

Technology Reports

iC Communications Application for FeliCa Ad-hoc Communications Function

SNS and social game services are faced with the challenge of increasing the number of users and invigorating communications among them, and must meet demands for a wider range of communication methods. Focusing on one of these communications methods, — iC communications functions — we have developed an iC communications appli that enables easy utilization of simple iC communications functions for data exchange with an i-mode browser or i-appli. This application is an easy-to-use iC communications function available to users as a new communications method.

Service & Solution Development Department

Naofumi Hara**Takeshi Kato**

Frontier Services Department

Takakiyo Nishikawa**Yoshiki Ito**

1. Introduction

In the mobile applications field in recent years, many user communications services such as SNS and social game services have been released. These community-based services face the challenge of increasing the number of users and invigorating communications among users, and must meet demands for a diverse range of user communication methods.

Here, iC communications appli that enables easy utilization of iC communications functions for data exchange with an i-mode browser or i-appli is developed by taking into account an iC communications function based on

FeliCa^{®*1} ad hoc communications^{*2}. At present, the iC communications function has been used to exchange data stored in handsets such as phone book, bookmark and image data. This iC communications appli makes it easy to mount iC communications functions onto an application, and thus offers a new communications method that enables users to swap items in social games, register friends or introduce friends to new content and so forth.

This article describes the objectives of the iC communications appli, its functions and operations, and presents examples of usage.

2. The Aim of iC Communications Appli

2.1 iC Communications Appli Overview

The iC communications appli packages the basic functions needed for iC communications, and provides a high level Application Program Interface (API)^{*3} to enable a range of applications to use iC communications. API application linkage functions are used to call the iC communications appli from an application, and return results to that application. We have incorporated browser-link functions and i-appli link functions into the API for the iC communications appli.

©2012 NTT DOCOMO, INC.

Copies of articles may be reproduced only for personal, noncommercial use, provided that the name NTT DOCOMO Technical Journal, the name(s) of the author(s), the title and date of the article appear in the copies.

*1 **FeliCa**[®]: A non-contact IC card technology developed by Sony Corp. A registered trademark of the company.

2.2 iC Communications Appli Features

The three main features of this software are as follows:

1) Available from Applications on the Browser

iC communications functions have been available from i-appli or native functions (telephone book, etc.)^{*4}, although they were not available from the i-mode browser. We have newly equipped an interface in the iC communications appli that enables reception of related browser data. This interface enables sending data from browser applications to iC communications appli using browser start-up.

This makes it possible to send and receive data with iC communications using browser applications in the same way as i-appli and native functions.

2) High Level API FeliCa Functions

Up to now, FeliCa communications needed to be established to include retry, verifying the other party, and communications complete functions in all applications for i-appli iC communications. Because this new software takes over those procedures and performs those processes, it makes it much easier to mount iC communications functions onto applications.

3) Guidance to Download Sites

To use iC communications between i-applis, it was necessary to install the i-appli software in both the receiving and transmitting handsets. In addition to sending data, the iC communications

appli also enables setting of information about the download site to get the i-appli to run, which means if the corresponding i-appli isn't installed, the iC communications appli will guide the user to the i-appli download site.

This means that users are prompted to download the i-appli merely by holding their phones close to each other, which can be a way of popularizing applications.

3. iC Communications Appli Operations Overview

3.1 Data Exchange via iC Communications between Browsers

The following describes the procedure for using the iC communications appli from an application running on a browser. **Table 1** shows the parameters that start up the iC communications appli from the browser, while **Figure 1** illustrates the screen transitions.

- Web site URL (fig. 1 (1)), transmission parameters to send to the receiving terminal (fig. 1 (2)), and parameters to return after the iC communications finish (fig. 1 (3))

are specified from an application running on browser, and the iC communications appli starts up.

