

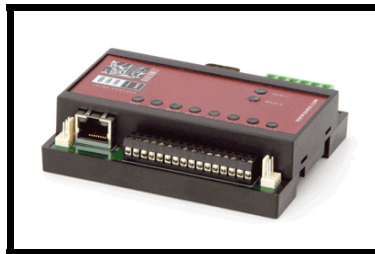
# BARIONET

TM



## BARIONET

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



## Development Kit Manual Technical Reference

Revision 1.46 20041008



THINK FURTHER



---

# Table of Contents

---

<b>Introduction.....</b>	<b>1</b>
Websaver.....	1
Prerequisites.....	1
Applications.....	1

---

<b>Files and Folders .....</b>	<b>2</b>
Unzip the Kit.....	2
ReadMe.txt.....	3
web.bat.....	4
bcl.bat.....	4
web2cob.exe.....	4
mimetype.ini.....	5
Root Folders.....	6
Folder stuff.....	6
Folder BCL.....	8
Tokenizer.....	8
File index.html.....	9
Frame menu.....	9
Frame config.....	9
Frame empty.....	9
Frame set uifstatus.html.....	9

---

<b>Barionet Memory.....</b>	<b>11</b>
-----------------------------	-----------

---

<b>HTTP Interfacing.....</b>	<b>12</b>
CGI commands.....	12
HTML tags.....	13



# 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 Development Kit (further referenced as Dev Kit) manual explains how to compile and upload your own User applications to 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.

### BCL documents

BCL programming is not covered in this manual. Please visit the Barix web site to obtain the BCL programmers manual.

### Prerequisites

- This manual is about the Barionet Development Kit  
Download the Kit from [www.barix.com](http://www.barix.com)
- This manual refers to the Barionet User Manual.  
Download the User Manual from [www.barix.com](http://www.barix.com)
- To edit the HTML pages you will need an editor tool such as Notepad or WordPad, we recommend tools such as Ultra Edit and Dreamweaver.
- To edit the graphics you will need a graphic tool, we recommend tools such as Photoshop and Fireworks.
- A PC, Network Interface Card with configured TCP/IP protocol, a browser such as the 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 hard disk (e. g. barionet, any name will do, the length of the path is not important)
- Download the

**Barionet\_Dev\_Kit\_b146\_20040930.zip**

Development Kit ZIP file form the Barix website.

- Unzip the file to the folder created.
- This will create a folder named:

**Barionet\_Dev\_Kit\_b146\_20040930**

- The content of this folder will look like this:

[..]	<DIR>	10/06/2004 19:20---
[BCL]	<DIR>	09/30/2004 15:31-a--
[stuff]	<DIR>	09/30/2004 15:31-a--
ReadMe	txt	5,049 10/06/2004 09:14-a--
mimetype	ini	573 09/30/2004 19:14-a--
barionet	bat	114 09/30/2004 15:31-a--
bcl	bat	210 09/30/2004 15:31-a--
bn146_20040930	rom	65,536 09/30/2004 15:31-a--
erase	bat	483 09/30/2004 15:31-a--
erasebcl	bat	104 09/30/2004 15:31-a--
web	bat	151 09/30/2004 15:31-a--
tokenizer	exe	70,145 08/26/2004 21:52-a--
empty	cob	63 01/13/2004 13:04-a--
web2cob	exe	45,056 08/22/2001 17:55-a--
SLEEP	EXE	8,816 09/08/2000 15:43-a--

## ReadMe.txt

The "ReadMe.txt" file contains informations about the Dev Kit.

Here an excerpt of the content:

```
=====
Barix Barionet Development Kit
-----
Version: 1.46
Date:    2004-09-30
Size:    325 KB (333,127 bytes)
Zip:     153 KB (157,460 bytes)
=====
```

### CONTENT

-----

This development kit contains all necessary files to create your own web interface and BCL application for the Barionet.

### NEW IN THIS VERSION:

-----

- More options for the serial ports in serial settings.
- NTP server configuration in network settings.
- Syslog server configuration in network settings.
- New PIC firmware 1.63 included.
- Standard BCL application:
  - + Digital I/O Tunnel.

### INSTALLATION

-----

Extract the ZIP file to a folder on your computer.

