



**Binary Runtime Environment for Wireless™**

**BREW SDK™ 2.0.0.38**  
发布声明



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

这些发布声明用于 Windows 平台的 BREW SDK™，软件版本 2.0.0。这些发布声明及其中所述之 BREW SDK 软件受版权保护，并保留所有权利。未经 QUALCOMM Incorporated 明确的书面许可，不得复制这些发布声明和 BREW SDK 软件，除非软件许可证中另有规定。

版权所有 ©2002 QUALCOMM Incorporated

保留所有权利

美国印制

本文档提供的所有数据和信息属于 QUALCOMM Incorporated 的机密，并已声明保留所有权利。接受本资料，即表示接受方同意本资料中包含的信息为机密信息，且未经 QUALCOMM Incorporated 的书面许可，不使用、复制，也不以任何方式向他人透露本资料的部分或全部内容。

美国政府可能会限制本技术的出口。严禁违反美国法律的转移行为。

Binary Runtime Environment for Wireless、BREW、BREW SDK、TRUE BREW、BREWStone、MobileShop、Eudora 和 PureVoice 是 QUALCOMM Incorporated 的商标。

QUALCOMM 是 QUALCOMM Incorporated 注册的商标及服务标志。

Bluetooth 是 Bluetooth SIG, Inc 的商标。

Microsoft、Windows、Visual Studio 和 Sound Recorder 是 Microsoft Corporation 在美国和（或）其他国家（地区）的注册商标或商标。

Macintosh 是 Apple Computer, Inc. 的注册商标。

UNIX 和 X-Windows 是 The Open Group 的商标。

Adobe、Acrobat 和 Photoshop 是 Adobe Systems Incorporated 的商标。

在此引用的所有商标和注册商标归其各自的所有者所有。

BREW SDK™ 2.0.0.38 Release Notes

70-D4152-2SC 2.0.0.38 版

October 24, 2002

## 简介—重要事项! 6

## 安装说明 7

最低系统要求 7

安装 BREW SDK 2.0.0 7

添加单个 SDK 组件 8

Microsoft Visual Studio 工具菜单设置 9

卸载全部 SDK 组件 9

卸载过程说明: 9

同一计算机上共存多个 BREW SDK 版本的影响 9

## BREW 2.0.0 API 更改内容 11

2.0.0.14 之前版本的 API 更改内容 11

IBitmap 11

IClipboard 12

动画图像 12

ISHELL\_RegisterHandler() 12

wParam 参数 13

IShell 13

IFont 14

多项目选择 14

AEERasterOp 14

ITEXTCTL\_GetInputMode() 15

ITAPI 15

IImageCtl, ClassID AEECLSID\_IMAGECTL 15

AEEAppInfo 15

新增接口 16

新增助手函数 16

优先级机制 16

屏幕保护小程序支持 17

IDateCtl 19

IDisplay	19
IDialog	20
ISTATIC_IsScrollable()	20
ITimeCtl	20
IPosDet	20
多媒体扩展	20
EVT_APP_STOP	21
2.0.0.14 版到 2.0.0.33 版的 API 更改内容	22
2.0.0.33 版到 2.0.0.38 版的 API 更改内容	24

## **BREW SDK 2.0.0 更改内容 25**

2.0.0.14 之前版本的 SDK 更改内容	25
BREW Emulator 中的键盘输入支持	25
BREW Emulator	27
BREW Device Configurator	28
TAPI 模拟	29
IMEDIA_Record() 模拟	29
BREW MIF 编辑器	29
网络速度模拟	29
GPS 模拟	30
BREW 联机帮助	30
BREW 2BitTool	30
BREW 资源编辑器	30
BREW MIF 编辑器	31
示例应用程序	31
2.0.0.14 版到 2.0.0.33 版的 SDK 更改内容	31
BREW 资源编辑器	32
BREW MIF 编辑器	32
BREW Emulator	32
BREW Compressed Image (BCI) 编辑工具	33
BREW Device Configurator	33
其它	33
2.0.0.33 版到 2.0.0.38 版的 SDK 更改内容	33
BREW MIF 编辑器	33
BREW 资源编辑器	34
其它	34

## 已知问题 35

## 简介—重要事项！

BREW SDK™ 2.0.0.38 发布声明包含安装说明、卸载说明，以及自 1.1 版以来对 BREW SDK™ 所进行的更改（包括 API）。安装 **BREW SDK 2.0** 之前，请先阅读这些发布声明（“自述文件”）！

它们是中文 BREW 2.0 SDK 特别“临时”版的发布声明。创建该临时版的目的是为了使您能够在我们的翻译 SDK 的同时，及早开始 2.0 应用程序的开发工作。请在 2002 年底访问 <http://www.qualcomm.com/brew>，您将可以获得有关最终版 BREW 2.0 SDK（全中文文档和软件）的详细信息。

此临时 SDK 包括：

1. 中文版 2.0 软件
2. 中文版新增接口参考资料（仅 1.1 版以后添加的 API）
3. 中文版 1.1 文档（包括 1.1 API 参考资料）
4. 英文版 2.0 联机帮助

因此，要开始 2.0 应用程序开发，您可以使用中文版软件 (1)，并通过中文版新增接口参考资料 (2) 掌握新增接口，通过中文版 1.1 API 参考资料 (3) 掌握先前的可用接口。此外，您还可以使用完整的英文版 2.0 联机帮助系统 (4)。当然，在最终发布时，我们将为您提供全套 2.0 版中文文档。

我们计划在 2002 年末或 2003 年初向运营商提供 BREW™ 2.0 以用于商业开发。而对于希望使用这些先进功能的开发者，现在就可以使用该版本 BREW SDK 2.0 进行开发工作。虽然目前尚无 2.0 手机面市，但 2002 年底有望上市几款。有关 BREW 2.0 手机的供应信息，请参阅外联网上的“电话资料”。使用 **SDK 2.0** 开发应用于 2.0 手机的应用程序。

本文档组织为以下几节：

章节标题	说明
安装说明	本节说明此版软件的安装和卸载说明。
BREW 2.0.0 API 更改内容	本节论述 BREW API 在此版中的更改内容。
BREW SDK 2.0.0 更改内容	本节论述 BREW SDK 在此版中的更改内容。

