Glympse URI Schema

Contents

Overview

The Glympse application for Android and iPhone register the "glympse:" protocol handler. The protocol handler allows Glympse to be launched (or brought to the foreground if already running) by simply invoking a URL. The query-string parameters passed to the Glympse URL allow the caller to automate certain functions of Glympse, including viewing a Glympse and sending a Glympse.

The Glympse URL can be invoked by the user or by other applications. For example, the URL could be a user-clickable hyperlink on an external or local web page, in an SMS text, in an email, etc. The URL can also be invoked from code by other applications on the device.

Some examples of how the Glympse protocol handler could get used.

On a Website

A restaurant website could place a link on their site that says 'Send us a Glympse to tell us you are on the way!". By clicking that URL, the Glympse application installed on the users phone would start with a pre-populated Glympse ready to send back to an email or phone number of someone at the restaurant.

My Favorite Glympse

Users could save a favorite Glympse as a Glympse URL, and then stick that URL wherever they want (say on the desktop of their Android or Springboard of their iPhone). One click and Glympse would start up pre-populated with that favorite Glympse.

Easy Application Integration

Any application that would like to add Glympse functionality can invoke a Glympse URL from within their application to start Glympse. The URL can contain parameters that specify what actions Glympse should perform. For example, the Contacts application could start Glympse and pre-populate the "To" field within Glympse to contain an email address or phone number of a user in Contacts. Or, the Maps application could

start Glympse with a pre-populated destination.

URL format

The most basic Glympse URL is one that will simply launch the Glympse client (or bring it to the foreground if it is already running):

`glympse:`	URL

Now, let's take a look at how we can add some parameters to the URL. Parameters are passed using standard URL encoded query-string formatting. Most Glympse parameters are passed as name/value pairs in the form of `name=value`. However, there is one case that allows just a nameless `value`. This condensed format is used when you want to specify a Glympse code to watch. In this one case, the parameter value is just the Glympse code. For example, to start the Glympse client and tell it to watch the user that sent the Glympse code of ABC-123, the URL would look like this:

`glympse:?ABC-123`	URL

Glympse supports multiple parameters. Here is an example of how to start Glympse and watch three different Glympse codes:

`glympse:?ABC-123&DEF-456&GHI-789`	URL

Just about every feature of Glympse can be automated via a parameter in the URL. Here is an example of how to start Glympse and pre-populate the "Send a Glympse" page with an email recipient:

`glympse:?src=sample&rec_type=email&rec_addr=sylvia@glympse.com`	URL

Here is that same URL, but we are setting the duration and adding a message and a destination:

`glympse:?src=sample&rec_type=email&rec_addr=`sylvia@glympse.com`&dur_min=30` `&dest_ll=33.812099,-117.918972&dest_name=Disneyland&msg_text=Meet+here!`	URL

Here are some rules about how URLs are processed by Glympse.
* Question marks is required after colon, if any arguments are passed. Slashes are not allowed.

`glympse:?ABC-123`	URL

* Multiple parameters are separated by ampersands (&)

`glympse:?src=sample&rec_type=sms&rec_addr=4255551212`	URL

* The value portion of each name/value pair should be URL encoded to escape any special characters.

For example, ampersands (&) should be encoded as %26, question marks (?) should be encoded as %3F, equals (=) should be encoded as %3D, spaces should be encoded as %20 or plus (+), etc.

`glympse:?src=sample&msg_text=Meet+here+for+drinks%3F`	URL

Parameters

The following table lists the parameters currently supported by the Glympse URL protocol handler. Unless noted otherwise, all parameters work on iPhone and Android.

