Smartphone Pre-installing Application Linking

Technology Reports

Smart Downloader for Smoother Android Application Installation

*Android™[*1] terminals do not come with all full applications installed, but have a pre-installed downloader for downloading each of these full applications. However, the standard downloader is inconvenient because it does not enable application linking in the same manner as other full Android applications. We have developed a solution that promotes smoother application download and use by incorporating a new downloader mechanism that enables application linking.*

Service & Solution Development Department **Mirai Hara**
Takeshi Kato

1. Introduction

Including carrier services, there is a demand for many applications to be pre-installed in Android terminals, although as a way of lowering testing costs and ensuring that the terminal has enough free memory space, downloaders that download the full applications are pre-installed, rather than the full applications themselves.

However, because the pre-installed downloaders are not able to accept intent[*2] (application launch requests) from other applications, the standard downloader cannot use the application linking functions available with the Android OS, which results in lost opportunities to use applications.

To resolve this issue, we have developed a smart downloader that offers downloader functions and operability that act as if the full application were pre-installed.

The smart downloader is able to receive intent from other applications on behalf of the full application when the application is not installed. The received intent is also temporarily saved, so that once the full application is detected as installed, the intent is sent to the application, which is effectively the same as application linking.

2. Forms of Pre-installed Applications on Android Terminals

There are four main forms that pre-installed applications can take on Android terminals – (a) the full application itself (b) shortcuts (c) the standard downloader (d) the smart downloader we have developed. **Table 1** shows a comparison of these pre-installed forms.

Because (a) is the only case from the above four types where the full application itself is actually pre-installed, this is the shortest path to commence application use, as there is no need to download and install it. However, pre-installing the application reduces available memory in the terminal, and the inability of preinstalled applications to receive update notices from the Android Market™[*3] is also a concern. Furthermore, because there are costs associated with testing all of the functions offered by the application in advance of terminal sale, the applica-

[*1] **Android™**: A trademark or registered trademark of Google, Inc., in the United States.

[*2] **Intent**: The messages and mechanism for exchange between functions within an application, or between different applications. Intent is also used to launch applications.

Table 1 Comparison of pre-installation forms

	(a) The application	(b) Shortcut	(c) Standard downloader	(d) Smart downloader
Terminal memory reduction	×	○	○	○
Update notifications from Android Market	△ Initial manual update required	○	○	○
Application linking (intent response)	○	×	×	○
Seamless path to application use	◎ No download required	×	×	○ Automatic launch from the passed intent
Other constraints	—	・Shortcut remains after the application is installed ・Must be positioned on the home screen	—	—

◎ very good ○ good △ not bad × bad

tion must be completed within a certain amount of lead time. before the terminal can be released onto the market, which present cost and time issues to terminal manufacturers and application developers and vendors. As a solution to these full application pre-installation issues, usually (b) shortcuts are pre-installed, and (c) the standard downloaders are pre-installed. Both (b) and (c) require smaller amounts of memory than the application itself, and because the application is downloaded from the Android Market, update notifications can be received. The application can also be completed right before the terminal goes on the market, which allows for more flexibility in development time.

Pre-installed shortcuts (b) provide links to download applications from the Android Market or Web sites, and can be positioned in the home application. But as issues that needed to be resolved, the shortcuts remain after the application is installed and must be deleted manually, or shortcuts must be positioned on the home screen.

The standard application downloader (c) also has a function that transitions to the Android Market or Web sites in the same way as shortcuts, but it is a general application. This downloader also has the same identifiers as the application (certification and package name), and because it has a lower version number than the corresponding full application, the Android system recognizes it as an older version of the application and overwrites it upon installing the downloaded full application. Also, because this is a general application, there is no need to position it on the home screen like a short cut.

Although pre-installation forms (b) and (c) solve the issue of reduced available memory that accompanies pre-installing applications, new issues arise because these are not applications themselves and cannot receive intent from other applications.

Intent is information used to launch an application. For example, as shown in **Figure 1**, when the user presses a share button while viewing an image using an image gallery application, applications that handle images such as Twitter[*4] or Facebook[*5] applications are launched. If intent cannot be received, then it is not possible to launch an application from another application. Compounding the issue of inability to receive intent, users may be hindered from accessing and using the application because they get bound up in operations to download it.

