This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. The article is believed to be still valid for the original topic scope.

Contents

Introduction to the WidSets SDK

To help the developing of WidSets widgets, a developer kit has been created. The SDK is available for download at Forum Nokia/WidSets SDK.

Writing the code for a WidSets widget requires only a text editor, for example, NotePad is sufficient. However, using a third-party editor that supports XML provides a number of utility functionalities, such as row numbering, syntax highlight, etc., and makes coding more convenient. The WidSets SDK comes with code syntax templates and code highlight support for UltraEdit and EditPlus editors.

Developer kit

- WidSets devkit is a light automated tool to help developers produce feature-enriched widgets.
- Widget developers no longer depend on the WidSets Web site for widget upload and deployment.
  - Development still requires access to a WidSets server, but you there is no need to use the Web site for development-related activities.
- A skinnable tool with many configuration possibilities.
- The unpacked devkit includes both the devkit application and the scripting API documentation.
- To run the devkit, open a command prompt, type in devkit and press enter. All the available options and commands are explained onscreen after this.

Devkit consists of the following components:

- WidSets Scripting Language (WSL) API - A reference to coding your own widgets using the WidSets Scripting Language.
- A command line emulator - Emulator, which acts as an end-user mobile phone. A platform for testing your creations.
- Some example widgets - Samples that you can use as reference to your own development.
- *Getting Started* documentation.

- API document about the features of the service.

Features

- **Built-in compiler for WidSets Scripting Language code**
  - ♦ Compiles the WidSets Scripting Language code when the developer saves the edited source file.
  - ♦ Injects the compiled code for the local use of the emulator.
- **Widget upload**
  - ♦ Checks the syntax of the xml-files.
  - ♦ Compiles the WSL code.
  - ♦ Packages the widget into a zip-file.
  - ♦ Deploys the package to the defined WidSets server.
  - ♦ Automatic widget deployment takes the XML and the resource changes to the WidSets server automatically.
- **Error messages**
  - ♦ WSL script errors.
  - ♦ Widget XML syntax errors.
  - ♦ Stylesheet error checking.
- **Synchronizing the WidSets client**.
  - ♦ Developer can force a synchronization between the mobile device and the web dashboard by calling a sync command.
- **Log window**
  - ♦ Log window that shows the debug details and possible errors (runw command).
- **WSL API documentation.**
  - ♦ Javadoc-styled documentation which describes the available constants and functions.
  - ♦ Deployed with the devkit, resides in the *devkit\api* folder.

Registering as a developer and downloading the SDK

Downloading the SDK is possible even if you have not registered as a WidSets user yet and created an account. However, being able to connect to the WidSets server and taking the SDK into full use requires a login with the account credentials.

To register, go to the developer site at http://www.widsets.com and click the "Register and download" button. Agree to Terms of Service. During the registration process you will need your phone handy to download the WidSet client via one of the methods that will be displayed. Follow the instructions given.

The SDK is distributed as a ZIP file. You can download the latest version from the Forum Nokia site. Remember to read Terms of Service the prior to downloading the SDK. The legal wording is known to differ slightly between client and SDK use

The main page for WidSets developers under the main site is located at http://www.widsets.com/developers.html.

To install the WidSets SDK, simply unzip the package into your local drive under a directory of your choice.

Development tools

The WidSets SDK is a light action packed set of tools, documents, and examples that get you into the world of mobile widgets.

Development requirements

- A personal computer
- A network connection
- Java runtime (1.4+)
- A text editor. (for example GNU Emacs, EditPlus, UltraEdit)
- The WidSets SDK

Supported platforms
The devkit has been developed to work under any platform that supports Java. If you run into any problems, please ensure that your runtime is at least 1.4.

Syntax highlighting
There are some syntax highlighting tools for the WidSets Scripting Language available, found in the devkit folder for the following editors GNU Emacs, EditPlus, UltraEdit. See Editor Syntax Configuration

Usage

- A command line tool.
- The most basic command is simply calling **devkit** in the devkit installation folder.
    - ♦ This command prints out all the options and commands available in the devkit.
    - ♦ Usage is explained next to each option/command.

Use cases:

- `devkit --proxy pryxyaddr:port --auth usr pwd --site default login usr pwd`
    - ♦ The above command defines both proxy and HTTP authentication as connection details, and logs the user to the default server.
- `devkit upload widclock`
    - ♦ The above command first checks the syntax of the widget in the *clock* folder, compiles the code, packages the widget to a zip-file and uploads it to the server.
    - ♦ **Assumption:** There?s a folder called *widclock* inside the *devkit* folder, which holds the example clock widget.

