QUALCOMM®

Binary Runtime Environment for Wireless™

BREW™ 3.0 Sample Applications Guide

brew™

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA. 92121-1714
U.S.A.

BREW™ 3.0 Sample Applications Guide

October 21, 2003

Contents

Introducing the BREW Sample Applications

The BREW Software Development Kit (BREW SDK™) includes several sample applications created with BREW that demonstrate the use of many interfaces and controls commonly used in BREW application development. The sample applications can be found in the <BREW\sdk\examples> directory. The purpose of these samples is to identify the BREW APIs and controls that allow each application to accomplish its intended task and describe the functionality of each of the BREW applications from the user's perspective.

NOTE: The sample source code provided with the BREW SDK has been written to demonstrate the usage of BREW data structures and functions. Despite the extensive testing that each sample application undergoes, the source code was not intended to pass the requirements of TRUE BREW™ Testing because the sample applications are not fully featured.

The following BREW sample applications are included:

Expense Tracker	Describes a simple expense tracking application that allows the device user to record new expenses (by payment type) as they are accrued. Expenses can be saved in files that can be loaded and viewed when needed.
MediaPlayer	Describes a multimedia application capable of playing sound and video clips, including audio clips that you recorded using the application. You can also display still graphic images. Standard controls, such as play, fast forward, rewind, stop, pause, and record, are demonstrated.
NetDiagnostics	Describes an application that allows the device user to perform echo tests using Transmission Control Protocol Internet Protocol (TCP/IP), User Datagram Protocol Internet Protocols (UDP/IP), and Hypertext Transfer Protocol (HTTP).

Road Warrior	Describes an application that accesses a web server and delivers real-time traffic data for major San Diego highways. The web server URL and highways are configurable.
WhiteBoard	Describes a “virtual whiteboard” application that allows the user to draw geometric shapes of various sizes and colors. The size of shapes is limited by the device screen size, and the choice of colors is limited by the device’s supported color depth.

NOTE: The <BREW\sdk\examples> directory also includes the HelloWorld sample application.

BREW documentation

This guide is part of an information set that includes the following documents:

BREW SDK User Docs	Describes the SDK and how to use its components.
BREW Programming Concepts	Provides key BREW programming principles.
BREW API Reference	Provides a reference for all BREW APIs.
Creating a BREW Application from Scratch	Provides a tutorial designed to guide new BREW application developers through the creation of their first BREW application. It also provides an overview of the components of the BREW SDK as they relate to application development.

Requesting new BREW features

Do you have ideas for features that would make the BREW SDK more valuable and useful to you? If so, send us email at brew-request@qualcomm.com. Each request is evaluated, and a member of the New Features Response Team will respond to your email.

For more information

Online information and support is available for BREW application developers. Please visit the BREW web site for details: www.qualcomm.com/brew/developer.

Expense Tracker

Expense Tracker is a travel expense-tracking application that allows the device user to record expenses for a defined period of time. The user can track the date of the transaction, payment types, and purchase amount, and enter a description of the expense. Expenses can be reviewed, either for the entire period or on a transaction-by-transaction basis. When all expenses have been entered in the database, the user can generate a text file report that can be viewed onscreen.

NOTE: The intended target device for this sample application is the Sharp Z-800. If you run this application on the Simulator, select the Sharp Z-800 device image for best results.

Expense Tracker specifications

The following table lists the interfaces and controls used in the development of Expense Tracker, along with the files you need to run the application on the handset.

Interfaces used	Controls used	Files needed on handset
IDatabase	IDateCtl	expensetracker.bar
IDBMgr	IDialog	expensetracker.mif
IDBRecord	IStatic	expensetracker.mod
IFile	ITextCtl	expensetracker.sig
IFileMgr		

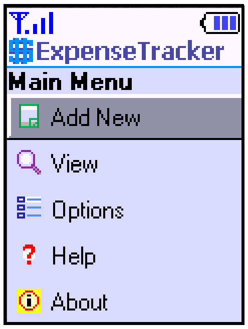
Running Expense Tracker on the handset

Before exploring the underlying code that makes Expense Tracker work, let’s take a look at the application from the user’s perspective; that is, how it works on a handset.

To run Expense Tracker

1. Run the BREW Simulator and make sure the MIF Directory setting is pointing to <BREW\sdk\examples>.
2. Choose the Expense Tracker application.

   BREW loads the Expense Tracker applet—that is, the Expense Tracker applet dynamic-link library (DLL)—and starts the application. A splash screen opens, and a screen similar to the following appears:



3. From the Expense Tracker Main Menu, you can perform the following functions:

Select this option	To do this
Add New	Enter a new expense record into the database. See Adding new expenses on page 9 for instructions.
View	View and, if needed, edit existing expense records for the current time period. See Viewing expenses on page 10 or Editing expense records on page 11 for instructions.
Options	Load and view an existing expense report and save the current expenses in the database into a flat file for viewing. See Saving and loading expense reports on page 12 for instructions.
Help	View online instructions for using Expense Tracker.
About	View copyright and version information for Expense Tracker.

4. To return to the previous screen, press **CLR**.
5. To stop the applet at any time, press **End**.

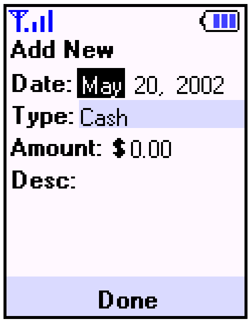
Adding new expenses

The Add New menu option lets you record expense information on a transaction-by-transaction basis. That is, each expense transaction is a separate record in the database. For each expense, you store the date that the transaction occurred, the type of payment used, the amount, and a brief description.

To add a new expense

1. Choose **Add New** from the Expense Tracker Main Menu.

   The Add New screen opens.



Add New

Date: May 20, 2002

Type: Cash

Amount: $0.00

Desc:

2. The **Date** defaults to today's date. If you need to change it, highlight the value you want to change, then press the up or down arrow keys to increase or decrease the value.