# 安装说明

本节将提供安装和卸载说明，以及在同一台计算机上共存多个版本 SDK 将产生的影响。

## 最低系统要求

要安装 BREW SDK 2.0.0，需要 PC 运行：

- Windows 2000（已认证）
- Windows XP（已认证）
- Windows NT（支持停止）

**注意：**如果要安装亚洲（中文、日文或韩文）版本的 BREW SDK，必须将其安装在相同语言的操作系统上。例如，安装日文版 BREW SDK 时，必须将其安装在日文版 Windows 2000 或 Windows XP 中。

要使用 BREW SDK 开发应用程序，需要满足以下最低系统要求：

- Microsoft Visual Studio (Visual C++) 6.0（或更高版本）

**注意：**在 Windows 2000 或 Windows XP 计算机上运行安装程序，必须拥有管理员权限。

## 安装 BREW SDK 2.0.0

BREW SDK 2.0.0 软件使用基于 web 的安装程序。磁盘需求将在安装过程中确定。

要安装 BREW SDK，请在 BREW SDK 2.0.0 部分的下载网页上单击**获取 SDK**按钮，然后按照屏幕上的说明执行操作：

“安装 BREW SDK 之前，请先阅读以下说明

- 单击下面的链接，复制一些启动下载的文件后，开始安装 BREW SDK。

- 此时将开始安装 SDK。按照“安装向导”的提示，您可以有选择地安装 BREW SDK 组件。如果需要安装或重新安装在首次安装时没有安装的组件，您可以返回至该网页并单击下面的链接，或者通过 Windows 控制面板的“添加/删除程序”选项完成此操作。
- 请注意，只有在首次运行此安装程序时，系统才会提供选择目标文件夹的选项。以后每次运行此安装程序时，都将进入“维护/更新”模式，您可以选择组件并安装到与以前相同的文件夹中。
- 如果需要卸载该版本 SDK，您可以使用 Windows 控制面板的“添加/删除程序”选项，并在弹出的对话框中选择“删除”选项。”

然后，单击底部显示为“单击启动 BREW SDK 2.0.0 预览版安装程序”的链接。

**注意：** 建议您在将该版本 BREW SDK 2.0.0 安装至计算机之前，先删除旧版的 BREW SDK 2.0.x（例如 2.0.0.14 或 2.0.0.33 版）。如果存在旧版 BREW SDK 2.0.0，则新版 BREW SDK 2.0.0 的安装程序将会识别，并显示对话框提示您。如果接受默认安装，则新版 SDK 2.0.0 将覆盖现有的 2.0.0 文件夹。默认的安装文件夹以及“开始->程序”菜单中的默认快捷方式文件夹都将获得更新，以表示新版 2.0.0 的组件和文件。

但是，如果不接受默认的目标文件夹和默认的快捷方式文件夹，并将它们重新命名，则此版 BREW SDK 2.0.0 将安装在您选择的新文件夹中，且不修改旧版的 2.0.0 文件夹。但在“控制面板”的“添加/删除程序”中，“BREW SDK v2.0.0”条目将指向最新版的 2.0.0。

## 添加单个 SDK 组件

由于 BREW SDK 使用基于 web 的安装程序，所以必须连接到因特网才能从 BREW 下载网站添加更多 SDK 组件。

如果需要安装在首次安装时没有安装的 SDK 组件，您可以单击下载网页上的**获取 SDK** 按钮，并按照屏幕上的安装说明进行操作。您也可以打开“开始->设置->控制面板->添加/删除程序”，然后选择 BREW SDK 应用程序，来执行相同的任务。

请注意，只有在首次运行此安装程序时，系统才会提供选择安装 BREW SDK 的目标文件夹选项。以后每次运行此安装程序时，都将进入“维护/更新”模式，您可以选择组件并安装到与以前相同的文件夹中。

## Microsoft Visual Studio 工具菜单设置

- 在从 1.1 版开始的旧版 BREW SDK 中，Visual Studio 工具菜单中添加了四种设置（两种用于 ARM 插件，另外两种分别为 MIF 编辑器和资源编辑器的快捷方式）。从本版 2.0.0 SDK 开始，Visual Studio IDE 的“工具”菜单中只添加用于 ARM 插件的两种设置。
- 如果您已经安装 1.1 SDK，然后又安装了 2.0.0 SDK，则 ARM 插件工具设置所指向的路径将更新为最新的 2.0.0 版安装文件夹。MIF 编辑器和资源编辑器的快捷方式不作更改。
- 在有些情况下，“工具”菜单中并未添加插件（如 Visual Studio 或 ARM ADS 1.0/1.1/1.2 版编译器）。有关如何获取、启用或删除这些插件的详细信息，请参阅 2.0.0 版 SDK 的《BREW SDK 用户指南》。

## 卸载全部 SDK 组件

如果需要卸载此版本的 BREW SDK，您可以打开开始 > 设置 > 控制面板 > 添加/删除程序，然后选择 BREW SDK 应用程序进行卸载。

### 卸载过程说明：

- 如果安装了 BREW SDK 2.0.0 然后将其卸载，则将删除 Visual Studio IDE 中的全部插件和工具设置。
- 如果已经安装 1.1 SDK，然后又安装了 2.0.0 SDK，但首先卸载 2.0.0 SDK，则由于计算机上仍存在其它版本的 SDK，插件和工具设置将保留不变且不被删除。此时您需要执行 1.1 SDK 安装的 bin 文件夹中的 EnvVarSt.exe（复制于 2.0 bin 文件夹），将变量设置为您需要指向的任何 SDK 版本。如有必要，您还必须打开 Visual Studio IDE 的“工具->自定义”菜单设置，修改插件指向的路径（如果遇到问题）。
- 另请注意，如果在 Dev Studio 中启用了插件并且正在使用它们，则由于这些文件正在使用中，卸载程序将无法删除它们。您必须禁用这些插件，并从 Visual Studio IDE 的“工具->自定义”菜单中将其删除。有关插件的详细信息，请参阅《BREW SDK 用户指南》。

## 同一计算机上共存多个 BREW SDK 版本的影响

- 计算机上共存多个版本的 SDK 不会影响任何 BREW SDK 组件（Emulator、MIF 等）的运行。它们将从各自版本的文件夹中执行。

- 但计算机上共存多个版本的 SDK 会影响 SDK 随附的 Visual Studio 插件和 BREWAppWizard。SDK 安装程序将这些插件和 AppWizard 安装在 Microsoft Visual Studio 文件夹的 Add-in(s) 和 Template 文件夹中。
- 插件和 AppWizard 取决于安装 BREW SDK 时 SDK 安装程序设置的 BREWDIR 环境变量。安装程序实际设置 BREWDIR 变量，并执行文件 EnvVarSt.exe，将这些设置应用于系统中所有激活的 Windows 应用程序。EnvVarSt.exe 文件也安装在 SDK 安装的 bin 文件夹中。
- BREWDIR 变量始终设置为最后安装的 BREW SDK。因此，如果在计算机上安装了 1.1 SDK，则 BREWDIR 将被设置为安装 1.1 SDK 的文件夹（例如：C:\Program Files\BREW v11）。现在，如果安装 2.0.0 BREW SDK，BREWDIR 变量将被修改为指向安装 2.0.0 的文件夹。
- 如果要使插件和向导指向旧版的 1.1 SDK，您需要将 EnvVarSt.exe 从 2.0.0 SDK 安装的 bin 文件夹复制到 1.1 SDK 的 bin 文件夹中。我们以前在 1.1 SDK 中附带了 EnvVarSt.exe 文件，但您现在需要使用 2.0.0 SDK 所附的最新 EnvVarSt.exe 将其替换。将文件复制到 1.1 SDK 的 bin 文件夹后，请双击该文件，它将重置 BREWDIR 变量并将更改应用于所有激活的 Windows 应用程序。如果要尝试在 DOS 提示下查看设置的 BREWDIR 变量值，您可以关闭该会话并打开新的 DOS 提示来查看更改。
- 如果需要切回 2.0.0 SDK，请在安装的 2.0.0 SDK\bin 文件夹中双击 EnvVarSt.exe 文件。

