

INTO THE new
BREW 2007 CONFERENCE

Writing uiOne™ Apps for uiOne HDK

Mike Mager, Managing Staff Engineer
QUALCOMM Incorporated





Agenda

- Brief uiOne™ Handset Development Kit (HDK) overview
- Features/Benefits of HDK app development
- Designing with the common framework
- Questions & Answers



Overview – What is the uiOne HDK?

1. uiOne-based common application framework
2. Set of uiOne applications (based on #1) and BREW[®] extensions that implement a reference UI

This presentation covers item #1



Present and forthcoming HDK apps

	HDK 1.X	HDK 2.0*
Call Handling	✓	✓
Power Up/Down	✓	✓
Clock	✓	✓
Calculator	✓	✓
Call Log	✓	✓
Contacts	✓	✓
Media Viewer	✓	✓
Music Player	✓	✓
Still/Video Camera	✓	✓
Messaging (SMS)	✓	✓
Settings	✓	✓
Themes	✓	✓
Voice Mail	✓	✓
Video Telephony	✓	✓
Sync DM		✓
Messaging (MMS)		✓
Browser		✓

* Not yet available





Opportunities for 3rd party app developers

- Enhance existing HDK apps
- Provide additional well-known apps
 - Instant Messaging
 - E-mail
- Develop new apps!



Features of uiOne HDK app development

- Primary development environment is uiOne SDK (TrigBuilder)
- Can still use Visual Studio .NET for actor/extension development & debug
- 3rd party apps can have HDK look & feel allowing seamless UI integration
- Framework provides many commonly used application components
 - Notes, popup menus, lists, softkeys, etc.



HDK App requirements

- Must provide a .tmf file in addition to .mod/.dll, .mif, .bar files
- For integration into the UI (e.g. Main Menu, Shortcuts) app must provide a mimetype in its .mif file.
 - `<MimeType Base="AEECLSID_APP" Handler="0x01031fd4" Value="settings"/>`
- App must be "AppFrame"-based



AppFrame

- All HDK apps are based on AppFrame
- Consists of a Trig Library and an Actor that make up most of the common framework
- Designing an app using AppFrame will ensure a cohesive user experience



TrigApplet

- IApplet-based code used by all HDK apps
- Slightly different than uiOne SDK's version
- No applet code to write!
- Simply use TrigApplet.lib as your .mod/.dll



.tmf file

- TrigApplet uses the .tmf file to load specified .bar files
- .bar files are loaded in order to build VFS
- Path to file can be absolute or relative to app's mod directory

0103331b.tmf

```
trigname:Calculator  
trigbar:fs:/shared/appframe.bar  
trigbar:fs:/shared/type2notes.bar  
trigbar:calculator.bar
```



Application Entry Point

- Standard entry point is TrigML[®] resource “startup/default”, normally provided by app
- By including appframe.bar in .tmf file:
 - “startup/default” provided to set up screen layout
 - New entry point for app becomes “pages/pgDefault/impl”
 - App now has access to common framework

Empty HDK App

- No pages/pgDefault/impl





AppFrame features

- Title and softkey elements
- Screen layout
- Popup notes
- Popup menus
- Page navigation & page history
- Lists & list items
- Scrollbars
- Text entry popup

Title and Softkey area elements

- Well-known variable locations:

Item	Path
Title Text	/var/titleString
Subtitle Text	/var/subtitleString
Title Icon image	/var/titleIcon
Left Soft Key	\$me/sk1
Right Soft Key	\$me/sk2

pages/pgDefault/impl:

```
<setdata when="_entry" res="/var/titleString"
value="'SampleApp'"/>
<setdata when="_entry" res="/var/subtitleString"
value="'First Page'"/>
<setdata when="_entry" res="/var/titleIcon"
value="theme/pages/pgDefault/icon"/>
<setdata when="_entry" res="$me/sk1"
value="Info"/>
<setdata when="_entry" res="$me/sk2"
value="{ $lang/softkeys/options }"/>
```



