BRUTUS

Installation Guide for VxWorks

5300 California Avenue • Irvine, CA 92617 USA • Phone: (949) 936-5000 • Fax: (949) 936-9265

REVISION HISTORY

Revision Number	*Date*	*By*	*Change Description*
1.0	4/22/04	J. Jordan	Created from Linux version
1.1	6/24/04	J. Jordan	Added section for hard disk formatting
1.2	6/25/2004	L. Liu	Added a recording section
1.3	7/8/2004	L. Liu	Clarify the recording.
3.0	9/13/2004	L. Liu	Modified for release 3
3.1	2/22/2005	D. Erickson	Modified for build system changes
3.2	7/28/05	R. Lewis	Updated boot ROM flashing instructions.
3.3	12/20/05	R. Lewis	Added information on using scripts.
3.4	1/23/06	R. Lewis	Corrected CFE instructions for flashing. Added VxWorks v6 instructions.
3.5	2/8/06	R. Lewis	Added missing VxWorks 6 environment variable setting.
3.6	3/27/06	R. Lewis	Miscellaneous edit corrections.
3.7	8/3/06	R Lewis	Added note about stopping VxWorks 5.5 registry when using VxWorks 6.x
3.8	11/8/06	R. Lewis	Corrected use of alternate startup commands (alt_go, disk_go, etc.)
3.9	11/28/06	R. Lewis	Added information on utilities and tests. Added explanation of Settop API run-time parameters.
4.0	8/12/08	R. Lewis	Removed platform setup information and added Nexus support.
4.1	6/5/09	R. Lewis	Fixed mistake in subst usage.

TABLE OF CONTENTS

TABLE OF CONTENTS **3**

INTRODUCTION **4**

PREREQUISITES **4**

USING THE BINARY DISTRIBUTION **5**

BUILDING THE REFERENCE SOFTWARE **6**

Use the Release Notes *6*
Nexus vs SettopAPI *6*
Unzipping the Source Code *6*
Building Brutus *7*
Installing Brutus configuration and other supporting files *8*
Building Utilities, Tests, and Examples *8*
Reference Software Compile-Time Options (SettopAPI/Non-NEXUS Platforms) *9*
Reference Software Compile-Time Options (NEXUS Platforms) *10*

RUNNING THE REFERENCE SOFTWARE **13**

BRUTUS APPLICATION 13
Downloading and running the Brutus application *13*
Downloading and running the Brutus application *13*
Customizations *13*
Formatting the Set-Top's Hard Disk Drive for PVR *14*
Recording to Hard Disk *14*
Alternate start up options *15*
NEXUS EXAMPLES 16
SETTOP API UTILITIES, TESTS, AND EXAMPLES 16
Starting up the applications *17*

RUNTIME OPTIONS **17**

Settop API Runtime Options when using VxWorks *17*
Nexus Runtime Options when using VxWorks *17*

INTRODUCTION

This document comes as part of the Broadcom Settop Reference Software Release. Refer to the 'Reference Software User Guide' for a high-level overview of the Reference Software. Refer to the document 'Reference Platform Setup Guide for VxWorks' for instructions on how to set up the build environment and configure the platform with the VxWorks bootloader and Operating System.

This installation guide contains specific instructions on how to configure, build, and install the Brutus application as part of the VxWorks Reference Software. By building and running Brutus, you will be exercising the entire Reference Software stack. This document also describes how to build Brutus for both SettopAPI (non-Nexus) and Nexus platforms. The instructions are common for all the platforms, but Nexus-specific changes are explicitly noted throughout this document.

This document contains instructions for building and installing the Brutus application under VxWorks V6 (Workbench 2.3). This document makes the assumption that the reader is familiar with the Workbench IDE and the VxWorks RTOS. This document also assumes you are using a Win32-based PC as the *host* system and a Set-Top-Box as the *target* device. Broadcom Corporation provides the following software components for this release:

- Boot ROM executable
- VxWorks run-time executable image for your target platform
- Pre-built reference Software executables
- Reference Software source code

The Boot ROM and the VxWorks run-time image are collectively referred to as the Board Support Package (or BSP). The sources to the BSP are released as a separate package. Please see the document 'Reference Platform Installation Guide for VxWorks' for details on the steps for loading the Boot ROM, VxWorks run-time image, and Brutus application.

