

BARIONET

TM



BARIONET

Universal network-enabled
automation interface in
home automation, commercial
control and monitoring applications



Development Kit Manual Technical Reference

Revision 1.02

Table of Contents

Introduction..... 1

Websaver.....	1
Prerequisites.....	1
Applications.....	1

Files and Folders2

Unzip the Kit.....	2
Root Files.....	2
Root Folders.....	4
File index.html.....	6
Frame menu.....	6
Frame config.....	6
Frame empty.....	6
Frame set uifstatus.html.....	7

Barionet Memory.....8

Page X1 firmware.....	8
Page WEB1 web application.....	8
Page WEB2 web help.....	8
Page WEB3 .. WEB7 user application.....	8

File Structure.....9

Root.....	9
Naming Convention.....	9

Sample User application CISCO.....10

Unzip the sample CISCO.....	10
-----------------------------	----

HTTP Interfacing.....11

CGI commands.....	11
Html tags.....	12

Introduction

Webserver

The Barionet contains a web server which allows visualization applications without any software on the host. The web server pages are loaded into a Flash memory to prevent loss of data in case of power interruption.

This manual explains how to compile and to upload your own User applications for the Barionet.

It starts with an overview of the files and folders used by the existing UI (User Interface) and it's general frame sets to enable you to understand the principal construction of the UI. This will help you to create your own User applications in a breeze.

Prerequisites

- This manual is about the Barionet Development Kit
Download the Kit from www.barix.com
- This manual refers to the Barionet User Manual.
Download the User Manual from www.barix.com
- To edit the HTML pages you will need an editor tool such as Notepad or WordPad, we recommend the use of better tools such as Ultra Edit and Dreamweaver.
- To edit the graphics you will need a graphic tool, we recommend the use tools such as Photoshop and Fireworks.
- A PC, Network Interface Card with configured TCP/IP protocol, a browser such as Internet Explorer.
- A tool to unzip the downloaded files.

Applications

- Customize the user interface to your own CD / CI
- Add custom pages to reflect the environment
- Web-based visualization of devices
- Home automation visualization
- Security, monitoring and surveillance applications

Files and Folders

Unzip the Kit

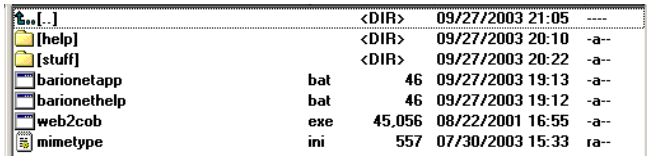
Create a folder on your PC's harddisk (eg. barionet, any name will do, the length of the path is not important)

The Development Kit comes as a ZIP file,

eg. barionet_devkit_10_20030927.zip

Unzip the file to the folder created.

The content should look like this:

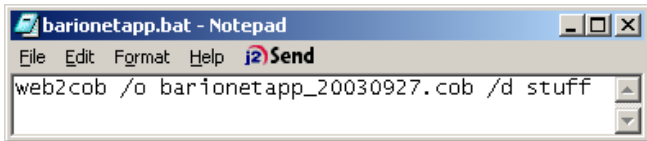


Root Files

Lets take a look on the root files.

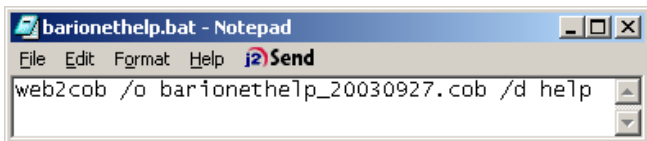
barionetapp.bat

The file barionetapp.bat is a batch file that starts the compilation of the folder **stuff**.



barionethelp.bat

The file barionethelp.bat is a batch file that starts the compilation of the folder **help**.



web2cob.exe

The file web2cob.exe is the actual compiler.

The command line :

```
web2cob /o barionetapp_20030927.cob /d stuff
```

calls the compiler **web2cob** to create an output file (**/o**) named **barionetapp_20030927.cob** out of the content of the directory (**/d**) named **stuff**.

To create your own User application we recommend to copy the batch file **barionetapp.bat** and rename it to **userapp.bat**.

