

A Historical Viewpoint

J2ME

Windows

- many variations
- more buggy
- more UI variations
- loose h/w integration
- low barrier to entry
- more tools and support
- varying developer programs
- many vendors
- multiple business relationships

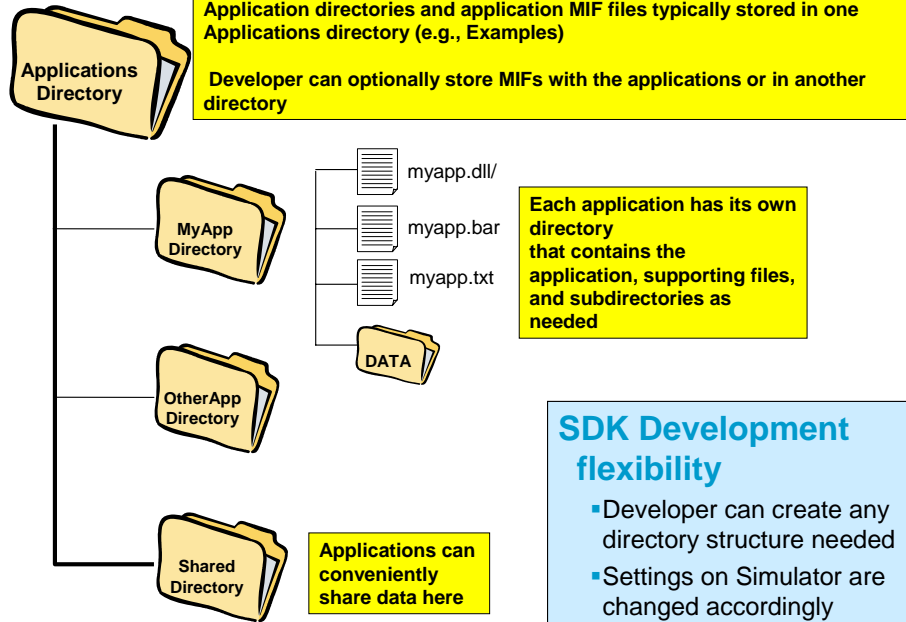
BREW <> Qualcomm

MacOS <> Apple

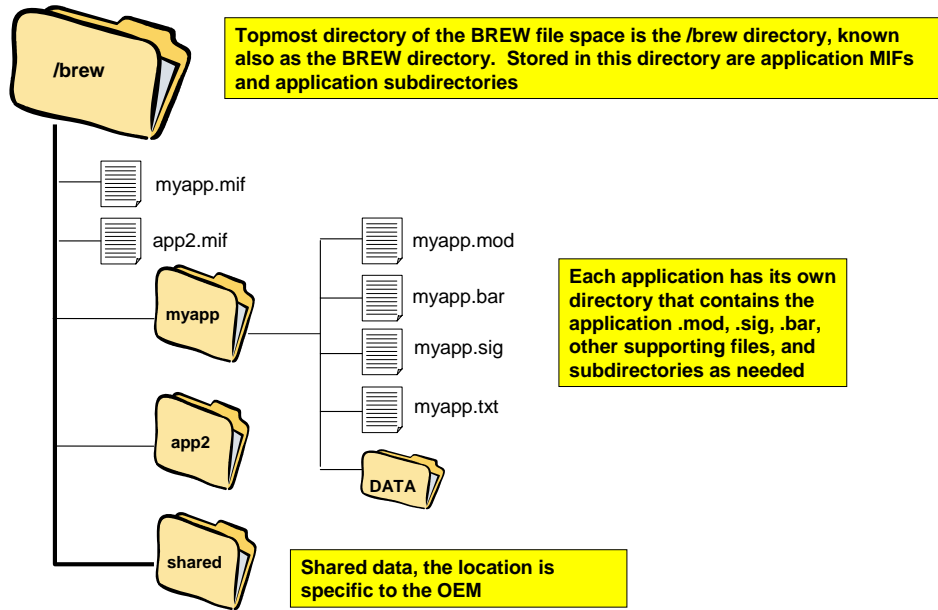
PalmOS <> Palm

- one standard implementation
- less buggy
- more consistent UI
- tight h/w integration
- higher barrier to entry
- fewer tools and support
- strong, deep developer prog.
- few vendors
- one business relationship

