++ plug-in, that provide the standard text edit field features with the addition of the color, style and font size attributes (set from the UI through Javascript APIs).

## 13.7 Address Book Panel

### 13.7.1 Address List View

This is implemented using a custom C++ plug-in. All texts, fonts, font size, font style and colors are customizable. The total width of the list view can be customized to fit any UI design. Each column can be resized or reorder independently. These options can be modified only during UI design. For the final product, all these parameters are fixed to specific values.

This panel displays the list of address book entries. All entries are normally shown in display mode. One of them can be switched to editing mode using the button options.

#### 13.7.1.1 Title Bar

The title bar is composed of a list of column headers. By clicking on a column header, you make the list view present its items sorted out in ascending or descending order based on the value of that email parameter. You can sort based on icon (sorted by group of same type, only one currently), nickname (sorted in alphabetical or reverse alphabetical order), first name (sorted in alphabetical or reverse alphabetical order), last name (sorted in alphabetical or reverse alphabetical order) or email address (sorted in alphabetical or reverse alphabetical order).

#### 13.7.1.2 Display Mode

Each entry displays the following non editable strings:

##### **Icon**

This represents the type of entry in the address book. Currently, only single users are supported. The icons are fully customizable.

##### **Nickname**

##### **First name**

##### **Last name**

**Email address**

13.7.1.3 Editing mode

**Icon**

Similar to the display mode (not editable).

**Nickname**

Text edit field to edit the nickname string.

**First name**

Text edit field to edit the first name string.

**Last name**

Text edit field to edit the last name string.

**Email address**

Text edit field to edit the email address string.

**Cancel button**

Cancels all changes and exits editing mode.

**Okay button**

Validates the changes and exits editing mode.

13.7.2 Use Button

Switch to the email editor panel, after populating the **to:** field using the information stored in the currently selected address book entry. This button is disabled if no entry is selected.

13.7.3 Edit Info Button

Switch the currently selected address book entry to editing mode. This button is disabled if no entry is selected.

13.7.4 New Button

Create a new empty address book entry and activate it in editing mode.

13.7.5 Delete Button

Delete the currently selected address book entry. This button is disabled if no entry is selected.

13.7.6 Close Button

Close the address book panel and switch back to the panel the user was coming from.

---

## 14 File Browser Panel

### 14.1 Storage Device Selection Panel

This device will present one button per storage device available. Availability of storage device is based on fully plug-and-play detection. Clicking on a specific device will open the file browser list view for the specific device. The following devices are examples of devices that could be supported:

- **CompactFlash**
- **USB Zip drive**
- **Network storage.** Using an FTP server.

### 14.2 File Browser List View Panel

This is implemented using a custom C++ plug-in. All texts, fonts, font size, font style and colors are customizable. The total width of the list view can be customized to fit any UI design. Each column can be resized or reorder independently. These options can be modified only during UI design. For the final product, all these parameters are fixed to specific values.

This panel displays the list of all files on the devices as filter based on a specific criteria.

#### 14.2.1 File Type Filtering Selector

The reference UI will offer two options to filter files based on type. Option to provide further customization are not defined yet and may still require some small amount of C++ custom work.

**Show only known types**

All file types supported by the software stack (see software component specification).

**Show all files**

Files that can not be displayed or played will generate an error message if you try to open them. But you can copy them or attach them to an email.

#### 14.2.2 Title Bar

The title bar is composed of a list of column headers. By clicking on any of them, you can make the list view present its items sorted out in ascending or descending order based on the value of that email parameter. You can sort based on icon (sorted by group of same type), name (sorted in alphabetical or reverse alphabetical order), last modification time and date (sorted from the most recent to the oldest, or reverse), size (sorted from the largest to the smallest, or reverse) or any optional attributes (sorted based on ordered values or different group of possible values)

All entries are normally showed in display mode. One of them can be switch to editing mode using one of the button options.