[*3] **Android Market™**: A trademark or registered trademark of Google, Inc., in the United States. To receive application update notifications from the Android Market, applications must be downloaded from the Android Market.

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

[*5] **Facebook**: A trademark or registered trademark of Facebook, Inc.

We have developed the new smart downloader (d) to solve these issues.

3. Smart Downloader Development

3.1 Smart Downloader Overview

The smart downloader transitions to the Android Market or Web sites in the same way as the standard downloader, however, (1) it features functions that respond to intent from other applications, and (2) delivers intent to the application once it has been installed to automatically launch it.

1) Intent Response

All Android applications have an application definition file called a manifest file that contains the package name as the application identifier[*6], as well as intent filter data that defines to what type of intent the application can respond. Smart downloaders basically have the same identifier and intent filter as full applications in their manifest files, which enable them to respond to intent from other applications on behalf of the application. Furthermore, because the intent that is received is to be used to launch the full application, it is temporarily stored.

2) Automatic Launch from the Passed Intent

If an application is installed on the Android OS, the OS issues an installation complete event. The smart downloader detects this once the application is installed, and sends the intent to the application to launch it. In this instance, the intent data saved in 1) is read and used, so that this data and any optional image data etc. that was passed to the smart downloader can be passed onto the application, as it is.

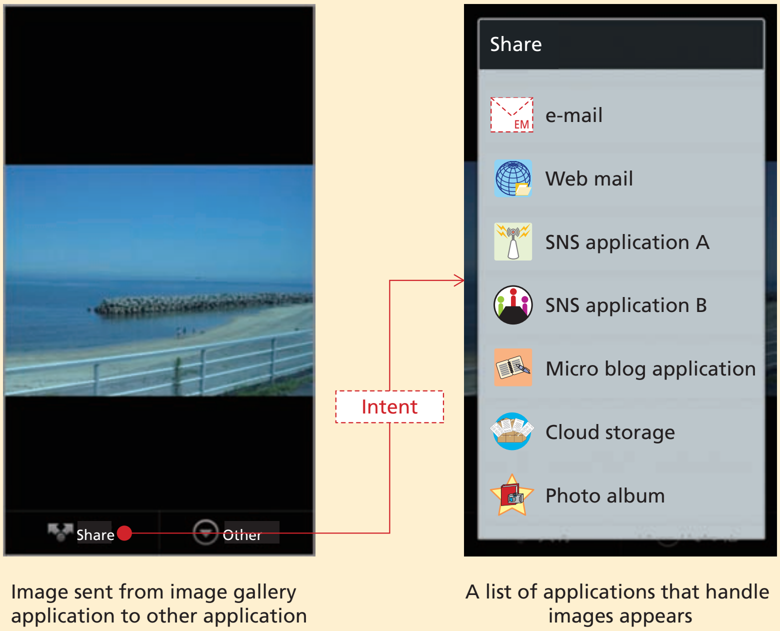


Image sent from image gallery application to other application

A list of applications that handle images appears

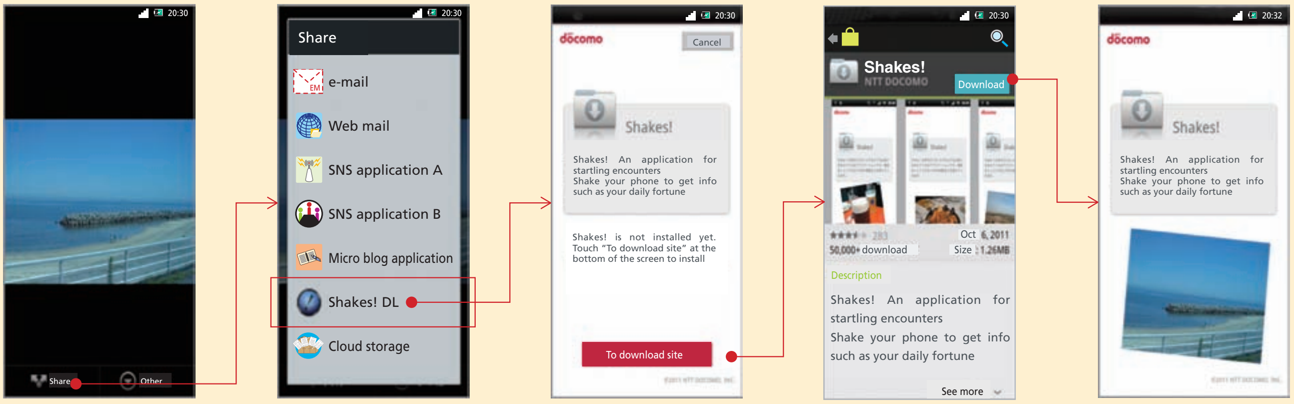
Figure 1 Example of intent

Figure 2 shows the smart downloader functions as the screen transitions that the user sees. For example, when the user presses the share button in an image gallery application, the smart downloader for the sample application Shakes! also appears in the list of sharing destinations in addition to the applications displayed in fig. 1 (fig. 2 (1) (2)). By selecting the Shakes! as the sharing destination, a screen prompting the user to download the Shakes! application by transitioning to the Android Market etc. appears (fig. 2 (3) (4)). After the application has been installed, it automatically launches with the image from the image gallery application displayed (fig. 2 (5)).