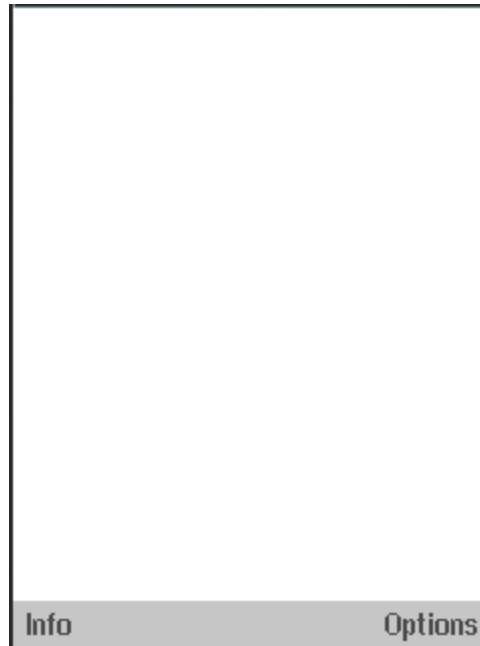
Screen Layout

- AppFrame actor has a string node to control layout: /actor/appFrame/layout

"normal"



"notitle"



"fullscreen"



Popup Menus

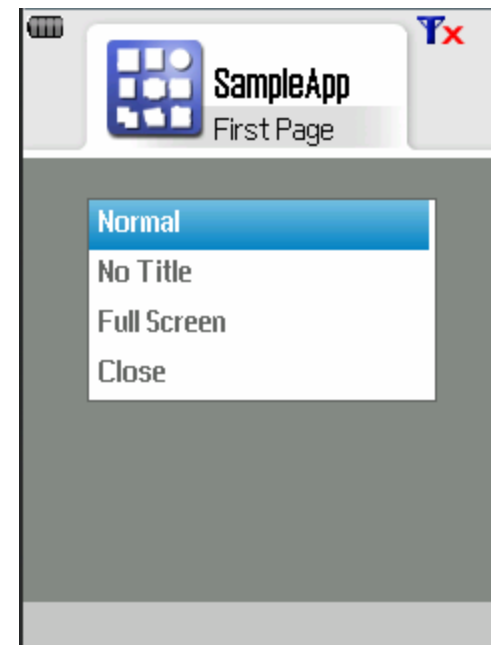
- Uses a list of resources to iterate over:

Resource Type	Path	Value
Text	lists/options/1/\$lang/caption	"Normal"
Text	lists/options/2/\$lang/caption	"No Title"
Text	lists/options/3/\$lang/caption	"Full Screen"
Text	lists/options/4/\$lang/caption	"Close"
Text	lists/options/5/\$lang/caption	"Option 5"
Text	lists/options/6/\$lang/caption	"Option 6"
List set	lists/options/all	1,2,3,4,5,6
List set	lists/options/normal	1,2,3,4

Popup Menus (cont.)

- Send "LoadPopupMenu" to AppFrame actor
 - Provide list to iterate over as event parameter
 - Provide "eventSink" to receive pmSelect event

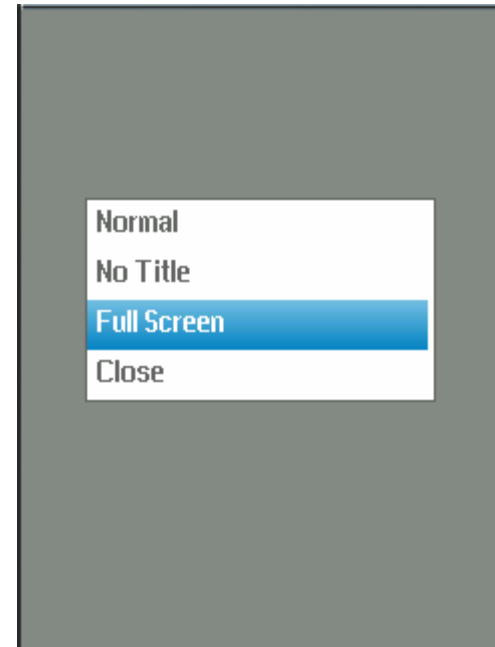
```
<!-- Load a popup menu on "Options" SK -->
<throw when="_keypress[_keysoft2]"
  event="LoadPopupMenu"
  target="/actor/appFrame">
  <param name="listRes"
    value="lists/options"/>
  <param name="listOrderRes"
    value="lists/options/normal"/>
  <param name="textNodeName"
    value="$lang/caption"/>
  <param name="eventSink"
    value="gp"/>
</throw>
```



Popup Menus (cont.)

