

Application Note

USSD Services for Interactive Mobile Users

Building User-Friendly Mobile
Telephony Applications Using
Dialogic[®] Distributed Signaling
Interface Components

USSD Services for Interactive Mobile Users

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

Application Note

Executive Summary

The application note describes the implementation of an interactive Unstructured Supplementary Service Data (USSD) service for GSM mobile users. Information is provided to describe how service providers who decide to use the power of USSD to provide mobile users with menu-driven, interactive services such as mobile account balance checking and top up, sports scores, and news and weather information can do so via current technologies and signaling.

An overview of the network architecture and a description of interactive USSD services are presented, as well as signaling requirements for such services. Sample code also shows an application that can be developed using the Dialogic® Distributed Signaling Interface (DSI) MAP Layer.

Finally, system design considerations are discussed, showing how high-performance USSD application platforms can be developed with Dialogic® DSI Components, enabling system designers to offer interactive USSD services worldwide.

USSD Services for Interactive Mobile Users

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

Application Note

Table of Contents

Introduction	2
Network Architecture for USSD Signaling	2
USSD Signaling Requirements	3
Example Menu-Driven USSD Application	3
How to Format USSD Text	5
Formatting a USSD Menu	5
Creating USSD Applications	6
MAP Services for USSD	6
MAP Parameters for USSD	7
Information for System Design: Working with MAP USSD Signaling	8
Code Modifications to the Dialogic® “MTU” Example Application	8
Dialogic® DSI Components for USSD Services	11
Summary	13
References	13
Acronyms	14
For More Information	14

USSD Services for Interactive Mobile Users

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

Introduction

Unstructured Supplementary Service Data (USSD) is a capability built into GSM phones, much like the Short Message Service (SMS).

USSD differs from SMS as follows:

- SMS uses a “store and forward” technique to deliver text messages:
A text message is first sent to a sender’s Short Message Service Center (SMSC) before the SMSC tries to deliver the text message to the recipient. The sender receives an indication of whether the text was successfully received by the SMSC; however, the text message is not guaranteed to reach the recipient instantly.
Note: The “SMS Router” concept can be used to improve SMS delivery time in some instances.
- USSD information is sent directly from a sender’s mobile handset to an application platform handling the USSD service. The USSD service can be located either in the sender’s mobile network or in a visited mobile network.

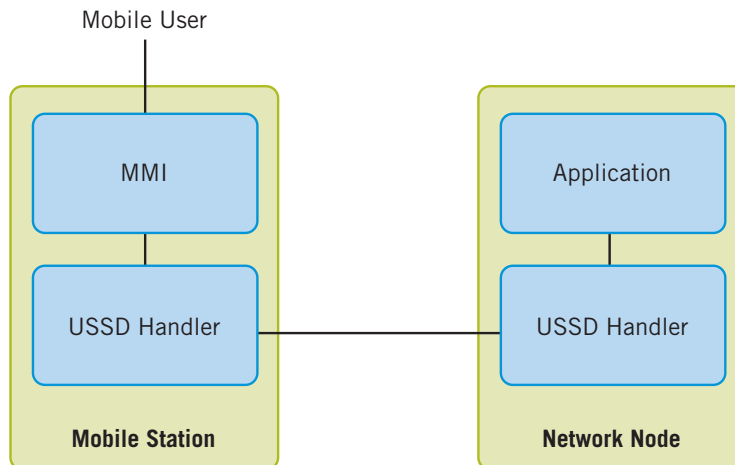
- A real-time “session” is initiated between the mobile user and the USSD application platform when the service is invoked, allowing data to be sent back and forth between the mobile user and the USSD application platform until the USSD service is completed.

This concept of a real-time data session is particularly useful if opting to build an interactive menu-driven application, such as a mobile-initiated “Balance Enquiry and Top Up” application, as described in the *Example Menu-Driven USSD Application* section.

Note: A USSD service could be invoked by either the mobile user or the USSD platform [GSM TS 23.090 Unstructured Supplementary Service Data (USSD) - Stage 2].

Network Architecture for USSD Signaling

USSD services use the existing architecture of GSM networks. A user dialing a USSD service code initiates a dialog with a USSD application residing in a mobile network, as detailed in Figure 1. The “Network Node,” shown in Figure 1, could be either an MSC, VLR, HLR, or other network entity, such as an application platform, which has access to the specific USSD service.