Then edit the file and change it as following:

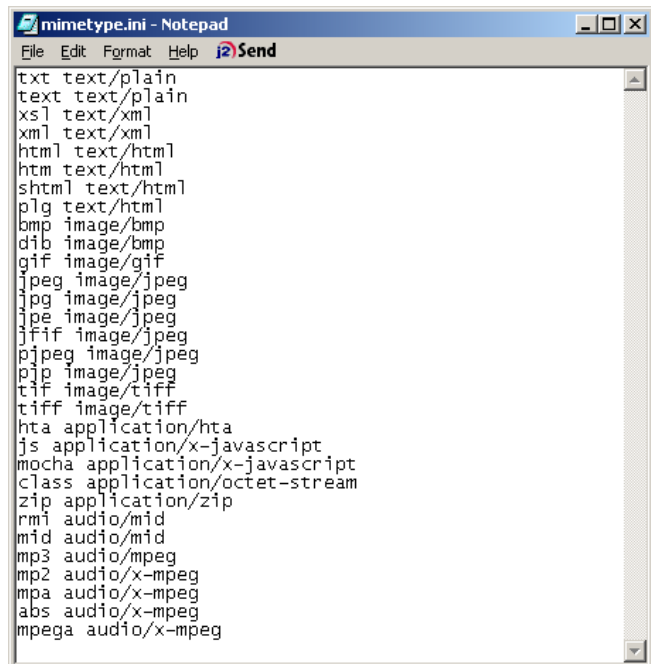
```
web2cob /o userapp.cob /d user
```

Now create a folder named **user**. Place your HTML files in here.

To compile your user application run the batch file userapp.bat

mimetype.ini

The file mimetype.ini is the MIME type definition file needed by the web server.



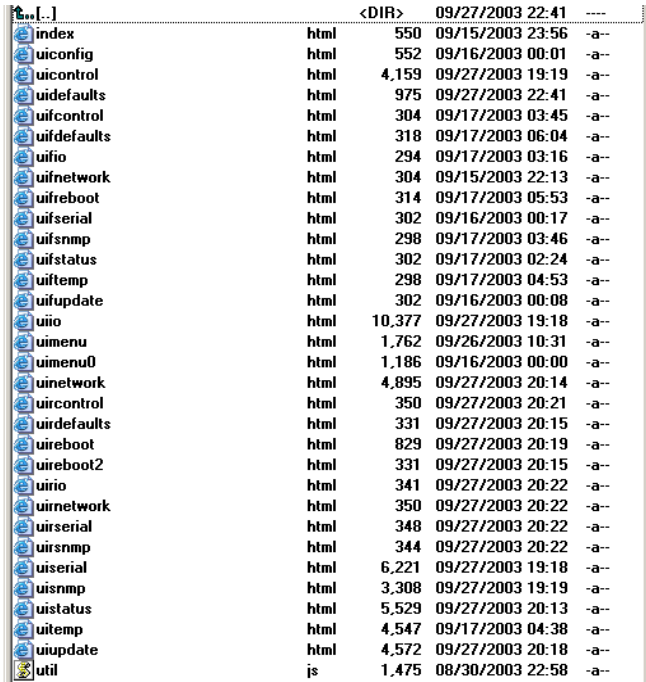
```
mimetype.ini - Notepad
File Edit Format Help i2 Send
txt text/plain
text text/plain
xsl text/xml
xml text/xml
html text/html
htm text/html
shtml text/html
plg text/html
bmp image/bmp
dib image/bmp
gif image/gif
jpeg image/jpeg
jpg image/jpeg
jpe image/jpeg
jfif image/jpeg
pjpeg image/jpeg
pjp image/jpeg
tif image/tiff
tiff image/tiff
hta application/hta
js application/x-javascript
mocha application/x-javascript
class application/octet-stream
zip application/zip
rmi audio/mid
mid audio/mid
mp3 audio/mpeg
mp2 audio/x-mpeg
mpa audio/x-mpeg
abs audio/x-mpeg
mpega audio/x-mpeg
```

Root Folders

Lets take a look on the root folders.

