



BREW[®] Application Manager and MobileShop[®] 2.1 Integration Guide

Restricted Distribution. Not to be distributed to anyone who is not an employee of either Qualcomm or a subsidiary of Qualcomm without the express approval of Qualcomm's Configuration Management.

Terms and Conditions of Usage

This document may include images, references to, or descriptions of devices or parts whose manufacture, use, sale, offer for sale, or importation into the United States is limited or prohibited by the February 5th, 2008 injunction against QUALCOMM Incorporated. This document should not be construed as an offer to sell such parts for use or importation into the U.S., nor should it be construed as assistance in making, using, selling, offering to sell, or the importation of any product in the U.S. containing such parts. This document is intended solely to provide information for those products that are outside the scope of the injunction. Recipient's download and/or use of the information in this document constitutes agreement with these terms.

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

Copyright © 2008 QUALCOMM Incorporated.
All Rights Reserved.

Not to be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm.

This documentation was written for use with BREW[®] Application Manager and MobileShop[®], software version 2.1. This documentation and the BREW Application Manager and MobileShop software described in it are copyrighted, with all rights reserved. This documentation and the BREW Application Manager and MobileShop software may not be copied, except as otherwise provided in your software license or as expressly permitted in writing by QUALCOMM Incorporated.

Any sample code herein is provided for your convenience, and has not been tested or designed to work on any particular system configuration. It is provided "AS IS" and your use of this sample code, whether as provided or with any modification, is at your own risk. Qualcomm undertakes no liability or responsibility with respect to the sample code, and disclaims all warranties, express and implied, including without limitation warranties on merchantability, fitness for a specified purpose, and infringement. Qualcomm reserves all rights in the sample code, and permits use of this sample code only for educational and reference purposes.

This technical data may be subject to U.S. and international export, re-export or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

MSM is a trademark of QUALCOMM Incorporated.

QUALCOMM, BREW, BREW SDK, the BREW logo, and MobileShop are registered trademarks of QUALCOMM Incorporated in the United States and may be registered in other countries.

Other product and brand names may be trademarks or registered trademarks of their respective owners.

BREW[®] Application Manager and MobileShop[®] 2.1 Integration Guide

80-DA257-1 Rev. A

July 29, 2008

Contents

1 Installing Application Manager	5
Before you begin	5
Required components	5
Installing Application Manager	6
2 Customizing Application Manager	7
Configuring device-specific information	7
TouchDisplays	8
BamEnabledDisplays	8
CancelOnClose	9
AppsUpdatedURL	9
KeyMapDisplay1-4	9
HapticTone and HapticResponseDuration	10
Understanding application shortcuts and statistics	11
Shortcut application interface commands	11
3 Integrating Application Manager into Device Builds	15
Application Manager integration	15
Integrating aeemenucltx	16
Integration verification	16
Appendix A PEK Tests	17
Appendix B Acronyms and Terms	18

Figures

Figure 1. Example bitwise representation	8
Figure 2. Bitwise representation of TouchDisplays setting	8
Figure 3. Bitwise representation of BamEnabledDisplays setting	9

Tables

Table 1. Required components	5
Table 2. Installation locations	6
Table 3. Sample KeyMapDisplay settings to operate Application Manager	10
Table 4. Default KeyMapDisplay settings for AVK keys	10
Table 5. Description of a line entry in applist.lst file	11
Table 6. Commands for shortcut application interface	11
Table 7. Application Manager files that must be built as BREW persistent files	15
Table 8. aeemenucltex files that must be built as BREW persistent files	16
Table 9. PEK tests	17

1 Installing Application Manager

This guide explains how to integrate the BREW® Application Manager (Application Manager) and MobileShop® on a device running BREW 3.1.5 or higher. It covers installation, customization, and integration of Application Manager and MobileShop on a BREW-enabled device.

This section describes how to install the Application Manager package, including the aeemenucltex.