- Once the iC communications appli starts, the Web site URL (fig. 1 (1)) and the transmission parameters (fig. 1 (2)) are sent to the receiving terminal via iC communications.
- After iC communications finish, the iC communications appli at the sending terminal starts the browser using the Web site URL (fig. 1 (1)) and the return parameters (fig. 1 (3)).
- The iC communications appli at the receiving terminal starts the browser using the Web site URL (fig. 1 (1)) and the transmission parameters (fig. 1 (2)) received.

The above procedure describes how data is exchanged between applications running on browsers using iC communications.

3.2 Data Exchange via iC Communications between i-applis

The following describes the procedure for using the iC communications appli between i-applis. **Table 2** shows

Table 1 Parameters to start the iC communications appli from the browser

	Name	Description
(1)	Web site URL	Web site displayed on both terminals after iC communications finish
(2)	Transmission parameters	Parameters added to the footer for the receiver (Receiver) to display a Web site. More than one can be described using & as a delineator
(3)	Return parameters	Parameters added to the footer for the receiver (Sender) to display a Web site. More than one can be described using & as a delineator

*2 **Ad hoc communications:** Direct data communications between mobile terminals that are not routed through a network. FeliCa, Bluetooth® and wireless LAN are some of the different technologies that can be used for this purpose.

*3 **API:** An interface that enables other software to use the functions provided by OS and middleware.

*4 **Native functions:** A general term used to describe functions included in mobile terminals other than i-appli functions.