Launching the SDK

To run the SDK, open a command prompt (Windows) or open new shell (Unix), type the command `devkit`, and press Enter. As a result, a list of all available options and commands with a short description will be displayed on the screen.

```
C:\sdk\devkit>devkit
What do you want to do?

Usage:
  devkit <options> command <arguments>

Options:
 --site SITEFILE          Configuration of widsets site to use
 --proxy HOST:PORT        Host and pointer of HTTP proxy server
 --noproxy                Don't use HTTP proxy
 --tunnel                 Enable tunneling (when on VPN with no HTTP proxy)
 --notunnel               Don't use tunneling
 --auth USER PASS         Username and password of website authentication
 --noauth                 Don't use website authentication
 --skin SKIN              Skin for emulator
 --force                  Force to use (old) existing of site config
 --client CLIENTJAR       Override embedded client with specified .jar

Commands:
  login USR PWD           Login to widsets site
  check [DIR]             Check the widget at current or referred directory
  upload [DIR]            Upload the widget at current or referred directory
  massupload FILE         Upload number of widgets into system using specified
                          script file
  massreplace FILE        Upload number of widgets into system using specified
                          script file. The user must own all the widgets being
                          replaced.
  sync                    Request dashboard synchronization, forces client to
                          start sync (you should have emulator or mobile client
                          started when calling this)
  writejad JADFILE        Write connectivity details to given .jad file
  compile FILE [LANG]     Checks given script
  readtheme FILE          Checks given stylesheet
  printtheme FILE         Checks and prints given stylesheet
  info                    Print out authentication information
  clean                   Cleans the record stores from current site
  run [DIR [LANG]]        Run the emulator. If DIR is given the emulator
                          will compile *.he files at the directory when needed
                          and replace corresponding widget scripts at dashboard.
                          By default the compiler localizes the scripts using
                          *.properties file with same name as the script, but
                          shared backup file file can be specified as second
                          argument
  runw [DIR [LANG]]       Otherwise like run but opens a separate window
                          for output


C:\sdk\devkit>
```

Note: This mechanism for loading language properties is for testing purposes only. The localization of the script can be verified this way. *@keyword* strings can be inserted into the code and the actual values for the keywords will come from the properties file.

DevKit uses the following syntax:

```
devkit <options> command <arguments>
```

If you are behind proxy, or VPN tunnel, you need to set the options described in the following step before the command (and its arguments) you are about to execute.

Let?s go through a few steps to setup the DevKit in your environment and login to the WidSets server with your WidSets account.

- Do this if your current network connection goes through a proxy (HTTP):

Set the proxy parameter:

```
devkit --proxy yourproxyaddress:port <COMMAND>
```

- If you have defined a proxy before and want to remove it (it is off by default):

```
devkit ?-noproxy <COMMAND>
```

- Do this if your current network connection goes though a VPN tunnel:

```
devkit ?-tunnel <COMMAND>
```

- Log in by defining your developer account username and password.

```
devkit login yourusername yourpassword
```

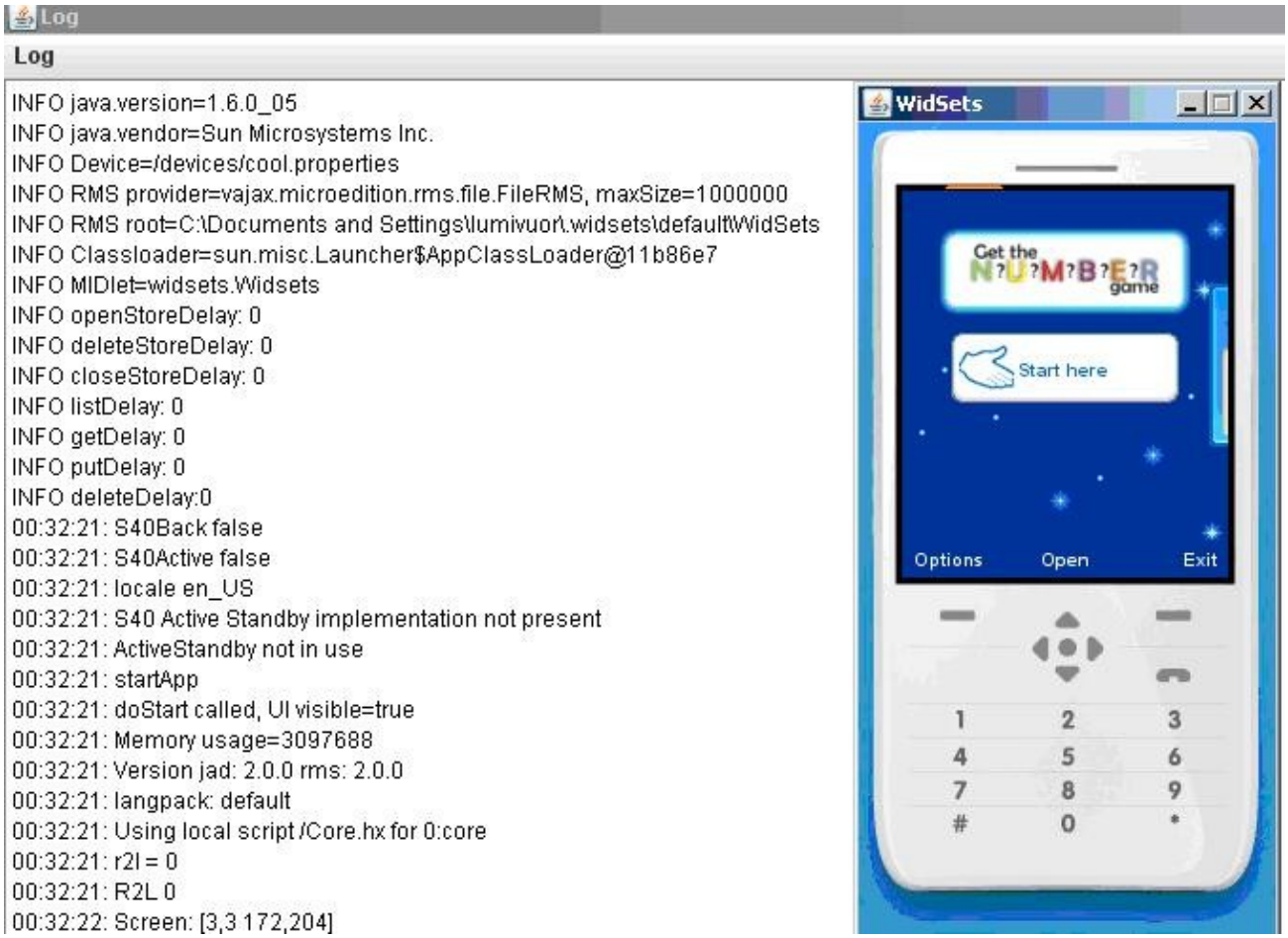
Note 1: Make a login.bat file with the @echo option set "off" to avoid retyping this every time you login and to hide your password.