Source: European Telecommunications Standards Institute (ETSI)

[TS 23.090]: “The Unstructured Supplementary Service Data (USSD) mechanism allows the Mobile Station (MS) user and a PLMN operator-defined application to communicate in a way which is transparent to the MS and to intermediate network entities.”

Figure 1. Handling of USSD

3GPP TS 22.090 details specific “MMI strings” or telephone numbers that could be dialed from a user’s “mobile station” (mobile handset) to initiate USSD services. Specifically, the following number range is defined as being allocated in GSM networks for USSD services residing in the user’s Home Public Land Mobile Network (HPLMN):

[3GPP TS 22.090]:

“1, 2 or 3 digits from the set (, #) followed by 1X(Y), where X=any number 0-4, Y=any number 0-9, then, optionally “* followed by any number of any characters”, and concluding with # SEND”.*

For example, a user could dial ***#123#** from the user’s mobile handset to reach a specific USSD service, where the USSD application itself resides in the user’s HPLMN.

In this way, the application developer could decide to build, install, and advertise use of a USSD service such as a menu-driven application (see the *Example Menu-Drive USSD Application* section) within a particular mobile network. The new USSD service would be available worldwide to subscribers to that mobile service via the USSD service number, which follows the rules detailed in 3GPP TS 22.090.

USSD Signaling Requirements

Example Menu-Driven USSD Application

One could decide to develop a mobile-initiated “Balance Enquiry and Top Up” application using USSD signaling, enabling a mobile user to interact with an application via the user’s handset, in order to view his/her current mobile account balance and top up as needed.

An example of such an application could be as follows:

1. A mobile user initiates the “Balance Enquiry and Top Up” service by dialing the USSD string defined by the service provider; for example, ***#123#**.

Note: See [3GPP TS 22.090] for more details regarding USSD string formats.

2. The USSD application receives the service request from the user and responds by sending the user a menu of options.
3. The user responds by selecting a “current balance” option.
4. The USSD application sends back details of the mobile user’s current account balance and also gives the option to top up the balance.
5. The user selects to top up his/her account.
6. The application responds by asking *how much credit to add?*
7. The mobile user responds with the amount to add.
8. The USSD application responds by sending an updated balance and ends the session.

Figure 2 shows an example of the MAP/TCAP message sequence required to realize the data transfers between a mobile user’s handset and the USSD application to implement the “Balance Enquiry and Top Up” service described above.