- “pmSelect” event thrown to \$eventSink when an item is selected

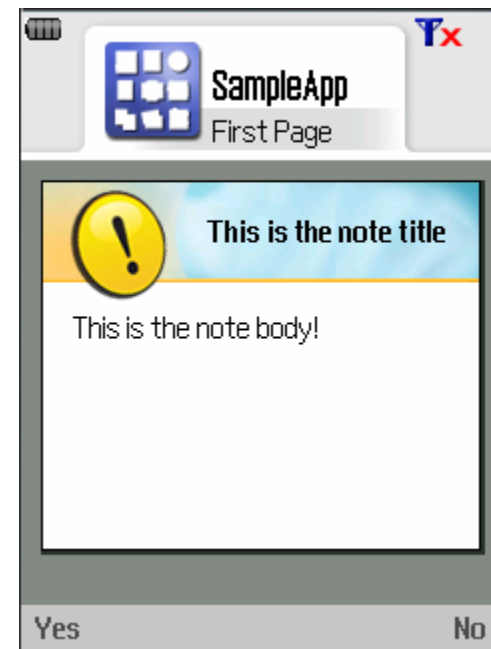
```
<group id="gp">
  <setdata when="pmSelect[@selectedIndex == '1']"
    res="/actor/appFrame/layout"
    value="normal"/>
  <setdata when="pmSelect[@selectedIndex == '2']"
    res="/actor/appFrame/layout"
    value="notitle"/>
  <setdata when="pmSelect[@selectedIndex == '3']"
    res="/actor/appFrame/layout"
    value="fullscreen"/>
  <throw when="pmSelect[@selectedIndex == '4']"
    event="ClosePopup"
    target="/actor/appFrame"/>
</group>
```



Popup Notes

- Send "LoadPopupNote" to AppFrame actor
 - Note resource must be located at `template/elements/type1note/[idNoteType]/note`
 - Several types provided by AppFrame, but apps can add their own

```
<group canfocus="_true">
  <!-- Load a note on the "Info" SK -->
  <throw when="_keypress[_keysoft1]"
    event="LoadPopupNote"
    target="/actor/appFrame">
    <param name="idNoteType"
      value="typeIDMinorNote"/>
    <param name="titleText"
      value="'This is the note title'"/>
    <param name="bodyText"
      value="'This is the note body!'"/>
  </throw>
</group>
```



Page Navigation & Page History

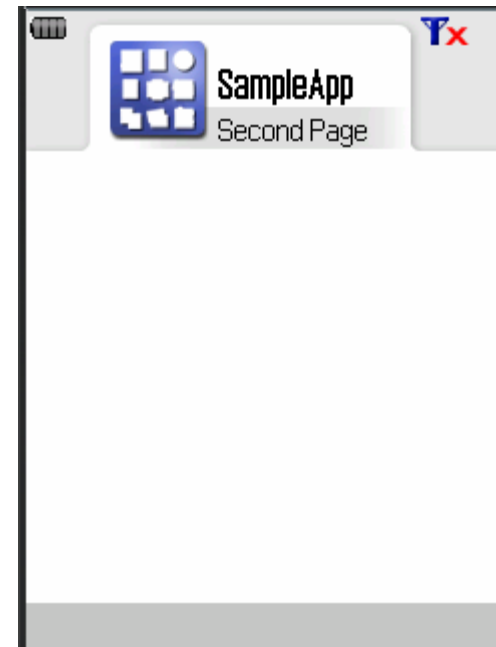
- AppFrame automatically keeps track of page history when using "LoadPage" and related events
- Pages defined at pages/[idPage]/impl

```
pages/pgDefault/impl:
```

```
<!-- on Select, go to the next page -->  
<throw when="_keypress[_keyselect]"  
    event="LoadPage"  
    target="/actor/appFrame">  
    <param name="idPage" value="pgTwo"/>  
</throw>
```

```
pages/pgDemo/impl:
```