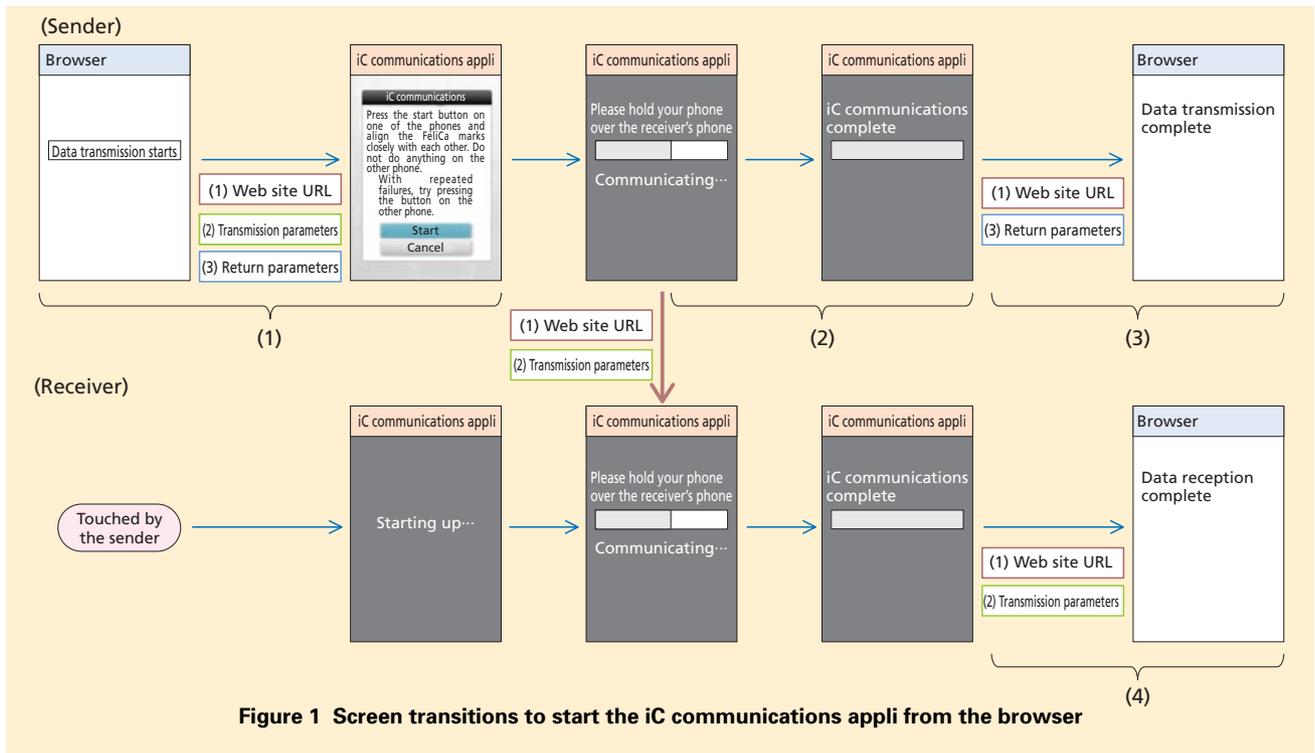


Figure 1 Screen transitions to start the iC communications appli from the browser

the parameters for starting up the iC communications appli from an i-appli, while **Figure 2** shows the screen transitions.

- The i-appli Application Description File (ADF)^{*5} (fig. 2(1)), the transmission parameters to be sent from the transmitting terminal (fig. 2 (2)), the return parameters after iC communications finish (fig. 2 (3)), the download URL for the i-appli (fig. 2 (4)) are specified from the i-appli, and the iC communications appli starts.
- Once the iC communications appli starts, the i-appli ADF (fig. 2(1)), the transmission parameters (fig. 2 (2)), and the download URL (fig.2 (4)) are sent to the receiving terminal

Table 2 Parameters to start the iC communications appli from an i-appli

	Name	Description
(1)	i-appli ADF	Specifies the i-appli to start when iC communications finish
(2)	Transmission Parameters	List of parameters to send to the other party. Multiple settings possible
(3)	Return parameters	The parameters listed to be sent after iC communications finish to start the i-appli specified in (1). Multiple settings possible
(4)	Download URL	URL for downloading the i-appli specified in (1) if the other party doesn't have it

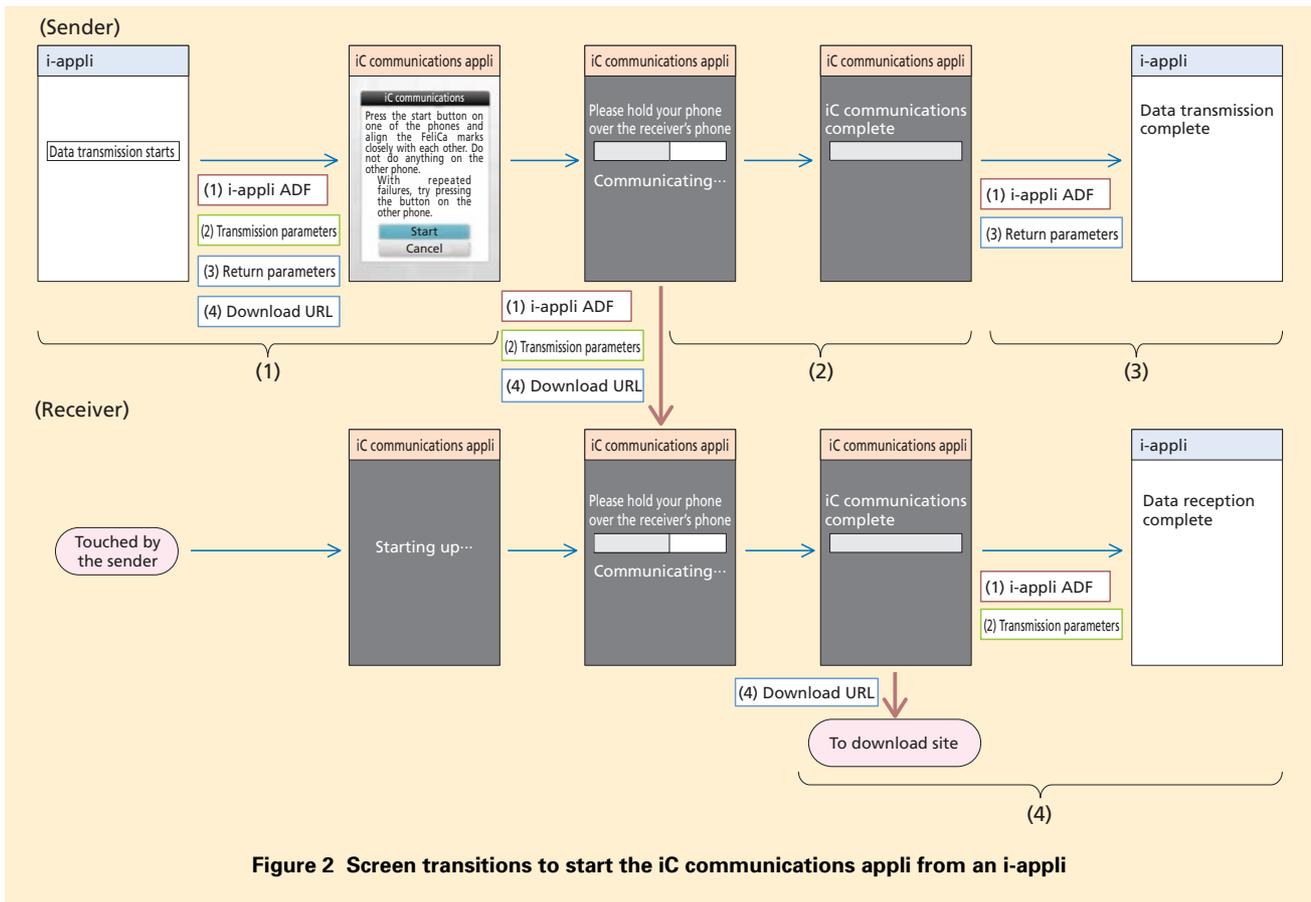
via iC communications.