   NOTE: To advance to the next field, press the right arrow key.

3. For **Type**, press the up or down arrow keys to scroll through the payment types until the correct one is displayed. The available types are Cash, Credit, Check, and Debit.
4. For **Amount**, use the number keys to enter a value, then press the right arrow key to accept the amount. As you enter the amount, the values are backfilled to the left. For example, to enter an amount of $200.35, you would enter `20035`. To enter an amount of 50 cents, you would enter `50`.
5. For **Desc**, enter up to 16 characters to describe the expense.
6. When you are finished, press **Select**.
7. To add the expense, press **Done**.

   The Expense Tracker Main Menu reopens.

Viewing expenses

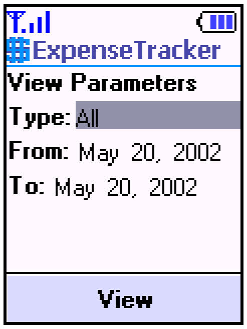
The View option on the Expense Tracker Main Menu allows you to view expenses recorded for a specified date range. You can view all transactions of a particular payment type or for all types.

NOTE: The expense records remain in the database until you save them to a flat file report. At that time, the database is cleared for the date range.

To view expenses

1. Choose **View** from the Expense Tracker Main Menu.

   The View Parameters screen opens.



ExpenseTracker

View Parameters

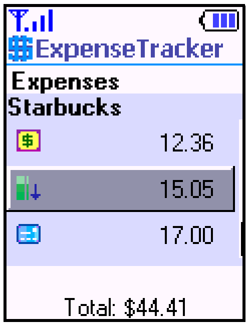
Type: All

From: May 20, 2002

To: May 20, 2002

2. For **Type**, select a payment type (All, Cash, Credit, Check, and Debit) for the expense. If you select a type other than All, only expenses of that type will be viewable.
3. In **From** and **To**, enter the range of dates for which you want to view expenses of the selected type.
4. When the correct parameters are shown, press **View**.

   The Expenses screen opens, showing the expenses for the specified range of dates, along with a total at the bottom.



ExpenseTracker

Expenses

Starbucks

12.36

15.05

17.00

Total: $44.41

5. From this screen, you can do the following:

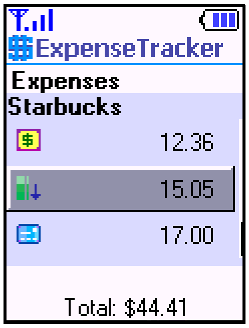
   - To see the description of each expense, press the up or down arrow keys.
   - To return to the View screen, press the **CLR** key.
   - To edit an expense, highlight it and press **Select**. See Editing expense records below.

Editing expense records

After you have added and viewed expenses, you can edit them, as needed, until you have saved them into a report.

To edit an expense

1. View the expenses you want to edit, as described in the previous procedure.



ExpenseTracker

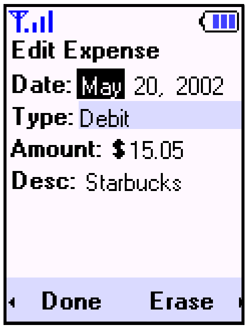
Expenses
Starbucks

$	12.36
	15.05
	17.00

Total: $44.41

2. Highlight the expense you want to edit and press **Select**.

   The Edit Expense screen opens.



Edit Expense

Date: May 20, 2002

Type: Debit

Amount: $15.05

Desc: Starbucks

3. Edit the information in the same way that you added it.

4. When you have finished editing the expense record, press **Select** and do one of the following:

   - Press **Erase** to delete the record.
   - Press **Done** to save the changes.

   The Expenses screen reopens and the changes are reflected in the total.

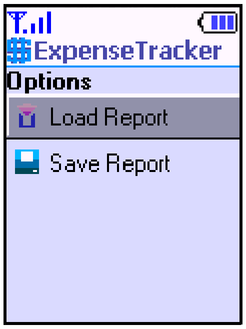
Saving and loading expense reports

When you have finished entering expenses for a particular time period—at the end of a business trip, for example—you can save the expenses into a flat file report for later viewing. After you have saved the report, the individual expense records are deleted to save space on your device. Saved reports can be loaded at any time to view the expense information.

To load an existing expense report

1. Choose **Options** from the Expense Tracker Main Menu.

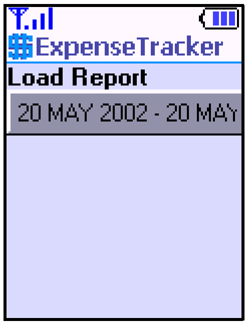
   The Options menu screen opens.



NOTE: If you haven’t added any expense records since the last time you saved a report, the Save Report option will not be shown in the menu.

2. Choose **Load Report**.

   The Load Report screen opens.



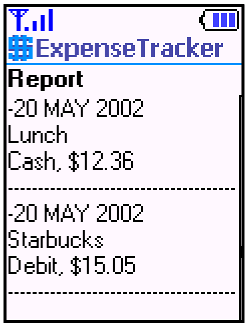
ExpenseTracker

Load Report

20 MAY 2002 - 20 MAY

3. Highlight the report you want to load and press **Select**.

   The Expense Report screen opens.



ExpenseTracker

Report

-20 MAY 2002
Lunch
Cash, $12.36

-20 MAY 2002
Starbucks
Debit, $15.05

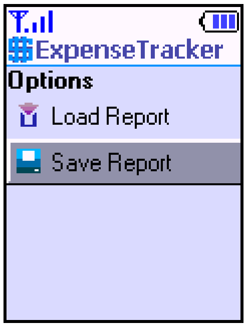
4. When you are finished viewing the report, press **Select**.

   The Expense Report screen closes and the Expense Tracker Main Menu screen reopens.

To save expenses into an expense report

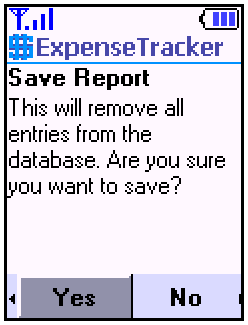
1. Choose **Options** from the Expense Tracker Main Menu.