This guide assumes you are using the Workbench Application environment as described in the 'Reference Platform Setup Installation Guide for VxWorks'.

PREREQUISITES

The reader should have the following:

- A Broadcom Reference Board.
    - The Reference Software is not guaranteed to run on other systems. It may require significant modification.
- A PC on which you have installed Workbench 2.3/VxWorks 6.1 and VxWorks 6.1 BSPs/Drivers for MIPS.
- Knowledge of how to connect the Set-Top and PC to the same Ethernet network (using static IP addresses).
- Connected a serial cable from the Set-Top to the PC.
- An FTP or TFTP server running on the PC that the Set-Top can use.
- Know how to run HyperTerm or similar terminal emulation software.

USING THE BINARY DISTRIBUTION

The Reference Software includes a binary distribution. This is a set of pre-built binaries (Brutus application only) that can be used to bring up a Reference Board without building the binaries from the sources. The binary distribution can also be used to verify that code built from the source is performing as it should.

If the Reference Software is being used for demonstration purposes then skip over the next section detailing how to build the code.

The reference software binaries are delivered as a zip file. You can unzip it using any Windows or command line zip utility.

1. Follow the information in the ‘Reference Platform Setup Guide for VxWorks’ for flashing the boot loader.
2. Set FTP home directory to point to the appropriate "binaries" directory.
3. Set Target Server Connection's "Kernel Image" to the "vxWorks" file in appropriate "binaries" directory.
4. Set Target Server Connection's "Advanced Target Server Options" to use directory "tgtsvr".
5. Follow the information in this "Brutus Installation Guide for VxWorks" for running the Brutus application.

Use the "Brutus Usage Guide" (STB_Brutus-SWUM300.pdf) for assistance in running the Brutus application.

Note: the binaries were built with limited build options. Any option that requires a license was not enabled.

BUILDING THE REFERENCE SOFTWARE

Use the Release Notes

Each release includes a set of Release Notes that define specific items of interest for a given release. Information in the Release Notes is more often newer than instructions in this document. Please read the Release Notes before attempting to build or run any of the Reference Software.

Nexus vs SettopAPI

In the past, the Reference Software for Settop Box supported SettopAPI middleware. SettopAPI middleware was designed for Settop Box applications, providing a high level API for creating Settop Box applications. After a while in the marketplace, the SettopAPI middleware was redesigned to address the next generation of Settop Box applications. Nexus middleware is the result of that redesign effort. Please see Nexus Design Document for details on the Nexus architecture.

To support the large set of applications written for to use SettopAPI middleware, we have created SettopAPI Shim layer for Nexus platforms. The SettopAPI Shim layer translates SettopAPI calls to Nexus calls, allowing SettopAPI applications to run on Nexus platform. Brutus application is an example of a SettopAPI application using SettopAPI Shim layer to run on Nexus platform.

Unzipping the Source Code

The reference software source is delivered as a zip file. You can unzip it using any Windows or command line zip utility:

When you unpack your source, you'll see three (or more) sub-directories that contain source code:

- BSEAV – Reference Software drivers, libraries and applications
- magnum – Porting interface for magnum platforms and some application utility code
- rockford – System code for Broadcom platforms
- nexus -- Application interface code for new settop environment (not on every platform)

These instructions assume you know where the BSEAV subdirectory is and can get to it.

Building Brutus

The VxWorks installation process should have created a "VxWorks Development Shell" icon on your desktop. Double click this icon to create a Command Prompt window (DOS box). If not, you can create a command prompt window by selecting "Run" in your "Start" menu and entering:

```
C:\WindRiver\wrenv.exe -p vxworks-6.1
```

The DOS command shell has a limitation in regards to the length of a command line. When the length is exceeded the command line is truncated before it is executed. This leads to strange build errors. Because the Brutus build is rooted, the full directory path is included when referencing any particular file. This leads to long file names that can over-run the command line length limit when referencing a large number of files in a single build instruction. This is particularly noticeable when the source is placed in a "deep" directory path. In order to minimize the path length, the DOS 'subst' command should be used to create a virtual drive. For example, if the source was extracted to: 'C:\broadcom\reference_code\bcm97325\latest' then the command would be:

```
C:\> subst j: c:\broadcom\reference_code\bcm97325\latest

C:\> j:

J:\>
```

This creates the virtual drive 'J' (be sure to use an unused drive letter). You should "change drive" to that new virtual drive before building.

