



SwiftPush – WAP Push Developer Toolkit (WAP 1.2.1)

Technical Specification Sheet

SwiftPush, NCL's WAP Push Developers Toolkit, is a java implementation of the protocols used to send bookmarks, configurations, phone browser operations to mobile phones over the short message service of cellular networks. SwiftPush makes internet navigation a more enjoyable experience for end users and mobile internet applications and services more accessible.

SwiftPush provides an API to developers for sending Cache Operations, Service Indicators, Service Loading, OTA Settings (Over The Air Mobile Internet Configuration) and Bookmarks. The API is simple to use as can be seen below.

SwiftPush supports the following operations:-

- **OTA Settings** - sending mobile internet configuration settings on the handset. These settings can be dialup settings (telephone number, IP address, home page, user name and passwords etc.) or GPRS settings (APN, IP address, etc.).
- **Bookmarks** - send favorite and easy install links to a handset using the SwiftPush. A bookmark contains a URL and a name. When it arrives on the phone the user has the option to save it to memory.
- **Service Indicators** - like bookmarks, containing text and a URL but the user is prompted to retrieve the data as specified by the URL. An example would be 'You have 3 new emails' and the URL would point to a Internet site from where the user can view the emails. These service indicators can be replaced.
- **Service Loading** operations is the process of sending a URL to a handset and the handset retrieves the internet page with or without user intervention. An example application would be a polling service whereby a (WAP) page automatically appears on the users phone and is prompted to make a selection from the list.
- **Cache Operations** are used to invalidate pages or groups of pages stored on the handsets cache. For example if a user had a 'WAP coupon', it could be removed once the user availed of it.

EXAMPLE

```
import ie.ncl.wap.push.*;
import ie.ncl.msg.*;

//Establish a SMS transport
Sender sender = Factory.getSender("mySender");
sender.connect();

//Create a WAP Push transport
WAPPush push = new WAPPush(sender);

//Send the objects.
Address me = new Address("+353812345678");
Bookmark bm = new Bookmark("Yahoo!",
    new URL("wap.yahoo.com"));
push.send( bm , me);
```

SwiftPush API is designed to be transport independent so it will run across any protocol/device supported by the underlying SwiftNote (SMSJDK) toolkit. SwiftPush API friendly as can be seen in the example above.

Requirements - SwiftNote (SMSJDK)

The bearer used to transmit is SMS and SwiftPush uses NCL's SwiftNote product (SMSJDK) to transmit the push requests. The SMS access can be over a cellular device or via a short message service centre (SMSC). See NCL's SwiftNote Technical Specification Sheet for more information.