- After iC communications finish, the iC communications appli starts the i-appli using the i-appli ADF (fig. 2 (1)) and the return parameters (fig. 2 (3)).
- Using the received ADF (fig. 2(1)) for the i-appli, the iC communications appli at the receiving terminal checks whether the corresponding i-appli is installed. If the appli is

installed, the i-appli is started using the received i-appli ADF (fig. 2(1)) and transmission parameters (fig. 2 (2)), if it is not installed the browser is started using the received download URL (fig. 2 (4)) to prompt the user to download appli.

The above procedure describes how data is exchanged between i-applis using iC communications.

*5 **ADF**: A type of information file that contains definitions and information for Star and DoJa applications.

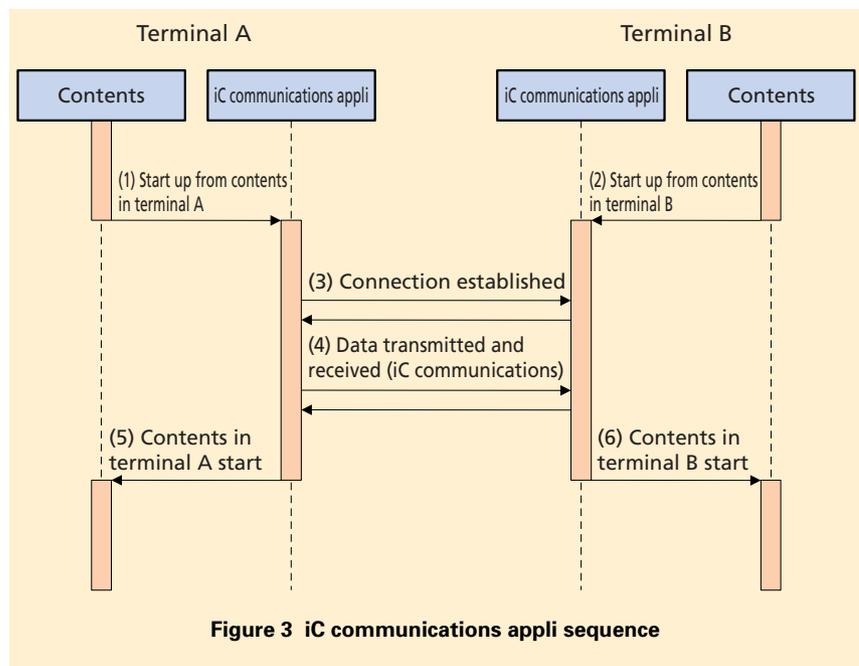


3.3 iC Communications Appli Operational Sequence

The iC communications appli sequence is illustrated in **Figure 3**.