Folder *stuff*

The folder **stuff** contains the current HTML pages used by the UI.



File Name	Type	Size	Date	Permissions
index	html	550	09/15/2003 23:56	-a-
uiconfig	html	552	09/16/2003 00:01	-a-
uicontrol	html	4,159	09/27/2003 19:19	-a-
uidefaults	html	975	09/27/2003 22:41	-a-
uifcontrol	html	304	09/17/2003 03:45	-a-
uifdefaults	html	318	09/17/2003 06:04	-a-
uifio	html	294	09/17/2003 03:16	-a-
uifnetwork	html	304	09/15/2003 22:13	-a-
uifreboot	html	314	09/17/2003 05:53	-a-
uifserial	html	302	09/16/2003 00:17	-a-
uifsnmp	html	298	09/17/2003 03:46	-a-
uifstatus	html	302	09/17/2003 02:24	-a-
uiftemp	html	298	09/17/2003 04:53	-a-
uifupdate	html	302	09/16/2003 00:08	-a-
uio	html	10,377	09/27/2003 19:18	-a-
uimenu	html	1,762	09/26/2003 10:31	-a-
uimenu0	html	1,186	09/16/2003 00:00	-a-
uinetwork	html	4,895	09/27/2003 20:14	-a-
uircontrol	html	350	09/27/2003 20:21	-a-
uirdefaults	html	331	09/27/2003 20:15	-a-
uireboot	html	829	09/27/2003 20:19	-a-
uireboot2	html	331	09/27/2003 20:15	-a-
uirio	html	341	09/27/2003 20:22	-a-
uinetwork	html	350	09/27/2003 20:22	-a-
uirserial	html	348	09/27/2003 20:22	-a-
uirsnmp	html	344	09/27/2003 20:22	-a-
uiserial	html	6,221	09/27/2003 19:18	-a-
uisnmp	html	3,308	09/27/2003 19:19	-a-
uistatus	html	5,529	09/27/2003 20:13	-a-
uitemp	html	4,547	09/17/2003 04:38	-a-
uiupdate	html	4,572	09/27/2003 20:18	-a-
util	js	1,475	08/30/2003 22:58	-a-

There is a default page called **index.html**. This page will be loaded when the web server is called by IP address only.



























To distinguish original UI files from User files we have used the following naming convention (you are free to use your own):

- All files of the UI start with the characters **ui**.
- All frames start with the characters **uif**.
- All response pages start with the characters **uir**.

There is a java script called **util.js**. This script is used to verify user entries in the configuration pages. Do not alter this file !

Folder *help*

The folder **help** contains the current HTML help pages and the graphics used by the UI.

<DIR>	09/27/2003 21:05	----
 barionet	gif	2,872 08/30/2003 14:16 -a-
 Barionetdev	gif	4,382 09/02/2003 19:34 -a-
 barix	gif	2,006 08/04/2003 08:02 -a-
 barix0	gif	1,800 08/25/2003 17:20 -a-
 col0	gif	43 07/31/2003 13:37 -a-
 col1	gif	43 08/26/2003 15:13 -a-
 lampe0	gif	116 09/02/2003 20:03 -a-
 lampe1	gif	116 09/02/2003 20:02 -a-
 ledbar	gif	51 08/26/2003 08:11 -a-
 menu	gif	1,608 09/26/2003 10:22 -a-
 menu0	gif	410 08/25/2003 17:01 -a-
 relay0	gif	116 09/02/2003 20:02 -a-
 relay1	gif	116 09/02/2003 20:03 -a-
 uigtbgrey	gif	118 09/15/2003 21:58 -a-
 uigtabwhite	gif	168 09/15/2003 22:00 -a-
 white	gif	35 08/26/2003 09:47 -a-
 uihcontrol	html	843 09/27/2003 19:34 -a-
 uihdefaults	html	602 09/27/2003 20:07 -a-
 uihio	html	485 09/27/2003 19:41 -a-
 uihnetwork	html	1,000 09/27/2003 19:30 -a-
 uihreboot	html	194 09/27/2003 20:02 -a-
 uihserial	html	672 09/27/2003 19:39 -a-
 uihsnmp	html	451 09/27/2003 19:42 -a-
 uihstatus	html	623 09/27/2003 19:55 -a-
 uihtemp	html	205 09/27/2003 19:43 -a-
 uihupdate	html	1,109 09/27/2003 20:10 -a-

The reason for the graphics being in here is the amount of Bytes used by the HTML pages inside the folder **stuff**.