3.2 Application Configuration

As described above, the smart downloaders have the same identifiers as applications, just as the standard downloader does, and these are overwritten and deleted once an application has been installed. This means that the downloaders themselves cannot perform operations to detect that application installation is complete, or pass saved intent data to the application after it has been installed. To overcome this issue, the smart downloader system consists of the linking downloader that

[*6] **Application identifier**: As well as package names, application certifications must match so that applications can be recognized as such.



Shakes! An application for startling encounters Shake your phone to get info such as your daily fortune

Shakes! is not installed yet. Touch “To download site” at the bottom of the screen to install

To download site

(1)Image sent to other application from image gallery application

(2)The uninstalled sample application Shakes! appears in the list of applications that handle images

(3)A screen prompting the application download appears when the user selects Shakes!

(4)The application is downloaded from the Android Market etc

(5)After installation is complete, the application automatically starts. The image passed to the smart downloader is used with the application

Figure 2 Smart downloader screen transitions

corresponds to each full application, and a bridge launcher application used by these linking downloaders. The bridge launcher stores the intent and detects when applications have been installed, and then passes the intent to the application. **Figure 3** shows a configuration example of a linking downloader and the bridge launcher.

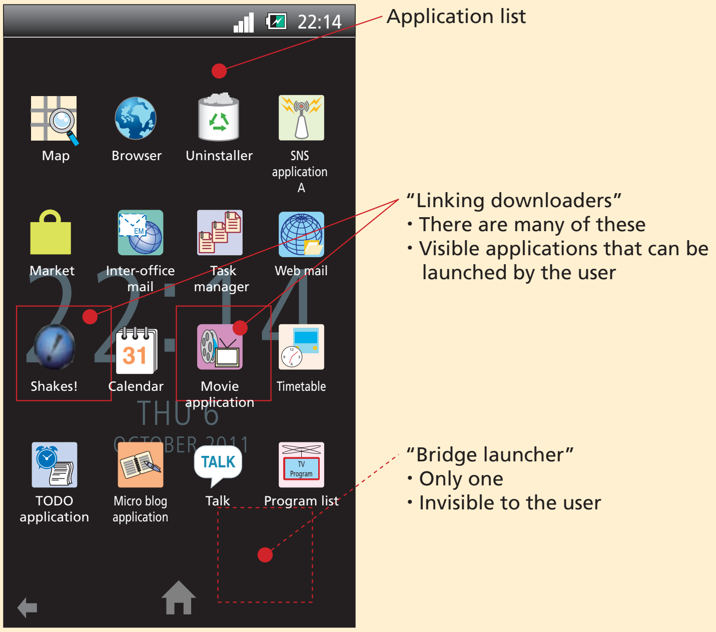


Figure 3 Example of linking downloader and bridge launcher configuration

3.3 Operating Sequence

Figure 4 shows the sequence of operations in the smart downloader.

To link with other applications, applications such as image gallery applications issue intent (launch requests) and optional data such as an image (fig. 4 (1)). The linking downloader responds to the intent on behalf of the full application yet to be installed.