The DOS window opened above has all the necessary environment variables set to build a VxWorks project but not those necessary to build the Brutus application. These need to be set manually. Change directory to the location of the build makefiles. Choose the PLATFORM variable based on the reference platform you are targeting. Set the SYSTEM variable to designate this as a VxWorks build. Set the BCHP_VER variable for the version of the main chip (i.e. A0, B0, etc.).

Here's an example for the bcm97325/B0 reference board:

```
J:\> cd BSEAV\app\brutus\build
J:\BSEAV\app\brutus\build> set PLATFORM=97325
J:\BSEAV\app\brutus\build> set SYSTEM=vxworks
J:\BSEAV\app\brutus\build> set BCHP_VER=B0
J:\BSEAV\app\brutus\build> set vxWorksVersion=6
```

In order to build for the Nexus API two additional environment variables must be set:

```
J:\BSEAV\app\brutus\build> set BUILD_SYSTEM=nexus
J:\BSEAV\app\brutus\build> set OS=vxworks
```

You can then use the 'make' command to build the reference code:

```
J:\BSEAV\app\brutus\build> make clean
J:\BSEAV\app\brutus\build> make
```

This will now build the Settop API or Nexus, including the porting interface, all libraries required by Brutus along with the Brutus application. Building everything can take a awhile...it's coffee time! The resulting executables (*libsettop.out* for Settop API, *libs.out* for Nexus, and *brutus.out*) will be found in the "BSEAV\bin" directory. These are files that are downloaded using the Workbench application.

The build process will also (optionally) make other libraries that are combined to make the *libsettop* and *libs* library files. These libraries are used when building stand-alone examples. Use the ".a" files when using the linker application ('cc' or 'ln') and the ".out" files when using the loader application ('ld').

You may also need to define environment variables to enable compile-time options. For instance, some audio decoders are not built by default because they require special licensing. See the *Build Options* section later in this document for more information.

Installing Brutus configuration and other supporting files

Brutus requires several files to be placed in the target server file system directory. These files control the run-time configuration and also provide other things such as logo bitmaps, pre-rendered fonts, etc.

The required files include:

- ❑ brutus.cfg
- ❑ bcmlogo.png
- ❑ channels.txt
- ❑ programguide.txt
- ❑ fonts sub-directory
- ❑ images sub-directory (and its sub-directories)

A 'makefile' target is provided for copying these files from the source tree to the target server file system directory. The PLATFORM and SYSTEM environment variables should already be set from above. The target server directory was specified in the setup process for the Target Server. You must use this same value in this step. Here's an example for installation:

```
J:\> cd BSEAV\app\brutus\build
J:\BSEAV\app\brutus\build> make install_files "TSFS_DIR=c:/tgtsvr/brutus"
```

This will copy the files to your target server file system directory. When this is completed successfully, you should see a banner like the following:

```
"*******************************************************"
"*"
"* Successful install of Brutus files to c:/tgtsvr/brutus."
"*"
"*******************************************************"
```

Building Utilities, Tests, and Examples

The Reference Software includes examples (Nexus) and examples, tests, and utilities (Settop API) as part of the documentation on how to build an application that utilizes the reference libraries.

Note: whereas the utilities in SettopAPI needed to be loaded with the libsettop.out file, utilities in Nexus are built to include the required Nexus and Magnum functions (i.e. load just the example).

Reference Software Compile-Time Options (SettopAPI/Non-NEXUS Platforms)

The SettopAPI and Brutus support various compile time options. Any required or commonly used options are documented in the release notes. However, there are many minor or optional settings which change frequently. The only way to know the complete list of compile time options is to search the makefiles.

Unless noted, all compile time options are simply on or off. This is usually documented as export OPTION=y. Some are documented throughout this guide. This list is a catalog of all options which are likely to be of interest.

Option	***Purpose***
PLATFORM	Selects the board to build for. See release notes.
BCHP_VER	Selects the chip revision of the main chip to build for. See release notes.
DEBUG	Set to yes or no. If yes, DBG interface is enabled. If no, DBG interface is compiled out. Defaults to yes.
ARCH	Set the compiler. See BSEAV/api/build/tools.mak for options.