Parameter	Description
None	Any value on the query-string that has no name is treated as a Glympse code to view. For example, the URL "glympse:?ABC-123" has a nameless value of "ABC-123". This value is assumed to be a Glympse code, and the Glympse application will attempt to view the Glympse with this code.
`src` `source` **Supported since** `glympse:`	This is a required value that specifies an application key for the caller. If you intend to use the Glympse protocol handler, please contact partners@glympse.com for an application key. For testing, you can use the Glympse protocol handler without a key. However, before releasing an application that uses the Glympse protocol handler, you will need to obtain a registered application key to prevent your application from getting blocked from calling the Glympse protocol handler.
`recipients` **Supported since** `glympse3:`	URL-encoded string containing JSON array of recipient objects. Each recipient object has the following fields populated: • `type` • `subtype` • `brand` • `name` • `address` The following recipient `types` are supported: <table><tr><td>`email`</td><td>Recipient is an email address.</td></tr><tr><td>`sms`</td><td>Recipient is a mobile phone number. Glympse will attempt to use the user's local phone number and operating system to send the invite if possible. If Glympse detects the device is not capable of sending SMS (like an iPod, iPad, or Android tablet), then the invite will be sent via the Glympse server's SMS system.</td></tr><tr><td>`group`</td><td>Recipient is a Glympse group. The `address` for this `type` must include a '!' as the first character.</td></tr><tr><td>`clipboard`</td><td>The Glympse URL is copied to the clipboard.</td></tr><tr><td>`twitter`</td><td>Post to Twitter. This assumes the user has Twitter setup</td></tr></table>

	locally.
`facebook`	Post to Facebook. This assumes the user has Facebook setup locally.
`link`	The Glympse URL is generated and is sent back to caller application on return URL.
`app`	The invite type should be used by 3rd party applications deeply integrated with Glympse. `subtype` field has to specify application ID in this case.

Example:

```
[
  {
    "type": "app",
    "subtype": "chaton",
    "brand": "chaton"
  }
]
```

`duration` **Supported since** `glympse3:`	When sending a Glympse, this specifies the total duration in milliseconds. This value is optional and defaults to the user preferred duration if not specified. dur_secs, dur_mins, and dur_hrs are mutually exclusive. You should only specify one of them.

`destination`
 	
Supported since `glympse3:`
	URL-encoded string containing JSON object with the following properties:
	• `name`
	• `latitude`
	• `longitude`
	
	Example:
	
	``
	{
	  "name": "Space Needle",
	  "latitude": 47.6200464,
	  "longitude": -122.349306
	}
	``

`message` **Supported since** `glympse3:`	Text to use for the message of the Glympse. Make sure this value is URL encoded.
`flags` **Supported since**	Set of flags defining the behavior of Send a Glympse wizard presented upon request of 3rd party application. The following flags are supported: • `FLAG_RECIPIENTS_EDITABLE` 0x0000000000000000L

glympse3:	• FLAG_RECIPIENTS_READ_ONLY 0x0000000000000001L • FLAG_RECIPIENTS_DELETE_ONLY 0x0000000000000002L • FLAG_MESSAGE_EDITABLE 0x0000000000000000L • FLAG_MESSAGE_READ_ONLY 0x0000000000000010L • FLAG_MESSAGE_DELETE_ONLY 0x0000000000000020L • FLAG_MESSAGE_HIDDEN 0x0000000000000040L • FLAG_DESTINATION_EDITABLE 0x0000000000000000L • FLAG_DESTINATION_READ_ONLY 0x0000000000000100L • FLAG_DESTINATION_DELETE_ONLY 0x0000000000000200L • FLAG_DESTINATION_HIDDEN 0x0000000000000400L • FLAG_DURATION_EDITABLE 0x0000000000000000L • FLAG_DURATION_READ_ONLY 0x0000000000001000L
screen **Supported since** glympse:	This optional value can be used to set the initial screen when Glympse is started (or when brought to the foreground if already running). This value is ignored if the query-string parameters also contain any parameters for sending a Glympse (like recipient, duration, destination, or message parameters) since those parameters infer that Glympse should start on the "Send a Glympse" screen. Possible values for this parameter are… | send	Start Glympse on the "Send a Glympse" screen. This value is assumed if any parameters for sending a Glympse (like recipient, duration, destination, or message parameters) are also specified on the query-string.
history	Start Glympse on the "History" screen. This can be useful if other applications want to have a "View your Glympse history" action.
favorites	Start Glympse on the "Favorites" screen. This can be useful if other applications want to have a "Send a favorite Glympse" action.
map (iPhone only)	Start Glympse on the "Map" screen.
settings (iPhone only)	Start Glympse on the "Settings" screen.
language (iPhone only)	Start Glympse on the Language select Settings screen.
about	Start Glympse on the "About" screen.
intro (iPhone only)	Start Glympse and start the first time user experience, "Introduction" screen.	