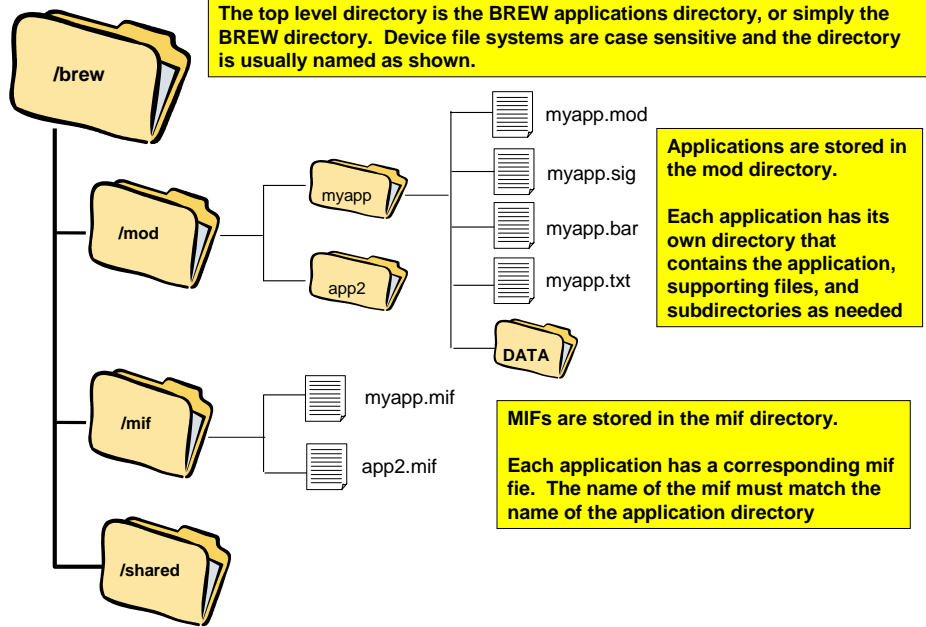
SDK Directory Structure



BREW Device Directory Structure Before 3.x



BREW 3.x Device Directory Structure



Variable Naming Conventions

◆ Follows Microsoft's "Hungarian" conventions

Prefix	Meaning	Example	Notes
p	pointer	pMe pIFile	Pointer to me (applet data area in memory) Pointer to IFile interface
pp	pointer to pointer	ppObj	Pointer to pointer to Obj
s	string	sURL	Name of URL
sz	null-terminated string	szResFile	Name of Resource File

Variable Naming Conventions

Prefix	Meaning	Example	Notes
n	integer	nResID nSize	Resource id number Size
w	word (unsigned int)	wParam	16 bit parameter
dw	double word (unsigned)	dwParam	32 bit parameter
c	character	cDigit	Character code for digit
b	boolean	bEnable	Enable / disable flag