Table 1: Common SettopAPI Options

Option	***Purpose***
ASF_SUPPORT	Compile ASF parser support. Code is only available for verified ASF licensees.
AVI_SUPPORT	AVI file format support. Needed for DivX support.
SUBTITLE_SUPPORT	Support for DivX subtitles.
RAP_WMA_SUPPORT	Compile WMA audio codec support. Code is only available for verified WMA licensees.
RAP_DDP_SUPPORT	Compile DDP audio codec support. Code is only available for verified DDP licensees.
RAP_AC3_SUPPORT	Compile AC3 audio codec support. Code is only available for verified AC3 licensees.
BHDM_DVO_ENABLE_ SUPPORT	Enable DVO output in HDMI porting interface.
STATIC_SETTOPAPI	Build the SettopAPI as a static library only.
BPROFILE_SUPPORT	SettopAPI's profiling engine.
PLAYBACK_IP_SUPPORT	Enable IP support in Brutus and SettopAPI.
MACROVISION_SUPPORT	Enable Macrovision™ in VDC. Defaults off. Code is only available for verified Macrovision licensees.
BCRYPTO_xxx_SUPPORT	Enable various security modules. See BSEAV/api/build/magnum for options.

Table 2: Various SettopAPI Settings

Option	***Purpose***
AUDIO_SUPPORT	Enable MP3 support in Brutus. Defaults on.
PVR_SUPPORT	Enable Playback/Record support in Brutus. Defaults on.
STATIC_BRUTUS	Build and link Brutus as a static binary (no shared libraries, including no libstdc++ or libc).
PLAYBACK_IP_SUPPORT	Enables IP support in Brutus and SettopAPI.
LIVEMEDIA_SUPPORT	Enables IP streaming support in Brutus.
DRM_SUPPORT	Compile Microsoft Digital Rights Management (DRM) 10 support. Code is only available for DRM licensees.
DIVX_DRM_SUPPORT	Compile DivX Digital Rights Management (DRM) support. Code is only available for DivX licensees.
POWERSTANDBY_SUPPORT	Set to n to disable low-power standby supports. Defaults to y on supported platforms.
DSG_SUPPORT	Enable DOCSIS Settop Gateway (DSG) feature in Brutus.
NETACCEL_SUPPORT	Enable the usage of NetAccel Module for receiving IP_UDP, IP_RTP, & IP_HTTP channels

Table 3: Various Brutus Settings

Reference Software Compile-Time Options (NEXUS Platforms)

The Broadcom Reference software supports various compile time options. Any required or commonly used options are documented in the release notes. However, there are many minor or optional settings which change frequently. The only way to know the complete list of compile time options is to search the makefiles.

Unless noted, all compile-time options are simply on or off. Broadcom usually documents this as export OPTION=y. Some are documented throughout this guide. This list is a catalog of all options which are likely to be of interest.

Option	***Purpose***
PLATFORM	Selects the board to build for. See release notes.
BCHP_VER	Selects the chip revision of the main chip to build for. See release notes.
DEBUG	Set to yes or no. If yes, DBG interface is enabled. If no, DBG interface is compiled out. Defaults to yes.
ARCH	Set the compiler. See BSEAV/api/build/tools.mak for options.

Table 4: Common SettopAPI Options

Option	***Purpose***
MEDIA_ASF_SUPPORT	Compile ASF parser support. Code is only available for verified ASF licensees.

MEDIA_AVI_SUPPORT	AVI file format support. Needed for DivX support.
SUBTITLE_SUPPORT	Support for DivX subtitles.
RAP_AC3_SUPPORT	Compile AC3 audio codec support. Code is only available for verified AC3 licensees.
RAP_DDP_SUPPORT	Compile AC3+ audio codec support. Code is only available for verified DDP licensees.
RAP_MPEG_SUPPORT	Compile MPEG audio codec support.
RAP_WMA_SUPPORT	Compile WMA audio codec support. Code is only available for verified WMA licensees.
RAP_WMAPRO_SUPPORT	Compile WMA Pro audio codec support. Code is only available for verified WMA licensees.
RAP_AACSBR_SUPPORT	Compile AAC audio codec support.
RAP_SRC_SUPPORT	Compile MPEG LSF/QSF audio codec support.
RAP_DDP_TO_AC3_SUPPORT	Compile Dolby Digital Plus to Dolby Digital conversion support.
BHDM_DVO_ENABLE_ SUPPORT	Enable DVO output in HDMI porting interface.
HDMO_DDP_PASSTHROUGH	Enable DDP pass-through on HDMI output. By default DDP is converted to AC3 on HDMI port.
PLAYBACK_IP_SUPPORT	Enable IP support in Brutus and SettopAPI.
MACROVISION_SUPPORT	Enable Macrovision™ in VDC. Defaults off. Code is only available for verified Macrovision licensees.
BCRYPTO_xxx_SUPPORT	Enable various security modules. See BSEAV/api/build/magnum for options.