USSD Services for Interactive Mobile Users

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

Application Note

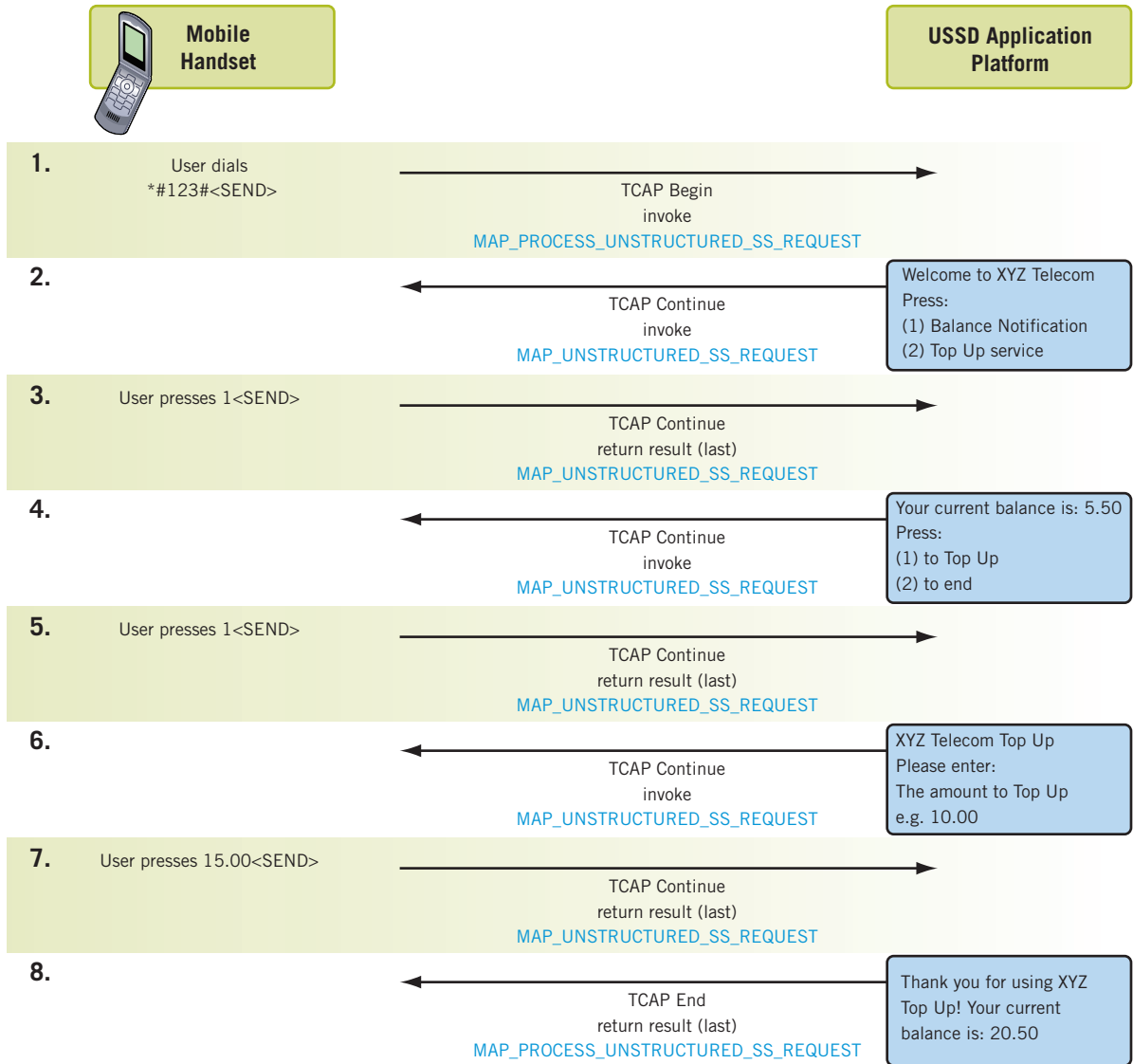


Figure 2. Mobile-Initiated "Balance Enquiry and Top Up" Service (adapted from Figure 6.2, [3GPP TS 24.090])

Notes on the USSD MAP/TCAP numbered message sequence shown in Figure 2:

1. The mobile-initiated USSD service commences with a mobile user dialing the USSD string (for example, *#123#). A TCAP dialogue is initiated following this, with a MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service component sent to the USSD application platform.
2. The USSD application platform receives the request to initiate a USSD service from the mobile user. The platform determines the specific USSD service requested by checking the USSD string dialed.

The USSD platform requests additional information from the mobile user (*balance notification or top up?*) via the MAP_UNSTRUCTURED_SS_REQUEST service.

3. The USSD platform receives the user's response within a MAP_UNSTRUCTURED_SS_REQUEST return result component.
- 4-7. The USSD application could request additional information from the mobile user several times for the same TCAP dialogue, each time the MAP_UNSTRUCTURED_SS_REQUEST service would be used, as shown in sequence 4 through 7.
8. Finally, the network USSD application platform ends the TCAP dialogue, sending MAP_PROCESS_UNSTRUCTURED_SS_REQUEST return result (*Thank you for using XYZ...*).

In the above example, the USSD platform requests further information from the mobile user via the MAP_UNSTRUCTURED_SS_REQUEST service. If the USSD platform did not require any further information from the mobile user, the USSD application could respond by sending a MAP_PROCESS_UNSTRUCTURED_SS_REQUEST return result component (as shown in Figure 2, point 8) and ending the TCAP dialogue.

For more information on mobile- (and network-) initiated USSD operations and the use of MAP USSD services, refer to [3GPP TS 24.090] in the *References* section.

How to Format USSD Text

In the “Balance and Top Up” service example shown in Figure 2, the network USSD application responds to inputs from the mobile user with various menus and questions. Figure 3 shows an example of a menu on a mobile handset.