Before you begin

To successfully integrate the Application Manager and its components, review the following documents:

- *BREW® Application Manager 2.1 Guide* (provided with the Application Manager package)
- *BREW® OEM Note: Verification of BREW Application Manager Integration* (provided on the BREW OEM extranet)
- *BREW® Porting Guide for MSM™ Platforms* (provided with the BREW Porting Kit) for more information about how to make the Application Manager part of the device build

Required components

[Table 1](#) lists the components required for integration of Application Manager.

Table 1. Required components

Component name	Description
Devices running BREW PK 3.1.5SP01	Devices require BREW 3.1.5 without the static version of either Application Manager or MobileShop. The devices must also pass the BREW Porting Evaluation Kit (PEK) Operational Acceptance Tests (OAT). See Appendix A on page 17 for the minimum tests that must be run.
BREW Application Manager 2.1.0	This package also contains Application Manager's aeemenucltex.
Device requirements	Minimum screen size requirements for running Application Manager are as follows: <ul style="list-style-type: none">■ Non-touch-enabled Application Manager: 128*128■ Touch-enabled Application Manager: 176*220 Devices that must run Application Manager when the flip/slider is closed must have four-to-six keys that can be used.

Installing Application Manager

Install Application Manager from the BREW OEM Extranet under Downloads. [Table 2](#) lists the various components provided with Application Manager and the default installation locations.

Table 2. Installation locations

Packages	Default installation location
BREW Application Manager and MobileShop	<install-dir>\BREW Application Manager\2.1.0\brewappmgr
aeemenucltx	<install-dir>\BREW Application Manager\2.1.0\aeemenucltx

The relevant Start menu entries contain a shortcut to open an Explorer window in the correct folder.

2 Customizing Application Manager

This section describes the steps necessary to customize Application Manager to meet your specifications:

- Configuring device-specific information, including displays enabled for touchscreen and Application Manager
- Customizing haptic behavior
- Understanding shortcut applications

Configuring device-specific information

The `oemcfg.txt` file is a device-specific configuration file that is customized by the OEM. A sample `oemcfg.txt` file is provided with Application Manager and is located in the `brewappmgr` folder.

The `oemcfg.txt` file contains settings that define device characteristics for Application Manager that are not available from other sources. The following is a sample `oemcfg.txt` file:

```
TouchDisplays=2
BamEnabledDisplays=3
CancelOnClose=false
AppsUpdatedUrl=OEMShortcut:AppsUpdated
KeyMapDisplay1=57393,57394,57397,57392
HapticTone=78
HapticResponseDuration=100
```

Some of the settings specify capabilities that are available when a particular BREW display is active. These settings are primarily used to dictate capabilities of different configurations for various devices.

The following are the displays supported by BREW:

- AEECLSID_DISPLAY1
- AEECLSID_DISPLAY2
- AEECLSID_DISPLAY3
- AEECLSID_DISPLAY4

The capability for each display is positional and is determined from the associated bit in the integer value of the capability, as the example in [Figure 1](#) shows.

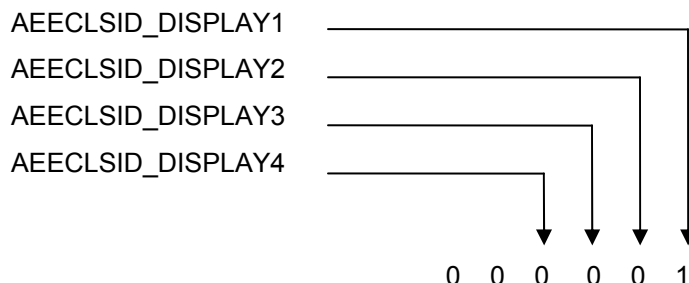


Figure 1. Example bitwise representation

If the bit associated with one of the BREW displays is set to 1, the setting is enabled for that display (e.g., AEECLSID_DISPLAY1 is set to 1 in [Figure 1](#)).

TouchDisplays

The TouchDisplays setting specifies the BREW displays with touchscreen capability. This setting results in Application Manager using a touch-enabled menu UI instead of the standard button menu UI.

[Figure 2](#) shows an example of the TouchDisplays setting that indicates AEECLSID_DISPLAY2 and AEECLSID_DISPLAY3 are touch-enabled and the other displays are not.