## BREW 2.0.0 API 更改内容

本节列举 BREW 2.0.0 版中所有的 API 更改。由于 2.0.0 版分为两个预览版以及最终版，可能有些开发者需要从较早的预览版移植到 2.0.0 版，所以这些信息也被分为三个部分。下表列举了用于识别各个版本的编译号。

版本	编译号	参阅部分
第一预览版	2.0.0.14	2.0.0.14 之前版本的 API 更改内容
第二预览版	2.0.0.33	2.0.0.14 版到 2.0.0.33 版的 API 更改内容
2.0.0 最终版	2.0.0.38	2.0.0.33 版到 2.0.0.38 版的 API 更改内容

### 2.0.0.14 之前版本的 API 更改内容

以下列举了 2.0.0.14 版中的 BREW API 更改。

#### IBitmap

引入 IBitmap 接口，以提供一种操作位图的方法。该接口引入了以下新的 API。有关这些 API 的详细信息，请参阅《BREW API 参考资料》。

#### IBitmap

```
IBITMAP_AddRef()
IBITMAP_Release()
IBITMAP_ConvertColor()
IBITMAP_DrawPixel()
IBITMAP_DrawHScanline()
IBITMAP_DrawVScanline()
IBITMAP_FillRect()
IBITMAP_BitBlt()
IBITMAP_DibBlt()
IBITMAP_GetPointer()
IBITMAP_GetDimensions()
IBITMAP_CreateCompatibleBitmap()
IBITMAP_GetPixel()
```

```
IBITMAP_SetTransparencyColor()
IBITMAP_GetTransparencyColor()
```

## IDisplay

```
IDISPLAY_GetDeviceBitmap()
IDISPLAY_CreateDIBitmap()
IDISPLAY_SetDestination()
IDISPLAY_GetDestination()
```

## IGraphics

```
IGRAPHICS_SetDestination()
IGRAPHICS_GetDestination()
```

## IShell

```
ISHELL_LoadResBitmap()
ISHELL_LoadBitmap()
```

## IClipboard

引入 **IClipboard** 接口，以支持剪切和粘贴操作。该接口包含以下函数。有关这些 API 的详细信息，请参阅《BREW API 参考资料》。

```
ICLIPBOARD_AddRef()
ICLIPBOARD_Empty()
ICLIPBOARD_GetSequence()
ICLIPBOARD_HasFormat()
ICLIPBOARD_EnumInit()
ICLIPBOARD_EnumNext()
ICLIPBOARD_Register()
ICLIPBOARD_GetData()
ICLIPBOARD_SetData()
```

## 动画图像

如果图像为动画图像，则可以向 **Image\_DrawFrame()** 传递值为 -1 的 **nFrame** 参数，以绘制当前帧。

## ISHELL\_RegisterHandler()

**ISHELL\_RegisterHandler()** 的行为在 1.1 版 BREW 的基础上进行了更改。现在，如果应用程序注册了处理程序，并使其本身成为多用途网络邮件扩展 (MIME) 类型的

“默认”处理程序，则重新启动手持设备即可保存该信息，而无需重新运行该应用程序。

例如：

应用程序 A 和应用程序 B 均为支持 "mailto" 统一资源定位器 (URL) 的电子邮件应用程序。

应用程序 A 和应用程序 B 在它们的模块信息文件 (MIF) 中均包含与其相关联的 "mailto"。

由于应用程序 A 是第一个被处理的 MIF 文件，因此是“默认”处理程序。

运行应用程序 B，它将使用其 ClassID 更新处理程序。

对于 1.1 版和其它旧版的 BREW，重置手持设备不保存应用程序 B 所作的注册更新。该行为现已更改，1.1 版以后的 BREW 通过重新启动手持设备可保存注册的最终更新。

即使删除了应用程序 B，应用程序 B 条目的注册仍将得到更新。

## wParam 参数

作为 EVT\_APP\_SUSPEND 事件一部分进行传递的 wParam 参数现可指定暂停应用程序的原因。当前包括的事件如下：

- AEE\_SUSPEND\_NORMAL：由于启动其它应用程序而暂停该应用程序。
- AEE\_SUSPEND\_EXTERNAL：由于外部请求而暂停该应用程序。它通常是 OEM/设备发起的暂停调用。

## IShell

对 IShell 接口进行了以下更改。有关这些 API 的详细信息，请参阅《BREW API 参考资料》。

- 引入 ISHELL\_StartAppletArgs()，以启动指定的 32 位 ClassID 所关联的小程序（通过向函数传递参数）。
- 引入 ISHELL\_DetectType()，以启用数据类型检测。当前，它可以检测基于 IMedia 的对象所对应的数据类型。
- SDK 以及支持定位的手持设备现在均支持 ISHELL\_GetPosition()。

## IFont

引入 IFont 接口，以支持字体的使用。与该接口一起引入了以下函数。 有关这些 API 的详细信息，请参阅《BREW API 参考资料》。

```
IFONT_AddRef()
IFONT_DrawText()
IFONT_MeasureText()
IFONT_GetInfo()
IFONT_QueryInterface()
IFONT_Release()
```

## 多项目选择

在 AEECLSID\_MENUCTL 和 AEECLSID\_SOFTKEY 类中引入选择多个项目的功能。通过设置新属性 MP\_MULTI\_SEL 可启用此功能。

为支持该功能，引入了以下方法：

```
IMENUCTL_EnumSelInit
IMENUCTL_EnumNextSel
IMENUCTL_SetSelEx
IMENUCTL_SetFocus
IMENUCTL_GetFocus
```

有关详细信息，请参阅 AEE IMenuCtl 属性和 IMenuCtl 接口文档。

## AEERasterOp

对 AEERasterOp 进行了以下更改：

- 不建议使用 AEE\_RO\_MASK（在 BREW 1.1 以前的版本中与 AEE\_RO\_TRANSPARENT 相同），并以 AEE\_RO\_OLDMASK 替换。
- 不建议使用 AEE\_RO\_MASKNOT，并以 AEE\_RO\_ANDNOT 替换。

引入以下定义以确保向后兼容性：

```
#define AEE_RO_MASK          AEE_RO_TRANSPARENT
#define AEE_RO_MASKNOT      AEE_RO_ANDNOT
```

以下为 BREW 2.0 AEERasterOp 枚举。

```
// 位图光栅操作类型
typedef enum
{
    AEE_RO_OR,
```

```

AEE_RO_XOR,
AEE_RO_COPY,
AEE_RO_NOT,
AEE_RO_OLDMASK,    // 在 BREW 1.1 以前的版本中, 与 AEE_RO_TRANSPARENT
相同; IBitmap 不支持
AEE_RO_MERGENOT,
AEE_RO_ANDNOT,
AEE_RO_TRANSPARENT,
AEE_RO_TOTAL
} AEERasterOp;

```

有关详细信息, 请参阅《BREW API 参考资料》中的 **AEERasterOp** 部分。

## ITEXTCTL\_GetInputMode()

引入 **ITEXTCTL\_GetInputMode()** API 以及 **AEETextInputModelInfo** 结构。有关详细信息, 请参阅《BREW API 参考资料》。