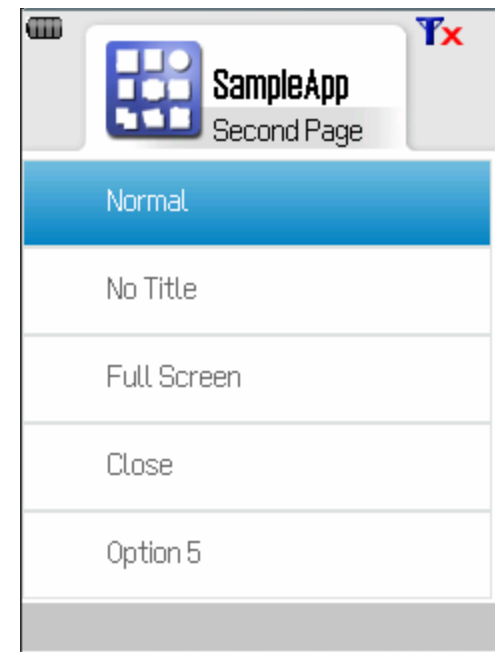
```
<setdata when="_entry"  
    res="/var/subtitleString"  
    value="'Second Page'"/>
```



Lists and List Items

- AppFrame list items designed to be used in griddata elements to make lists

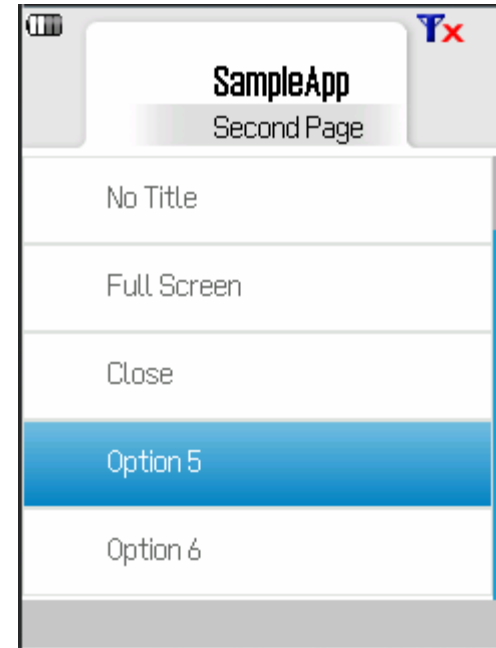
```
<!-- include a list -->
<griddata id="gd" rows="5" cols="1"
  repeatover="lists/options/all">
  <include res="template/elements/typeF1List/impl">
    <param name="idListItem" value="$$"/>
    <param name="idListItemText"
      value="lists/options/$$/$lang/caption"/>
  </include>
</griddata>
```



Scroll bars

- AppFrame scrollbar element can be attached to any scrollable visible element

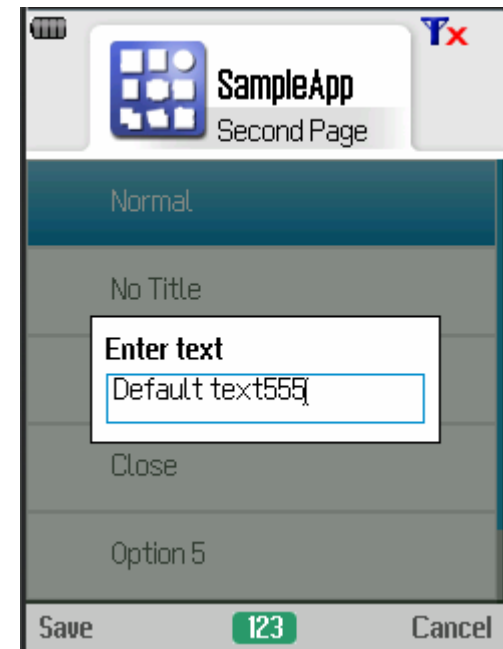
```
<!-- attach a scroll bar -->  
<include res="template/elements/vScrollBar/impl">  
  <param name="scrollerOf" value="gd"/>  
</include>
```