   The Options menu screen opens.



2. Choose **Save Report**.

   The Save Report verification screen opens.



Save Report

This will remove all entries from the database. Are you sure you want to save?

3. To save the report and delete the expense records, press **Yes**.

   The report is saved, the individual expense records are deleted, and the Options menu screen reopens.

What Expense Tracker demonstrates

Expense Tracker demonstrates how to create a BREW application that utilizes file handling and database capabilities, as well as various GUI controls provided in the BREW SDK. The Expense Tracker interface uses several menus that allow the user to access various program capabilities, including adding new expenses to the expense database, performing database queries, editing and erasing expenses, and generating expense reports for onscreen viewing.

Users enter, search, edit, and erase expense records using a set of forms that allow information to be entered on the phone screen. These forms demonstrate a variety of BREW controls, including IMenuCtl, IDateCtl, ITextCtl, IStatic, and a custom control called CAmountCtl that was designed specifically for this application. Forms are implemented programmatically using the IDialog interface which uses form controls created with the BREW Resource Editor.

Expense Tracker also demonstrates implementation of a BREW database application that uses the IDatabase, IDBMgr, and IDBRecord interfaces, as well as IFileMgr and IFile for file handling. The sample application supports localization using the BREW Resource Editor to define string and image values used by the application.

MediaPlayer

MediaPlayer is a multimedia application capable of playing sound and video clips, and displaying still graphic images. The application can also record and play back QUALCOMM PureVoice (QCP) audio clips if you have the appropriate CMX DLLs loaded. Up to 32 multimedia files can be stored for use with this application. MediaPlayer includes standard controls, such as play, fast forward, rewind, stop, pause, and record. MediaPlayer supports a wide variety of audio, video, and still graphic formats including QCP, MPEG Audio Layer 3 (MP3), Musical Instrument Digital Interface (MIDI), BREW Compressed Image (BCI), Bitmap (BMP), and Packet Mode Data (PMD), which contains both audio and video.

NOTE: The intended target device for this sample application is the Sharp Z-800. If you run this application on the Simulator, select the Sharp Z-800 device image for best results.

MediaPlayer specifications

The following table lists the interfaces and controls used in the development of MediaPlayer, along with the files you need to run the application on the handset.

Interfaces used	Controls used	Files needed on handset
IMedia	IDisplay IFile IFileMgr IMenuCtl IStatic	mediaplayer.bar mediaplayer.mif mediaplayer.mod mediaplayer.sig You also need to include media files in a directory called “media” on the handset.

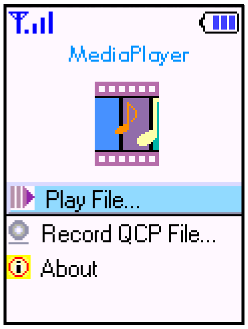
Running MediaPlayer on the handset

Before exploring the underlying code that makes MediaPlayer work, take a look at the application from the user's perspective; that is, how it works on a handset.

To run MediaPlayer

1. Run the BREW Simulator and make sure the MIF Directory setting is pointing to <BREW\sdk\examples>.
2. Choose the MediaPlayer application.

   BREW loads the MediaPlayer applet DLL and starts the application. A splash screen opens, and a screen similar to the following appears:



3. From the MediaPlayer Main Menu, you can perform the following functions:

Select this option	To do this
Play File	Select an audio, video, or still graphic file to run on the handset. See Viewing or playing a multimedia file below for instructions.
Record QCP File	Record a QCP audio file using a microphone to emulate the device mouthpiece. See Recording a QCP audio file on page 21 for instructions. The recorded file can be played using the Play File menu option.
About	View copyright information for MediaPlayer.

4. To stop the applet at any time, press **End**.

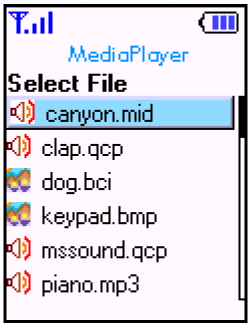
Viewing or playing a multimedia file

The Play File menu option lets you select the audio, video, or still graphic file to be run on the handset screen. Each of these types of media are described below.

To play an audio file

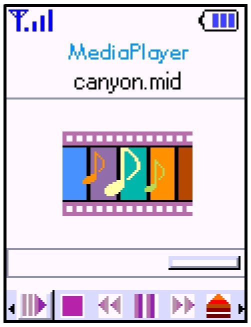
1. Choose **Play File** from the MediaPlayer Main Menu.

   The Select File screen opens.



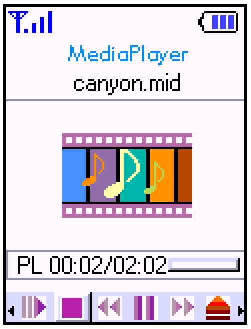
2. Highlight the audio file (MIDI, QCP, or MP3) you want to play and press **Select**.

   A screen similar to the following opens.

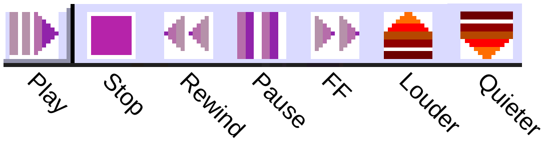


3. To play the audio clip, press **Select**.

   The sound clip plays and the status bar at the bottom displays the time elapsed.



The other controls are described below.



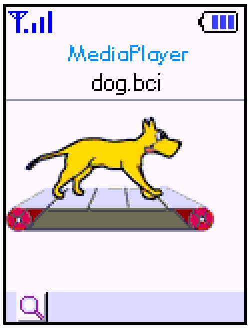
4. When you are finished playing the media file, press **CLR**.

   The Select File screen reopens.