## ITAPI

引入以下 **ITAPI** API。有关这些 API 的详细信息, 请参阅《BREW API 参考资料》。

```

ITAPI_SendSMS()
ITAPI_IsVoiceCall()
ITAPI_IsDataSupported()
ITAPI_OnCallEnd()
ITAPI_OnCallStatus()

```

## ImageCtl, ClassID AEECLSID\_IMAGECTL

引入 **ImageCtl** (ClassID **AEECLSID\_IMAGECTL**), 允许调用程序在图像中显示可滚动视图。有关此接口的详细信息, 请参阅《BREW API 参考资料》。

## AEEAppInfo

**AEEAppInfo** 结构稍有更改, 并添加了两个新字段。这两个字段用于替换以前未使用的填充值。

```

uint16 wAppType;
uint32 dwLastUsedTime;

```

**wAppType**: 该 16 位值的用途是供运营商定义特殊的应用程序类型。

**dwLastUsedTime:** 该 32 位值包含用户使用小程序的持续时间（秒）。它可用来为将来的自动禁用功能确定最近最少使用（LRU）的应用程序列表。

## 新增接口

引入以下新接口。有关这些接口的详细信息，请参阅《BREW API 参考资料》。

IBitmap  
 IFont  
 IClipboard  
 ISprite  
 ITransform  
 IBTSDP (Bluetooth)  
 IBTDIOPORT (Bluetooth)

## 新增助手函数

引入以下新的助手函数。有关这些助手函数的详细信息，请参阅《BREW API 参考资料》。

JULIANTOSECONDS()  
 STRLCPY()  
 STRLCAT()  
 WSTRLCPY()  
 WSTRLCAT()  
 FASSIGN\_STR()  
 FASSIGN\_INT()

## 优先级机制

对 BREW 的优先级机制引入以下更改：

- 未更改现有的优先级。现有优先级（如 PL\_FILE 和 PL\_NETWORK）仍将保留并继续按原方式使用。所有 MIF 和应用程序均向后兼容优先级。
- 从 BREW 2.0 开始，采用以 ClassID 本身作为优先级的规则。这表示如果应用程序需要使用特定的保护类时，必须将该类明确声明为 MIF 中的依。
- 对于每个扩展类，保护扩展和非保护扩展的概念说明如下。

保护扩展：必须在该模块的应用程序或类的 MIF 依存部分中明确声明这些，才能创建此扩展的实例。

非保护扩展：任何模块都可以创建该类，而无需将其明确声明为 MIF 中的依存。

例如：如果将扩展类 E 开发为保护类，则应用程序必须将其声明为 MIF 中的依存才能使用该类。

同样的规则也适用于通过注册获取的类。例如，如果 ISHELL\_GetHandler() 返回保护的 ClassID，则请求程序必定已将该类声明为 MIF 中的依存才能创建它。BREW 的图像浏览器和 SoundPlayer 是非保护的，因此可以任意创建。

- 对于模块导出的每种扩展类，您可以在 MIF 中指定它是否为保护类。已增强的 BREW MIF 编辑器，可以在“扩展”选项卡下指定每个导出类是否为保护类。
- MIF 编辑器列出可用扩展，以从中选择作为模块的依存，使得扩展用户不必手动输入该扩展所依存的类的 ClassID。
- 只有在类当前可用且请求程序拥有创建该类所需的优先级时，函数 ISHELL\_QueryClass() 才为该返回 TRUE。
- 这种模式的结果是不能部分保护类。即，要么保护整个类，要么不保护整个类，而不可能只在接口或类的几个选定函数上应用优先级机制。

## 屏幕保护小程序支持

引入对屏幕保护小程序的支持。请注意下列事项：

### 注册屏幕保护程序

通过 ISHELL\_RegisterHandler 函数可以向外壳注册屏幕保护小程序。该函数的调用方式如下：

```
ISHELL_RegisterHandler(pShell, AEECLSID_SCREEN_SAVER,  
MT_SCREEN_SAVER, <屏幕保护小程序的 ClassID>)
```

其中：

AEECLSID\_SCREEN\_SAVER 位于 AEEClassIDs.h 中  
MT\_SCREEN\_SAVER 是 AEEMimeTypes.h 中的 ASCII 字符串

屏幕保护程序的超时值应设置如下：

```
AEEScreenSaverInfo si;  
si.wSecs = <超时值>  
ISHELL_SetPrefs(pShell, AEECLSID_SCREEN_SAVER, SSINFO_VER, &si,  
sizeof(si));
```

其中：

**SSINFO\_VER** 位于 **AEEShell.h** 文件中，为首选资源的版本。  
**AEEScreenSaverInfo** 是在 **AEEShell.h** 中定义的一个结构，包含有关秒数和范围的信息。请注意，通过 **ISHELL\_GetPrefs** 函数可以查询该结构。

*AppManager - 应用程序管理器查找和显示可用屏幕保护应用程序列表的方式*

应用程序管理器（或其它任何应用程序）可通过 **ISHELL\_EnumApplet** 和 **ISHELL\_EnumNextApplet** 函数查找支持屏幕保护程序功能的应用程序。屏幕保护应用程序必须支持其 **AEEAppInfo** 中的 **AFLAG\_SCREEN\_SAVER** 标记。它允许应用程序管理器为用户枚举和显示可用屏幕保护程序的列表。如下所述，应用程序还应设置 **AFLAG\_HIDDEN** 标记，从而不出现在应用程序管理器所显示的应用程序列表。

*编写屏幕保护应用程序*

除了以下几点之外，屏幕保护应用程序与其它任何 **BREW** 应用程序都相同：

- 它应设置 **MIF** 中 **AEEAppInfo** 结构的 **AFLAG\_SCREEN\_SAVER** 和 **AFLAG\_HIDDEN** 位。
- 如果正常启动，它将不接收按键事件（例如 **EVT\_KEY**）。
- 如果作为屏幕保护程序启动，它将接收 **EVT\_APP\_START** 消息。
- 它还应支持 **EVT\_APP\_CONFIG** 启动消息。这使得它可以执行任何必要的配置。

*屏幕保护程序交互*

当屏幕保护程序计时器到期而无其它用户活动时，在下列情况下将启动屏幕保护程：

- 当前活动的 **BREW** 应用程序指示它“可以”关闭或暂停。
- 没有正在进行中的活动的声音/数据调用。

- 如果按下按键，屏幕保护程序将按以下方式关闭：
  - 如果屏幕保护程序下无其它 **BREW** 应用程序，则将从按键事件返回 **FALSE**，以便原始设备制造商 (OEM) 使用该键。
  - 如果存在被暂停的 **BREW** 应用程序，则在应用程序重新启动时将按键传递给它们。

## IDateCtl

IDateCtl 发生以下更改。

IDateCtl 接口中引入以下新的 API 以及新属性 **DP\_CENTERED**。有关详细信息，请参阅《BREW API 参考资料》。

```

IDATECTL_SetFont()
IDATECTL_GetFont()
IDATECTL_SizeToFit()

```

为 IDATECTL\_GetDateString 引入以下附加 dwDateFormat

```
DFMT_INT_YYYY_MM_DD "2001.06.01"
```