##### 14.2.2.1 Display Mode

Each entry display the following non editable strings:

**Icon**

This represents the type of file. Currently, the following file types are identified: image, text/html, sound, video or other. All icons are fully customizable.

**Filename**

This displays the name of the file (after removing potential “.” filetype extensions)

**Last modification time and date**

This use the same format than the one used for the mailbox list view

**Size**

This use the same format than the one used for the mailbox list view

**Custom attribute Display mode**

Depending the how the file type filtering is done, all files may also shared extra attributes that the user may be interested to know of, for example music information for audio files.

#### 14.2.2.2 Editing mode

**Icon**

Similar to the display mode (not editable).

**Filename**

Text edit field to edit the nickname string.

**Last modification time and date**

Similar to the display mode (not editable).

**Size**

Similar to the display mode (not editable).

**Custom attribute Display mode field**

Text edit field to edit some specific custom attributes of a file.

**Cancel button**

Cancels all changes and exits editing mode.

**Okay button**

Validates the changes and exits editing mode.

#### 14.2.3 Open Button

Clicking on the Open button opens the document viewer that corresponds to the specific file type of the currently selected file, or opens a media content handler corresponding to the specific media type, similarly to what would happen on a regular Web page after clicking on a link to the file. This button is disabled if no file is selected.

#### 14.2.4 Edit Info Button

Switch the currently selected file entry to editing mode. This button is disabled if no entry is selected.

#### 14.2.5 Copy To Button

Clicking on this button will open the storage device selection panel (the button is disabled if no file is selected). This panel includes the following options:

**Storage device selection list**

This is a list of the currently available storage device. The user can only select one of them.

**Cancel button**

This will abort the operation and close the storage device selection panel

**Okay button**

This will copy the currently selected file at the top level of the selected storage device, then close the panel.

**14.2.6 Attach Button**

Add the current file to the attachment list of the email being currently edited. This button is disabled if no file is selected.

**14.2.7 Delete Button**

Delete the currently selected file. This button is disabled if no file is selected.

**14.2.8 Close Button**

Close the address file browser panel and switch back to the panel the user was coming from (if applicable).

---

## **15 Error Messages**

The error messages are listed in a chapter of the *UI Customization* guide.



# BeIA 2.0 Client Platform Specification

## The Browser

This document describes BeIA's integrated browser and the technologies it supports. This document does not describe third-party plug-ins that are licensed along with the browser (such as **Macromedia Flash** or **RealPlayer G2**).

---

## 1 Overview

BeIA's integrated Web browser is based on **Opera 4.0** from **Opera Software**; **Opera 5.0** is under development and should be available for BeIA 2.0. The browser understands **HTML 4.0**, **Cascading Style Sheets (CSS1 & CSS2)**, and **Javascript 1.3 (ECMAScript)**. Some aspects of **DHTML** are supported. Content can be fetched using **HTTP 1.0/1.1** (including **SSL 2/3/TLS**), **FTP**, and local file access.

---

## 2 Hypertext Markup Language (HTML) 4.0

BeIA can parse and display **HTML 4.0** documents.

### 2.1 HTML elements

All HTML 4.0 elements, except `label` and `legend`, are supported. For a complete list of elements, go to

- <http://www.w3.org/TR/html40/index/elements.html>

### 2.2 HTML attributes

All HTML 4.0 attributes, except as noted below, are supported. For a complete list of attributes, go to

- <http://www.w3.org/TR/html40/index/attributes.html>

Unsupported attributes: `abbr`, `accept`, `accept-charset`, `accesskey`, `align`<sup>1</sup>, `axis`, `char`, `charoff`, `charset`, `cite`, `content`, `coords`<sup>2</sup>, `defer`, `dir`, `disabled`, `for`, `frame`, `headers`, `hreflang`, `option`<sup>1</sup>, `lang`, `longdesc`, `profile`, `readonly`, `rel`<sup>3</sup>, `rev`, `rules`, `scheme`, `scope`, `shape`<sup>2</sup>, `standby`, `tabindex`, `title`, `type`<sup>4</sup>, `version`, `width`<sup>5</sup>

<sup>1</sup> not supported when used with `legend` or `label`; supported in all other cases

<sup>2</sup> supported on `area`, but not on `a`

<sup>3</sup> `rel="stylesheet"` is supported

<sup>4</sup> supported when used with `button`, `input`, `li`, `ul`, `ol`

<sup>5</sup> not supported when used with `pre`.