The linking downloader that responds to the intent requests that the bridge launcher store the intent (fig. 4

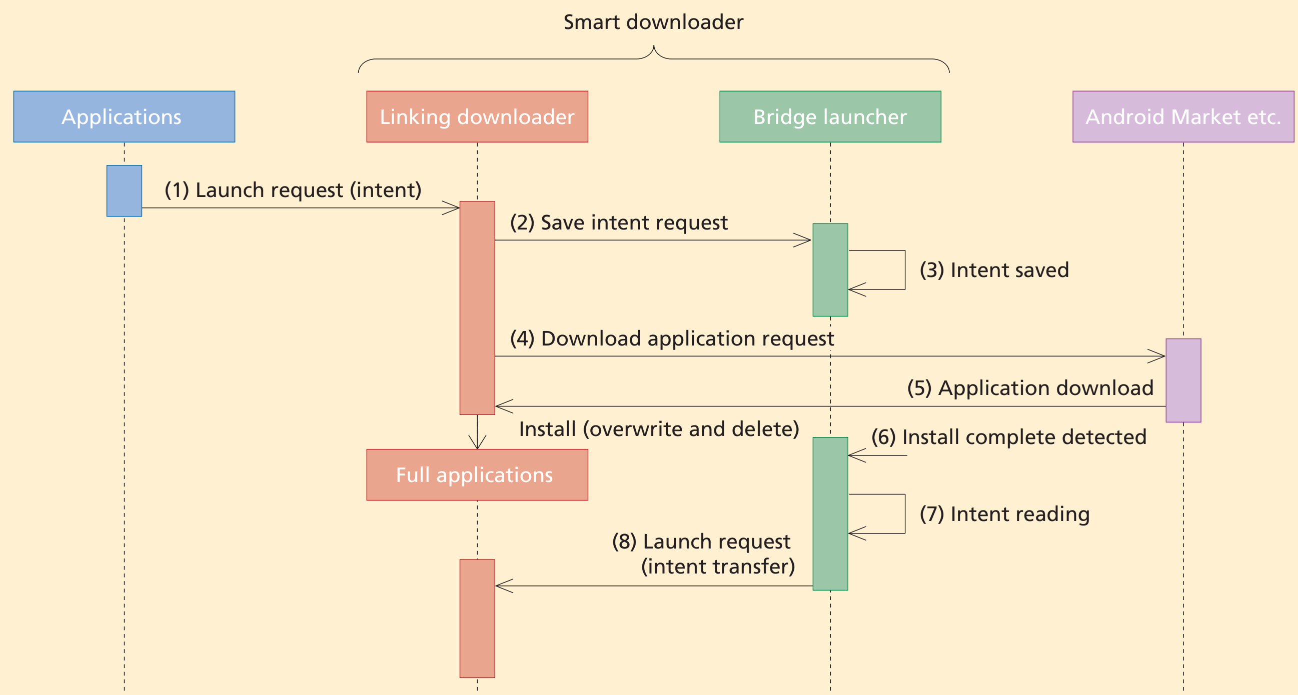


Figure 4 Smart downloader operations sequence

(2)), which it does (fig. 4 (3)).

After the bridge launcher stores the intent, the linking downloader prompts the user to download and install the application from the Android Market etc. (fig. 4 (4) (5)). When the application has been installed the linking downloader is overwritten and hence deleted. Remaining, the bridge launcher detects the application installation complete event (fig. 4 (6)), reads the stored intent (fig. 4 (7)), and sends it to the application (fig. 4 (8)). The application is launched with any optional data such as image passed to the linking downloader at fig. 4 (1) in accordance with the intent described. The combined operations of the linking downloader and the bridge launcher are the basic process of the smart downloader.

Furthermore in fig. 4 (1), intent can also be issued to launch an application when the user taps an application icon, even though there is no accompanying optional image data etc. In this case, the basic sequence of operations is the same as that above.

4. Conclusion

This article has described a smart downloader that offers a new pre-installation solution for Android terminals. The smart downloader performs intent transfer processing by combining an individual linking downloader with the shared bridge launcher application, and achieves application linking that was previously impossible with the standard downloader. This means that even more users can be lured into using a wider range of download-type pre-installed services. At the time of writing this article, the bridge launcher is to be pre-installed in Autumn/Winter 2011-2012 models, and the smart downloader system has been adopted for 13 applications.

We plan to continue to make the smart downloader even easier for service providers to adopt, by providing development packages and so forth.