### UPDATING YOUR BARIONET

-----

Change the working directory to the installation folder and execute the command:

```
barionet <IP address>
```

where <IP address> is the IP address of your Barionet. This will create two COB files barionetweb.cob and barionetbcl.cob and transfers both to your Barionet. You can also use

```
web <IP address>
```

or

```
bcl <IP address>
```

to update each part of the application separately.

### Recommendation:

Barix recommends strongly to read this informations before running any of the included batch files.

### web.bat

The file web.bat is a batch file that starts the compilation of the folder **stuff** which contains the web UI, web help, graphics, MIB, MIB.zip, PIC firmware and version file.

Content:

```
@echo off
echo # Creating Web COB file...
web2cob /o barionetweb.cob /d stuff
echo # Transferring to Barionet...
tftp -i %I put barionetweb.cob WEBI.
```

### bcl.bat

The file bcl.bat is a batch file that starts the compilation of the folder BCL which contains the BCL application and its web setup.

Content:

```
echo # Tokenizing BCL...
cd BCL
..\tokenizer barionet.bas
cd ..
echo # Creating BCL COB file...
web2cob /o barionetbcl.cob /d BCL
echo # Transferring to Barionet...
tftp -i %I put barionetbcl.cob WEB4
```

### 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 in order to include the correct mime type.

Content:

```
mib text/plain
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
jff 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 two root folders “stuff” and “BCL”.

### Folder *stuff*

The folder **stuff** contains the web UI, web help, graphics, MIB, MIB.zip, PIC firmware and version file.

Content:

 Barionet	MIB	2,109	08/26/2004	21:52-a-
 Barionet_MIB	zip	862	09/24/2004	08:04-a-
 d0	gif	116	09/22/2003	13:43-a-
 d1	gif	116	09/22/2003	13:43-a-
 index	html	572	10/01/2004	00:22-a-
 picfw	hex	10,083	09/10/2004	11:43-a-
 picfw	ver	58	09/10/2004	11:44-a-
 uiconfig	html	574	10/01/2004	00:22-a-
 uicontrol	html	6,833	09/30/2004	23:16-a-
 uidefaults	html	952	10/01/2004	00:48-a-
 uidefaults2	html	311	10/01/2004	00:48-a-
 uifappsetup1	html	306	07/19/2004	01:58-a-
 uicontrol	html	304	09/17/2003	03:45-a-
 uidefaults	html	318	09/17/2003	06:04-a-
 uifio	html	294	09/17/2003	03:16-a-
 uifnetwork	html	304	10/01/2004	00:21-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-
 uiftime	html	298	09/23/2004	21:20-a-
 uifupdate	html	302	09/16/2003	00:08-a-
 uigconf	gif	507	09/30/2004	23:47-a-
 uigd	gif	4,382	09/22/2003	13:43-a-
 uigg	gif	43	09/22/2003	13:43-a-
 uigl	gif	131	09/20/2004	08:19-a-
 uiglogo	gif	2,478	09/30/2004	23:38-a-
 uigmconf	gif	1,943	07/13/2004	16:49-a-
 uigmstat	gif	410	09/22/2003	13:43-a-
 uigs	gif	138	09/20/2004	08:15-a-
 uigstat	gif	778	10/01/2004	00:05-a-

 uicontrol	html	2,074	09/30/2004	19:33-a--
 uihdefaults	html	734	09/22/2003	13:43-a--
 uihio	html	1,030	09/30/2004	19:59-a--
 uihnetwork	html	1,276	09/30/2004	19:53-a--
 uihreboot	html	243	09/22/2003	13:43-a--
 uihserial	html	1,228	09/30/2004	19:47-a--
 uihsnmp	html	1,175	09/30/2004	19:36-a--
 uihstatus	html	795	10/01/2004	00:26-a--
 uihtemp	html	450	09/30/2004	16:44-a--
 uihtime	html	666	09/30/2004	19:40-a--
 uihupdate	html	1,374	09/30/2004	18:52-a--
 uiio	html	7,743	09/30/2004	23:16-a--
 uimenu	html	1,748	10/01/2004	00:15-a--
 uimenu0	html	964	10/01/2004	00:19-a--
 uinetwork	html	3,917	09/30/2004	23:16-a--
 uireboot	html	804	10/01/2004	00:47-a--
 uireboot2	html	458	10/01/2004	00:47-a--
 uireboot3	html	225	10/01/2004	00:47-a--
 uisaved	html	202	09/30/2004	15:50-a--
 uiserial	html	6,779	09/30/2004	23:16-a--
 uisnmp	html	5,262	09/30/2004	23:16-a--
 uistatus	html	7,570	10/01/2004	00:45-a--
 uistatus2	html	7,570	10/01/2004	00:45-a--
 uistatusn	html	7,520	10/01/2004	00:45-a--
 uitemp	html	2,509	09/30/2004	23:16-a--
 uitime	html	6,009	09/30/2004	23:16-a--
 uiupdate	html	1,343	09/30/2004	23:16-a--
 updateex	html	787	08/26/2003	21:06-a--
 util	js	1,649	05/05/2004	19:29-a--

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

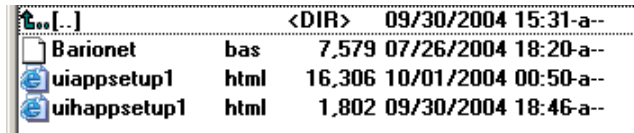
To distinguish original UI files from User files we have used the following naming convention (you are free to use your own):

- 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 *BCL***

The folder BCL contains the BCL application and its web setup.



Icon	Name	Type	Size	Date	Time
<DIR>	<DIR>			09/30/2004	15:31-a-
Folder icon	Barionet	bas	7,579	07/26/2004	18:20-a-
HTML icon	uiappsetup1	html	16,306	10/01/2004	00:50-a-
HTML icon	uihappsetup1	html	1,802	09/30/2004	18:46-a-

The reason for the web setup HTML pages of this particular BCL application being inside the folder **BCL** is the following fact. Overwriting the BCL application cob file (usually in WEB4) would automatically also overwrite the web setup pages and therefore simplifies the process and prevents from having the “wrong” UI for the loaded BCL application.

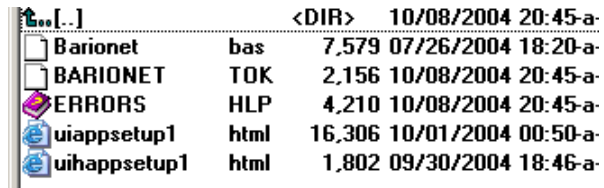
**Tokenizer**

The file bcl.bat contains the following command lines :