1) Starting the iC Communications Appli

Contents sends multiple parameters including the i-appli ADF and transmission parameters, and starts the iC communications appli (fig. 3 (1) (2)). In fig. 3, it is assumed that communications are bidirectional (data is transmitted from both terminals A and B), however, for unidirectional communications (only terminal A sends data, terminal B only receives), the iC communications



appli in terminal B does not start (fig. 3 (2)) from contents, but starts when terminal B is held close to terminal A (fig. 3 (3)).

2) Performing iC Communications

The iC communications connection is established by holding the FeliCa marks over each other on the two terminals (fig. 3 (3)). Then, data is transmitted and received by iC communications (fig. 3(4)). In the unidirectional communications case above, the terminal that does not have any data to send sends empty data so that the communications can be treated as bidirectional, which simplifies the communications procedure.

3) Contents Start Up

When iC communications have finished, the original contents, with the multiple parameters included as “iC communications success or failure” and “the data received from the other party,” are automatically launched in both terminals by the iC communications applis (fig. 3 (5) (6)).

4. Mounting Compatible Content

4.1 Mounting in Browser

Figure 4 shows an example of the iC communications appli source code to use the iC communications appli from i-mode browser contents. Using the browser button as a trigger, the iC communications appli must be started with parameters attached. Also if the

browser fails to start the i-appli (=the corresponding appli does not exist), users can be prompted to download the iC communications appli by specifying the above URL with the A tag^{*6} and href properties^{*7} at the bottom of fig. 4 as the iC communications appli download page, because it is possible to go to any desired URL.

4.2 Mounting in i-appli

To use iC communications appli from an i-appli, mounting is possible by importing the open-source library^{*8} shown in **Figure 5**, and adding the iC communications appli ADF and parameter name and value pairs as a post-script to the i-appli source code.

Furthermore, detailed materials and libraries for mounting on a browser or

```
<OBJECT declare id= "application.a"
type= "application/x-jam"
data= "URL for iC communications appli ADF (jam file)" >
<PARAM name= "FT_VERSION"
value= "1.0.0" >
<PARAM name= "FT_URL"
value= "http://aaa.co.jp/" >
<PARAM name= "FT_URL_PARAM"
value= "userid=123456" >
<PARAM name= "FT_RELOAD_PARAM"
value= "send=userid" >
</OBJECT>
<A ista= "#application.a"
href= "URL to display if iC communications appli is not installed" > start up. </A>
```