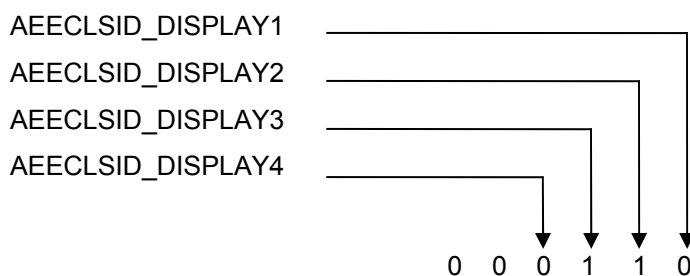


Figure 2. Bitwise representation of TouchDisplays setting

This example is represented as TouchDisplay = 6 in the oemcfg.txt file. When set to 0 (default), the displays are not touch-enabled. When set to 1, the display is touch-enabled.

BamEnabledDisplays

The BamEnabledDisplays setting specifies the displays that can support Application Manager operation. Displays should not be enabled that are smaller than the minimum specified size or expose insufficient keys to allow proper Application Manager operation.

Figure 3 shows an example of the BamEnabledDisplay setting that indicates AEECLSID_DISPLAY1 and AEECLSID_DISPLAY2 are capable of supporting the operation of Application Manager in some manner and the other displays are not.

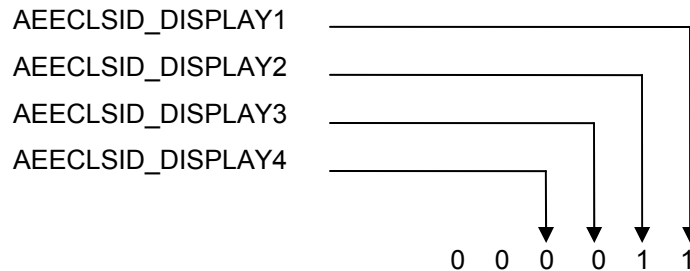


Figure 3. Bitwise representation of BamEnabledDisplays setting

This example is represented as BamEnabledDisplays = 3 in the oemcfg.txt file. The default value for this setting is 1 (Application Manager only runs on DISPLAY1, the primary display).

CancelOnClose

The CancelOnClose setting is a Boolean value that determines whether Application Manager does one of the following when the flip or slider is closed:

- Cancels a non-download operation or finishes downloading a new application, and closes Application Manager
- Finishes downloading a new application and transfers the operation of Application Manager to the secondary display

This setting is dependent on the BamEnabledDisplays setting on page 8, meaning that the operation only continues if the newly-activated display is enabled for an Application Manager operation. If a flip event is received while a download is in progress, rather than exiting the application per BamEnabledDisplays setting on page 8, the download completes and the settings take effect. The default value for this setting is true (i.e., operations will be canceled on flip or slide).

AppsUpdatedURL

The AppsUpdatedURL setting is a string value that specifies the return communication (postback URL) used by Application Manager to notify an OEM shortcut application that the application list (applist.lst) has been updated. The default value for this setting is empty.

KeyMapDisplay1-4

Application Manager requires four navigation keys—up, down, select, or clear—to operate on a display. If those four keys are mapped differently on a device for the active display, the four keys must be remapped as up, down, select, or clear. If the display has the required keys available (or

if Application Manager does not run on that display because it has less than four keys), the keys do not need to be specified.

The KeyMapDisplay settings are integer values for mapping the keys. To map the keys, set the integer values for KeyMapDisplay1, KeyMapDisplay2, KeyMapDisplay3, and KeyMapDisplay4 to the AVK code associated with those keys. When that display is active, Application Manager uses those codes for up, down, select, and clear.

Table 3 shows sample settings that map forward, rewind, play, and stop to up, down, select, and clear.

Table 3. Sample KeyMapDisplay settings to operate Application Manager

AVK codes	Description
57458	AVK_FFWD key used as the up key
57459	AVK_RWD key used as the down key
57454	AVK_PLAY key used as the select key
57460	AVK_STOP key used as the clear key

This example is represented as KeyMapDisplay1=57458,57459,57454,57460 in the oemcfg.txt file.

