

USSD Center

Developers Guide

Draft 1.A

26th January 2009

**CONTENTS**

[**1.** **DOCUMENT INFORMATION** 3](#_Toc223876428)

[**1.1.** **Document History** 3](#_Toc223876429)

[**1.2.** **References** 3](#_Toc223876430)

[**1.3.** **Glossary of Terms** 3](#_Toc223876431)

[**2.** **INTRODUCTION** 4](#_Toc223876432)

[**3.** **USSD BROWSER** 5](#_Toc223876433)

[**4.** **USER SESSION VARIABLES** 6](#_Toc223876434)

[**5.** **SERVICE CREATION** 8](#_Toc223876435)

[5.1. Service Operation 8](#_Toc223876436)

[5.2. Subscriber Access 9](#_Toc223876437)

[5.3. External Connections 9](#_Toc223876438)

[**6.** **USER PROFILE** 10](#_Toc223876439)

[**7.** **SERVICE CREATION ENVIRONMENT** 11](#_Toc223876440)

[**8.** **CONCLUSION** 12](#_Toc223876441)

1. **DOCUMENT INFORMATION**
	1. **Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Comments** |
| 26th Jan 2009 | Draft 1A | Moses Mwangi | Document created |
| 3rd March 2009 | Draft 1B | Amos Njenga | Document updated with Service Creation |

* 1. **References**

|  |  |  |
| --- | --- | --- |
| **Reference** | **Source** | **Description** |
| Opencode\_USSDBrowser\_User\_Guide\_ED001908USR\_271.pdf | Opencode | USSD Browser users guide |
|  |  |  |

* 1. **Glossary of Terms**

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| SCE | Service Creation Environment |
| USSD | Unstructured Supplementary Services Data |
| HTTP | Hyper Text Transfer Protocol |
| SOAP | Simple Object Access Protocol |
| VXML | Voice eXtensible Markup Language |
| USSDB | USSD Browser |
| USSDGW | USSD Gate Way |
| MAP | Mobile Application Part |
| VLR | Visitor Location Register |
|  |  |

1. **INTRODUCTION**

Safaricom has installed a new Programmable USSD Platform to cater for future growth and features a Service Creation Environment.

This document is meant as a developers guide for integrating external applications to the new USSD platform.

1. **USSD BROWSER**

The USSD Browser (USSDB) is the core application of the USSD platform that processes, executes and manages USSD menu sessions between mobile users and USSD value added services.

The USSD Browser uses application modules, called connectors, which implement different protocols, to interact with external applications. Examples of protocols and/or connectors implemented are:

* HTTP\_CLIENT
* HTTP\_SERVER
* SOAP\_CLIENT
* ORACLE\_CLIENT
* MYSQL\_CLIENT
* SMPP\_SERVER

The USSDB provides a web interface for creation of service menus. The menus are stored as VXML files and are interpreted by a proprietary VXML interpreter.

While creating a service on the new USSD platform, there are two models are possible:

* Create all static menus on the USSDB and only trigger to external applications for dynamic data.
	+ Advantages:
		- Faster service response hence better user experience.
		- Ability to reuse menus on different USSD services
	+ Disadvantages:
		- Menu maintenance can be tricky as it is done centrally on the browser.
* Forward all requests to an external application for processing. No menus are created on the USSDB.
	+ Advantages:
		- Service menu maintenance does not require access to the USSDB
	+ Disadvantages:
		- Every request triggers to external application.
		- Does not utilize the rich features of the new USSD platform
1. **USER SESSION VARIABLES**

The USSDB maintains a set of user session variables for each session opened by a mobile user. These session variables can be availed to external applications and can also be used on the USSDB to control the service menu logic and/or flow.