### 3 Cascading Style Sheets (CSS)

BeIA can direct the formatting of HTML documents according to **Cascading Style Sheets Level 1 (CSS1)** and **Level 2 (CSS2)**

#### 3.1 Level 1 (CSS1)

BeIA supports all of **CSS1**. For the specification, go to

- <http://www.w3.org/TR/REC-CSS1>

#### 3.2 Level 2 (CSS2)

Except for the components listed below, BeIA supports all of **CSS2**. For the specification, go to

- <http://www.w3.org/TR/REC-CSS2>

Unsupported features: **Aural CSS**

Unsupported rules: @page

Unsupported properties: caption-side, clip, cursor, direction, font-size-adjust, font-stretch, marker-offset, text-shadow, unicode-bidi, all outline properties (outline, outline-color, outline-style, outline-width)

Unsupported property values: font:system, list-style-type:\*, display:marker, text-align:<string>, visibility:collapse, overflow:scroll

Unsupported selectors: :first-child, :focus, :lang(), multiple pseudo-classes in a single simple selector.

### 4 Javascript and ECMAScript

BeIA can execute **ECMAScript** code embedded in HTML content.

(ECMAScript is a version of the **Javascript Core** that's being standardized through the ECMA standards body. ECMAScript does not include browser-related objects such as **DOM 1** and **DOM 2**).

#### 4.1 ECMA-262 Version 2 and Javascript 1.3

BeIA supports the entire **ECMA-262 Version 2** standard. This standard corresponds, more or less, to **Javascript 1.3**.

Except for the RegExp object and **UTF-16** support, the Javascript 1.3 implementation is complete. For a description of Javascript 1.3, go to

- <http://developer.netscape.com/docs/manuals/js/client/jsref/index.htm>

#### 4.2 ECMA-262 Version 3 and Javascript 1.4

The BeIA Javascript interpreter supports most of the upcoming **ECMA-262 Version 3**. This version corresponds to Netscape's **Javascript 1.4 Core**. The features that are missing from BeIA's implementation of ECMA-262 Version 3 are:



- Exception handling has not been sufficiently tested.
- `locale` methods have not been implemented.

---

## 5 DHTML

**DHTML** is a hodgepodge of cooperating—or in some cases competing—standards. Since it's a collection of technologies rather than a single standard, it isn't possible to completely define the state of DHTML support at this time. However, as a natural result of its implementations of HTML 4.0, CSS1, CSS2, and Javascript 1.3—and barring a few key features—BeIA is compatible with most commonly used elements of DHTML.

### 5.1 DOM Objects

Most of the objects specified by the **Document Object Model (DOM) Version 1** and **Version 2** are present and accessible from ECMAScript code. The most significant limitation in BeIA's implementation is:

- **Any changes made to DOM objects that would require laying out a portion of an HTML document after it has been loaded will not work.**

For example, Javascript code that dynamically changes the absolute position or visibility of a particular HTML element, or that dynamically writes new HTML code into part of a page after that page has been loaded, will have no effect or may cause an exception to be thrown.

### 5.2 No layer Extension

BeIA doesn't support the `layer` extension (popularly supported by Netscape Navigator). There are no plans for this support.



# BeIA 2.0 Client Platform Specification

## Browser Plug-ins

This document describes the plug-ins that are supported by the BeIA implementation of **Opera 4.0**.