Variable Naming Conventions

- ◆ **UPPERCASE - Constants**

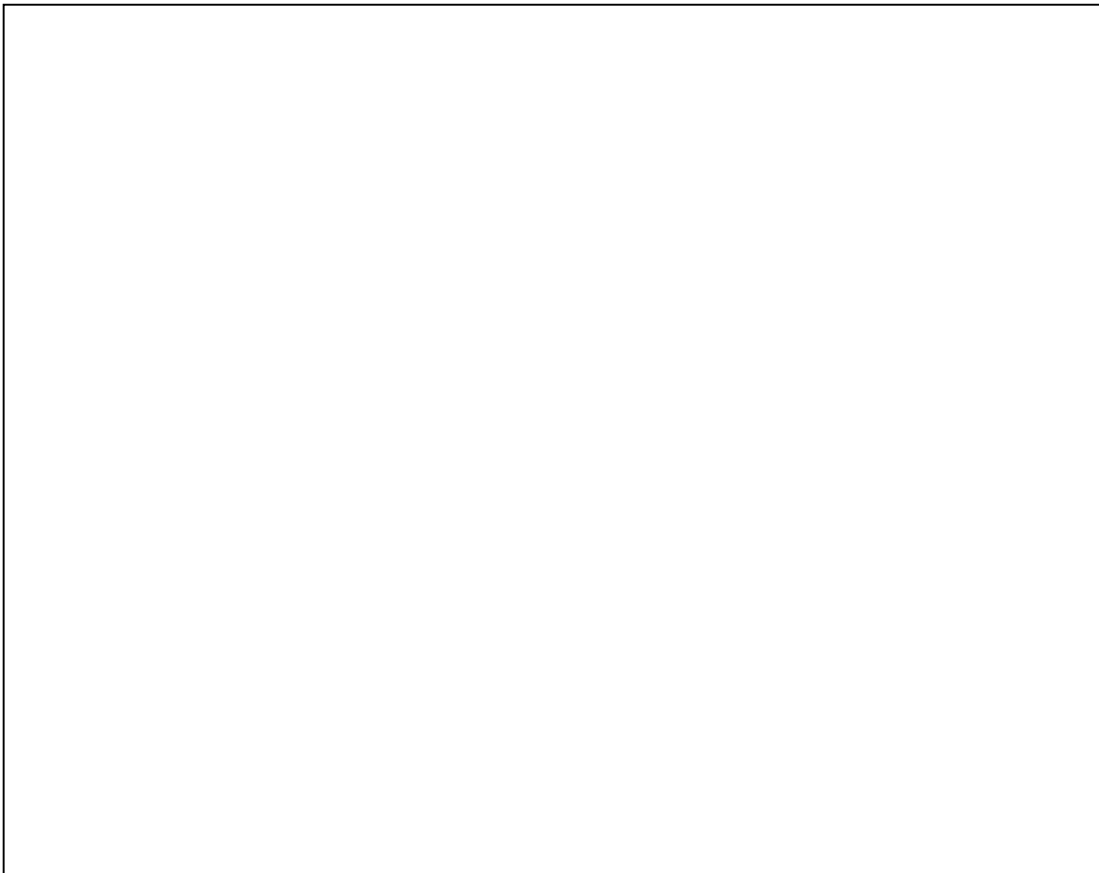
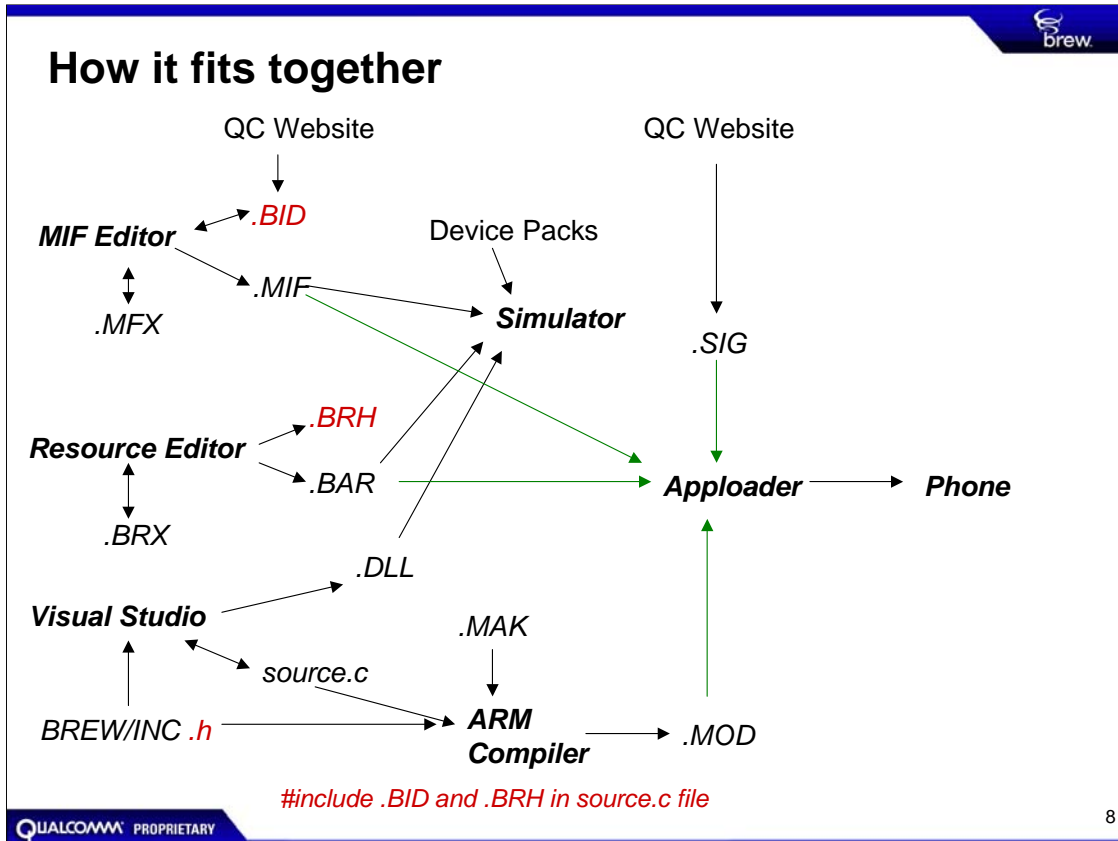
- ◆ **Accessing variables**