Figure 4 Example of source code for mounting in a browser

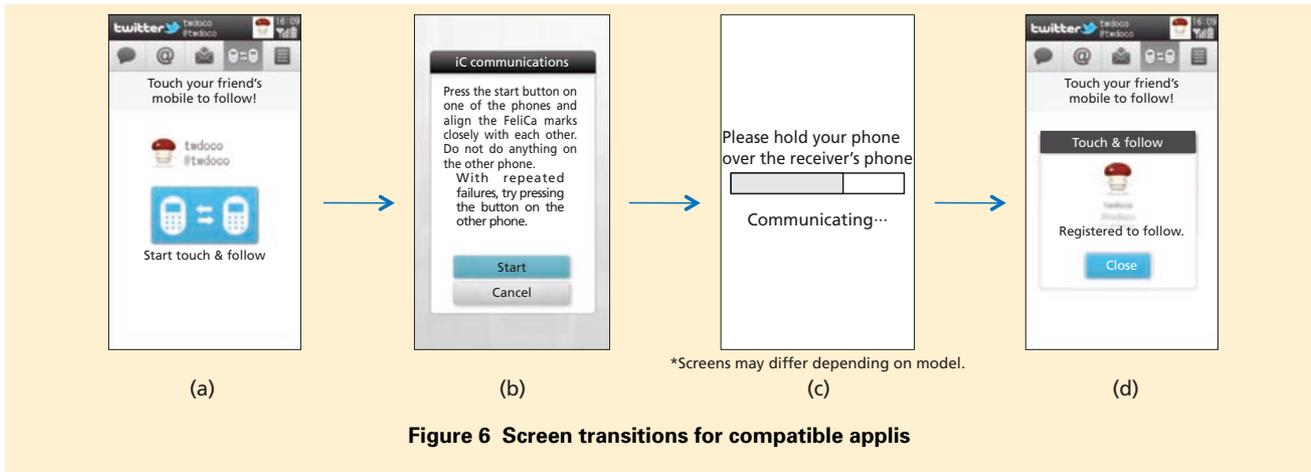
```
//Import classes provided by iC communications appli library.
import jp.co.nttdocomo.ss.i.adhocappli.LaunchUtil;
:
//Some parameters are specified using Hashtable
// which stores String instance parameter name and value pairs.
Hashtable parameters = new Hashtable();
parameters.put("sendItemID", "1000");
parameters.put("number", "3");
Hashtable rebootParameters = new Hashtable();
rebootParameters.put("send", "item");
try{
LaunchUtil.launchApplication(
"URL for ADF (jam file)",
parameters,
"URL for appli download site",
rebootParameters );
}catch(IllegalArgumentException iaex){
}
```

Figure 5 Mounting example using library in i-appli

*6 **A tag**: A type of tag used in HTML documents, also called a link tag. These enable transitioning to the link described.

*7 **Href properties**: Indicates the location of the details about the link to which the A tag transitions.

*8 **Library**: a collection of high-versatility programs that can be reused/recycled.



on an i-appli can be downloaded from NTT DOCOMO Web page[1].

5. Usage Example

As an example of i-appli using the iC communications appli, we present Twitter^{*9} for i-appli. Twitter for i-appli uses the iC communications appli to enable users to exchange Twitter user IDs so that they can follow each other's accounts. **Figure 6** shows the screen transitions. By pressing the Twitter for i-appli button (fig. 6 (a)), the iC communications appli (fig. 6 (b)) starts, and when the user presses the button in the iC communications appli, iC communications start (fig. 6 (c)), and then when communications finish, Twitter for i-appli automatically starts (fig. 6 (d)).

In order to follow a friend's account with the conventional Twitter client,

you must ask the friend's account name and then conduct a user search, however, the iC communications appli makes it possible to follow another person's account without any of those complicated operations.

6. Conclusion

This article has described an iC communications appli for data exchange functions using iC communications with applications running on browsers or i-applis.

The iC communications appli makes it possible to use iC communications from a browser, and because it offers packaged mounting functions for iC communications, it enables easy incorporation into a wide range of services. Furthermore, if the receiving terminal does not have the appropriate

i-appli installed, the user is guided to a download site, which means this system can be used to further popularize iC communications services.

We plan to continue into the future by expanding services for compatibility with data exchange between applications running on Android^{TM*10} terminals, and continue to advance developments related to promoting the use of Osaifu-keitai in smartphones as well as i-mode terminals.

REFERENCE

- [1] NTT DOCOMO : "Information for i-mode Content Creators Let's Make i-mode Content."
<http://www.nttdocomo.co.jp/english/service/developer/make/content/felica/data/index.html>

*9 **Twitter**: A registered trademark of Twitter, Inc. in the United States and other countries.

*10 **Android**TM: An open source platform developed by Google, Inc. mainly targeted at mobile terminals. AndroidTM is a trademark or registered trademark of Google, Inc in the United States.