When compiled the resulting cob file is already 57 Kbytes.

It is not allowed to be bigger than 64 Kbytes. Otherwise it would overwrite another memory page when uploaded to the Barionet. Further details in chapter **Barionet Memory** on page 8.

File *index.html*

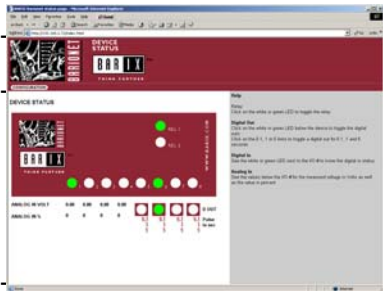
Lets take a look on the index.html file.

```
<html>
<head>
<title>BARIX Barionet status page</title><meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
</head>
<frameset rows="156,*" frameborder="NO" border="0" framespacing="0">
  <frame src="uimenu0.html" name="menu" scrolling="NO" noresize marginwidth=5
marginheight=5 frameborder=0 framespacing=0>>
  <frame src="uifstatus.html" name="config" marginwidth=5 marginheight=5 noresize
frameborder=0 framespacing=0>>
  <frame name="empty" scrolling="no" noresize></frame>
</frameset><noframes></noframes>
</html>
```

It contains three frames:

menu

config



empty (invisible, for responses of CGI commands only)

Frame *menu*

The frame **menu** loads the HTML page **uimenu0.html** which gives the user the navigation to the Barionet product web page (www.barionet.com), the Barix web site (www.barix.com) and to the configuration pages.

Frame *config*

The frame **config** loads a frame set called **uifstatus.html** which is described further below.

Frame *empty*

The frame **empty** is invisible and is used mainly for responses of CGI commands.

Barionet Memory

The Barionet memory is a Flash memory divided into eight pages. Each page has a size of 64 Kbytes. Do not try to load files bigger than 64 KB. This might result in overwritten memory in another page.

Page	Function
XI	Firmware
WEB1	Web application
WEB2	Web help
WEB3	User application
WEB4	User application
WEB5	User application
WEB6	User application
WEB7	User application

To upload pages use the web update in the configuration menu or the following commands (tftp works faster than http, do not forget to exchange the shown IP address with the real address) :

Page XI
firmware

To upload the firmware use this command:

```
tftp -i 192.168.0.222 PUT barionetware.ROM XI
```

Page WEB1
web application

To upload the web application use this command:

```
tftp -i 192.168.0.222 PUT barionetapp.cob WEB1
```

Page WEB2
web help

To upload the web help use this command:

```
tftp -i 192.168.0.222 PUT barionethelp.cob WEB2
```

Page WEB3 .. WEB7
user application

To upload an user applications use these commands:

```
tftp -i 192.168.0.222 PUT user1.cob WEB3
.
.
tftp -i 192.168.0.222 PUT user5.cob WEB7
```

File Structure

The folders help and stuff are compiled separately. The file barionethelp.cob and barionetapp.cob are then uploaded into two different memory pages. Because no folder was created within the folders help and stuff all files contained in the two cob files are seen by the web server as placed in the root.