支持 IDATECTL\_SetProperties，且属性掩码的低 8 位指定控件的日期格式。默认日期格式为 DFMT\_DD\_MON\_YY。

## IDisplay

IDisplay 发生以下更改：

- **IDisplay 多个实例：**IDisplay 现在可支持多个实例。这表示每次使用 **AEECLSID\_DISPLAY** 调用 **ISHELL\_CreateInstance()** 时，都将返回一个应用程序全局 IDisplay 实例。该实例可以维护状态信息，如字体、颜色、目标位图和剪切矩形（新增）。而且，还可以复制 IDisplay 然后修改，使得小程序的不同部分均由单独的 IDisplay 进行绘制，从而避免每次绘制操作后都需要恢复状态数据。为了支持该行为，IDisplay 接口中添加了 **IDISPLAY\_Clone()**。此外，还添加了新的 **ClassID AEECLSID\_DISPLAYCLONE**，以生成当前应用程序全局 IDisplay 的副本。
- **IDisplay 剪切矩形：**添加 **IDISPLAY\_SetClipRect()** 以设置限制所有 IDisplay 绘制函数的剪切矩形。要将剪切矩形重置为全屏幕，请在 (**AEERect\***) 参数中传递 **NULL**。此外还添加了 **IDISPLAY\_GetClipRect()**。

## IDialog

已增强的 IDialog 允许应用程序注册事件处理程序，以接收对话框未处理的所有事。在以前的版本中，使用 IDIALOG\_SetEventHandler() 注册的事件处理程序只能接收来自对话框的事件（EVT\_DIALOG\_END、EVT\_COMMAND）。这种增强的目的是为了扩展功能，使得事件处理程序能够接收当前活动对话框未处理的输入事件（即按键事件）。

有关此功能增强的详细信息，请参阅《BREW API 参考资料》中的 IDIALOG\_SetProperties()、IDIALOG\_GetProperties() 和 IDIALOG\_SetEventHandler() 文档。

## ISTATIC\_IsScrollable()

ISTATIC\_IsScrollable() 是 IStatic 接口中引入的新 API，同时还引入了新属性：ST\_ENABLETAB 和 ST\_ENABLE\_HLGHT。有关详细信息，请参阅《BREW API 参考资料》。

## ITimeCtl

ITimeCtl 接口中引入了以下新的 API 以及新属性 TP\_LEFT\_JUSTIFY。

```
ITIMECTL_SetFont()
ITIMECTL_GetFont()
ITIMECTL_SizeToFit()
```

## IPosDet

已增强的 IPosDet 接口可支持全球定位系统 (GPS) 定位和向量信息。该接口中添加了以下新的 API。有关这些 API 的详细信息，请参阅《BREW API 参考资料》。

```
IPOSDET_GetGPSConfig()
IPOSDET_SetGPSConfig()
IPOSDET_GetGPSInfo()
```

## 多媒体扩展

在 BREW 2.0 中，开发者（特别是 OEM）可通过从 IMedia 派生需要的内容处理程序接口，来编写自己的可插入式多媒体扩展作为 BREW 接口。IMedia 是一种抽象基，允许接口启用多媒体内容（音频/视频）的播放和录制。

对于移动基站调制解调器 (MSM™) 平台，BREW 2.0 的 CMX 与 2.2 版 Compact Media Extension™ (CMX™) 相兼容，即表示已实现以下接口：

**IMediaMIDI:** 播放标准的乐器数字化接口 (MIDI) 格式。

**IMediaMP3:** 播放 MPEG 音频层 3 (MP3) 格式。

**IMediaQCP:** 播放和录制 QUALCOMM PureVoice™ (QCP)。

**IMediaPMD:** 播放 WebAudio 内容（包括图像、MIDI/QCP 音频和卡拉 OK 文本）。

**IMediaMIDIOutMsg:** 通过 MIDI 设备播放直接的 MIDI 消息。

**IMediaMIDIOutQCP:** 通过 MIDI 设备播放 QCP。它属于前景音乐，可以同时播放四个 QCP 文件，非常适用于游戏。

请注意，应用程序开发者通常需要处理 **IMedia API**。如果要对接口执行一些特定操，他们可能需要访问这些具体的接口。

## EVT\_APP\_STOP

为 **EVT\_APP\_STOP** 添加支持关闭后台应用程序的参数（以前为空）。该事件现传递指向布尔值的指针（初始值为 **TRUE**）。如果需要异步关闭应用程序（即在后台运行并在稍后终止其本身），则可以将该布尔值设置为 **FALSE**。稍后，应用程序便可以调用 **ISHELL\_CloseApplet** 来关闭自身。

例如：

```
static void MyApp_ReadyToClose(MyApp * pme)
{
    pme->m_bReadyToClose = TRUE;
    ISHELL_CloseApplet(pme);
}

static boolean MyApp_HandleEvent(MyApp * pme, AEEEvent evt, uint16
wp, uint32 dwParam)
{
    switch(evt){
        ...

        case EVT_APP_STOP:
            if(!pme->m_bReadyToClose){
                boolean * pb = (boolean *)dwParam;
                if(pb)
                    *pb = FALSE;
                MyApp_StartClosing(pme);
            }
            return(TRUE);

        ...
    }
}
```

此更改向后兼容所有旧版本。一般的应用程序不必留意该布尔值。

## 2.0.0.14 版到 2.0.0.33 版的 API 更改内容

以下列举了自 2.0.0.14 版 BREW 至 2.0.0.33 版 BREW 的 API 更改：

- 为对话框添加了 **IStatic** 和 **ImageCtl** 支持。资源编辑器可用来创建包含 **IStatic** 和（或）**ImageCtl** 的对话框，**BREW API** 可用来显示这些对话框。
- 添加了新函数 **ISHELL\_GetDeviceInfoEx()**，以获取扩展设备信息。
- 在 **BREW API** 和 **SDK** 中添加了对中文编码 (GB-2312) 的支持。
- **BREW** 支持限制文件总数及模块占用文件空间总量的功能。每个模块都可在 **MIF** 中指定这些限制。**BREW** 中的文件系统 **API** 将确保模块在写入文件系统时不会超出这些限制。该功能有助于在文件空间较少的情况下测试应用程序。为支持此功能，**IFileMgr** 接口中添加了 **API** **IFILEMGR\_GetFileUseInfo()**。
- **BREW** 支持一种机制，可跟踪共享目录中的文件，并在不需要这些文件时将其删除。需要使用共享目录中文件的应用程序可通过 **API** **IFILEMGR\_Use()** 告知 **BREW**。当应用程序不再需要共享目录中的文件，可通过 **IFILEMGR\_UnUse()** 告知 **BREW**。删除任何应用程序时，**BREW** 都将检查共享目录中是否存在任何需要清除的文件。共享目录中没有注册用户的文件将被删除。为实现此功能，**IFileMgr** 接口中添加了两个新的 **API**：**IFILEMGR\_Use()** 和 **IFILEMGR\_UnUse()**。
- **BREW** 允许检索文件的扩展信息，其中包括各个文件所关联的长字符串说明。为实现此功能添加了以下 **API**：**IFILEMGR\_GetInfoEx()**、**IFILE\_GetInfoEx()**、**IFILEMGR\_SetDescription()** 和 **IFILEMGR\_EnumNextEx()**。
- 添加了名为模拟时钟控件 (**IAClockCtl**) 的新控件。
- 添加了对 **Bluetooth** 的支持。
- 在 **IDialog** 接口中添加了新函数 **IDIALOG\_Redraw()**。该函数用于刷新对话框。
- 已增强的 **IGraphics** 接口可支持绘制椭圆弧和圆角矩形。在 **IGraphics** 接口中添加了两个新的 **API**：**IGRAPHICS\_DrawEllipticalArc()** 和 **IGRAPHICS\_DrawRoundRectangle()**。