Text Entry popup

- AppFrame includes highly configurable text entry popup

```
<setdata when="_entry" res="/var/pteBuf/0/buffer"
  value="/var/buf" />
<throw when="_keypress[_keyselect]" event="LoadPopup"
  target="/actor/appFrame">
  <param name="res"
    value="template/elements/popupTextEntry/impl"/>
  <param name="caption" value="'Enter text'"/>
  <param name="list" value="/var/pteBuf"/>
  <param name="length" value="20"/>
  <param name="preloadedText" value="'Default text'"/>
  <param name="inputmode" value="_multitap;_numeric"/>
  <param name="fieldMode" value="0"/>
  <param name="sk1" value="Save"/>
  <param name="sk2" value="Cancel"/>
  <param name="keyHandler" value="path/to/keyhandler"/>
</throw>
```





Other Useful Components

- Shared Trig Libraries
 - ContactsExt: Contact picker
 - MViewerLib: Viewing media
- More than 60 Actors
 - DebugMsgActor, MediaActor, CallLogActor, etc.
- ActorFrame
- URL lookup table



Advanced Features

- Themes/Skins
- Notifications
- Main Menu & Shortcuts integration



Skeletons vs. Skins

- Skeleton
 - Functional logic (TrigML)
 - Text resources
- Skins
 - Display logic (TrigML)
 - Image resources
- Can change look & feel by only changing skin .bar files

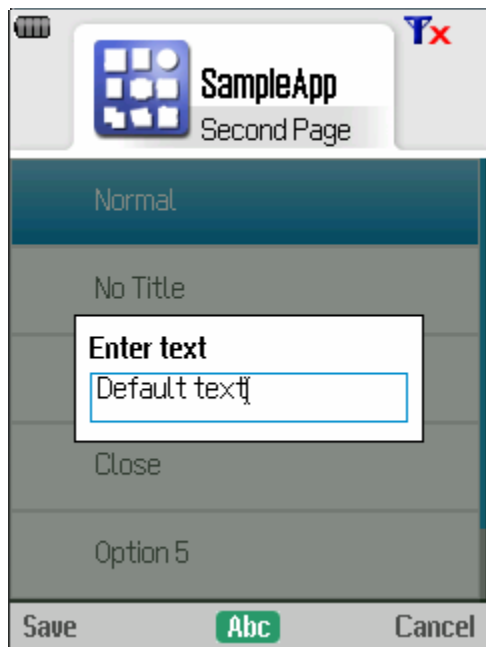


Themes/Skins

- Whenever a “skel” bar file is loaded, a corresponding “skin” bar file load is attempted, too
 - appframe.bar -> appframe_vixenskin.bar
- When changing themes, new “skin” bar file is loaded
- App should provide separate skin bar files as necessary

Changing Themes

- When using AppFrame elements, app looks like the rest of the UI when themes change





Notifications

- Any notification that needs to appear over another app is considered a type2note
- Include `fs:/shared/type2notes.bar` in the `.tmf` to allow your app to display notes (`.tmf` entry needed for HDK 1.1, not 1.2+)
- See Developer's Guide for more info on creating custom notifications

Main Menu & Shortcuts integration

- Add entry for new app to a few lists in PhoneApp and Settings trigs

Path	Path	Value
PhoneApp	lists/menuItems/[newApp]/actionURL	App's URL + ":Top" (e.g. "clock:Top")
PhoneApp	lists/menuItems/[newApp]/gridIcon	Image for Icon view
PhoneApp	lists/menuItems/[newApp]/listIcon	Image for List view
PhoneApp	lists/menuOrder/[newApp]/defaultMenuOrder	Add new list item called [newApp] and adjust order as needed
PhoneApp	\$lang/shortcuts/[newApp]	Name of app
Settings	lists/shortcutItems/shortcuts	Add new list item called [newApp], modify resources and order appropriately

```
<!-- launch an app -->
<throw when="foo" event="SendURL" target="/actor/shell">
  <param name="url" value="{lists/menuItems/$app/actionURL}" />
</include>
```



More Information

- For more information on developing for uiOne HDK, please contact us at hdkrequests@qualcomm.com
- Immediately following this presentation
 - uiOne HDK Case Study by Opera Software ASA



Questions?

