Archived:New Features in WidSets

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

The article is believed to be still valid for the original topic scope.

Since 3.1.0

Maintenance release

- Channel API works again
- Minor bug fixes

Since 3.0.0

API Features

SHA-1

- ByteArray ByteArray.sha1()

Computes digest from this array using SHA1 algorithm.

Feature support

- boolean isSupported(String feature)

Returns true if used phone supports requested feature.

Color morphing

- int morphColor(int sourceColor, int targetColor, int progress)

Morphs between two colors using given progress [0..1000].

Browser opening

- int openBrowser(String url)

Opens web browser, or other suitable view for content located in url.

UI Features

Animation

Helper class for calculating animations using Linear, Quad or Sine interpolation.

Surface

New container for Components allowing free positioning and sizing. Can be used with Animation to animate Components.

- Component.setLocation(int x, int y)

Sets absolute location of component on Surface.

Component focusing

- boolean Component.isShown()

Is component shown on screen.

(C) Copyright Nokia 2014. All rights reserved.

- Shell.setFocused(Component focused, boolean ensureVisibility)

Same as setFocused(Component) but by setting ensureVisibility = false, viewport does not scroll to this component.

Scrollable exact positions

- int, int Scrollable.getCurrentPosition()
- void Scrollable.setCurrentPosition(int x, int y)

Server Features

HTTP Service 2.0

- Status code and headers can now be returned
- Exec -action added

See HTTP service V2

Since 2.0.0

API Features

Multi-setters for arrays

- ByteArray ByteArray.set(index, value1, value2, ...)
- IntArray IntArray.set(index, value1, value2, ...)
- ByteArray ByteArray.fill(filling)
- IntArray IntArray.fill(filling)

Float parsing

- long parseFloat(string, fraction)

Parses the string argument as floating point number.

Binary resources

- ByteArray getResource(resourceName)

Loads binary widget resource as ByteArray.

Calling serially

- void callSerially(Runnable runnable)

Executes specified operation later, by placing it into the user interface event queue.

Image operations

- Image(int width, int height) constructor
- Image(IntArray rgb, int width, int height, boolean processAlpha)
- Graphics Image.getGraphics()
- Image.getRGB(int[] rgbData, int offset, int scanlength, int x, int y, int width, int height)

Allows creation of Images without Canvas.

Graphics clipping and translating

- void Graphics.translate(int x, int y)
- int, int Graphics.getTranslate()
- void setClip(int x, int y, int width, int height, boolean absolute)
- int, int, int, int getClip()

Allows

- Changing the area of canvas on which drawing will take place
- Drawing sections of image on canvas

(C) Copyright Nokia 2014. All rights reserved.

UI Features

List widget Shells

- List getShells()

Returns list containing all Shells that the widget has in view stack. It is easier to keep track of shells on stack, or keeping reference to all of them.

Getter for focused Style

- Style Style.focused()

Returns focused Style-variant.

Custom component decoration

- void Style.setPainter(PaintCallback callback)

Allows widget code to render a custom component border and background.

Channel API

Completely new framework that allows communication between widgets and phones.

- Public and Private channels can be formed
- Widget can join or leave channels, and list members
- Each member and channel can contain their own context-data (Value) that can be used to set states and hold other information
- Messages (containing service name and data-value) can be broadcasted to all channel members or privately to a selected member

Allows the creation of interactive multi-user widgets without the need for external 3rd party services (such as PHP/perl scripts to hold and serve the messages between widgets)

- Chat / Instant Messaging
- Multiplayer games (use public channel for game lobby to find people to play with, then create a private channel for in-game communication)
- Different kinds of widgets can be connected if they understand messages sent by each other

Note. Messages are not persistently stored, so in to receive a message you have to be on the channel it was sent to.

Note. Current Channels implementation is first draft and not completely tested, and so may contain errors.

TODO: Example widget

Server features

Header support to HTTP-service

HTTP request headers can now be defined by creating *headers* name-value list to the argument structure.

See HTTP service

Token authentication changes

- Changed the signed request format

Removed 'userid'. Now TOKEN is always in plaintext

- Three parameters are now added to the request
  - **token**, that is, user identifier
  - **seed**, a random seed such as a timestamp
  - **sig**, the signature
- Hash algorithm is md5
- Configuration of token auth is now simpler in *widget.xml*
- Example PHP implementation for server side (Partner Web site)

See Token authentication

(C) Copyright Nokia 2014. All rights reserved.

Since 1.0.7

UI Features

Camera

Camera -class replaces previous captureImage() function.

Example code: Archived:Example Camera in WidSets

View

Possibility to dynamically create Views.

TODO: Example.

Size expression parsing

Parses size-expression used with Views.

- int parseSize(String expr, int parentSize)

TODO: Link to apidoc.

Style space consumption

Calculates the total amount of space consumed from specified edges of this style.