Table 5: Various SettopAPI Settings

Option	***Purpose***
AUDIO_SUPPORT	Enable MP3 support in Brutus. Defaults on.
PVR_SUPPORT	Enable Playback/Record support in Brutus. Defaults on.
STATIC_BRUTUS	Build and link Brutus as a static binary (no shared libraries, including no libstdc++ or libc).
PLAYBACK_IP_SUPPORT	Enables IP support in Brutus and SettopAPI.
LIVEMEDIA_SUPPORT	Enables IP streaming support in Brutus.
MSDRM_PD_SUPPORT	Compile Microsoft Digital Rights Management (DRM) 10 support. For Portable Devices. Code is only available for DRM licensees.
MSDRM_ND_SUPPORT	Compile Microsoft Digital Rights Management (DRM) 10 support. For Network Devices. Code is only available for DRM licensees.
DIVX_DRM_SUPPORT	Compile DivX Digital Rights Management (DRM) support. Code is only available for DivX licensees.
POWERSTANDBY_SUPPORT	Set to n to disable low-power standby supports. Defaults to y on supported platforms.
NETACCEL_SUPPORT	Enable the usage of NetAccel Module for receiving IP_UDP, IP_RTP, & IP_HTTP channels

Table 6: Various Brutus Settings

RUNNING THE REFERENCE SOFTWARE

BRUTUS APPLICATION

Downloading and running the Brutus application

In order to run the Brutus application it must be downloaded to the target device using the Workbench Application. This requires that the vxWorks run-time image has been loaded and is running and that the target server is connected.

Downloading and running the Brutus application

Use the Workbench IDE to download applications. Perform the following steps to download the Brutus application.

- ❑ Select the running target server name from the Target Manager window.
- ❑ Right click and select "Download" option (looks like a target with an arrow pointing to it)
- ❑ Navigate to the BSEAV/bin directory and select either *libsettop.out* (SettopAPI) or *libs.out* (Nexus) for download. Hit the "OK" button.
- ❑ Repeat the above two steps but instead download the *brutus.out* file.
- ❑ Create a target shell by clicking on the "Host Shell" icon under the "Target" menu (looks like an arrow pointing to a lowercase 'i').
- ❑ Run the application by typing `"go"` on the target shell (at the "->" prompt). Brutus should start.
- ❑ Please read the Workbench documentation for more detailed information about the environment.

The Brutus application displays output on both the target shell and serial port. If the "--exec" option is used ("--exec" option is automatically selected with the "go" command) then the application will display a "Brutus>" prompt on the serial port once it completes the initialization process. This prompt can be used to issue commands. See the "Brutus Usage Guide" for more information on the commands supported.

Note: If the target server is disconnected you will get the error message: "WTX Exchange Error Transport disconnect" when you try to download the library file. In this case, just click on the "ok" button, reconnect to the target server, and download the object again. Also, the Host Shell may need to be restarted after resetting the board/target server. Just close the Host Shell window and open a new one by selecting "Host Shell" under the "Target" menu.

Customizations

After installing on the Set-Top, you'll need to customize your Brutus configuration in the target server directory.

- ❑ Edit `'channels.txt'` and `'programguide.txt'` for your head-end. See the "Brutus Usage Guide" (Channel Map section) for more help.
- ❑ Edit `'brutus.cfg'` with any special options. See the "Brutus Usage Guide" (Configuration section) for more help.

Note: the default 'brutus.cfg' file uses the target server for audio, fonts, pictures, scripts and image files and uses the disk (/ata0) for videos. The target server files can be copied to the disk for stand-alone testing (use "disk_go" instead of "go" for start up command) but the "videos" directory should not be set to use the target server as it is too slow to support typical record and playback operations.