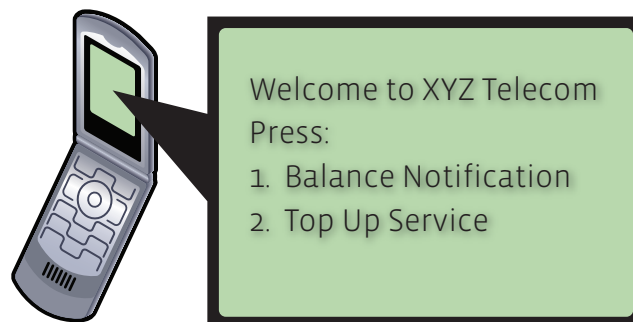


Figure 3. USSD Menu Example on a Mobile Handset

Formatting a USSD Menu

How Is Menu Information Sent to the User?

The menu, as it appears on the screen of the mobile handset, is actually a straightforward text string broken up using carriage return <CR> characters as shown in Figure 4.

The text string:

```
Welcome to XYZ Telecom<CR>Press:<CR>1. Balance Notification <CR>2. Top Up Service
```

Will be presented on the handset as ...

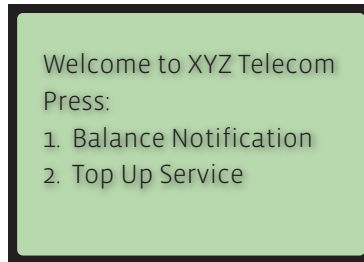


Figure 4. Formatting a USSD Menu for Display on a Handset

Creating USSD Applications

MAP Services for USSD

As shown in Figure 2, MAP services “MAP_UNSTRUCTURED_SS_REQUEST” and “MAP_PROCESS_UNSTRUCTURED_SS_REQUEST” are used to deliver text that forms the menus, questions, and answers between a network USSD application platform and a mobile user’s handset.

Note: MAP services PROCESS_UNSTRUCTURED_SS_REQUEST and UNSTRUCTURED_SS_REQUEST are defined in [3GPP TS 29.002].

These two MAP services both use parameters “USSD Data Coding Scheme” and “USSD String” as detailed in Figure 5.

UNSTRUCTURED_SS_REQUEST			PROCESS_UNSTRUCTURED_SS_REQUEST		
Parameter	Class	Context	Parameter	Class	Context
Primitive type octet	M	V1, V2	Primitive type octet	M	V1, V2
Timeout (default = 600 seconds)	O	V1, V2	Timeout (default = 600 seconds)	O	V1, V2
Invoke ID	M	V1, V2	Invoke ID	M	V1, V2
USSD Data Coding Scheme	M	V1, V2	USSD Data Coding Scheme	M	V1, V2
USSD String	M	V1, V2	USSD String	M	V1, V2
Alerting pattern	O	V2	Alerting pattern	O	V2
MSISDN	O	V2	MSISDN	O	V2
			Ellipsis	O	V2

Figure 5. USSD MAP Service Definitions from [MAP Programmer’s Manual]

USSD Services for Interactive Mobile Users

Application Note

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

MAP Parameters for USSD

USSD Data Coding Scheme

As defined in 3GPP TS 29.002, this parameter details “the alphabet and the language used for the unstructured information in an Unstructured Supplementary Service Data operation.”

Encoding is according to 3GPP TS 23.038 section “Cell Broadcast Data Coding Scheme”. A typical encoding scheme used is as follows:

“GSM 7 bit default alphabet”, encoded within the parameter as “00001111”.

The following code sample shows how the “GSM 7 bit default alphabet” USSD Data Coding scheme parameter could be encoded within a customer application:

```
/* USSD coding parameter */
bit_set(req.pi, MAPPN_USSD_coding);
/*
 * USSD coding set to 'GSM default alphabet' 00001111
 * see 3GPP TS 23.038 'Cell Broadcast Data Coding Scheme'
 * for further detail
 */
bit_to_byte(req.ussd_coding.data, 0x1, 0);
bit_to_byte(req.ussd_coding.data, 0x1, 1);
bit_to_byte(req.ussd_coding.data, 0x1, 2);
bit_to_byte(req.ussd_coding.data, 0x1, 3);
bit_to_byte(req.ussd_coding.data, 0x0, 4);
bit_to_byte(req.ussd_coding.data, 0x0, 5);
bit_to_byte(req.ussd_coding.data, 0x0, 6);
bit_to_byte(req.ussd_coding.data, 0x0, 7);
req.ussd_coding.num_bytes = 1;
```

For more information, see [3GPP TS 23.038] listed in the *References* section.