Table 4 shows the default settings for the AVK keys.

Table 4. Default KeyMapDisplay settings for AVK keys

AVK_codes	Key
57393	AVK_UP
57394	AVK_DOWN
57397	AVK_SELECT
57392	AVK_CLR

HapticTone and HapticResponseDuration

OEMs can customize haptic behavior for actionable items on the screen by configuring the following settings:

- **HapticTone:** Tone played when touching an actionable item on the screen. This setting is a valid value to be passed for ISound_PlayTone, normally defined in AEEISound.h.
- **HapticResponseDuration:** Duration of the haptic response in milliseconds (i.e., the tone)

To disable the haptic feature

Do one of the following:

- Remove either of the two haptic configurable variables from the oemcfg.txt file.
- Set either value to zero.

Understanding application shortcuts and statistics

The applist.lst file contains a list of applications managed by Application Manager, along with additional information for each application. This information can be used by the OEM shortcut application to send a notification to Application Manager for new shortcut settings and enables an OEM shortcut application to retrieve a list of installed applications.

The following is an example of an applist.lst file:

```
UPGRADE01|0x102DB98|0|0|0
INSTALL04|0x102DBA6|0|0|0
INSTALL03|0x102DBA4|0|0|0
HELLO_RECALL_M_01|0x102D134|0|1|886799017
HELLO_ENABLE_01|0x102CA95|0|0|0
HELLO_LICENSE_02|0x102C300|0|0|0
Hello World App|0x1009FF0|0|0|0
```

Each line entry in the file represents the data for a single application. The information is separated into five columns separated by a | (pipe). The applist.lst file is created on the device and stored in fs:/mod/brewappmgr.

Table 5 shows the information contained in the five columns of a line entry in the applist.lst file.

Table 5. Description of a line entry in applist.lst file

Column	Description
1	Application name
2	Application class ID
3	Specifies whether the application has a shortcut: 0 = no shortcut 1 = a shortcut
4	Number of times Application Manager launched the application
5	Last time Application Manager launched the application (in seconds) since January 6, 1980

Shortcut application interface commands

Application Manager handles the commands described in Table 6 from external applications that are initiated using the BREW API ISHELL_PostURL or an equivalent mechanism.

Table 6. Commands for shortcut application interface

Command	Description	Notes
BAM:ListApps	Application Manager creates the applist.lst file.	After the file is created, Application Manager uses ISHELL_PostURL to post the command, specified by the AppsUpdatedURL setting, in the oemcfg.txt file.

Command	Description	Notes
BAM:Shortcut	<p>Application Manager notes in the applist.lst file whether the application specified has a shortcut. The following is the full command followed by an explanation of the parameters:</p> <pre>BAM:Shortcut=<clsid>&active=<true or false></pre> <p>clsid = class ID in base-10 decimal (not base-16 hexadecimal) of the application that is being updated</p> <p>true = has a shortcut</p> <p>false = remove active shortcuts</p> <p>For example:</p> <pre>BAM:Shortcut=123567890&active=true</pre>	<p>After Application Manager updates the applist.lst file, it uses ISHELL_PostURL to post the command, specified by the AppsUpdatedURL setting, in the oemcfg.txt file.</p>
BAM:LaunchApp	<p>Application Manager launches the specified application. The following is the full command followed by an explanation of the parameter:</p> <pre>BAM:LaunchApp=<clsid></pre> <p>clsid = class ID in base-10 decimal (not base-16 hexadecimal) of the application to launch</p>	<p>Because Application Manager launches the application, the applist.lst file is updated with the number of launches and the time of the last launch.</p> <p>There is no response from Application Manager when it receives this command.</p>

Creating a shortcut

Application Manager tracks BREW applications with associated shortcuts and notifies users that a shortcut exists when removing one of these applications. For a shortcut application to specify to Application Manager that a shortcut exists, the following occurs:

- The shortcut application posts a command to have Application Manager create or update the applist.lst file, using the following command with no arguments:

```
BAM>ListApps
```

- When Application Manager receives this message, it enumerates the BREW applications resident on the device and writes the list to applist.lst in the fs:/mod/brewappmgr folder. When this operation is complete, Application Manager uses ISHELL_PostURL to post the command specified in the oemcfg.txt file for AppsUpdatedURL.
- The shortcut application reads the applist.lst file to obtain the list of BREW applications available on the device and allows the user to create a shortcut to a BREW application.
- To provide information to Application Manager about the shortcut, the shortcut application must send the following post URL command:

```
BAM:Shortcut=<clsid>&active=true
```

The clsid value is obtained from the applist.lst file for the application to which the shortcut is being created.