- int Style.consumes(int dirs)

```
int styleWidth = getStyle("myStyle").consumes(LEFT|RIGHT);
```

Middle soft key support

Middle softkeys (MSK) can now be labeled, too. Changes:

- getSoftKey() asks for SOFTKEY_MIDDLE
- Prompt(Image icon, String message, MenuItem softOK, MenuItem softMiddle, MenuItem softBack) added

HyperText

New component added which is able to limitedly support markup. HyperText can contain text sections rendered with different styles and links that can be clicked.

Example code: Archived:Example HyperText in WidSets

TODO: Link to apidoc.

Menu opening

Menus can be now explicitly opened using *open()*-method.

- void Menu.open()

Simple image loading

Shortcut for loading images from the Internet (longer version would involve using an image-service)

- Picture getPicture(String url)
- Picture getPicture(String url, int boundx, int boundy, String format)

(C) Copyright Nokia 2014. All rights reserved.

Shows a placeholder image while the image is being loaded.

Note. *You cannot get the encoded binary version of the image this way if you'd like to save it to the Store.*

New Component flags

To be used with components container in a Flow

- HEXPAND - If component has preferredWidth = 0, component shall use all remaining space on current line
- HLIMIT - If component has preferredWith = 0, component actual width won't be wider than the remaining space on the current line

Graphics arc drawing

Drawing of outlined and filled arcs added.

- void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)
- void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)

TODO: Link to apidoc

API features

Hex String functions

- ByteArray decodeHex(String string)
- String encodeHex(ByteArray array)

Value read/write operations

- Value InputStream.readValue()
- void OutputStream.writeValue(Value value)

Date/time formatting

Formatting of time and date according to current locale.

- String format(int format)

```
Calendar cal = new Calendar();
String date = cal.format(FORMAT_DATE);
String time = cal.format(FORMAT_TIME);
String dateAndTime = cal.format(FORMAT_DATETIME);
```

URL encoding

Encodes the specified string using url encoding. Before encoding is applied, string is converted into UTF-8 character encoding.

- String encodeURL(String string)

Note. *When using the HTTP-service, parameters given in params-array are URL-encoded at server-side.*

Registered user checking

Returns true if user has registered a username with WidSets. User who has not registered has anonymous username like *D43285454*.

- boolean isRegistered()

(C) Copyright Nokia 2014. All rights reserved.

Sound player

Sound playing fixed and refactored as a Player class with following features:

- Creating Player from binary resource provided with the widget
- Creating Player from binary array loaded from the Internet
- Playing, pausing and re-starting
- Setting number of loops the sound will be played

Example code: Archived:Example Sound Player

TODO: Apidoc link

Widget Features

Parameter change notification

In a case where the widget's parameters were changed, either from script or from server, script will get notified if it declares following function

- void parametersChanged()

If function is not declared, the widget is re-started so that it will notice and use the new changed parameters (just like the 0.99 version did).

Widget help

Help text can be added to *widget.xml* by declaring following nodes inside the info-section:

```
<help default="en" showFirstTime="true">
  <text lang="en">
    English help text goes here
  </text>
  <text lang="fi">
    Sama suomeksi.
  </text>
</help>
```

By setting attribute *showFirstTime* to *true* the help text is shown to user the first time this widget is opened. Language used in help text is the same as the device (and WidSets client) language. If that language is not found then default is used, in this case English.

Help item can be also be shown in option menus by using *OPEN_HELP* action id constant.

Class templates

Generic base classes can be defined at server side so that one script can be used with multiple widget configurations and skins - templating.

Currently only one base class is available.

- Syndication - See Syndication widgets

More classes to be added.

Instead of declaring the script .he just define

```
<resources>
  <class src="Syndication"/>
</resources>
```


(C) Copyright Nokia 2014. All rights reserved.

Dynamic service invoking

Services can be invoked without declaring them in the *widget.xml*.

- void call(Object state, String serviceVersionAction, Value argument, SuccessCallback onSuccess, FailureCallback onFailure)

serviceVersionAction consist of the service name string (same as used in *widget.xml*), version number of the service schema (currently 1), and service action (previously given standalone in service-call).

```
call(null, "http/1/get", ["url" => "http://www.google.com"], ok, nok);

void ok(Object state, Value ret)
{
  printf("result: "+ret);
}

void nok(Object state, String error)
{
  printf("error: "+error);
}
```

TODO: Link to apidoc

Localization

Widget strings can be localized for different languages.

Example code: Archived:Example Localization in WidSets

Sharing widget

You can open the widget-sharing dialog by using *SEND_WIDGET* action id.

```
Menu getMenu(Shell shell, Component focused)
{
  Menu menu = new Menu()
    .add(SEND_WIDGET, "Share");

  return menu;
}
```

This opens a dialog in which the user can enter the phone number to which the widget is sent. If the phone number is registered for WidSets user's account, a sharing message will be sent to this user's inbox. If user does not have a WidSets account, a new account is created, the widget is added to the dashboard, and a SMS message containing the WidSets client download link is sent to the phone number given.

Platform Changes

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.