USSD String

USSD characters are packaged using a 7 bit encoding scheme, much like SMS characters, as defined in 3GPP TS 23.038.

The Dialogic® source code sample software, “MAP Test Utility”, contains a function example called “MTU_str_to_def_alph()”.

The function MTU_str_to_def_alph() converts an ascii string into “GSM 7 bit default alphabet” encoding, following rules defined in [3GPP TS 23.038] (also see Note: *Support for USSD Character Packing* below). This function can be useful, such as if creating a USSD string text to send over a GSM network based upon user input.

To test usage of the MTU_str_to_def_alph() function, the following code sample (based upon MTU source code) encodes a USSD string menu text similar to that which was previously discussed:

```
/* USSD string parameter */
bit_set(req.pi, MAPPN_USSD_string);

/*
 * USSD string
 */
mtu_args.message=
"Welcome to XYZ Telecom\n Press:\n 1. Balance Notification\n 2. Top Up Service";

num_da_chars = MTU_str_to_def_alph(mtu_args.message,
                                &req.usssd_string.data[req.usssd_string.num_bytes],
                                &da_len,
                                MAX_DATA_LEN - req.usssd_string.num_bytes);
```

Note the use of “\n” to encode a carriage return, as required for menu type text displays.

Note: Support for USSD Character Packing — 3GPP TS 23.038 (defined in the “USSD packing” section) details two USSD text cases that the supplied example function `MTU_str_to_def_alph()` does not support. In the two cases, the user could extend the `MTU_str_to_def_alph()` function source code as follows:

1. “If the total number of characters to be sent equals $(8n-1)$, where $n=1,2,3$, etc., then there are 7 spare bits at the end of the message. To avoid the situation where the receiving entity confuses 7 binary zero pad bits as the @ character, the carriage return or <CR> character ... shall be used for padding in this situation, just as for Cell Broadcast.”
2. “If <CR> is intended to be the last character and the message (including the wanted <CR>) ends on an octet boundary, then another <CR> must be added together with a padding bit 0.”

See [3GPP TS 23.038] for more information on these two cases, in order to develop a complete `str_to_def_alph()` function.

Information for System Design: Working with MAP USSD Signaling

Code Modifications to the Dialogic® “MTU” Example Application

The following section has an example of a function that can be used to send the service `MAP_PROCESS_UNSTRUCTURED_SS_REQUEST` to initiate a USSD session, as previously shown in Figure 2.

The code is based upon the structure of existing code for other MAP services supplied for the MAP Test Utility (MTU) example, as part of the Dialogic® User Part Development package [MTU and MTR User Guide].

This code sample indicates how to set the MAP service name for `MAP_PROCESS_UNSTRUCTURED_SS_REQUEST` and includes the USSD specific parameters “USSD Data Coding Scheme” and “USSD String” previously described.