This section describes the session variables that are available to the service creator.

|  |  |
| --- | --- |
| **Variable** | **Description** |
| ORIG | MSISDN of the originator of the USSD user session(the mobile user) |
| DEST | Service code of initially activated service |
| SESSION\_ID | Internally assigned session ID |
| USSD\_PARAMS | USSD string that user has entered, without servicecode |
| USSD\_PARAM\_COUNT | Number of USSD parameters, not counting servicecode |
| USSD\_PARAM\_0 | Service code of initially activated service |
| USSD\_PARAM\_N(where N is a number >= 1) | Subsequent USSD parameter |
| NOW\_DATE | Current date in format YYYY-MM-DD |
| NOW\_TIME | Current local time in format HH:MM:SS |
| IS\_RESUMED | Indicates whether a session is new or resumed( It will have "true" value if resumed, otherwise -"false" ) |
| <NODE\_ID>\_choice | Contains the choice user has made in a menu nodewith ID <NODE\_ID>Example:If the ID is Menu1, then the variable Menu1\_choicewill contain the user input.If the ID is not Menu, then the variable choice will beempty. |
| <NODE\_ID>\_menu | Name of the chosen option in a menu with ID<NODE\_ID> |
| <variable\_name>\_raw | Each variable defined in Assign Node or VariableReturn has its raw (absolutely unprocessed) valuestored in the variable. |
| PREVIOUS\_NODE\_ID | ID (the part after #) of the previous node |
| PREVIOUS\_NODE\_URI | Fully qualified URI of the previous node |
| SESSION\_START\_URI | URI to the beginning of the session – going there islike restarting the session |
| LOAD\_MPS\_TOTAL | Overall messages (network in and out) per second. |
| LOAD\_MPS\_SC\_<Service Code> | Load of MPS for the given Service Code. If the code does not exist, the return value will be "-1". |
| MAP\_USSD\_PARAMS | Contains a USSD string sent by mobile user in thecase where more parameters are sent between USSDGateway (USSDGW) and USSD Browser applications |
| MAP\_IMSI | Contains IMSI received in MAP request |
| MAP\_MSISDN | Contains the MSISDN of the originator. This parametercan be used if USSDGW reformatting function is usedand if %ORIG% is set to IMSI, instead of to MSISDN. |
| MAP\_VLR | Contains VLR sent in MAP request to USSDGW |
| MAP\_HLR | Contains HLR sent in MAP request to USSDGW |
| MAP\_MAPVER | Contains MAP version (1 | 2) |

Other variables that the USSD Browser can avail to external applications include:

* Menu Constants
* Session variables collected during menu browsing
* Session variables assigned in ASSIGN nodes
* User profile variables
* Substrings of session variables
1. **SERVICE CREATION**

This section gives implementation details of third party applications including application layout. To enable the service to be created on the Gateway, the following will have to be established:

* + Service Code through which the service will be accessed
	+ Full HTTP URL of the application
	+ Whether the subscriber is charged or not
	1. **Service Operation**

All requests forwarded to 3rd Party Applications will be through a simple HTTP GET using HTTP\_CLIENT connector with the all or some of the following parameters forming part of the request.

The USSD Browser Application extracts available USSD parameters related to request. The required parameters are then appended to the URL as defined in the third party application specification. Unless other parameters are specifically requested by a 3rd party application developer, through the proper and predefined request channels, the following parameters will be availed to any 3rd party application:

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| ORIG | MSISDN of the originator of the USSD user session (the mobile user phone number in the 12 digit format “254 7xx xxxxxx”). |
| DEST | Service code of initially activated service including the initial asterisk (“\*”) and trailing hash (“#”) |
| SESSION\_ID | This is a session ID generated by the USSD Gateway. |
| USSD\_PARAMS | USSD string that user has entered, without service code |

The action taken by the Browser application in Safaricom side will depend on the response from the third party application and nature of the application

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Request** | **Response from Third Party Application** | **Action by this service** |
| 1 | \*SC\*<parameters># |  | Forward to the application and wait for a response. Generate a CDR or charge as defined for the service |
| 2 |  |  “CON <message>”  | Forward the <message> part and wait for a response. Generate a CDR or charge as defined for the service |
| “END <message>” | Forward the <message> part and close the dialog. Generate a CDR or charge as defined for the service |
| Any other response | Generate a CDR or charge as defined for the service |