`initial_nickname` **Supported since** `glympse3:`	**[iOS only]** Initial nickname to set for the user if it isn't already set. If the user has set their nickname then this parameter is ignored. This can be useful if the app has just prompted the user to download the Glympse app you want the initial name to be set without having to go through the first time user experience in the Glympse app.
`ret_url` **Supported since** `glympse2:`	**[iOS only]** This optional value specifies a URL to navigate to after a Glympse has successfully been sent. This parameter is intended to be used with other parameters for sending a Glympse (like recipient, duration, destination, or message parameters). A common use for this parameter is when the calling application would like control returned to it after a Glympse has been sent. The URL specified can have any schema, so it is possible to take the user to a web site via an http URL, another application using a custom schema, or can even be used to call another glympse URL for nested Glympse actions. Make sure this value is URL encoded. The following parameters will be appended and passed back after the glympse is created. {table below}

`gly_recipients`	URL-encoded string containing JSON array of recipient objects. Each recipient object has the following fields populated: ● `type` ● `subtype` ● `name` ● `address` ● `url` ● `message`
`gly_duration`	Duration of the Glympse in milliseconds.

`ret_cancel_url` **Supported since** `glympse2:`	**[iOS only]** This optional value specifies a URL to navigate to after the user cancels a "Send a Glympse" operation. This parameter is intended to be used with other parameters for sending a Glympse (like recipient, duration, destination, or message parameters). The parameter is similar to ret_url, except that this URL is invoked when the user chooses not to send the Glympse that the caller passes in on the query-string. Make sure this value is URL encoded.

Parameters (Deprecated)

Parameter	Description
`rec_addr[#]` **Supported since** `glympse:`	Recipient address. This can be a phone number or an email address. This value is ignored if the recipient type (rec_type) is Twitter or Facebook. The [#] part of the name is an optional index value if more than one recipient is being specified. See the Sending to multiple recipients section below for more information.
`rec_type[#]`	Type of recipient. This value is required if sending to Twitter or Facebook and

Supported since `glympse:`	recommended if sending to a phone number or email address. If this value is not specified, then Glympse will examine the text in rec_addr to determine if the recipient is phone number or an email address.
	<table><tr><td>`email`</td><td>Recipient is an email address.</td></tr><tr><td>`sms`</td><td>Recipient is a mobile phone number. Glympse will attempt to use the user's local phone number and operating system to send the invite if possible. If Glympse detects the device is not capable of sending SMS (like an iPod, iPad, or Android tablet), then the invite will be sent via the Glympse server's SMS system.</td></tr><tr><td>`group`</td><td>Recipient is a Glympse group. The rec_addr for this rec_type must include a '!' as the first character.</td></tr><tr><td>`clipboard`</td><td>The Glympse URL is copied to the clipboard.</td></tr><tr><td>`twitter`</td><td>Post to Twitter. This assumes the user has Twitter setup locally.</td></tr><tr><td>`facebook`</td><td>Post to Facebook. This assumes the user has Facebook setup locally.</td></tr></table> The [#] part of the name is an optional index value if more than one recipient is being specified. See the Sending to multiple recipients section below for more information.
`rec_name[#]` **Supported since** `glympse:`	Friendly display name of the recipient specified in rec_addr. This value is ignored if the recipient is Twitter or Facebook, and optional if the recipient is a phone number or email address. If this value is specified, then it will be shown along with the raw value set in rec_addr. The value is intended to provide a friendly context for the recipient. Make sure this value is URL encoded. The [#] part of the name is an optional index value if more than one recipient is being specified. See the Sending to multiple recipients section below for more information
`dur_secs` **Supported since** `glympse:`	When sending a Glympse, this specifies the total duration in seconds. This value is optional and defaults to the user preferred duration if not specified. dur_secs, dur_mins and dur_hrs are mutually exclusive. You should only specify one of them.
`dur_mins` **Supported since** `glympse:`	When sending a Glympse, this specifies the total duration in minutes. This value is optional and defaults to the user preferred duration if not specified. dur_secs, dur_mins and dur_hrs are mutually exclusive. You should only specify one of them.
`dur_hrs` **Supported since** `glympse:`	When sending a Glympse, this specifies the total duration in hours. This value is optional and defaults to the user preferred duration if not specified. dur_secs, dur_mins and dur_hrs are mutually exclusive. You should only specify one of them.
`dest_addr`	Destination. Currently, the only supported format is: latitude,longitude

Supported since `glympse:`	
`dest_name` **Supported since** `glympse:`	Friendly display name of destination. Optional, but highly recommended if dest_addr is specified. If no name is provided, a name will be automatically generated using the latitude and longitude. Make sure this value is URL encoded.
`msg_text` **Supported since** `glympse:`	Text to use for the message of the Glympse. Make sure this value is URL encoded.