USSD Services for Interactive Mobile Users

Application Note

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

Code Sample to Send

MAP_PROCESS_UNSTRUCTURED_SS_REQUEST

```
static int MTU_process_uss_req (dlg_id, invoke_id) /* USSD */
u16 dlg_id; /* dialogue ID */
u8 invoke_id; /* invoke ID */
{
    MTU_DLG *dlg; /* dialogue data structure */
    MTU_MSG req; /* structured form of request message */
    u8 da_len; /* length of formatted u-data */
    u8 num_da_chars; /* number of formatted*/

    dlg = &(dlg_data[dlg_id]);

    /*
     * The following parameters are set in the
     * MAP_PROCESS_UNSTRUCTURED_SS_REQUEST:
     * ussd-DataCodingScheme
     * ussd-string - this will be entered by the user e.g. *#123#
     */

    memset((void *)req.pi, 0, PI_BYTES);
    req.dlg_id = dlg_id;
    req.type = MAPST_PRO_UNSTR_SS_REQ_REQ;
    req.invoke_id = invoke_id;
    bit_set(req.pi, MAPPN_invoke_id);

    /* USSD coding parameter */
    bit_set(req.pi, MAPPN_USSD_coding);

    /*
     * USSD coding set to 'GSM default alphabet' 00001111
     * see 3GPP TS 23.038 'Cell Broadcast Data Coding Scheme'
     * for further detail
     */

    bit_to_byte(req.ussd_coding.data, 0x1, 0);
    bit_to_byte(req.ussd_coding.data, 0x1, 1);
    bit_to_byte(req.ussd_coding.data, 0x1, 2);
    bit_to_byte(req.ussd_coding.data, 0x1, 3);
    bit_to_byte(req.ussd_coding.data, 0x0, 4);
    bit_to_byte(req.ussd_coding.data, 0x0, 5);
    bit_to_byte(req.ussd_coding.data, 0x0, 6);
    bit_to_byte(req.ussd_coding.data, 0x0, 7);

    req.ussd_coding.num_bytes = 1;

    /* USSD string parameter */
    bit_set(req.pi, MAPPN_USSD_string);
}
```

```
/*
 * USSD string
 */

req.ussd_string.num_bytes = 1; /* USSD string, allow byte for data length */
num_da_chars = MTU_USSD_str_to_def_alph(mtu_args.ussd_string,
                                         &req.ussd_string.data[req.ussd_string.num_bytes],
                                         &da_len,
                                         MAX_DATA_LEN - req.ussd_string.num_bytes);

/*
 * fill in the ussd_string, the number of formatted default alphabet
 * characters
 */

req.ussd_string.data[req.ussd_string.num_bytes - 1] = num_da_chars;
req.ussd_string.num_bytes += da_len;

/*
 * Operation timeout - 15 seconds
 */
bit_set(req.pi, MAPPN_timeout);
req.timeout = 15;

MTU_send_srv_req(&req);

return(0);
} /* end of MTU_process_uss_req() */
```

Encoding Application Context for USSD Version 2 Operations

The following code sample can be used to add support within MTU for the USSD version 2 application context, as defined in [3GPP TS 29.002].

Add the following definition to mtu.c:

```
/* new application context networkUnstructuredSsContext-v2 */
static u8 networkUnstructuredSsContextV2[AC_LEN] =
{
    06,          /* object identifier */
    07,          /* length */
    04,          /* CCITT */
    00,          /* ETSI */
    00,          /* Mobile domain */
    01,          /* GSM network */
    00,          /* application contexts */
    19,          /* map-ac networkUnstructuredSs */
    02           /* version 2 */
};
```

USSD Services for Interactive Mobile Users

Application Note

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

Add the following code to mtu.c function “MTU_open_dlg” to make use of the USSD version 2 application context:

```
case MTU _ PROCESS _ USS _ REQ:  
/*  
 * Send MAP _ PROCESS _ UNSTRUCTURED _ SS _ REQUEST  
 */  
for (i=0; i<AC _ LEN; i++)  
    req.applic _ context[i] = networkUnstructuredSsContextV2[i];  
break;
```

Dialogic® DSI Components for USSD Services

Dialogic® DSI SS7 Boards, Dialogic® DSI Protocol Stacks, and Dialogic® DSI Signaling Servers can be used to create the network elements required to develop a USSD service such as the mobile “Balance Enquiry and Top Up” application described in this application note.

As an example, Figure 6 shows a Dialogic® DSI-based USSD platform developed using high-density PCIe signaling cards. The USSD platform uses a pair of servers and signaling cards that act as a single SS7 end point, providing availability and a high degree of fault tolerance.

Depending on the amount of USSD signaling traffic anticipated, multiple TDM Low Speed Links (LSL) (64 kbps) could be deployed. For systems with larger throughput requirements, TDM High Speed Links (HSL) could be deployed, with each signaling link giving approximately 2 Mbps of signaling throughput.