- 对 BREW 安全 API 进行了以下增强：
  - SSL 支持
  - 新增 IHashCTX 接口： 该接口的函数用于散列数据。
  - 通过新增以下函数增强了 IRSA 接口： IRSA\_Decrypt()、IRSA\_Encrypt()、IRSA\_ModExp() 和 IRSA\_VerifySig()。
  - 通过新增以下函数增强了 IX509Chain 接口： IX509CHAIN\_GetOpt()、IX509CHAIN\_RemoveOpt() 和 IX509CHAIN\_VerifyV()。
- 通过新增以下函数增强了 IHTMLViewer 接口：  
IHTMLVIEWER\_GetViewState() 和 IHTMLVIEWER\_SetViewState()。
- 通过新增以下函数增强了 IImage 接口： IIMAGE\_SetFrameSize()。
- 通过新增以下函数增强了 INetMgr 接口： INetMgr\_SetOpt() 和 INetMgr\_GetOpt()。 这些函数可以提供配置特定网络选项的功能。
- 新增接口 IRamCache。该接口利用 TTL 提供有限大小和有限项目的 LRU 缓存，并在堆上存储其中的记录。
- IShell 增强： 在 IShell 接口中新增了以下函数：
  - ISHELL\_GetResSize()。
  - 添加了若干通知注册函数，使得应用程序可以注册在发生特定事件时接收通知，如： ISHLL\_OnInit()、ISHELL\_OnExit()、ISHELL\_OnLowRAM()、ISHELL\_OnLowStorage() 和 ISHELL\_OnModUnload()。
  - ISHELL\_RegisterSystemCallback()。
  - ISHELL\_SetTimeEx()。它将使用 AEETCallback 对象而不是函数指。
- 在 ITextCtl 接口中新增了两个函数： ITEXTCTL\_EnumModelnit() 和 ITEXTCTL\_EnumNextMode()，以枚举设备上可用的文本输入模式。
- 新增 IVocoder 接口以捕获和播放声码器帧。
- BREW 现在执行严格的堆验证。重用已释放的指针，多次释放同一指针或其它类似操作将导致立即重置设备。

## 2.0.0.33 版到 2.0.0.38 版的 API 更改内容

以下列举了从 2.0.0.33 版到 BREW 2.0.0 最终版的 API 更改。

- 此版本的 BREW API 参考资料文档中增加了 SSL 错误代码列表。
- ISHELL\_LoadResImage() API 现在可用来加载 JPEG 图像。 有关详细信息，请参阅“BREW SDK 2.0.0 更改内容”部分的“BREW 资源编辑器”。

## BREW SDK 2.0.0 更改内容

本节列举 BREW 2.0.0 版中所有的 BREW SDK 更改。由于 2.0.0 版分为两个预览版以及最终版，可能有些开发者需要从较早的预览版移植到 2.0.0 版，所以这些信息也被分为三个部分。下表列举了用于识别各个版本的编译号。

版本	编译号	参阅部分
第一预览版	2.0.0.14	2.0.0.14 之前版本的 SDK 更改内容
第二预览版	2.0.0.33	2.0.0.14 版到 2.0.0.33 版的 SDK 更改内容
2.0.0 最终版	2.0.0.38	2.0.0.33 版到 2.0.0.38 版的 SDK 更改内容

### 2.0.0.14 之前版本的 SDK 更改内容

以下列举了 2.0.0.14 版中的 BREW SDK 更改。

#### BREW Emulator 中的键盘输入支持

添加了使用 BREW Emulator 进行 BREW 键盘输入的支持。

BREW Emulator 现在可以传递任何字符事件，但不能传递按键事件，除非按键位于设备图像上。这使得用户能够快速地将字符消息发送到应用程序，以配合使用文本和菜单控件。如果设备图像具有键盘布局，通常的消息序列如下：

```
EVT_KEY_PRESS  
EVT_KEY  
EVT_CHAR  
EVT_KEY_RELEASE
```

如果设备图像中没有某个字符的按键，则只发送 EVT\_CHAR 和 EVT\_UPDATE\_CHAR 消息，而不发送“按键”事件。副作用是，字母上方的 0-9 按键现对应于字符 0-9，您不能按照习惯的方式按下这些按钮。要使用键盘按下这些按钮，请使用开启 NUMLOCK 的数字小键盘。

EVT\_CHAR 是在发送字符时发出的消息。它出现在发送 EVT\_KEY 消息以后。EVT\_CHAR 消息的 wParam 是一个 AECHAR，对应于键入的字符和所使用的设备图像编码。如果加载 LG 设备图像，它将是 KSC6501 编码字符。如果使用 Unicode 设备图像并输入中文，则为中文 Unicode 字符（确保使用中文字体，否则将显示为空白框）。EVT\_CHAR 消息的 dwParam 是修改键状态的位标记掩码。AEEVCodes.h 中包含这些标记的定义和使用该位掩码获取按键状态的宏。

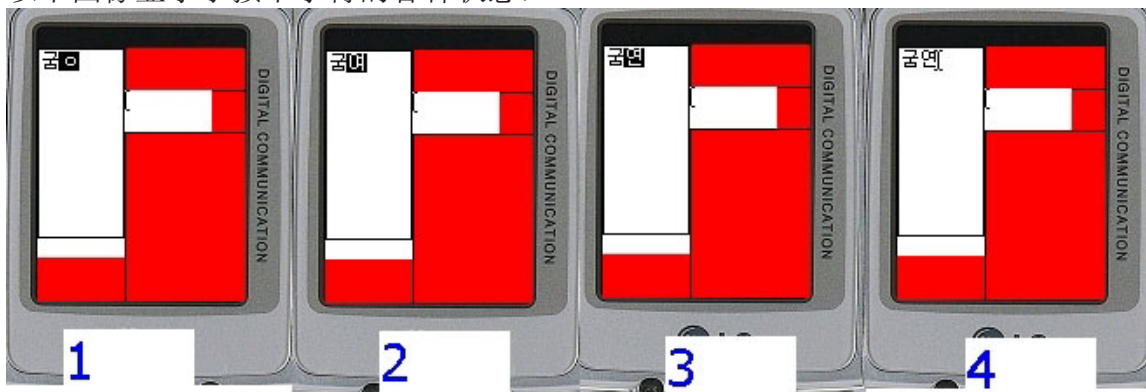
EVT\_UPDATECHAR 是在更新字符时发送的消息。如果字符需要多次击键才能显示，则在发送 EVTCHAR 消息之前应首先发送该消息。许多亚洲语言需要多次击键才能完成一个字符。设置字符后将发送 EVT\_CHAR。EVT\_UPDATECHAR 则将其 wParam 值设为正在处理的编码字符的 AECHAR 值。对于没有按键映射的设备图，典型的情况如下：

```
EVT_UPDATECHAR
EVT_UPDATECHAR
EVT_UPDATECHAR
EVT_CHAR
```