Root

[-] 1	<DIR>	09/20/2003	01:47	--
barionet	gif	2,872	08/30/2003	14:16
barionetdev	gif	4,382	09/02/2003	19:34
base	gif	2,005	08/04/2003	08:02
base0	gif	1,800	08/25/2003	17:20
ca0	gif	43	07/31/2003	13:37
ca1	gif	43	08/26/2003	15:13
comp0	gif	116	09/02/2003	20:03
comp1	gif	116	09/02/2003	20:02
dbw	gif	51	08/26/2003	08:11
menu	gif	1,508	09/26/2003	10:22
menu0	gif	410	08/25/2003	17:01
relap0	gif	116	09/02/2003	20:02
relap1	gif	116	09/02/2003	20:03
sigabgsep	gif	118	09/15/2003	21:58
sigabwhite	gif	168	09/15/2003	22:00
white	gif	35	08/26/2003	09:47
index	html	550	09/15/2003	23:56
uiconfig	html	552	09/16/2003	00:01
uicontrol	html	4,159	09/27/2003	19:19
uicontrols	html	975	09/27/2003	12:41
uicontrol	html	304	09/17/2003	03:45
uicontrols	html	318	09/17/2003	06:04
uinfo	html	294	09/17/2003	03:16
uiframework	html	304	09/15/2003	22:13
uifreeboot	html	314	09/17/2003	05:53
uifreeboot	html	302	09/16/2003	08:17
uifreeboot	html	298	09/17/2003	03:46
uifreeboot	html	302	09/17/2003	02:24
uifreeboot	html	298	09/17/2003	04:53
uifreeboot	html	302	09/16/2003	08:08
uifreeboot	html	1845	09/27/2003	19:34
uifreeboot	html	602	09/27/2003	20:07
uifreeboot	html	405	09/27/2003	19:41
uifreeboot	html	1,008	09/27/2003	19:38
uifreeboot	html	134	09/27/2003	20:02
uifreeboot	html	672	09/27/2003	19:39
uifreeboot	html	451	09/27/2003	19:42
uifreeboot	html	623	09/27/2003	19:55
uifreeboot	html	205	09/27/2003	19:42
uifreeboot	html	1,109	09/27/2003	20:10
uifreeboot	html	18,377	09/27/2003	19:10
uifreeboot	html	1,762	09/26/2003	10:31
uifreeboot	html	1,186	09/16/2003	00:00
uifreeboot	html	4,895	09/27/2003	20:14
uifreeboot	html	350	09/27/2003	20:21
uifreeboot	html	331	09/27/2003	20:15
uifreeboot	html	829	09/27/2003	22:43
uifreeboot	html	341	09/27/2003	20:22
uifreeboot	html	350	09/27/2003	20:22
uifreeboot	html	331	09/27/2003	20:15
uifreeboot	html	348	09/27/2003	20:22
uifreeboot	html	344	09/27/2003	20:22
uifreeboot	html	6,221	09/27/2003	19:10
uifreeboot	html	3,308	09/27/2003	19:19
uifreeboot	html	5,525	09/27/2003	20:13
uifreeboot	html	4,547	09/17/2003	04:38
uifreeboot	html	4,572	09/27/2003	20:10
uifreeboot	js	1,475	08/30/2003	22:58

Naming Convention

This explains why you should not use file names already used in another page. We recommend the following naming convention for user applications:

- u_name.html for content pages
- uf_name.html for frame sets

Alternately you can create folders within the user folder. These folder are then placed in the root directory.

Sample User application CISCO

To give you a quick start we have prepared a sample user application called cisco.zip (download from www.barionet.com) This application was actually used on a CISCO road show.

Unzip the sample CISCO

Unzip the file to the folder where the compiler is located. The content should look like this:

[.]	<DIR>	09/28/2003	02:22	----
[cisco]	<DIR>	09/28/2003	02:12	-a--
[help]	<DIR>	09/27/2003	20:10	-a--
[stuff]	<DIR>	09/27/2003	22:44	-a--
barionetapp	bat	46	09/27/2003 19:13	-a--
barionethelp	bat	46	09/27/2003 19:12	-a--
cisco	bat	31	09/28/2003 02:13	-a--
web2cob	exe	45,056	08/22/2001 16:55	-a--
mimetypes	ini	557	07/30/2003 15:33	ra--