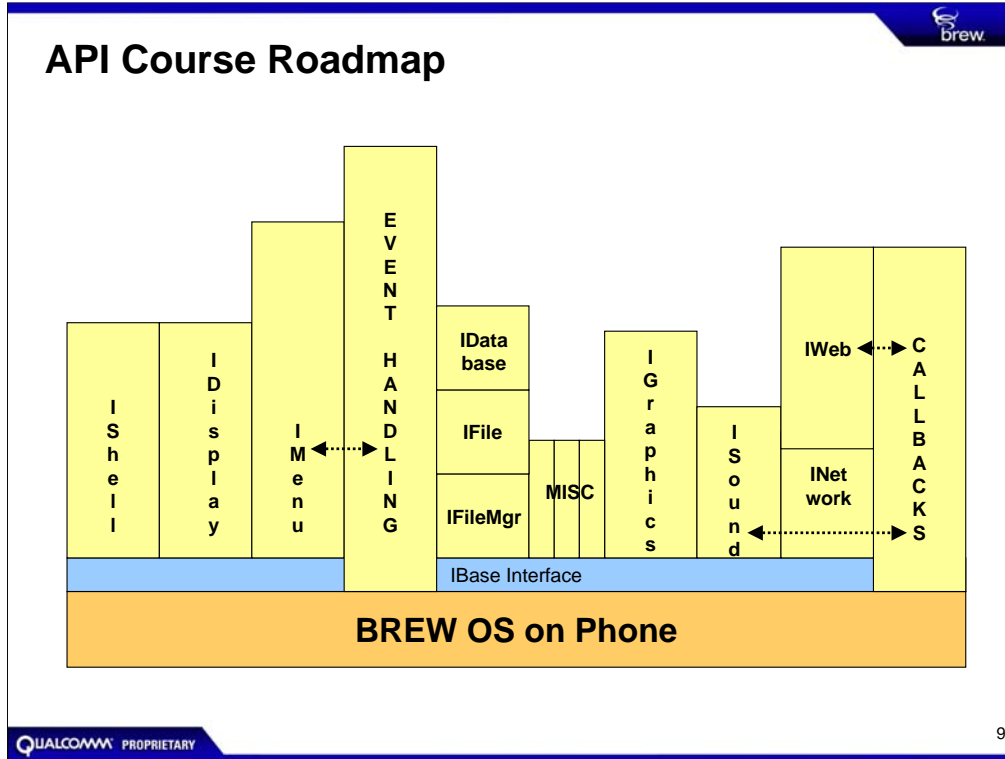
 - pMe->pIMenuCtl -- access pointer to menu control interface
 - pMe->myVariable

- ◆ **Accessing members of structures**

 - s.m – m is member of structure s
 - pMe->DeviceInfo.cxScreen

m_ conventions sometimes used to indicate member
pMe->a.m_pIDisplay





IGraphics Data Structure Examples (AEEGraphics.h)



◆ Structures and unions

```
typedef struct _circle {
    int16 cx, cy; // Center of the circle
    int16 r;      // Radius of the circle
} AEECircle;
```

◆ Enumerated types

```
// Enum for Stroke Styles
enum {
    STROKE_SOLID,
    STROKE_DOTTED
};
```

◆ Constants

```
// Flags used for handling clipping region
#define AEE_GRAPHICS_NONE 0x00
#define AEE_GRAPHICS_FRAME 0x02
#define AEE_GRAPHICS_CLEAR 0x04
#define AEE_GRAPHICS_FILL 0x08
```

ISHELL_CreateInstance()

- ◆ Used to create instances of classes, especially BREW interfaces

```
int ISHELL_CreateInstance(IShell * pIshell,
                          AEECLSID cls,
                          void ** ppobj);

// create main menu
if (ISHELL_CreateInstance(pMe->pIshell,
                          AEECLSID_MENUCTL,
                          (void **) pMe->pIMainMenu)
    == SUCCESS){
```