这表示三次击键形成一个字符，提交该字符时将发送 EVT\_CHAR 消息。

EVT\_UPDATECHAR 不移动光标，但会与 MultiTap 一样保持突出显示字符。EVT\_CHAR 将提交字符，并把光标从插入点向前移动。这意味着最后一个 EVT\_UPDATECHAR 的 wParam 应等于 EVT\_CHAR 的 wParam。

以下图像显示了按下字符的各种状态：



1. 它显示了调用 EVT\_UPDATECHAR 后的状态。wParam 为突出显示的字符。由于字符尚未完成，所以仍突出显示该选择。

2. 它显示了下一个 EVT\_UPDATECHAR 调用。wParam 是外观为 (o) 的字符的 KSC6501 值。该选择仍旧突出显示。

3. 它显示了下一个 EVT\_UPDATECHAR 调用。该字符刚好完成，但尚未发送 EVT\_CHAR 消息。

4. 它显示 EVT\_CHAR 消息已发送，且光标已前移。步骤 3 和 4 中的 wParam 值相同。

在按下相关按键时，EVT\_KEY、EVT\_KEY\_PRESS、EVT\_KEY\_RELEASE 和 EVT\_KEY\_HELD 现使用 dwParam 作为修改键的状态快照。这些 dwParam 的位定义如下：

```
// 修改键的位
#define KB_LSHIFT 0x00000001 // 左 Shift 键
#define KB_RSHIFT 0x00000002 // 右 Shift 键
#define KB_LCTRL 0x00000004 // 左 Ctrl 键
#define KB_RCTRL 0x00000008 // 右 Ctrl 键
#define KB_LALT 0x00000010 // 左 Alt/菜单键
#define KB_RALT 0x00000020 // 右 Alt/菜单键
```

```
// 粘性修改键的位
#define KB_CAPSLOCK 0x00010000 // Caps Lock 键
#define KB_SCROLLLOCK 0x00020000 // Scroll Lock 键
#define KB_NUMLOCK 0x00040000 // Num Lock 键
```

在 AEEVCodes.h 中：

另有一个助手宏为 IS\_KEY\_DOWN，它使用相关按键的位字段和位。

用法示例：

```
IS_KEY_DOWN(wParam, KB_LSHIFT);
TRUE 表示发送消息时按下了左 shift 键。
FALSE 表示未按下该键。
```

ITextCtl 将处理 EVT\_CHAR 和 EVT\_UPDATECHAR。EVT\_CHAR 在光标位置添加字符并将其提交。EVT\_UPDATECHAR 在光标位置添加字符，并保持突出显示字，直至完成该字符。

IMenuCtl 现可处理 EVT\_CHAR 消息。当菜单控件被激活并有 EVT\_CHAR 消息发送时，菜单将从以键入字符开始的当前选择中查找第一个匹配。这种比较不区分大小，所以键入 'a' 将可匹配 'Amphibian'。

## BREW Emulator

对 BREW Emulator 进行了以下增强：

- “文件”菜单现可列出最近使用的设备。
- 不再要求明确地设置 MIF 目录。默认情况下，将在应用程序目录本身中搜索 MIF。但是，您也可以选择从“工具”菜单访问“设置”对话框来明确设置 MIF 目录。
- BREW SDK 不再附带各种颜色的 MIF。所有 MIF 都直接置于 Examples 目录中。它们对应于 256 色。
- BREW Emulator 现还可接受设置文件的命令行参数。该路径可以是绝对路径，也可以是相对路径。
- 查看 > 实际屏幕大小菜单选项已被取消。
- 将按键事件（EVT\_KEY、EVT\_KEY\_PRESS 等）发送到应用程序时，dwParam 始终设置为零。

## BREW Device Configurator

对 BREW Device Configurator 进行了以下增强：

- 设备对象（按键和屏幕）矩形按（左边，顶部，宽度，高度）方式描述
- 按键对象属性：
  - 已清除重载代码。
  - 已删除按键符号和按键操作。
- 屏幕对象属性：
  - 取消“定义纵横比”复选框以及“尺寸”（实际屏幕）宽度和高度
  - 使用用户绘制的屏幕宽度和高度。必须定义像素计数（每行和每列的像素）。BREW Emulator 将使用像素计数作为宽度和高度对屏幕矩形进行 StretchBlt。
- OEM 字体：设备可以关联 OEM 字体（.bbf 文件）。任意时刻的活动字体应为 Windows 字体或 OEM 字体。Windows 字体为默认字体。
- DeviceID：设备可以在详细信息 > 设备属性 > 常规对话框中关联一个 ID。

## TAPI 模拟

通过**设置 > TAPI 模拟 > 启动呼叫**启动进入呼叫时，将中止应用程序，并显示消息“呼叫进行中...应用程序已中止。”不提示用户。

外发呼叫（使用 `ITAPI_MakeVoiceCall()` 初始化）的模拟与之相同。

如果在 TAPI 正忙时尝试启动新呼叫，则将显示消息框，指示 TAPI 正忙于呼叫或停止呼叫。

## IMEDIA\_Record() 模拟

已增强的 BREW SDK 可支持 QCP (PureVoice) 记录。

您可以使用 `ClassID AEECLSID_MEDIAQCP`（创建 `IMediaQCP` 接口）来创建 `IMedia` 接口，并调用 `IMEDIA_Record()` 来记录 QCP 文件。通过麦克风即可完成向前链接和向后链接记录，而播放则始终通过扬声器进行。

## BREW MIF 编辑器

对 BREW MIF 编辑器进行了以下增强：

- 模块版本现可存储于特定模块的 MIF 中。
- 使用 `ISHELL_GetAppVersion()` 可访问存储在 MIF 中的模块版本。
- `ISHELL_ShowCopyright()` 已更新为显示模块版本。

## 网络速度模拟

- **Device Configurator** 现在允许设置设备上可用套接字的数量（TCP 和 UDP 分别设置）。
- **BREW Emulator** 现在允许用户模拟各种网络条件。BREW Emulator 中可配置以下参数。
  - 传输速度
  - 接收速度
  - 通信量上行时间
  - 通信量下行时间

- PPP 上行时间
- PPP 下行时间

## GPS 模拟

BREW Emulator 中可使用 GPS 模拟来显示 BREW 应用程序如何接收 GPS 或 GNSS 数据。使用 BREW SDK 随附的 National Marine Electronics Association (NMEA) Logger Utility 可记录该活动。Emulator 中的设置决定了输入所代表的直接数据通信是从 GPS 设备和计算机间的串行连接，还是从以前记录的文件中接收到。

## BREW 联机帮助

- 集成帮助：提供 Microsoft Visual Studio 插件，使得开发者可以从 Visual Studio 环境搜索 BREW API 参考资料帮助中的上下文相关帮助。
- 所有 SDK 工具现在都具有程序式主题联机帮助（可通过“帮助”菜单访问）。