### 1 RealPlayerG2

A plug-in for **RealPlayerG2 6.0** from **RealNetworks** is provided. This is the complete integrated version of **RealAudio** and **RealVideo**.

**RealPlayerG2 8.0** is under development.

#### 1.1 Size

At 2.1 MB, **RealPlayerG2** is the largest integrated component in Opera. Currently, the **RealPlayerG2** plug-in can't be whittled down by removing unneeded components (codecs and plug-ins)—it's all or nothing.

Note that **RealPlayerG2 8.0** will require around 3.5 MB of storage, and significantly more RAM than **RealPlayerG2 6.0**.

#### 1.2 Delivery and Markup Formats

The **RealPlayerG2** plug-in supports **RTP**, **RTSP**, **PNM**, and **HTTP** delivery, and **SMIL** markup.

#### 1.3 Codecs

The following **RealPlayerG2** codecs are supported: **14\_4**, **28\_8**, **cokr**, **cook**, **ddnt**, **dnet**, **drv1**, **drv2**, **dspr**, **rncolor**, **rv10**, **rv20**, **sipr**

#### 1.4 Plug-ins (and associated MIME types)

The following **RealPlayerG2** plug-ins are provided. When known, a plug-in is followed by the list of media formats, identified by MIME signatures, that it can handle.

**audplin**: audio/x-pn-windows-acm, x-pn-aiff, x-pn-alaw, x-pn-au, x-pn-dvi4, x-pn-g721, x-pn-g723, x-pn-gsm610, x-pn-mulaw, x-pn-wav

**authmgr**

**bascauth**

**httpfsys**: application/octet-stream; audio/rn-mpeg

**meta**: audio/x-pn-realaudio

**plusplin**: application/x-pn-plusurl

**pnxres**

**pxcgif2****pxcjpeg2****pxff**: application/vnd.rn-realpixmap, vnd.rn-realpixmapstream2**pxgf**: application/vnd.rn-gifstream2**pxgr**: application/vnd.rn-gifstream, vnd.rn-gifstream2**pxjf**: application/vnd.rn-jpegstream**pxjr**: application/vnd.rn-jpegstream**pxlive**: application/vnd.rn-realpixmapstream2**pxrend2**: application/vnd.rn-realpixmapstream, vnd.rn-realpixmapstream2**rarender**: audio/x-pn-multirate-realaudio, x-pn-realaudio

**rmffpln**: application/x-pn-imagemap, x-pn-realevent, x-pn-realmedia;  
 audio/x-pn-realaudio;  
 video/x-pn-multirate-realvideo, x-pn-realvideo;  
 logical-application/x-pn-multirate-imagemap, x-pn-multirate-realevent;  
 logical-audio/x-pn-multirate-realaudio, x-pn-realaudio;  
 logical-video/x-pn-multirate-realvideo

**rn5auth****rtffpln**: application/vnd.rn-realtex, x-pn-realtex**rtrender**: application/vnd.rn-realtex, x-pn-realtex

**rvrend**: application/x-pn-imagemap;  
 video/x-pn-multirate-realvideo, x-pn-realvideo

**sdpln**: application/sdp**smlffpln**: application/rma-driver, smil, vnd.rn-rmadriver**smlrendr**: application/rma-driver, vnd.rn-rmadriver**smmrendr**: application/x-pn-realad, x-pn-realevent**smplfsys**

**vidpln**: application/x-ms-asf, x-pn-avi-plugin, x-pn-quicktime-stream;  
 audio/x-pn-alaw, x-pn-compressed-quicktime-audio, x-pn-g721, x-pn-g723, x-pn-gsm610,  
 x-pn-mulaw, x-pn-wav, x-pn-windows-acm;  
 video/x-pn-icm-plugin, x-pn-jpeg-plugin, x-pn-qtvideo-stream

**vivff**: audio/g723-lo, g723-silent, x-g723-hi, x-siren;  
 video/vnd.vivo, vnd.vivo2, x-h263

---

## 2 Macromedia Flash 4.0

**Opera 4.0** incorporates **Flash 4.0**, the flash-shockwave player from **Macromedia**.

---

### 3 PersonalJava 3.0.1

The **PersonalJava 3.0.1** plug-in (language) is based on the **PersonalJava Application Environment Specification 1.1.1**. The full specification can be found here:

<http://java.sun.com/products/personaljava/spec-1-1-1/index.html>

#### 3.1 Size

BeIA's **PersonalJava** implementation is ~800KB on Flash, and ~4MB in memory.

