



Your Partner In Your Mobile & Online Presence

Last Invention® USSD Application Development Platform (Gateway 1) API **User Manual**

lastinvention.co.za



CONTENTS

1. Version Control	3
2. USSD Developers Guide - Introduction	4
3. USSD Functional Mechanism Overview	5
3.1 Internals of the Last Invention® USSD Daemon	5
4. XML Samples	6
4.1 Example 1	6
4.2 Example 2	6
4.3 Example 3	6
5. XML Parameters	7
5.1 <request> tag	7
5.2 <headertext> tag	7
5.3 <options> tag	7
5.4 <option> tag	7
5.5 Attributes	7
6. Example Session	8
7. Parameters added in query string	9
8. Process Flow	10
9. USSD Timeouts and Character Limitations	11
9.1 MT (Mobile Terminating) Traffic	11
9.2 MO (Mobile Originating) Traffic	11
9.3 Network Session Timeouts	11
9.4 Network Charge Periods (charged to USSD dialler)*	11
10. Integration	12
11. Retrieval of your project's Last Invention platform information	13
12. Support	15
13. Trademark notice	16



1. Version Control

Revision Number	Revision Date	Author	Changes
0.1	20 Feb 2015	Thato Dipudi	Initial Document Release
0.2	21 Feb 2015	Thato Dipudi	Added "Process Flow" topic
0.3	22 Nov 2016	Thato Dipudi	Updated file name, gateway number and base USSD code
0.4	04 May 2017	Thato Dipudi	Updated document name and trademark information
0.5	22 Apr 2018	Thato Dipudi	Added section: Retrieval of your project's Last Invention platform information



2. USSD Developers Guide - Introduction

Unstructured Supplementary Service Data, abbreviated as USSD, is a capability of all GSM phones and all networks.

Response times for interactive USSD-based services are usually quicker than those used for SMS.

As an analogy, USSD is similar to telnet, while SMS is similar to e-mail.

USSD operates by using a number such as *120*123# to trigger a menu structure which is displayed on the users handset. Our USSD URL Caller solution allows for developers to create their own menu structure with the use of HTTP and XML. A simple communication with our server allows for the information to be delivered to the handset.



3. USSD Functional Mechanism Overview

This section of the document explains in general how the Last Invention® USSD Daemon (application processing) goes about handling USSD sessions. USSD implementation centres around *URL based requests and responses*.

Upon creation of a new USSD campaign:

- The client is required to create the URLs for the USSD daemon to submit the first request and subsequent responses to. Later on in this doc, examples of the XML and an example USSD session is given to aid in understanding of the system.

3.1 Internals of the Last Invention® USSD Daemon

Upon the USSD Daemon calling the first “USSD Request” URL, the daemon will read the XML located at the URL. The Daemon will then construct a menu structure for the subscriber and will create a response URL based on the subscriber’s selected menu option.

This will continue for each “USSD response” until the USSD session has been completed.

In the next section, an example of the XML, a breakdown of the XML parameters and a simple example of a USSD session is given.



4. XML Samples

Below are the examples to demonstrate the XML required to be generated by clients for their USSD campaigns. Here we can see a simple 3 question campaign and the XML which will represent each question.

4.1 Example 1

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <headertext>Can you answer the question?</headertext>
  <options>
    <option command="1" order="1" callback="Developer's Application URL"
      display="true">Yes I can!</option>
  </options>
</request>
```

4.2 Example 2

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <headertext>Can you answer the second question?</headertext>
  <options>
    <option command="1" order="1" callback="Developer's Application URL"
      display="true">Yes I can!</option>
    <option command="2" order="2" callback="Developer's Application URL"
      display="true">Take me back to the first one!</option>
  </options>
</request>
```

4.3 Example 3

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <headertext>You passed the test. Congratulations!</headertext>
</request>
```



5. XML Parameters

5.1 <request> tag

This tag encapsulates the entire XML structure.

It has no attributes.

It has the following subtags: <headertext> and <options>.

<headertext> is mandatory, <options> is optional. Each subtag may only appear ONCE in the XML.

5.2 <headertext> tag

This tag contains the message that will be displayed on the mobile user's handset. It has no attributes.

It has no subtags.

5.3 <options> tag

This tag contains any menu options that are to be displayed on the mobile user's handset. It has no attributes.

It has the subtag <option>, which is mandatory and can be specified multiple times (for each menu option).

The <options> tag itself is optional. Should it not be present, the USSD service will only display the text in the <headertext> tag, and then terminate the session.

5.4 <option> tag

This tag specifies a single menu option. The inner text of this tag is the menu text that will be displayed on the mobile handset.

It has no subtags.

It has the following attributes: *command*, *order*, *callback* and *display*.

5.5 Attributes

command - the character sequence that the user must enter on his handset to select the menu option. Must be convertible to a 16-bit signed integer.

order - the order amongst the menu options that this menu option will be shown. Must be convertible to a 16-bit signed integer.

callback - the URL that will be called when this menu option is selected by the mobile user. It is assumed that the URL will also return XML data (in this format) that will be used to determine the response to be sent to the mobile user's handset.

display - if set to true, the menu option will be displayed on the user's handset; if set to false, the menu item will not be displayed on the user's handset.



6. Example Session

Using the example XML on the **XML Sample** section above and assuming that the client has configured the USSD service to call a URL that returns the first example XML response, when the mobile user initiates a session:

```
Can you answer the question?  
1 - Yes I can!
```

The mobile user enters 1.