* 1. **Subscriber Access**

The Controlling USSD service, which resides on the USSD Browser, is designed as a versatile session service (with no interactive menu) that is open to all Safaricom subscribers regardless of profile (PrePay, PostPay or even Hybrid) or location HPLMN, Roaming) unless otherwise advised by the CP or due constraints imposed by any specific standards that apply to the service or are enforced in the offering of the service.

The decision to continue or end a session is derived from the command prefix from the 3rd party application. The Subscriber Menu navigation is therefore fully with the 3rd party application through the proper use of the “CON” and “END” commands.

* 1. **External Connections**

The service uses an external HTTP\_CLIENT **connection** to send the received USSD string the third party application as described in the above section. The connection must first be created as a connector to the third party provided URL.

1. **USER PROFILE**

The USSD Center has a user profile database that can be used by services created on the USSD Browser to provide a rich experience to users e.g. multi language services.

The user profile repository can be updated by external systems or by local services and scripts. Updating of the profile database does not impact live running services

The table below describes the user profile variables that are stored in the user profile repository:

|  |  |
| --- | --- |
| **User Profile Variable** | **Description** |
| USER\_MSISDN | Originator MSISDN |
| USER\_PROFILE\_ID | Profile ID of the subscriber |
| USER\_PROFILE\_NAME | Profile name |
| USER\_IMSI | Subscriber IMSI |
| USER\_INFO1-10 | Fields for storing data from different USSD menus |
| USER\_LAST\_MODIFIED | Date user was last modified |
| Custom Fields | On demand fields can be added |

The user profile repository can be used to provision services and also control the service call flow. For example, the USSDB can allow access to a certain USSD service to subscribers with a certain user profile setting only (provisioning).

A common user profile variable is language ID; it can be used to provide a multi language USSD menu. Since it is possible to update the user profile fro a USSD service logic, it is possible to provide a link in the service menu for subscriber self administration.

1. **SERVICE CREATION ENVIRONMENT**

The USSD Browser features a web-based SCE (Service Creation Environment) that can be used for creating and managing USSD menus. The menus thus created are stored on the USSD Browser as VXML files that are executed by a proprietary VXML interpreter each time the menu is accessed.

While creating a USSD service on the SCE, it is possible to configure the menu on the USSD Browser itself, or to pass the request to an external application (of course with additional parameters/variables if needed). This logic is applied per USSD transaction, so that for the same service, it is possible to create the first level menu (for example) on the USSDB but pass the second level request through to the external application.

The external application that needs to receive USSD requests from the USSDB must expose either a HTTP or SOAP web service interface. The interface must provide for the parameters/variables that it desires to receive from the USSDB. The parameters/variables are passed either as HTTP GET or POST variables or SOAP call parameters.

The external application must respond in real time for the USSD session not to timeout. The response from the external application is assigned to a variable in the USSDB, and it can be returned to the mobile as a menu or it can be used on the USSDB to construct a menu to be sent to the mobile.

For application initiated USSD sessions, the interface can be either HTTP or SMPP.



1. **CONCLUSION**

The USSD Browser provides higher flexibility while interfacing with external applications. It always acts as a client to external applications for Mobile Initiated sessions and as service for application initiated sessions. All communications between the USSDB and external applications is synchronous. The response to each request should contain the actual data to be sent to the subscriber.

The USSDB also affords the external applications a wide array of parameters/variables that the external application can use to create a very rich user experience. For example, it is possible to user the MAP\_VLR session variable to provide a different service flow depending on whether the subscriber is roaming or not. The user profile variables can be used to provide a Multilanguage service, for example.

For each service, it is important to analyze and decide which login will be handled on the USSDB and which will be handled by an external application.