Formatting the Set-Top's Hard Disk Drive for PVR

In order to use the Set-Top for Personal Video Recorder (PVR) functions, the Serial ATA driver and the 'dosFs' file system support is required. The boot loader and VxWorks image supplied with the Reference Software for VxWorks includes support for the hard disk drive (SATA and dosFs driver). Note that the 'dosFs' supplied with the BSP is a modified version with additional features to support PVR operations (direct I/O, large disks, larger blocking factors, etc.).

This section explains how to initialize and mount the dosFS from an ATA disk.

1. Mount dosFS from an ATA hard disk.

```
-> usrAtaConfig(0,0,"/ata0")
```

2. If the disk is not formatted, type the following command to format the disk with a dosFS filesystem:

```
-> dosFsVolFormat("/ata0", 0x312, 8192)
```

Note: this will erase any information previously residing on the drive.
Also note that this file system is **not** compatible with the Windows DOS file systems and cannot be mounted as a usable disk on your PC.

For convenience, using the "formatDisk()" shell command will initialize the primary disk using the commands listed above and create the "videos" directory listed below.

3. Create the "videos" directory. This is the directory where the Brutus application will create all its recording files (specified in the `brutus.cfg` file). Create the directory with:

```
-> mkdir("/ata0/videos")
```

You can list the files or directories on the disk using the ll(directory) command:

```
-> ll("/ata0")          or          -> ll("/ata0/videos")
```

Optionally, you can use the copy(from, to) command to copy file(s) to/from PC to the hard disk.

```
-> copy ("c:/sourcefile.txt", "/ata0/destfile.txt")
```

This will copy the file from the PC to the hard disk through the FTP protocol. You need to have the FTP server running on your PC. You can use the "cp" command to use the target server.

Recording to Hard Disk

In order to record a stream, press the "Record" button on the remote or enter the Brutus command "goto(record)". In recording screen, enter the title of the movie in the name screen, followed by save button. Then proceed to the record button.

Note that when recording using `brutus.cfg` (after reloading brutus), the application assumes the recorded file name to be brutus1 and creates three files: `brutus1.info` stores the file title and PID values; `brutus1.nav` is the index file used for rewind/fast forwarding; and `brutus1.mpg` is the data file. All the files are located in "/ata0/videos" directory. You can look at the files using the following command:

```
-> ll ("/ata0/videos")
```

The next file recorded will be named as 'brutus2', and so on.

The recording can be stopped by pressing the “stop” button on the remote or entering the Brutus command “stop”.

Alternate start up options

Starting Brutus with the “go” command runs the application with the following (five) arguments:

- -cfg /tgtsvr/brutus.cfg
- -ch /tgtsvr/channels.txt
- --exec

Starting up with different start up parameters

In order to utilize some of the optional command line arguments of the Brutus application, a set of alternate start up functions are available (used instead of the “go” function).

Command	Parameters	Description
scripts	none	Same as “go” function but without the “--exec” parameter. (see below)
alt_go	arg5,- arg10 (6 arguments)	Args are used in addition to the standard arguments specified by the “go” function but does not include the “--exec” argument.
disk_go	arg5 – arg10 (6 arguments)	This is used to specify a configuration file (“-cfg /ata0/brutus.cfg”) and channels file (“-ch /ata0/channels.txt”) located on the hard drive. Args are used in addition to the listed “-cfg” and “-ch” parameters.
vxworks_main	arg1 – arg10 (10 arguments)	Arguments are passed directly to “main”. Note: this function does not prepare the disk for recording. If disk recording is desired, use the “setupDisk()” function prior to using the “vxworks_main” function.

Examples

go()

Run with default arguments.
Note: the “go” function is equivalent to:
vxworks_main(“-cfg”,“/tgtsvr/brutus.cfg”,“-ch”,“/tgtsvr/channels.txt”,”--exec”)

alt_go(“-scr”,”/tgtsvr/tests”)

Run with additional arguments.
Note: the “alt_go” function (with the arguments listed above) is equivalent to:
vxworks_main(“-cfg”,“/tgtsvr/brutus.cfg”,“-ch”,“/tgtsvr/channels.txt”,“-scr”,”/tgtsvr/tests.txt”)

disk_go("--exec")

Run using configuration and channels files located on the disk and including the "--exec" option.
Note: the "disk_go" function (with the argument listed above) is equivalent to:
  vxworks_main("-cfg","/ata0/brutus.cfg","-ch","/ata0/channels.txt","--exec")

vxworks_main("--help")

List all the run time options.

vxworks_main("--help-cfg")

List all the configuration options.