#### 3.2 Validation

BeIA's **PersonalJava** implementation is validated with **Sun's PJCK**.

#### 3.3 Optional components

The **PersonalJava Application Environment Specification** allows optional support for a specified set of packages, classes, and features. BeIA's **PersonalJava** implementation supports all optional packages except `java.rmi` and `java.sql`, and all optional features except locales other than `en_US`. To be clear, this means that BeIA *does* support the following optional packages: `applet`, `awt`, `beans`, `io`, `lang`, `lang.reflect`, `math`, `net`, `text`, `util`, and `util.zip`.

**Note:** The unsupported packages and features can be included at the customer's requested.

---

## 4 MediaPlayer

BeIA includes a version of **MediaPlayer** (Be Incorporated), a lightweight, extensible audio playback application (video is currently unsupported). Although it's implemented as a plug-in, **MediaPlayer** also has hooks to present a media bar in **Opera**'s main window. The media bar contains a timeline with a “scrubbable” thumb (to move quickly forward or backward through the data), start/stop/pause buttons, and a volume control.

#### 4.1 Supported Media Formats, Codecs, and File Formats

See the lists in “**7.1.1** Media plug-ins” in the *Architectural Overview* document.

---

## 5 Beatnik

BeIA includes the Beatnik® Audio Engine (BAE(TM)). The Beatnik Audio Engine is able to play Beatnik's Rich Music Format (RMF) files RMF files. RMF provides a richer audio experience than MIDI in a fraction of the file size needed by MP3 data of similar playback length.



# BeIA 2.0 Client Platform Specification

## The Development Environment

This document describes the BeIA development environment.

---

## 1 Develop Using BeOS 5

The BeIA development platform is based on BeOS 5, a complete modern desktop operating system that's compatible with most modern hardware. For the full compatibility list, go to

- [http://www.be.com/support/guides/beosreadylist\\_intel.html](http://www.be.com/support/guides/beosreadylist_intel.html)

BeOS 5 can easily be installed in a multi-partition environment coexisting with Windows or Linux, and can even be installed as a simple file in a Windows partition; see the Be Web site ([www.be.com](http://www.be.com)) for details.

### 1.1 Modern C++ Architecture Framework

The BeOS API offers a modern, consistent, small framework to develop applications. The BeOS API is documented in the BeOS Book, which is included in HTML format on every Be computer.

Online sample code and information is freely available at:

[http://www-classic.be.com/developers/developer\\_library/index.html](http://www-classic.be.com/developers/developer_library/index.html)

#### 1.1.1 Posix API Compatibility

The BeOS API is built around **Posix** compatibility. If you're familiar with the **Posix** API, you should be comfortable with the Be API.

---

## 2 BeIDE - the Integrated Development Environment

BeOS 5 includes **BeIDE**, Be's Integrated Development Environment. Some of the features of **BeIDE** are listed in the following sections.

### 2.1 Easy to Learn and Use

BeIDE comes with a set of project templates (or "Stationary") for creating applications, libraries, add-ons (plug-ins), and so on.

To create a new project, you choose a Stationary template (or start from scratch) to bring up a new **BeIDE** project window into which you drag and drop your application sources.

The project window lets you reorder your source files, and organize them into groups. You can also add header files (and other files of interest) to the project—this is particularly handy if you want to search-and-replace text across all files in your project.