- Application Manager once again updates the applist.lst file indicating that the specified BREW application now can be launched using a shortcut.
- Application Manager uses ISHELL_PostURL to post the command specified in the oemcfg.txt file for AppsUpdatedURL.

Deleting a shortcut

The shortcut application uses a mechanism similar to creating a shortcut for indicating to Application Manager that a shortcut is no longer associated to an application.

- The shortcut application posts the following command:

```
BAM:Shortcut=<clsid>&active=false
```

Application Manager then updates the applist.lst file to remove the shortcut from the application. The <clsid> value is obtained from the applist.lst file for the BREW application.

- After the applist.lst file is updated, Application Manager uses ISHELL_PostURL to post the command specified in the oemcfg.txt file for AppsUpdatedURL.

Application Manager shortcut deletion notifications

If a device user deletes or unsubscribes to an application that has a specified shortcut, the following occurs:

1. Application Manager checks the information in the applist.lst file to determine whether a shortcut exists for an application.
2. If a shortcut exists, the device user is warned that a shortcut exists and is asked to confirm the deletion.
3. If the user confirms the deletion of the application, Application Manager updates the applist.lst file to remove the specified application.
4. Application Manager uses ISHELL_PostURL to post the command that is specified in the oemcfg.txt file for AppsUpdatedURL.
5. After receiving the posted command, the shortcut application reads the applist.lst file to ascertain the change.
6. After determining that the BREW application and associated shortcut no longer exist, the shortcut application removes the OEM shortcut.

Launching a shortcut application

After receiving an indication that a BREW application needs to be launched from a shortcut, the OEM application can do one of the following:

- Launch the BREW application directly using whatever means available (presumably ISHELL_StartApplet or some similar API). Under this scenario, the OEM application would handle launch errors, etc. In addition, the Application Manager application tracking mechanism would not track the number of launches or the last launch time.

- Use ISHELL_PostURL or a similar mechanism to post a command similar to the following with the value for <clsid> coming from the applist.lst file:

```
BAM:LaunchApp=<clsid>
```

Application Manager starts up and handles the launch of the specified application and presents to the user any launch failures. In addition, the application tracking feature of Application Manager properly tracks the number of launches and the last launch time for the BREW application.

Using application tracking statistics

The applist.lst file contains tracking information maintained by Application Manager on each launch of a BREW application. An OEM application needs to read the contents of this file to extract the application tracking statistics for each BREW application.

As noted in [Launching a shortcut application](#) on page 13, BREW application launches must be handled by Application Manager (either through the UI or from the BAM:LaunchApp command) for the application tracking feature to function. The fourth field in each applist.lst line entry specifies the number of times that the application has been launched after being downloaded or installed. The fifth field specifies the date and time of the most recent launch (represented as the number of seconds since January 6, 1980). A value of zero (0) in these fields indicates that Application Manager has not tracked any launches of this application after it was downloaded.

3 Integrating Application Manager into Device Builds

After the customization is complete, you can integrate the Application Manager and its components. This is described in the section on integrating the Application Manager in the *BREW® Porting Guide for MSM™ Platforms* provided with the BREW PK. Qualcomm recommends integrating the components using the persistent file system.

Application Manager integration

Integrate Application Manager into your device build by adding the application and its resources to the BREW persistent file system. [Table 7](#) shows the resources and pathnames where they must be located in the BREW file system.