Sending to multiple recipients

There are three parameters related to specifying a recipient: rec_addr, rec_type, and rec_name. To enable sending to multiple recipients, these three parameters support an optional index number. Here is an example of sending to two recipients…

`glympse:?src=sample&rec_type1=email&rec_addr1=`sylvia@glympse.net`&rec_name1=Silvia` `&rec_type2=sms&rec_addr2=425-555-1212&rec_name2=Nicole`	URL

If no index value is specified, then a value of 1 is assumed. The order of the parameters don't matter (you could specify rec_addr2 before rec_addr1 if you wanted). The index numbers do not need to be consecutive (you could specify rec_addr1 and rec_addr3 without a rec_addr2). Once all the parameters have been processed by Glympse, the recipients will be sorted by index value and added to the recipient list UI in this order.

Calling the Glympse protocol handler on Android

The following is an example of how to launch Glympse from an Android application.

```Java
// Define our application key value.
String app_key              = "sample";

// Define the recipient, duration, message, and destination values.
String rec_addr             = "sylvia@glympse.net";
String rec_name             = "Sylvia";
int    dur_mins             = 15;
String msg_text             = "Off to the tower!";
String dest_name            = "Eiffel Tower";
double dest_latitude        = 48.8585;
double dest_longitude       = 2.2947;

// Build the Glympse URL with correct URL encoding.
String url = String.format(
    "glympse:?src=%s&rec_addr=%s&rec_name=%s&dur_mins=%d&msg_text=%s" +
    "&dest_name=%s&dest_addr=%f,%f",
    URLEncoder.encode(app_key, "UTF-8"),
    URLEncoder.encode(rec_addr, "UTF-8"),
    URLEncoder.encode(rec_name, "UTF-8"),
```

```
    dur_mins,
    URLEncoder.encode(msg_text, "UTF-8"),
    URLEncoder.encode(dest_name, "UTF-8"),
    dest_latitude,
    dest_longitude);


// Invoke the URL to launch Glympse.
startActivity(new Intent(Intent.ACTION_VIEW, Uri.parse(url)));
```

Determine if Glympse is installed on iOS

It is possible to determine if the Glympse app is installed on the device. This allows an app to gray out the option or possible direct the user to install Glympse from the app store.

```
// If Glympse is not installed                                         ObjC
if(NO == [[UIApplication sharedApplication] canOpenURL:
    [NSURL URLWithString:@"glympse:"]])
{
    // Redirect the user to the app store to install Glympse
}
```

Calling the Glympse protocol handler on iOS

The following is an example of how to launch Glympse from an iOS application.

```
// Define our application key value.                                   ObjC
NSString *app_key       = @"sample";

// Define the recipient, duration, message, and destination values.
NSString *rec_addr       = @"sylvia@glympse.net";
NSString *rec_name       = @"sylvia";
NSString *rec_type       = @"sms";
int      dur_mins        = 15;
NSString *msg_text       = @"Off to the tower!";
NSString *dest_name      = @"Eiffel Tower";
double    dest_latitude  = 48.8585;
double    dest_longitude = 2.2947;

// Build the Glympse URL with correct URL encoding.
NSString *url = [NSString stringWithFormat:
    @"glympse:?src=%@"
    @"&rec_addr=%@"
    @"&rec_name=%@"
    @"&rec_type=%@"
    @"&dur_mins=%d"
    @"&msg_text=%@"
    @"&dest_name=%@"
    @"&dest_addr=%f,%f",
    [Helpers urlEncode:app_key],
    [Helpers urlEncode:rec_addr],
    [Helpers urlEncode:rec_name],
    [Helpers urlEncode:rec_type],
```

```
    dur_mins,
    [Helpers urlEncode:msg_text],
    [Helpers urlEncode:dest_name],
    dest_latitude,
    dest_longitude];

// Invoke the URL to launch Glympse.
[[UIApplication sharedApplication] openURL:[NSURL URLWithString:url]];
```

Calling the Glympse protocol handler from an HTML web page

The following is an example of how to launch Glympse from HTML.

```
<a href="glympse:?src=sample&rec_addr=sylvia@glympse.net">
    Click here to send me a Glympse!
</a>
```
HTML