```
Can you answer the second question?  
1 - Yes I can!  
2 - Take me back to the first one!
```

The mobile user enters 2.

```
Can you answer the question?  
1 - Yes I can!
```

The mobile user enters 1.

```
Can you answer the second question?  
1 - Yes I can!  
2 - Take me back to the first one!
```

The mobile user enters 1.

You passed the test. Congratulations!

The USSD session ends.

For **free text** answers, add 1 option with no answer text and set the display to false. The returned input will be returned in the “request” parameter in the query string. The example below shows how to declare a free text answer type question:

```
<?xml version="1.0" encoding="utf-8" ?>  
<request>  
  <headertext>Please enter you name.</headertext>  
  <options>  
    <option command="1" order="1" callback="Developer's Application URL"  
      display="false"></option>  
  </options>  
</request>
```

If the user returns “Mark”, the USSD service will append the request parameter to your callback url. Below would be the final outcome of the question:

http://Developer_Application_URL?msisdn=132123123&request=Mark&provider=CellC&ussdSessionId=16473528



7. Parameters added in query string

The USSD Service will always add the following parameters to the query string of any URLs to call. Developers may use these parameters in any way that they deem fit.

msisdn = mobile number of the user in international format (e.g. 27820000000).

request = the input that the user submitted from their device. This is used for Free Text answers.

provider = indicates which mobile network operator that the *msisdn* is on . The possible values are: Vodacom, MTN, CellC or Telkom_8ta

ussdSessionId = indicates the USSD session ID. This is a random number assigned by the mobile network.

Below is an example of a URL called in a USSD session:

`http://Developer_Application_URL?msisdn=132123123&request=Testing&provider=CellC&ussdSessionId= 5528945429`

Note:

The above example will be URL encoded.



8. Process Flow

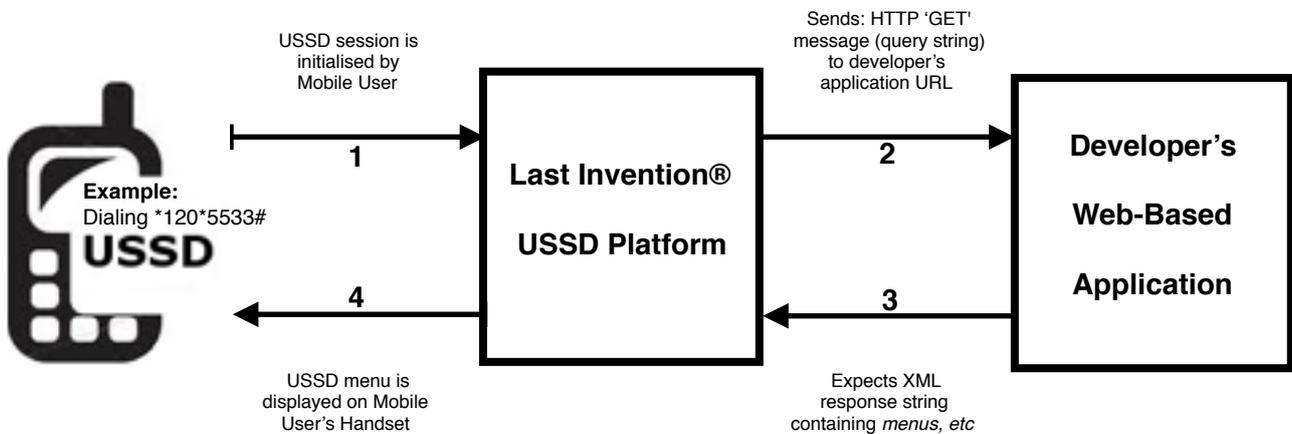


Figure 9.1: Process Flow

- (1) Last Invention® USSD Platform receives initial USSD menu request from the mobile user (dialler) and starts the USSD menu session using this API specification.
- (2) Last Invention® USSD Platform sends HTTP GET request message containing request data (**query string**) to predefined URL (Developer's Application URL).
- (3) Last Invention® USSD Platform waits for XML response string from developer's application.
- (4) XML response (from developer's application) contains USSD menu information that is sent back to the mobile subscriber, via Last Invention® USSD Platform.

Important notes:

HTTP GET Request Data (sent from Last Invention® USSD Platform to Developer's Application URL on USSD session initiation):

- Below is an example of a URL called in a USSD session:
`http://Developer_Application_URL?msisdn=132123123&request=Testing&provider=CellC&ussdSessionId= 5528945429`
- Also see "Parameters added in query string" section of document.

XML Response String (Developer's Application back to Last Invention® USSD Platform):

See the following sections of the document:

- XML Samples
- XML Parameters
- Example Session



9. USSD Timeouts and Character Limitations

9.1 MT (Mobile Terminating) Traffic

Network	First Menu Character	Remaining Menu's
All	140	160

9.2 MO (Mobile Originating) Traffic

Network	First Return Characters	Subsequent Return Characters
All	160	160

9.3 Network Session Timeouts

Network	Minutes
Vodacom	3
MTN	4
CellC	2
Telkom_8ta	3

9.4 Network Charge Periods (charged to USSD dialler)*

Network	Minutes / Seconds
Vodacom	60 cents per minute
MTN	20 cents per 20 seconds
CellC	20 cents per 20 seconds
Telkom_8ta	20 cents per 20 seconds

*All figures inclusive of VAT



10. Integration

USSD can be used for a multitude of purposes, including:

- Contests
- Polls
- Voting

Due to the fact that USSD calls a URL, it also passes the users MSISDN through to your URL allowing you to make use of any of our other mobile solutions and run business logic.

Example:

- Upon receipt of a USSD call, an SMS can be sent to the user.



11. Retrieval of your project's Last Invention platform information

You can retrieve your project's Last Invention platform account information via the following API.

Retrievable information:

- a. Your company name (as stored on the platform) => entity_company_name.
- b. Your unique project identifier => entity_project_id.
- c. The package of the service your project is using => entity_package.
- d. The number of USSD application menus => ussd_no_of_menus. Note that for Developer Program services, this tag will have a value of 0.
- e. The cell phone number of the project administrator => admin_number.
- f. The email address of the project administrator => admin_email_address.
- g. The project status (ACTIVE / INACTIVE) => account_status.
- h. The number of SMS credits in the account => sms_credits_balance. For the developer program, we keep it at 5 because we may have system generated SMS alerts that the account admin needs to receive (example is when the account reaches their low-threshold of 100 session credits).
- i. The number of USSD session credits in the account => ussd_sessions_credits_balance.
- j. The date and time that the project was created on the Last Invention platform => date_created.

A. You need to call the URL ...

http://ussd.lastinvention.co.za/ussd/clientaccount/?ussd_code=xxx&proj_id=yyy
Where...

xxx = URL encoded USSD code, and

yyy = your project Id

You need to request your project Id from the support department. Please keep the project Id secure.

B. The response from our server will be ...