```
cd BCL
..\tokenizer barionet.bas
cd ..
```

The second line calls the tokenizer to create a TOK file. Such a file is much smaller in size and save considerably the memory needed inside the Barionet. See next picture for comparison.

It also creates a file called ERRORS.HLP if it does not exist in the folder. This help file will, as its name implies, help you with clear text messages inside the Syslog messages when errors are being displayed.



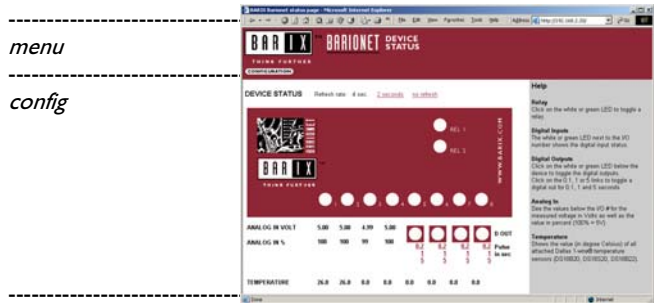
Icon	Name	Type	Size	Date	Time
<DIR>	<DIR>			10/08/2004	20:45-a-
Folder icon	Barionet	bas	7,579	07/26/2004	18:20-a-
Folder icon	BARIONET	TOK	2,156	10/08/2004	20:45-a-
Help icon	ERRORS	HLP	4,210	10/08/2004	20:45-a-
HTML icon	uiappsetup1	html	16,306	10/01/2004	00:50-a-
HTML icon	uihappsetup1	html	1,802	09/30/2004	18:46-a-

## 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="1|6,*0" 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 src="UntitledFrame-3" name="empty" scrolling="no" noresize></frame>
</frameset><noframes></noframes>
</html>
```

It contains three frames:



*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 configuration pages, the mainBarix web site ([www.barix.com](http://www.barix.com)), and to the Barix support web page (<http://www.barix.com/en/support/support.php>).

## 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.

## Frame set *uifstatus.html*

The frame set **uifstatus.html** contains two frames:

## Barionet Development Kit Manual VI.46

```
<html>
<frameset cols="650,*" frameborder="NO" border="0" framespacing="0">
  <iframe src="uistatus.html" marginwidth=5 marginheight=5 frameborder=0 framespacing=0>
  <iframe src="uihstatus.html" marginwidth=5 marginheight=5 frameborder=3 framespacing=5>
</frameset></noframes></noframes>
</html>
```

The left frame loads the HTML page `uistatus.html` (the visualization of the Barionet).