Releasing Interfaces

◆ **WARNING: Only release interface once**

```
if (pMe->pIMainMenu){  
    IMENUCTL_Release(pMe->pIMainMenu);  
    pMe->pIMainMenu = NULL;  
}
```

ISHELL_GetDeviceInfo() Example

◆ Example – Screen size from Simon game

```
AEEDeviceInfo di; // local variable allocated on stack
di.wStructSize = sizeof(AEEDeviceInfo);

// Get the device info and set the screen rect
ISHELL_GetDeviceInfo(pMe->pIShell, &di);

pMe->ScreenRect.x = 0;
pMe->ScreenRect.y = 0;
pMe->ScreenRect.dx = di.cxScreen;
pMe->ScreenRect.dy = di.cyScreen;

pMe->QuadRect[0].x = 0;
pMe->QuadRect[0].y = 0;
pMe->QuadRect[0].dx = di.cxScreen / 2;
pMe->QuadRect[0].dy = di.cyScreen / 2;
```

IDisplay_DrawText in Detail

```
IDISPLAY_DrawText(pMe->pIDisplay,
                  AEE_FONT_BOLD,
                  szWideHelloString,    // here is our string to print
                  -1, 0, 0, 0,
                  IDF_ALIGN_CENTER | IDF_ALIGN_MIDDLE);
```

Prototype:

```
int IDISPLAY_DrawText (
IDisplay * pIDisplay,           // pointer to IDisplay
AEEFont Font,                  // font (constant)
const AECHAR * pcText,        // pointer to wide buffer
int nChars,                    // number of chars to display
int x,                         // x position on screen
int y,                         // y position on screen
const AEERect * prcBackground, // pointer to clipping area
uint32 dwFlags                 // optional flags
)
```

IDisplay_DrawText Flags

Specifies the flags that can be used for drawing the text. This can be a logical OR of one of the items selected from each of the following entries:

- One of the horizontal alignment flags
(IDF_ALIGN_LEFT, IDF_ALIGN_CENTER, IDF_ALIGN_RIGHT)
- One of the vertical alignment flags
(IDF_ALIGN_TOP, IDF_ALIGN_MIDDLE, IDF_ALIGN_BOTTOM)
- One of the text format flags
(IDF_TEXT_UNDERLINE, IDF_TEXT_INVERTED)
- One of the rectangle format flags
(IDF_RECT_FRAME, IDF_RECT_FILL, IDF_RECT_INVERT)

these flags work on the **prcBackground** rectangle, using the CLR_USER_BACKGROUND as the fill color and CLR_USER_FRAME as the frame color. If any of the [Flags for the rectangle](#) are specified, then this rectangle is also used as filling.

IDisplay_DrawRect Example



```

// Draw rectangle in top half of screen; filled with Red and with a frame in
// Blue.

AEERect rct;
SETAEERECT (&rct, 0, 0, di->cxScreen, di->cyScreen/2);

clrFrame = MAKE_RGB (0, 0, 0xff); // Blue color
clrFill = MAKE_RGB (0xff, 0, 0); // Red color
dwFlags = IDF_RECT_FILL | IDF_RECT_FRAME;

// Draw rectangle with values initialized above
IDISPLAY_DrawRect (pME->a.m_pIDisplay, &rct, clrFrame, clrFill, dwFlags);

```


IDisplay Exercise



Timer Example

```
ISHELL_SetTimer(pMe->pIShell,  
                1000,           // time in milliseconds  
                PlaybackTimerCB, // call back function  
                (void*)pMe);    // parameter to callbck  
return;  
}
```

```
static void PlaybackTimerCB(void *pUser){  
    myApp* pMe = (myApp*)pUser;  
    int CurrentQuad = pMe->Sequence;  
    RenderQuad(pMe, CurrentQuad);  
}
```

Message Box and Prompt Functions

- ◆ ISHELL_MessageBox()
- ◆ ISHELL_MessageBoxText()
- ◆ ISHELL_Prompt()
- ◆ ISHELL_ShowCopyright()

Message Box Example

```
/* ISHELL_MessageBox: instructs the shell to display  
a message box to the user. A message box is a simple  
window with a title and text.
```

```
Title and message contained in resource file.
```

```
*/
```

```
char szResFile[] = APP_RES_FILE;
```

```
ISHELL_MessageBox(pMe->pIShell,  
                  szResFile,  
                  IDS_MESSAGEBOX_TITLE,  
                  IDS_MESSAGEBOX_TEXT);
```