Upload the sample CISCO

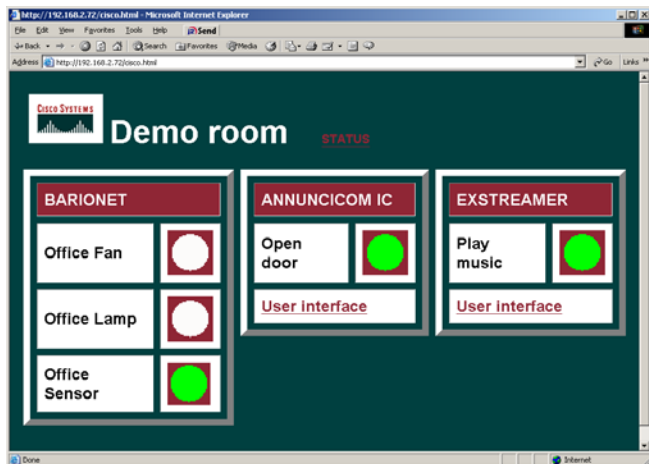
To upload the sample application use this command:

```
ftpp -i 192.168.0.222 PUT cisco.cob WEB3
```

Load the sample CISCO

Type in the address field of your browser:
<http://192.168.0.222/cisco.html>

The page loaded should look as follows :



HTTP Interfacing

Custom web pages can be generated and uploaded by the user, and are then served by the web server built into the Barionet. Outputs can be controlled via CGI commands, and the status of all I/O functions can be used to dynamically create content in HTML/XML pages. With this functionality, one can build an XML or HTML representation of whatever is connected to the Barionet without changing the firmware of the device.

CGI commands

The relays and digital outputs can be controlled via the following cgi command:

<http://ip/rc.cgi?o=x,y>

where x is the output number according to the following table:

x	Function
3:1	Relay 1
3:2	Relay 2
6:1	Digital output 1
6:2	Digital output 2
6:3	Digital output 3
6:4	Digital output 4

and y is the action (what to do):

y	Function
0	Set output to inactive
1	Set output to active
t	Revert (toggle) output
pn	Pulse output for n*100ms

Example: switch on Relay 1:

<http://ip/rc.cgi?o=3:1,1>

HTML tags

Certain HTML tags will be interpreted by the web server in the Barionet and replaced by a string formatted to the users needs using the values of the inputs or outputs. These tags can then be used to simply display values, add the values into XML documents, but also to generate links, file names for graphic symbols etc etc.

To be interpreted, a file must have the following tag in the first 512 bytes (typically, the header):

```
&L(0,"*");
```

The tag may be embedded in a comment, it just needs to be present in the file.

The following tags address the I/O functions:

```
&LIO(1,"format",I/O);  
&LIO(2,"format",I/O,mul,ofs,div,exp);
```

The I/O addresses for this and other functions are given in the Table in the "protocol" section, and consist of a module and a pin number, separated by colon. The format string is a typical "C" style string which is used to generate the output, and must contain an "unsigned" data type with optional digit number for the standard inputs and outputs. The second function type is used to display analog values in a certain range, by first multiplying them with a constant, then adding an offset, then dividing by another constant, and then displaying with a potential decimal point. Here are some samples of how to use these tags:

```
&LIO(1,"%u",6:1);  
this function reads back the status of output I and will yield either "0" or "1"
```

```
&LIO(1,"%04u",4:2);  
this function reads the value of the analog input #2, and will yield a four-digit number in the range 0..1023, with leading zeros
```

```
&LIO(2,"%u.%02u",4:1,500,0,1023,100)  
this function reads analog input #1, multiplies it *500, does not add an offset, then divides by 1023 (so the result is 0..500), and then displays the value as a number in the range 0.00 to 5.00 with two decimal digits after the decimal point. Please note that this function mainly makes sense for the analog inputs.
```

Barionet Development Kit Manual VI.02

© 2003 Barix AG, Zürich, Switzerland.

Portions © Lantronix Inc, Irvine, CA, USA.

All rights reserved.

The newest information about our devices and the latest version of this manual is available via download from our website, www.barix.com.

We explicitly reserve the right to change and improve the product without notice.

All trademarks mentioned or used are belonging to their respective owners. Barix and Barionet are registered trademarks.

Barix AG

Wiesenstrasse 17
8008 Zürich

SWITZERLAND

Phone: +41 | 380 7850

Fax: +41 | 274 2849

Internet

web: <http://www.barix.com>

email: sales@barix.com