```
<?xml version="1.0" encoding="utf-8" ?>
<ussd_acc_info>
  <entity_company_name>Your company name on our system</entity_company_name>
  <entity_project_id>Your project id</entity_project_id>
  <entity_package>MyOwn_SME_USSD</entity_package>
  <ussd_no_of_menus>8</USSD_no_of_menus>
  <admin_number>27824638004</admin_number>
  <admin_email_address>name@domain.co.za</admin_email_address>
  <account_status>ACTIVE</account_status>
  <sms_credits_balance>1000.0</sms_credits_balance>
  <ussd_sessions_credits_balance>572.0</ussd_sessions_credits_balance>
  <date_created>2019-08-16 14:02:01.786330</date_created>
</ussd_acc_info>
```



If there is no account for the enquiry, the response will be ...

```
<?xml version="1.0" encoding="utf-8" ?>
<ussd_acc_info>
  <entity_company_name></entity_company_name>
  <entity_project_id></entity_project_id>
  <entity_package></entity_package>
  <ussd_no_of_menus></USSD_no_of_menus>
  <admin_number></admin_number>
  <admin_email_address></admin_email_address>
  <account_status>Account does not exist</account_status>
  <sms_credits_balance></sms_credits_balance>
  <ussd_sessions_credits_balance></ussd_sessions_credits_balance>
  <date_created></date_created>
</ussd_acc_info>
```

NB:

The xml structure may be amended later. Check the latest API document for any changes.



12. Support

We've tried to put all the information we could in this Guide, but we are sure you'll have some questions.

If you would like to chat to us, please contact on any of the following channels:

- Phone: +27 (0) 82 463 8004
- WhatsApp: +27 (0) 82 463 8004
- Fax: +27 (0) 86 547 5309
- USSD Client Platform (South Africa only): Dial *120*5533#
- Skype name: lastinvention

Email:

- Sales: sales@lastinvention.co.za
- Support: support@lastinvention.co.za
- General Info: info@lastinvention.co.za

Business Hours:

- Time Zone: GMT + 2 (Pretoria, South Africa)
- Monday - Friday: 09H00 to 17H00
- Saturday: 09H00 to 12H00
- Closed on Sunday and Public holidays ([Holiday Schedule](#))

Please also visit our website and submit a support ticket at:

Secure Client Portal: <https://my.lastinvention.co.za/>

Main website: <https://lastinvention.co.za/>



13.Trademark notice

Last Invention and **MyCVMate** are registered trademarks of The Last Invention (Pty) Ltd.