## BREW 2BitTool

BREW 2Bit Tool 提供了命令行接口 (CLI) 和图形用户界面 (GUI)，使用户能够将 4 位 BMP 转换为 2 位 BMP（反之亦然）以用于图像编辑。2Bit Tool 通过覆盖现有 BMP 或创建新的 BMP，将已转换的 BMP 保存为特定文件。

## BREW 资源编辑器

对 BREW 资源编辑器进行了以下增强：

- 资源编辑器的命令行支持可使用 .bri 文件并将其编译成 .bar 文件。为方便使用还添加了若干其它命令行选项。
- 资源编辑器现在始终处理相对于 .bri 文件位置的路径，而“不是”相对于资源编辑器启动位置的路径。
- 资源编辑器允许创建指向不存在的图像文件的资源。出现这种情况时将发出警告。但只有当您试图编译包含不存在的图像文件的 .bar 文件时才会出错。

## BREW MIF 编辑器

对 BREW MIF 编辑器进行了以下增强：

- MIF 编辑器允许 MIME 类型长度超过 32 字节。
- MIF 编辑器提供了一种方法，使您在提交应用程序进行认证之前能够在 BREW Emulator 和手持设备上测试许可方案的 *使用次数*。
- 新增了“弹出”和“屏幕保护程序”设置。
  - “弹出”指示启动小程序时不清除手持设备上的设备屏幕。如果未选择该项，BREW 将在启动小程序之前清除设备屏幕的内容。
  - 创建屏幕保护应用程序时应使用“屏幕保护程序”设置。

## 示例应用程序

BREW SDK 包含了使用 BREW 创建的若干示例应用程序，以演示 BREW 应用程序开发中许多常用的接口和控件的用法。这些示例的用途是识别帮助应用程序完成各自目标任务的 BREW API 和控件，并从用户角度说明各个 BREW 应用程序的功能。包括以下示例应用程序：

- HelloWorld
- Expense Tracker
- MediaPlayer
- NetDiagnostics
- Road Warrior
- WhiteBoard

**注意：**使用 Media Player 示例应用程序读取和播放 QUALCOMM Pure Voice (QCP) 音频文件时要求已加载相应的 CMX DLL。

## 2.0.0.14 版到 2.0.0.33 版的 SDK 更改内容

以下列举了从 2.0.0.14 版到 2.0.0.33 版的 SDK 更改。

## BREW 资源编辑器

- 支持创建对话框中的图像控件 (IImageCtl)。
- 编译 .bri 文件时生成的头文件仅在需要时才进行更新。之前，每次编译 .bri 文件都要更新头文件。
- 在保存 BRI 文件的过程中，所有图像路径都将转换为相对路径。
- 添加了对 GB2312 编码的支持。

## BREW MIF 编辑器

- 更新了“常规”选项卡，允许用户指定文件总数和文件空间总量的限制。它包括指定应用程序的文件系统使用量（应用程序所需的文件计数和文件空间总量）的功能
- 添加了对 SHELL\_INIT 通知的支持（类：AEECLSID\_SHELL 通知掩码：为 NMASK\_SHELL\_INIT）。它允许应用程序在初始化 BREW 时注册通。
- 添加了对 GB2312 编码的支持。
- 添加了预览 JPEG (.jpg, .jpeg) 图像的功能。

## BREW Emulator

- 添加了对 GB2312 编码的支持。
- 已增强文件 > 加载设备、文件 > 更改小程序目录以及“文件”菜单下加载最近使用的设备图像等，以相应地更新 BREW\_Emu.dat。以前是通过工具 | 设置菜单获得此功能。
- 除非在工具 | 设置对话框中选中了“指定 MIF 目录”，否则使用文件 | 更改小程序目录更改小程序目录将更新 MIF 目录，使其指向同一目录。如果已选中“指定 MIF 目录”并在“初始 MIF 目录”中输入了有效目录，则将该目录用作 MIF 目录。
- “输出窗口”可调节大小，并记忆它先前的状态和在屏幕上的位置。如果用户选择的条目与上一个相同，则它将自动滚动至列表的底部。

## BREW Compressed Image (BCI) 编辑工具

- 添加此功能，除了可以使用 **BMP** 和 **PNG** 图像类型（已有功能），还可使用 **JPEG**（.jpg 和 .jpeg）来创建 BCI。

## BREW Device Configurator

- 每个设备都关联了以下两个参数：
  - 允许创建的文件总数。有效范围：有效范围： 7 – 65535
  - 允许写入的最大空间。有效范围： 20480 – 4294967295

如果 MIF 中未指定，BREW Emulator 将使用这些值。增强的“设备属性”对话框中包含了“文件系统”选项卡，其中的编辑字段可指定这些值。

- 添加了对 **GB2312** 编码的支持。

## 其它

该 SDK 现随附了 **Samsung SPH-E1000**（彩色）设备。

各种设备图像文件及其依存文件的命名方式现为：

- 对于无速度模拟的设备图像： **Make\_Model.qsc**（例如 **LGE\_VX-10.qsc**）。
- 对于具有速度模拟的设备图像： **Make\_Model\_SpeedEmu.qsc**（例如 **LGE\_VX-10\_SpeedEmu.qsc**）。
- **BMP** 文件名格式： **Make\_Model\_Up[Down].bmp**。

## 2.0.0.33 版到 2.0.0.38 版的 SDK 更改内容

以下列举了从 2.0.0.33 版到 BREW 2.0.0 最终版的 SDK 更改。

## BREW MIF 编辑器

- “**MIF 编辑器许可证**”选项卡已更新，其中包括了“演示版”价格类型选。这使得开发者可以在 **SDK** 上测试应用程序的“演示版”。
- 开发应用程序时，必须确保 **MIF** 文件名不完全由数字构成。**MIF** 文件名必须包含字母字符，并且必须以字母字符开始。

下表显示了有效和无效的 MIF 文件名。

有效的 MIF 文件名	无效的 MIF 文件名
helloworld.mif	400.mif
b2b.mif	1056.mif
good4u.mif	9988.mif

当应用程序已经过数字签名并置于运营商服务器上供终端用户下载时，**BREW** 将自动分配并使用数字式 MIF（如 400.mif）。**BREW** 以特殊的方式处理数字式 MIF，如果试图在应用程序开发过程使用它，则应用程序可能出现严重错误。

## BREW 资源编辑器

添加 JPEG 图像时，资源编辑器现可保存图像类型为图像/jpeg。由于此更改，ISHELL\_LoadResImage() API 现可加载 JPEG 图像。

## 其它

术语“预览版”现已从产品（文档、“帮助 - 关于”对话框等等）中删除，本版本现命名为“2.0 商业版”。

Visual Studio 插件 Automated ARM Compiling（ARM 自动编译）现已更新，在生成应用程序的 ARM 生成文件时，它将包含 APCS 选项 norwpi，以及链接选项 -rwpi。通过这些更改，可以在使用 ADS 1.1 或 1.2 编译器时检测动态应用程序中的全局数。

## 已知问题

有关 BREW 2.0.0 已知问题的列表，请访问 BREW 外联网：  
<https://brewx.qualcomm.com/developer/sdk/download.jsp>。