To play a video file

1. From the Select File screen, highlight the video file (BCI) you want to play and press **Select**.

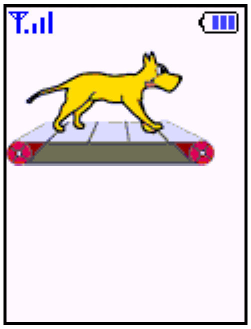
   The video plays.



2. To clear the screen and show only the video, press the **Select** key to select the **Zoom** (magnifying glass) toolbar button.

   NOTE: The Zoom button is available only for BCI and BMP files.

   The video plays without displaying the information at the top of the screen or the button bar.



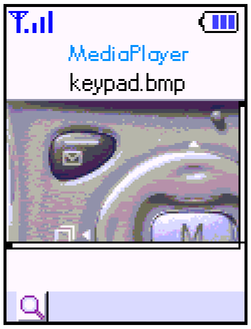
3. Press any key to return to the previous view.
4. When you are finished playing the media file, press **CLR**.

   The Select File screen reopens.

To view a still image

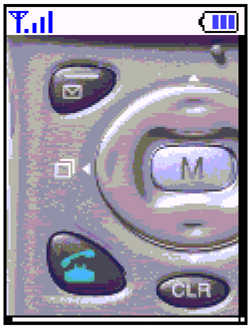
1. From the Select File screen, highlight the image file (BMP) you want to view and press **Select**.

   A screen similar to the following appears.

2. To clear the screen and show only the graphic, press the **Select** key to select the **Zoom** toolbar button.

   The image appears without displaying the information at the top of the screen or the button bar.

3. Press any key to return to the previous view.
4. You can also use the arrow keys to scroll the graphic left, right, up, and down.
5. When you are finished viewing the still graphic image, press **CLR**.

   The Select File screen reopens.

6. To exit MediaPlayer, press **End**.

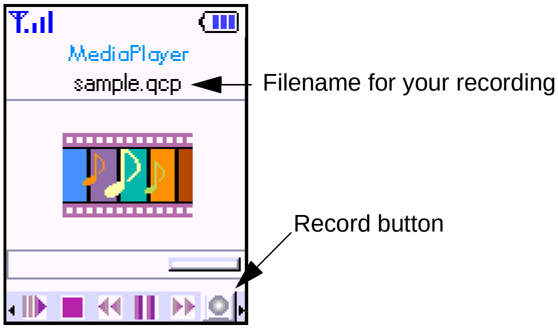
Recording a QCP audio file

MediaPlayer allows you to record your voice or any other sound through the device mouthpiece and then, using the Play File menu option, replay the recorded sounds. There is no limit to the length of each recording; however, due to the limited amount of memory available on a handset, the durations should be relatively short.

To record a QCP file

1. Choose **Record QCP File** from the MediaPlayer Main Menu.

   The Record screen opens.



   Notice that there is a filename at the top of the screen. This is the filename to be used for the recording. Also notice that the button bar now includes the green Record button.

2. Highlight the Record button and press **Select**.

   The Record screen shows the elapsed time of the recording above the button bar.

3. When you want the recording to stop, highlight the **Stop** button and press **Select** again.

   The recording is saved with the name that appears at the top of the screen. This file is now available for selection when you choose the Play File option from the MediaPlayer Main Menu.

What MediaPlayer demonstrates

MediaPlayer demonstrates how to create a BREW multimedia application. The BREW programmer used IMedia and IMedia-based interfaces to build a media player that plays and records all multimedia formats, including audio and video. Currently, only QCP recording is supported, and it requires that you have the appropriate CMX DLLs loaded. MediaPlayer uses IImageCtl and IImage interfaces to display images. The programmer offers the following basic guidelines to help you create similar types of applications.

- Create your own interface (abstract base class) and use it in the application. MediaPlayer has an IWindow interface that is inherited by all the windows. Each window is capable of redrawing itself and handling the events sent to it. There are three windows:
  - The Main window manages the main menu.
  - The FileList window allows the user to select from the list of media files.
  - The Player window enables playback and recording controls.
- Use various controls, such as IMenuCtl, IImageCtl, and IStatic. MediaPlayer also shows how you can use IDisplay to implement your own progress bar control.
- Enable cooperative multitasking through the ISHELL_Resume() API, which is used to break a big task into smaller chunks. In MediaPlayer, the IWindow redraw occurs asynchronously. MediaPlayer also demonstrates usage of asynchronous timers in the application.
- Use user-defined events in your application. For example, MediaPlayer uses user-defined events to create IMedia-based classes.

NetDiagnostics

NetDiagnostics is an application that allows the device user to perform echo tests using the TCP/IP and UDP/IP protocols to a user-specified host or IP address, and HTTP tests that can return a specified URL, as well as information such as transmission speed.

NOTE: The intended target device for this sample application is the Sharp Z-800. If you run this application on the Simulator, select the Sharp Z-800 device image for best results.

NetDiagnostics specifications

The following table lists the interfaces and controls used in the development of NetDiagnostics, along with the files you need to run the application on the handset.

Interfaces used	Controls used	Files needed on handset
INetMgr	IHtmlViewer	netdiagnostics.bar netdiagnostics.mif netdiagnostics.mod netdiagnostics.sig
ISocket		
IHtmlViewer		
IWeb		
IWebOpts		
IWebResp		
ISource		
ISourceUtil		
IPeek		
IGetLine		

Running NetDiagnostics on the handset

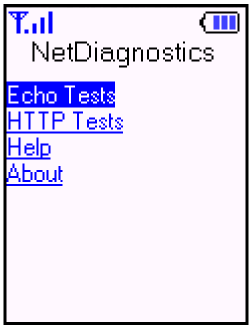
Before exploring the underlying code that makes NetDiagnostics work, take a look at the application from the user’s perspective; that is, how it works on a handset.