Using scripts

The "--exec" option enables a command prompt on the serial port and disables the ability to use scripts (also disables the "scripts" button in the "admin" screen). Use the "scripts" command (instead of the "go" command) to run the Brutus application to enable the use of scripts. Note that this disables the command prompt on the serial port meaning you must use the remote to control the Set-TOP.

scripts()

Run with default arguments.

Note: the "scripts" function is equivalent to:
  vxworks_main("-cfg","/tgtsvr/brutus.cfg","-ch","/tgtsvr/channels.txt")

NEXUS EXAMPLES

The Nexus reference software includes a set of examples used to demonstrate specific functions of Nexus or just to demonstrate how one might build a stand-alone application using the Nexus software layer.

These can be found in the 'nexus/examples'.

The examples are statically linked against the Nexus library (libnexus.out) to create a single object. This means the 'libsettop.out' object need not be loaded prior to loading these modules.

SETTOP API UTILITIES, TESTS, AND EXAMPLES

The Settop API reference software includes a set of utilities, tests, and examples used to demonstrate specific functions of the Settop API or just to demonstrate how one might build a stand-alone application using the Settop API. The files are separated into:

- utilities – applications used to demonstrate particular functions of the Settop API.
- tests – applications used to test (new) features of the Settop API.
- examples – applications used to demonstate how to write a simple stand-alone application.

These can be found in the 'BSEAV/api/utils', 'BSEAV/api/tests', and 'BSEAV/api/examples' directories.

The utilities are statically linked against the Settop library (libsettop.out) to create a single object. This means the 'libsettop.out' object need not be loaded prior to loading these modules. The examples and tests are dynamically linked against the Settop library. The Settop library needs to be be loaded before loading these applications. These can be linked into a single object using the 'ldmips' tool:

```
C:\> ldmips –r –o  outputfile.out libsettop.out example.out
```

Starting up the applications

A wrapper function is provided for most of the applications to allow them to be run from the host shell. This is done with the "go" command. Unlike the Brutus application's "go" command, this command takes a single parameter, comprised of space-separated application parameters to be passed to the application's "main" function. The first parameter is used as the application name. If the application takes no parameters then no parameters are needed (just use 'go'). For example, the following can be used to start the 'decode_still_picture' test application:

```
-> go "decode /tgtsvr/pictures/still1.bmg /tgtsvr/pictures/still2.bmg"
```

Note: the Brutus application includes the steps necessary to setup the disk for PVR use. This includes mounting the file system (usrAtaConfig) and setting up a disk cache to optimize the disk performance. This is done each time the application is run. The utilities, tests, and examples do not include these steps. If the particular application is to use the disk for recording or playback, then the file system needs to be mounted, each time the application is run, prior to using the 'go' command:

```
-> usrAtaConfig(0,0,"/ata0")
-> go "playwave /tgtsvr/gd.wav"
```

RUNTIME OPTIONS

Settop API Runtime Options when using VxWorks

Runtime options can be supplied to the Settop API layer to change its behavior. This is done by using the function "bsettop_set_config" (after loading the object modules and before running the application). This is typically used for enabling debug prints. For example:

```
-> bsettop_set_config("msg_modules","decode")
```

This will enable BDBG_MSG prints for the supplied module ('decode' in this example). Other definitions are used to control optional functions. For example:

```
-> bsettop_set_config("no_watchdog","yes")
```

These options are only set for the next run, must be used on each subsequent run, and cannot be set after the application has started. See the 'Reference Software Debug Guide' for more information on this function.

Nexus Runtime Options when using VxWorks

Runtime options can be supplied to the Nexus layer to change its behavior. This is done by using the function "NEXUS_SetEnv" (after loading the object modules and before running the application). This is typically used for enabling debug prints. For example:

```
-> NEXUS_SetEnv("msg_modules","decode")
```

This will enable BDBG_MSG prints for the supplied module ('decode' in this example). Other definitions are used to control optional functions. For example:

```
-> NEXUS_SetEnv(“no_watchdog”,”yes”)
```

These options are only set for the next run, must be used on each subsequent run, and cannot be set after the application has started. See the ‘Reference Software Debug Guide’ for more information on this function.