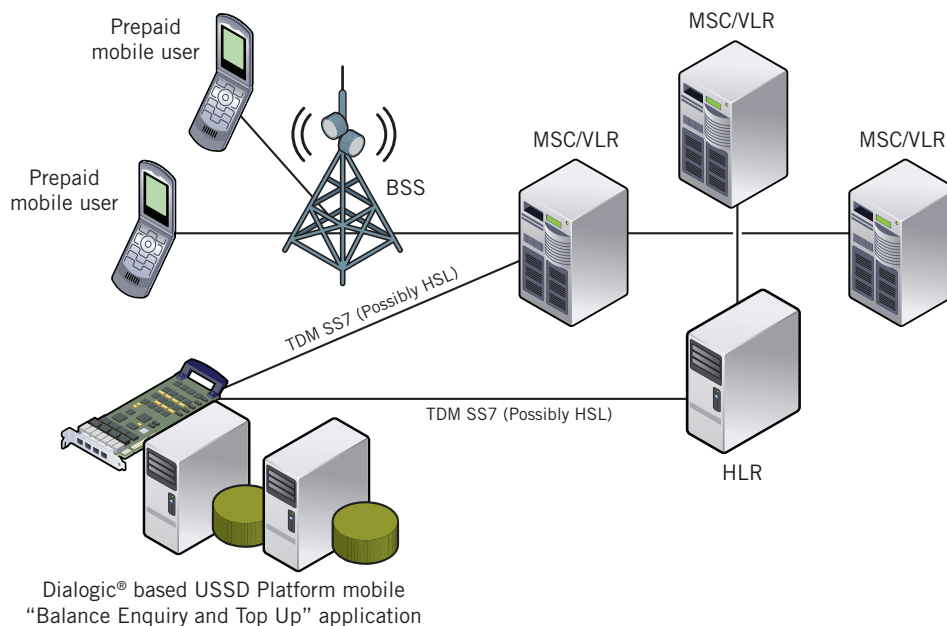


Figure 6. Implementing a USSD Application Platform with Dialogic® DSI Components — TDM SS7 Connections Example

USSD Services for Interactive Mobile Users

Application Note

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

Alternatively, the Dialogic® DSI SIGTRAN Stack could be used to connect the USSD application platform to the mobile network. Figure 7 shows a similar example deployed using Dialogic® DSI SUA Layer, where the example includes two application server processes for resilience. For SIGTRAN deployments, throughput is not limited by signaling link bandwidth as in the TDM world; instead, high-speed Ethernet provides bandwidth for high-performance MAP USSD applications.

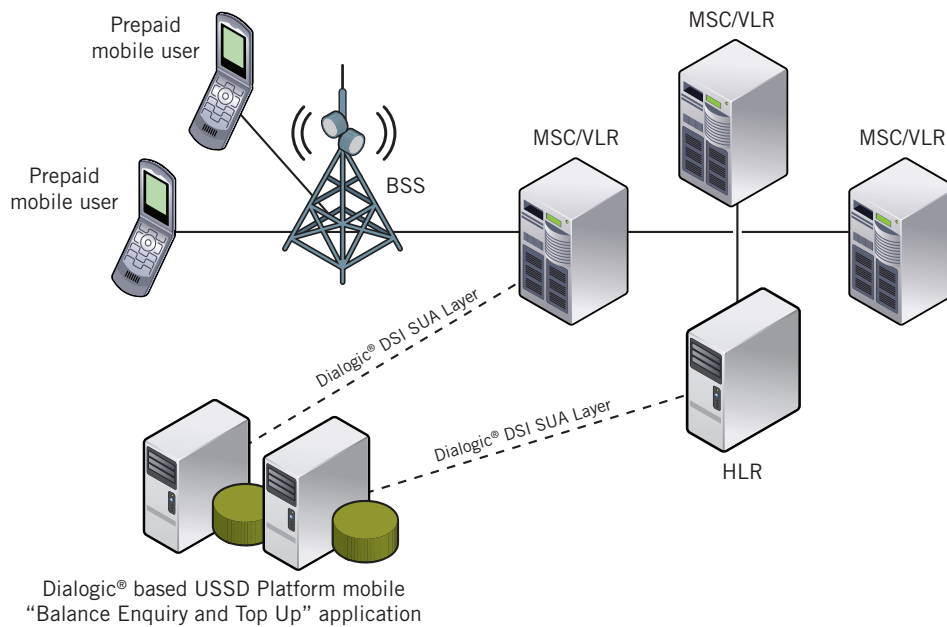


Figure 7. Implementing a USSD Application Platform with Dialogic® DSI Components — DSI SIGTRAN Stack Connections Example

As examples, Dialogic® DSI Protocol Stacks could be used to develop systems over:

- SS7 TDM using Dialogic® DSI SS7 Stack - DSI MAP Layer, DSI TCAP Layer, and DSI SCCP Layer
- SIGTRAN using Dialogic® DSI SIGTRAN Stack - DSI SUA Layer, DSI M3UA Layer, or DSI M2PA Layer