To run NetDiagnostics

1. Run the BREW Simulator and make sure the MIF Directory setting is pointing to <BREW\sdk\examples>.

2. Choose the NetDiagnostics application.

   BREW loads the NetDiagnostics applet DLL and starts the application. A splash screen opens, and a screen similar to the following appears:



3. From the NetDiagnostics Main Menu, you can perform the following functions:

Select this option	To do this
Echo Tests	Perform echo tests on the device using the TCP/IP or UDP/IP protocol. See Performing echo tests below.
HTTP Tests	Test connection response information for a user-specified URL. See Performing HTTP tests on page 26 for instructions.
Help	View online instructions for using NetDiagnostics.
About	View copyright and version information for NetDiagnostics.

4. To stop the applet at any time, press **CLR**.

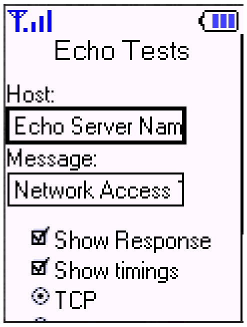
Performing echo tests

The Echo Tests menu option lets you specify a series of parameters for testing, such as a destination host or IP address, a text message to be transmitted, and the type of networking protocol to be tested. When the tests are run and network access is tested, you have the option of viewing response information, as well as response time information.

To run the echo tests

1. Choose **Echo Tests** from the NetDiagnostics Main Menu.

The Echo Tests screen opens.



Echo Tests

Host:

Echo Server Nam

Message:

Network Access

☑ Show Response

☑ Show timings

● TCP

2. To change the destination host or IP address, highlight **Host** and press **Select**.

   The address becomes an editable field. You can perform the following text editing functions:

To perform this edit	Do this
Enter text characters	Press the right and left arrow keys and the text characters on the device keypad.
Enter numbers	Perform these steps: • Press **Select** to highlight **Done**. • Press the right arrow to highlight **Multitap**. • Press **Select**. • Highlight **Numbers** and press **Select**. • Now press the device keypad to type numbers.
Enter symbols	Perform these steps: • Press **Select** to highlight **Done**. • Press the right arrow to highlight **Multitap**. • Press **Select**. • Highlight **Symbols** and press **Select**. The screen shows a series of symbols with their associated keys on the device keypad. • To see more symbol choices, press the down arrow key. • Press the device keypad to enter the associated symbol.

3. To change the message to be transmitted, highlight **Message** and press **Select**.

The message becomes an editable field and the same editing functions as Host are available.

4. The **Show Response** check box indicates whether you want to view the response from the host. You can change the selection by highlighting the check box and pressing **Select**.

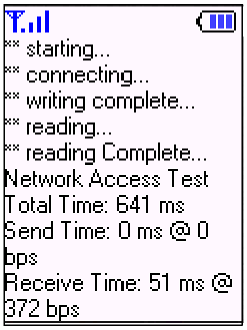
5. The **Show timings** check box indicates whether you want to view response time information, such as the total transmission time (in milliseconds), send time (in milliseconds at a specified number of bits per second), and receive time (for example 10 ms @ 2000 bps). You can change the selection by highlighting the check box and pressing **Select**.

6. Select either the **TCP** or **UDP** Internet Protocol. This is a radio button-type setting. You must select one option or the other; selecting both is not possible. To change the selection, highlight the protocol you want to use for the tests and press **Select**.

NOTE: To view onscreen definitions of the fields, highlight **Info** and press **Select**. When you are finished, press **CLR**.

7. When you are finished selecting testing parameters, press **Start**.

The tests are run and a screen similar to the following appears.



```
** starting...
** connecting...
** writing complete...
** reading...
** reading Complete...
Network Access Test
Total Time: 641 ms
Send Time: 0 ms @ 0
bps
Receive Time: 51 ms @
372 bps
```

8. To clear the test screen, press **CLR**.

The NetDiagnostics Main Menu reopens.

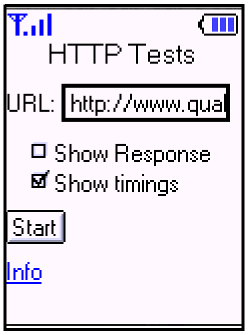
Performing HTTP tests

The HTTP Tests menu option lets you specify a series of parameters for testing the HTTP connection to a specified URL. When the tests are run, you have the option of viewing response information, as well as response time information.

To run the HTTP tests

1. Choose **HTTP Tests** from the NetDiagnostics Main Menu.

   The HTTP Tests screen opens.



HTTP Tests

URL: http://www.qua

□ Show Response

☑ Show timings

2. To change the target URL destination, highlight **URL** and press **Select**.

   The URL becomes an editable field. You can perform the same text editing functions described in the previous procedure.

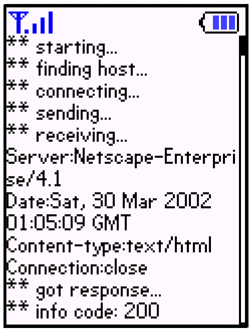
3. The **Show Response** check box indicates whether you want to view the HTTP connection response from the target URL. You can change the selection by highlighting the check box and pressing **Select**.

4. The **Show timings** check box indicates whether you want to view response time information. You can change the selection by highlighting the check box and pressing **Select**.

   NOTE: To view onscreen definitions of the fields, highlight **Info** and press **Select**. When you are finished, press **CLR**.

5. When you are finished selecting testing parameters, press **Start**.

   The tests are run and a screen similar to the following appears.



```
** starting...
** finding host...
** connecting...
** sending...
** receiving...
Server:Netscape-Enterpri
se/4.1
Date:Sat, 30 Mar 2002
01:05:09 GMT
Content-type:text/html
Connection:close
** got response...
** info code: 200
```

6. To clear the test screen, press **CLR**.

The NetDiagnostics Main Menu reopens.

What NetDiagnostics demonstrates

NetDiagnostics demonstrates how to create a BREW application that utilizes the various BREW network interfaces. The GUI is driven by the IHtmlViewer interface, which allows HTML documents to be viewed on the phone screen. Each application screen is generated by accessing an HTML file stored in the phone's file system, and displaying it in the IHtmlViewer object. The user navigates the interface by selecting the various links displayed in the HTML page.