The right frame loads the HTML page `uihstatus.html` (the help for the visualization of the Barionet)

Take time to look into this two files.

Check the first line with the DHTML tag `&L(0,"*");`

```
<html>&L(0,"*");
<head><meta http-equiv="refresh" content="2; url=uistatus.html"></head>
<body bgcolor=#FFFFFF link=#8F2635 vlink=#8F2635><br>
<font size="3" face="Arial, Helvetica, sans-serif"><b>DEVICE STATUS</b></font>
<div id="Layer1" style="position:absolute; left:10px; top:64px; width:602px; height:104px; z-index:2"></div>
<div id="Layer2" style="position:absolute; left:170px; top:254px; width:29px; height:22px; z-index:2"></div>
<div id="Layer3" style="position:absolute; left:220px; top:254px; width:29px; height:22px; z-index:2"></div>
<div id="Layer4" style="position:absolute; left:271px; top:254px; width:29px; height:22px; z-index:2"></div>
<div id="Layer5" style="position:absolute; left:322px; top:254px; width:29px; height:22px; z-index:2"></div>
<div id="Layer6" style="position:absolute; left:372px; top:254px; width:29px; height:22px; z-index:2"></div>
<div id="Layer7" style="position:absolute; left:423px; top:254px; width:29px; height:22px; z-index:2"></div>
<div id="Layer8" style="position:absolute; left:474px; top:254px; width:29px; height:22px; z-index:2"></div>
<div id="Layer9" style="position:absolute; left:524px; top:254px; width:29px; height:22px; z-index:2"></div>
<div id="Layer10" style="position:absolute; left:428px; top:84px; width:29px; height:22px; z-index:2"><A href=/rc.cgi?o=3:1,t target="empty"></A></div>
<div id="Layer11" style="position:absolute; left:428px; top:132px; width:29px; height:22px; z-index:2"><A href=/rc.cgi?o=3:2,t target="empty"></A></div>
<div id="Layer15" style="position:absolute; left:372px; top:330px; width:39px; height:22px; z-index:2">
  <div align="right"><b><font size="2" face="Arial, Helvetica, sans-serif">
    <A href=/rc.cgi?o=6:1,t target="empty"></A><br>
    <A href=/rc.cgi?o=6:1,p1 target="empty">0.1</A><br>
    <A href=/rc.cgi?o=6:1,p10 target="empty">1</A><br>
    <A href=/rc.cgi?o=6:1,p50 target="empty">5</A></font></b></div>
</div>
<div id="Layer17" style="position:absolute; left:423px; top:330px; width:39px; height:22px; z-index:2">
  <div align="right"><b><font size="2" face="Arial, Helvetica, sans-serif">
    <A href=/rc.cgi?o=6:2,t target="empty"></A><br>
    <A href=/rc.cgi?o=6:2,p1 target="empty">0.1</A><br>
    <A href=/rc.cgi?o=6:2,p10 target="empty">1</A><br>
    <A href=/rc.cgi?o=6:2,p50 target="empty">5</A></font></b></div>
```

For information regarding CGI commands and HTML tags used refer to the chapter on page .

## 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 cob files bigger than 64 KB. This might result in overwritten memory in another page. If splitting is not possible just consider the following page as used.

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

Do not try to load a cob file containing a single file being bigger than 64 KB. Unpredictable erroneous behaviour will be the result.

To upload files into the Barix Barionet the TFTP protocol can be used but a short pause of approximately 3 seconds after each upload has to be allowed for in order for the Barix Barionet to store the file internally.

### WARNING:

If this timing is not obeyed, wrong or corrupt files are uploaded to wrong destinations using TFTP unpredictable behaviour can be observed, which might render a Barix Barionet unusable.

**Therefore Barix recommends the supplied batch files and to leave the usage of TFTP to advanced programmers only.**

## 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 1 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.46

© 2004 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](http://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

Seefeldstrasse 303  
8008 Zürich  
SWITZERLAND

Phone: +41 43 433 22 11  
Fax: +41 1 274 28 49

Internet

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

email: [sales@barix.com](mailto:sales@barix.com)