Table 7. Application Manager files that must be built as BREW persistent files

From the installation package location...	Copy files...	Into the device location...
<install-dir>\BREW Application Manager\2.1.0\brewappmgr	brewappmgr.mif	fs:/mif/brewappmgr.mif
	brewappmgr\brewappmgrcommon.bar	fs:/mod/brewappmgr/br ewappmgrcommon.bar
	brewappmgr\brewappmgrln.bar	fs:/mod/brewappmgr/br ewappmgrln.bar
	brewappmgr\brewappmgrls.bar	fs:/mod/brewappmgr/br ewappmgrls.bar
	brewappmgr\oemcfg.txt	fs:/mod/brewappmgr/oe mcfg.txt
	brewappmgr\brewappmgr.mif	fs:/mod/brewappmgr/br ewappmgr.mif

The source files can be built statically. See the section on enabling BREW features in the *BREW® Porting Guide for MSM™ Platforms*. It is recommended that this process be automated through a suitable build system (e.g., make).

Integrating aeemenuctlex

If the device is touch-enabled, integrate the aeemenuctlex into the BREW persistent file system, but do not integrate the aeemenuctlex statically. [Table 8](#) shows the resources and the pathname where it must be located in the BREW file system.

Table 8. aeemenuctlex files that must be built as BREW persistent files

From the installation package location...	Copy files...	Into the device location...
<install-dir>\BREW Application Manager\2.1.0\aeemenuctlex	aeemenuctlex.mif	fs:/mif/aeemenuctlex.mif
	aeemenuctlex\aeemenuctlex.mod	fs:/mod/aeemenuctlex/aeemenuctlex.mod

Integration verification

After integrating Application Manager and its components, ensure that all the test cases pass as described in the *BREW® OEM Note: Verification of BREW Application Manager Integration* located on the BREW OEM extranet. See also [PEK Tests](#) on page 17 for more information.

Appendix A PEK Tests

Table 9 lists the recommended minimum set of PEK tests that need to be run to verify the porting of BREW and the integration of Application Manager to a device.

Table 9. PEK tests

PEK tests	Description
OATPointer	Tests the touch support on the device.
OAT Device	Tests system configuration and device configuration parameters, and logs configuration values. You must proactively verify the values to determine whether a test has passed or failed.
OAT Download	Tests the BREW application download and control functionality. It verifies the download, disable, remove, and application recall functionality.
OAT Web	Tests IWeb, IWebOpts, IWebResp, and IWebUtil interfaces.

Appendix B Acronyms and Terms

This document uses the following acronyms and terms.

BAM	BREW Application Manager
OAT	Operational Acceptance Tests
PEK	Porting Evaluation Kit
PK	Porting Kit
SDK	Software Development Kit

Index

A

Acronyms and terms, 18

Application

- shortcuts, 11

- touchmenuctl, 16

- tracking statistics, 14

Application Manager

- customizing, 7

- installing, 6

- integrating, 15

- shortcut deletion notifications, 13

- verifying integration, 17

AppsUpdatedURL setting, 9

B

BamEnabledDisplays setting, 8

C

CancelOnClose setting, 9

Creating shortcuts, 12

Customizing Application Manager, 7

D

Deleting shortcuts, 13

Device-specific configuration, 7

Disabling haptic feature, 10

Displays supported by BREW, 7

F

File, oemgfg.txt, 7

H

Haptic behavior. disabling, 10

HapticResponseDuration setting, 10

HapticTone setting, 10

I

Installing Application Manager, 6

Integrating

Application Manager, 15
touchmenuctl, 16
verification, 16, 17

K

KeyMapDisplay setting, 9

L

Launching shortcut application, 13

O

oemcfg.txt file, 7

P

PEK Tests, 17

R

Required components, 5

S

Setting

AppsUpdatedURL, 9
BamEnabledDisplays, 8
CancelOnClose, 9
HapticResponseDuration, 10
HapticTone, 10
KeyMapDisplay, 9
TouchDisplays, 8

Shortcuts

about, 11
creating, 12
deleting, 13
deletion notifications, 13
interface commands, 11
launching, 13

Statistics, application tracking, 14

T

TouchDisplays setting, 8

V

Verifying

BREW porting, 17
integration, 16, 17