For GSM MAP signaling, a number of Dialogic® DSI Components are available including:

- Dialogic® DSI SS7AM1 Network Interface Boards (Advanced Mezzanine Card for AdvancedTCA)
- Dialogic® DSI Signaling Servers
- Dialogic® DSI SS7HDE Network Interface Boards (PCIe SS7 board)
- Dialogic® DSI SS7HDC Network Interface Boards (cPCI SS7 boards)
- Dialogic® DSI SS7HDP Network Interface Boards, Dialogic® SPCI4 Network Interface Boards (PCI SS7 Boards)

A choice of products is available based upon factors such as the level of integration needed, expected transaction rate, and the level of fault tolerance required. The same Application Programming Interface (API) is used across the Dialogic® DSI Component family, allowing easy migration from board-based to server-based solutions, or from TDM to IP (SIGTRAN) as service needs change. For more information on the range of Dialogic® DSI Components available, see the *For More Information* section.

USSD Services for Interactive Mobile Users

Application Note

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

Summary

This application note described interactive USSD services for GSM mobile users and showed how such a service could be provided using GSM MAP signaling.

An example mobile-initiated “Balance Enquiry and Top Up” application was described, detailing the MAP services and parameters required by a network-based USSD application, and showed how USSD text strings are handled by the application for sending to mobile users’ handsets.

Finally, system designs were considered, providing USSD service designers with the option to develop and deploy USSD applications with Dialogic® DSI Boards, Dialogic® DSI Servers, and Dialogic® DSI Protocol Stacks.

References

[3GPP TS 22.090] Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS); Unstructured Supplementary Service Data (USSD); Stage 1

[3GPP TS 23.038] Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS); Alphabets and language-specific information

[3GPP TS 23.090] Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS); Unstructured Supplementary Service Data (USSD); Stage 2

[3GPP TS 24.090] Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS); Unstructured Supplementary Service Data (USSD); Stage 3

[3GPP TS 29.002] Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS); Mobile Application Part (MAP) specification

[MAP Programmer’s Manual] *Dialogic® SS7 Protocols MAP Programmer’s Manual*
<http://www.dialogic.com/support/helpweb/signaling/software3.htm>

[MTU and MTR User Guide] *Dialogic® SS7 Protocols MAP Test Utility (MTU) and Responder (MTR) User Guide*
<http://www.dialogic.com/support/helpweb/signaling/software3.htm>

USSD Services for Interactive Mobile Users

Building User-Friendly Mobile Telephony Applications Using Dialogic® Distributed Signaling Interface Components

Application Note

Acronyms

3GPP	3rd Generation Partnership Project
API	Application Programming Interface
HLR	Home Location Register
HPLMN	Home Public Land Mobile Network
MAP	Mobile Application Part
MSC	Mobile Switching Center
MSISDN	Mobile Subscriber ISDN Number
MTU	MAP Test Utility
SCCP	Signaling Connection Control Part
SMS	Short Message Service
TCAP	Transaction Capabilities Application Part
USSD	Unstructured Supplementary Service Data
VMSC	Visited MSC

For More Information

3GPP / ETSI publications — <http://webapp.etsi.org/key/queryform.asp>

Dialogic® DSI Components Information — <http://www.dialogic.com>

Dialogic® DSI Software and Documentation — <http://www.dialogic.com/support/helpweb/signaling/>

www.dialogic.com

Dialogic Corporation

9800 Cavendish Blvd., 5th floor
Montreal, Quebec
CANADA H4M 2V9

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH PRODUCTS OF DIALOGIC CORPORATION OR ITS SUBSIDIARIES ("DIALOGIC"). NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Dialogic may make changes to specifications, product descriptions, and plans at any time, without notice.

Dialogic is a registered trademark of Dialogic Corporation. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners. Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement their concepts or applications, which licenses may vary from country to country.

Any use case(s) shown and/or described herein represent one or more examples of the various ways, scenarios or environments in which Dialogic products can be used. Such use case(s) are non-limiting and do not represent recommendations of Dialogic as to whether or how to use Dialogic products.

© 2007, 3GPP™ TSs and TRs are the property of ARIB, ATIS, ETSI, CCSA, TTA and TTC who jointly own the copyright in them. They are subject to further modifications and are therefore provided to you "as is" for information purposes only. Further use is strictly prohibited.