The application performs two types of network connectivity tests:

These tests	Allow the user to do this
Echo Tests	Send TCP or UDP messages to a specified echo server. The user specifies the host, message text, transmission type (TCP or UDP), and the response parameters in an onscreen HTML form. After the test is initiated, the application displays the status of the transmission, returned message, and transmission statistics.
HTTP Tests	Access an Internet address using the HTTP protocol. The user specifies the URL and display parameters in an onscreen HTML form. After the test is initiated, the application displays the status of the transmission, returned HTML source code, and transmission statistics.

Network connectivity is handled using the INetMgr, ISocket, IWeb, IWebOpts, and IWebResp BREW API interfaces. HTML text is parsed and displayed by the IHtmlViewer, ISource, ISourceUtil, IPeek, and IGetLine Interfaces.

Road Warrior

Road Warrior is an application that connects to the California Department of Transportation web server and delivers real-time traffic data for major San Diego highways. The web server URL and highways are configurable.

NOTE: The intended target device for this sample application is the Sharp Z-800. If you run this application on the Simulator, select the Sharp Z-800 device image for best results.

Road Warrior specifications

The following table lists the interfaces and controls used in the development of Road Warrior, along with the files you need to run the application on the handset.

Interfaces used	Controls used	Files needed on handset
IApplet IDisplay IFile IFileMgr IImage INetMgr IShell ISocket ISource IWeb IWebResp	IMenuCtl IStatic	roadwarrior.bar roadwarrior.dat roadwarrior.mif roadwarrior.mod roadwarrior.sig

Running Road Warrior on the handset

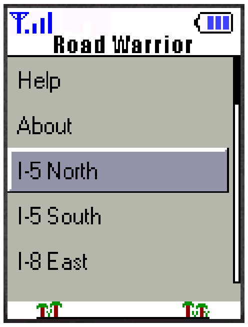
Before exploring the underlying code that makes Road Warrior work, take a look at the application from the user's perspective; that is, how it works on a handset.

To run Road Warrior

1. Run the BREW Simulator and make sure the MIF Directory setting is pointing to <BREW\sdk\examples>.

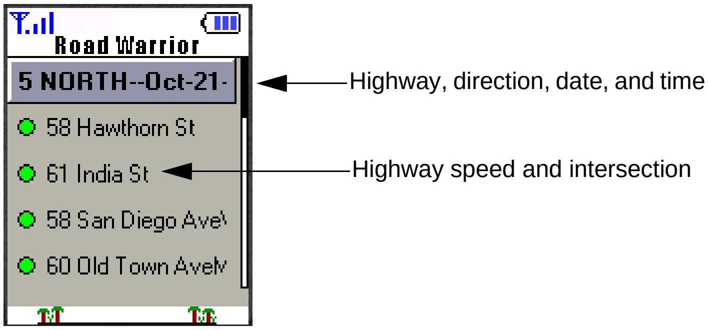
2. Choose the Road Warrior application.

   BREW loads the Road Warrior applet DLL and starts the application. A splash screen opens, and a screen similar to the following appears:



3. Highlight the highway and direction for which you want to acquire speed information, and press **Select**.

   The application connects to the web server, and the following screen opens.

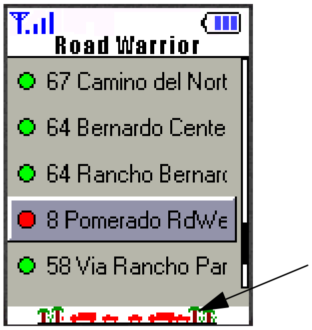


5 NORTH--Oct-21-
58 Hawthorn St
61 India St
58 San Diego Ave\
60 Old Town Ave\

Highway, direction, date, and time

Highway speed and intersection

The scrolling banner along the top shows the selected highway and direction, and the date and time the data was taken from the freeway. The number to the left of each intersection is the current speed of traffic. As you scroll down through the intersections, the graphic display at the bottom of the screen whimsically approximates the amount of traffic at each intersection, as shown here.



Flow of traffic at selected intersection

4. To stop the applet at any time, press **End**.

What Road Warrior demonstrates

Road Warrior is an application that provides the user with real-time traffic speeds. It presents data for San Diego County freeways only, but it can be used as a model for similar applications in other geographical areas.

Road Warrior demonstrates how to use the IWeb interface to get information from a publicly available web server, and then parse the resulting HTML to prepare the data for display to the user. It also demonstrates the following:

- It shows how to provide feedback to the user of a potentially time-consuming network operation that is currently in progress, and how to allow the user to cancel out of the network connection.
- It uses static and animated images to help beautify an otherwise all text-based application.
- It shows how suspend and resume of a BREW application should be handled, including management of the active states of controls, and the stopping and starting of animations.
- It uses the IMenuCtl interface to display information and get user input, including custom key press handling.
- It provides an example of how to persistently save user preferences across different running sessions of the application.
- It demonstrates how to perform simple file reading, including a mechanism to read large blocks of text in at one time to improve file system performance.

- It queries the BREW device at runtime to determine screen color depth, and configures menu control styles based on that information.
- It provides an example of how to put strings in string resources so that the application can be easily translated into other languages.
- It shows techniques for minimizing the number of keystrokes the user has to press: type-to-select and remembering last used items.
- Most importantly, Road Warrior demonstrates a real-life application that has passed TRUE BREW Testing and is being used in commercial BREW environments.

WhiteBoard

This “virtual whiteboard” application allows the user to draw geometric shapes of various sizes and colors. All or part of a drawing can be erased at any time. The size of shapes is limited by the device screen size, and the choice of colors is limited by the color depth supported by the device.

NOTE: The intended target device for this sample application is the Sharp Z-800. If you run this application on the Simulator, select the Sharp Z-800 device image for best results.

WhiteBoard specifications

The following table lists the interfaces and controls used in the development of WhiteBoard, along with the files you need to run the application on the handset.