**BeIDE** lets you open and build more than one project at the same time. In addition, a project can contain “sub-projects”—you can use **BeIDE** as a simple build control system.

## 2.2 Easy Application Building

In addition to building and running your entire application, menu items are provide for preprocessing, linking, compiling, checking your code’s syntax, and so on. You can also build your application and launch it in the debugger through a menu item selection.

Re-compiling is intelligent and selective: If a source file hasn’t changed (and doesn’t depend on any files that have changed), it won’t be re-compiled.

BeIDE presents a graphical interface to the compiler and linker options. Setting up the proper options to create debuggable code can be performed with a single menu selection.

If you’re running on a multi-processor machine, BeIDE automatically spawns a compilation thread for each processor, thus parallelizing the compile.

## 2.3 Syntax Highlighting Editor

BeIDE provides a source code editor that understands **C**, **C++**, and **Java**, that performs brace matching and auto-indentation, lets you set your own key bindings, and so on.

## 2.4 Multi-File Find Capability

The source code editor has a find mechanism that can search across all your project’s files, understands regular expression syntax, maintains a history of your previous searches, and can run in batch mode in which it presents you with a list of matches.

## 2.5 Navigation

BeIDE and the source code editor provides a number of hooks that make it easy to navigate between your source code, system header files, and the BeOS technical documentation easier:

- You can pop up a list of a file’s functions and go to a desired function by choosing it from the list.
- Go to an errant line of code by clicking on an error message in **BeIDE**’s compile message window.
- Bring up documentation for a selected keyword through a single keyboard gesture.
- Similarly, you can select a header file name and jump to the file through a single gesture.
- Easily switch between a source file and its similarly-named header file through a single gesture.

---

## 3 bdb - the Be Source Code Debugger

**bdb** is the BeOS 5 source code level graphical debugger. Some of its features are described below.



### 3.1 Graphical User Interface

“If you can see it, you can act on it” style of interface. All elements of your application—files, functions, addresses, variables, registers, threads, even your application’s frame stack—are presented in hierarchically arranged lists. Click on an element to see its value, edit it, set a breakpoint, and so on.

**bdb** also lets you quickly open your application’s source files, and edit them with an editor of your choice.

### 3.2 Program Control

**bdb**’s main control bar contains the expected program control buttons: **Run**, **Stop**, **Kill**, **Step into function**, **Step over function**, **Step out of function**. All of these actions can also be taken through menu shortcuts.

You can easily attach **bdb** to a currently running program, or attach it to a program that has crashed to examine its state.

### 3.3 Breakpoints

**bdb** lets you set a wide range of break conditions. You can tell your application to break when...

- it’s about to execute a “marked” line of code
- a variable or memory location changes
- a library is about to be loaded
- a thread is created
- an exception is thrown

Setting a breakpoint is as simple as clicking the mouse.

### 3.4 Assembly Code Support

Although **bdb** is primarily a source code debugger, it also knows how to display the contents of registers, can show the assembly code for a function, and lets you set breakpoints in assembly code instructions when the source code isn’t available.

### 3.5 Remote Debugging

**bdb** lets you debug a compressed application that’s running on a remote device.

---

## 4 gcc Compiler and Linker

BeOS 5 provides **gcc 2.9** compiler and linker.

## 5 Shell Commands and Tools

BeOS 5 provides a version of the **Bash** shell (the “Bourne again shell”), and a suite of over 250 **UNIX** and **gnu** command line tools, from **awk** to **zgrep**.

---

## 6 Remote Program/Device Monitor (Spy)

**Spy** lets you to monitor activity on a remote device. You can look at CPU and memory usage across the entire system, or look at individual teams and threads. In addition, **Spy** lets you copy files (including executables) onto the monitored device during the test cycle.

---

## 7 Profiling

The **bprof** and **profile** tools are available.

---

## 8 Java Development Tools

The following tools are available:

- The **javac** Java compiler
- The **jdb** Java debugger