Note 2: Usually when the SDK prints out "Connection to [URL] could not be established in timely manner." it means that you are behind a firewall and need to configure a http proxy.

- Starting the emulator:

To see the SDK emulator in action, type:

```
devkit run
or to enable log window;
devkit runw
```



Connectivity

- The site to which the emulator connects to can be defined using the **--site option**.
- There are multiple WidSets environments.
- Currently the default is the WidSets developer site.
- The connectivity details are kept in a configuration file.
    - The config file can be found in the users' home directory under the *.widsets* folder
    - Example: *C:\Documents and Settings\Developer\.widsets*
- There can also be additional config files for connecting to multiple WidSets servers.
    - Create a folder in *~\.widsets\insert_your_username_here*
    - Insert a config file there.

An example configuration file

```
# needed values
login.url=http\://developer.widsets.com/dl/devkitjad       # url where the jad file is read (conn
upload.login.url=http\://developer.widsets.com/login.dox   # url used for login
upload.url=http\://developer.widsets.com/actualDevUpload   # url used when uploading widgets
sync.url=http\://developer.widsets.com/updatePhone.dox     # url used for calling a synchronizati
timeout.socket=8000                                        # timeout for socket connection
timeout.connect=4000                                       # timeout for http connection
```

```
via=83.145.232.75\:443                                    # tunnel ip used for tunneling socket
tools.version=0.1

# these can be created with the devkit tool from the above
username=your_username                                    # username
password=your_password_here                               # password
proxy=yourproxy.site.com\:8080                            # proxy configuration
device=cool                                               # skin used in the emulator
tools.version=0.1
window.emulator=1076,266
window.log=0,0,500,700                                   # log window dimensions
last.login=1171015273343

# JAD-properties from the system
MicroEdition-Configuration=CLDC-1.0
Manifest-Version=1.0
MIDlet-Jar-URL=lift.jar
MIDlet-Name=WidSets
MIDlet-1=WidSets, icon.png, widsets.Widsets
MIDlet-Vendor=WidSets
MIDlet-Version=
MIDlet-Jar-Size=0
MicroEdition-Profile=MIDP-2.0
queue=lift/lift/Developer/queue                           # which queue is used (user specific v
url=socket\://developer.widsets.com\:2081                 # socket url from the jad-file
url2=http\://developer.widsets.com\:2080/comm             # http url from the jad-file
key=1ff3243219b133f008202e821f14df90ec9c76c5374fee0d6152f61dc0677175 # encryption key value used
```

Resetting the devkit

On some occasions the developers may run into RMS related problems. To clear the RMS of the WidSets client running under the emulator, user can either manually delete the contents of the skin folder or just simply delete the whole folder

An example of a folder used for RMS: *C:\Documents and Settings\username\.widsets\default\cool*

- ◊ **devkit login username password**
  Empties the RMS
- ◊ **devkit clean**
  Cleans the RMS

If there are further problems, you may need to login to the server again.

Additional notes on developing

- Detailed information about the stylesheet syntax is deployed in the scripting API documentation under *devkit/api/widsets/api/stylesheet.html*.
- Check the scripting API for details on the available function usage.
- Syntax Highlighting available for UltraEdit32 to ease the WidSets Scripting Language script developing.

See also

- **WidSets SDK**
- WidSets Client
- WidSets Scripting Language
- Configuring an editor for syntax highlighting
- Widget examples
    - WidClock
    - Memory Game
    - Filter test
    - Hello World
    - UITest