Interfaces used	Controls used	Files needed on handset
IBitmap IDisplay IGraphics IImage	IMenuCtl IStatic	whiteboard.bar whiteboard.mif whiteboard.mod whiteboard.sig

Running WhiteBoard on the handset

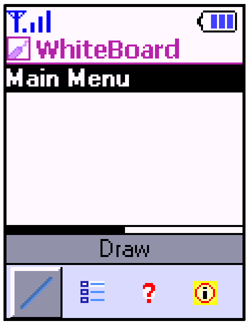
Before exploring the underlying code that makes WhiteBoard work, take a look at the application from the user’s perspective; that is, how it works on a handset.

To run WhiteBoard

1. Run the BREW Simulator and make sure the MIF Directory setting is pointing to <BREW\sdk\examples>.

2. Choose the WhiteBoard application.

   BREW loads the WhiteBoard applet DLL and starts the application. A splash screen opens, and a screen similar to the following appears:



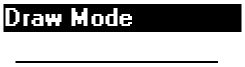
3. From the WhiteBoard Main Menu, you can perform the following functions:

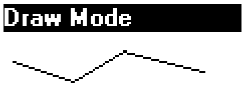
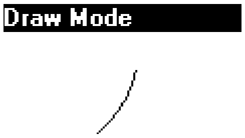
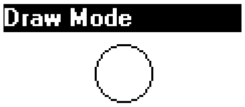
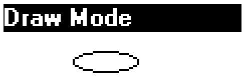
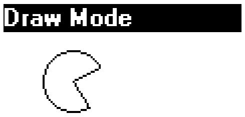
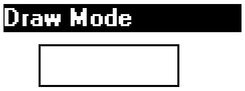
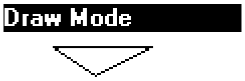
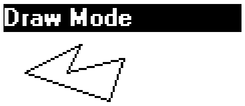
Select this option	To do this
Draw	Draw on the whiteboard using the shapes and colors selected using the Option menu option. See Drawing on the whiteboard on page 38 for instructions.
Tools	Specify the shape and color for drawing and the fill color. See Setting draw options on page 34 for instructions.
Help	View online instructions for using WhiteBoard.
About	View copyright and version information for WhiteBoard.

4. To stop the applet at any time, press **End**.

Setting draw options

The Tools menu option lets you select the shape that you want to draw, and the border and fill colors. The following shapes are available. Instructions for drawing each shape follow.

Shape	Example
Line	

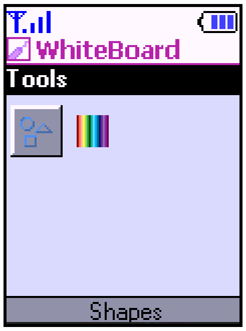
Shape	Example
Polyline	
Arc	
Circle	
Ellipse	
Pie	
Rectangle	
Triangle	
Polygon	

NOTE: In addition to the shapes shown above, those shapes that are enclosed, such as circles, rectangles, and polygons, can be filled with black or any other color. Lines and the borders of shapes can also be assigned a color.

To select a shape

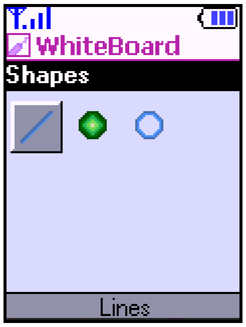
1. Choose **Tools** from the WhiteBoard Main Menu.

   The Tools Menu screen opens.



2. With Shapes highlighted, press **Select**.

   The Shapes Menu screen opens.



3. Select the type of shape you want to draw. The following table describes the shapes available under the three menu options on the Shapes menu.

Select this menu option	To draw these shapes
Lines	Line Polyline Arc

Select this menu option	To draw these shapes
Filled Shapes	Circle Ellipse Rectangle Triangle Polygon
Unfilled Shapes	Circle Ellipse Pie Rectangle Triangle Polygon

4. When you have made your shape selection, do one of the following:

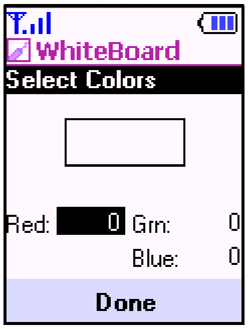
- If you want to change the border color, or if you selected an enclosed shape, such as a circle, and want to change the fill color, see To select a border or fill color below.
- If you do not want to change the color, press **CLR**.

  The WhiteBoard Main Menu opens.

To select a border or fill color

1. From the Tools menu, choose **Color**.

The Select Colors screen opens.



WhiteBoard

Select Colors

Red: 0 Grn: 0

Blue: 0

The selected shape appears in the screen; in this case, an unfilled Rectangle shape was selected. If a filled shape had been selected, the screen would also show the Mode field which allows you to switch between Border color and Fill color.

2. To change the border color, select Border in the Mode field and specify the RGB values for the color you want to use.

3. To change the fill color, select Fill in the Mode field and specify the RGB values for the color you want to use.
4. When you are finished selecting colors for the shape, press **Done**.

   The WhiteBoard Main Menu screen opens.

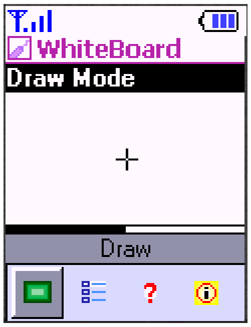
Drawing on the whiteboard

After you have used the Tools to select a shape and, optionally, colors, you are ready to use the Draw menu option to draw geometric shapes on the whiteboard. You can change the drawing shape at any time and shapes can overlap.

To draw on the whiteboard

1. Choose **Draw** from the WhiteBoard Main Menu.

   The WhiteBoard screen goes into Draw Mode.



2. Using the arrow keys, move the crosshairs to draw the selected shape. Each shape requires a different drawing method, as described in the procedures that follow.

   NOTE: When you are in Draw Mode, the crosshairs move 1 pixel each time you press one of the arrow keys. To increase the distance of each movement, press one of the number keys. For example, to have the crosshairs move 7 pixels each time you press an arrow key, press the **7** key.

3. When you are finished drawing, press **CLR**.

   The WhiteBoard Main Menu opens.

   NOTE: The **CLR** key can also be used to delete a drawing before you have committed it to the white board. For more information, see Undo on page 42.

To draw a line

1. Move the crosshairs to the point where you want the line to start and press **Select**.

   A small flashing dot appears.

2. Move the crosshairs to the point where you want the line to end and press **Select**.

   A line connects the two points.

To draw a polyline

1. Move the crosshairs to the point where you want the polyline to start and press **Select**.

   A small flashing dot appears.

2. Move the crosshairs to the point where you want the first line to end and press **Select**.

   Another dot appears.

3. Repeat step 2 until you have reached the point where you want the polyline to end.

4. Press **Select** a second time at the last point.

   A polyline connects the dots.

To draw an arc

1. Move the crosshairs to the point where you want the center of the arc to appear and press **Select**.

   A small flashing dot appears.

2. Move the crosshairs to the point where you want the radius and starting angle of the arc to appear and press **Select**.

   Another dot appears.

3. Move the crosshairs to the point where you want the ending angle of the arc to appear and press **Select**.

4. Press **Select** again.

   An arc connects the dots.

NOTE: WhiteBoard draws arcs in a counterclockwise direction. Therefore, if you specify a location for the starting angle of the arc that is lower on the Y axis than the ending angle, the resulting arc will be less than 180 degrees. If the location for the starting angle of the arc that is higher on the Y axis than the ending angle, the resulting arc will be more than 180 degrees.

To draw a circle

1. Move the crosshairs to the point where you want the center of the circle to appear and press **Select**.

   A small flashing dot appears.

2. Move the crosshairs to the point you want to use for determining the radius of the circles along the X axis and press **Select**.

   A circle is drawn.

To draw an ellipse

1. Move the crosshairs to the point where you want the center of the ellipse to appear and press **Select**.

   A small flashing dot appears.

2. Move the crosshairs to the point you want to use for determining the semi-X and semi-Y axis of the ellipse and press **Select**.

   An ellipse is drawn.

To draw a pie

1. Move the crosshairs to the point where you want the center of the pie to appear and press **Select**.

   A small flashing dot appears.

2. Move the crosshairs to the point where you want the radius and starting angle of the pie to appear and press **Select**.

   Another dot appears.

3. Move the crosshairs to the point where you want the ending angle of the pie to appear and press **Select**.

4. Press **Select** again.

   A pie connects the dots.

NOTE: Like arcs, WhiteBoard draws pies in a counter-clockwise direction. Therefore, if you specify a location for the starting angle of the pie that is lower on the Y axis than the ending angle, the resulting pie will be less than 180 degrees. If the location for the starting angle of the pie that is higher on the Y axis than the ending angle, the resulting pie will be more than 180 degrees.

To draw a rectangle

1. Move the crosshairs to the point where you want one corner of the rectangle to appear and press **Select**.

   A small flashing dot appears.

2. Move the crosshairs to the point where you want the opposite corner to appear and press **Select**.

   A rectangle is drawn.

To draw a triangle

1. Move the crosshairs to the point where you want one angle of the triangle to appear and press **Select**.

   A small flashing dot appears.

2. Move the crosshairs to the point where you want the second angle to appear and press **Select**.

   Another dot appears.

3. Move the crosshairs to the point where you want the third angle to appear and press **Select**.

   A triangle connects the three points.

To draw a polygon

1. Move the crosshairs to the point where you want one angle of the polygon to appear and press **Select**.

   A small flashing dot appears.

2. Repeat step 1 until you have reached the point of the last angle and press **Select** again.

   A polygon connects the points.

Some final notes about drawing

In addition to drawing, WhiteBoard offers a few features that enhance the usefulness of the application, including an Undo feature, a Scrolling feature that allows the user to extend the drawing horizontally beyond the width of the device screen, and a Clipping feature.

Undo

When you draw a given shape, a triangle for example, the positions for the angles are saved in the device’s memory until the final shape is drawn. If you press **CLR** before the shape is completed (that is, before you set the third point), all points for the shape are deleted.

Scrolling

When you are in Draw Mode, you can extend your drawing beyond the confines of the device’s screen width. The scroll bar at the bottom of the screen allows you to scroll twice the width of the device screen.

Clipping

The Clip feature allows you to delete a rectangular portion of your drawing. To clip a drawing, finish drawing the current shape and then press **Select** twice at one corner of the clipping rectangle. Move the crosshairs to the opposite corner of the rectangle and press **Select** again. All drawing within the selected rectangle is deleted. You can delete a small portion of the screen or the entire screen using this feature.

What WhiteBoard demonstrates

WhiteBoard demonstrates how to create a BREW application that utilizes the graphics capabilities of the BREW API. The sample application allows the user to render simple geometric shapes to the phone's display in a scrolling, rectangular, drawing canvas. The user can select shapes, and set the border and fill color of each. A cursor is used to plot points on the drawing canvas to draw the selected geometric shape.

The application allows users to select filled and unfilled drawing shapes by navigating through a series of Icon List menus. A form is also available to allow the user to set and view the color settings for a particular shape. After the shape and color settings have been selected, the user can draw the shape on the canvas by plotting points which represent the required vertices of the shape to be drawn.

Extensive use is made of the BREW IGraphics interface for drawing lines, polylines, arcs, circles, ellipses, rectangles, triangles, pie, and polygons of various dimensions. The canvas is implemented using two graphics buffers, the screen display and a separate offscreen IBitmap buffer. The screen display graphics buffer is used to display the drawing cursor and plotted points. Shapes are rendered to the IBitmap buffer and updated to the phone screen buffer when needed. Using the AEE_RO_XOR raster operation, the cursor image and points drawn on the phone screen with the underlying image, cursor, and point movement is made to appear transparent on the screen.

The IBitmap offscreen buffer is wider than the phone screen to allow the canvas to be scrolled in the horizontal direction.

Index