

## Sending and Receiving Data Messages

The GTalk Service includes functionality to transmit data messages between applications running on different devices. These data messages are handled separately from normal text chat messages and are invisible to users.

**The functionality described in this section was removed prior to the version 1.0 release of the Android SDK. This is largely because of the security implications associated with the ability to remotely execute code on a target device. It is expected that this API will be exposed for developer access in future releases of Android, although it may differ from the implementation described here.**

GTalk data messages are a mechanism that lets you broadcast Intents over the air (OTA) to remote user devices. On the target device, the GTalk Service extracts the Intent from the received message and rebroadcasts it locally, where it's handled by the Intent resolution mechanism in the same way as locally broadcast Intents. The process is illustrated in Figure 9-1.

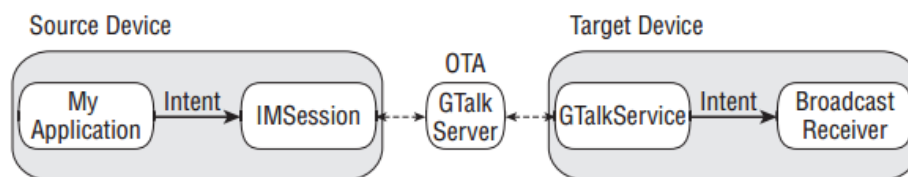


Figure 9-1

The result is an interface for broadcasting Intents on remote devices using instant messenger contacts. The broadcast Intent will be received by any Broadcast Receiver registered for the action represented by the Intent.

*By extending the reach of your applications beyond the scope of the device on which they're running, you take on additional responsibilities to ensure that your applications are well behaved, and to take all possible precautions to ensure that your applications aren't open to exploitation by those looking to use this mechanism maliciously.*

Data messages are an excellent way to support multi-user applications on distributed mobile devices, thanks to the low latency and rapid response times provided by the instant messaging architecture.

### Transmitting Data Messages

The best practice is to create custom actions to use when broadcasting an Intent to a remote device, such as the one shown in the snippet below:

```
public static final String ACTION_OTA_ELIMINATE = "com.paad.ota_eliminate_action";
```

The next snippet shows how to create a simple Intent that will be packaged within a data message to transmit the above action to a remote device:

```
Intent intent = new Intent(ACTION_OTA_ELIMINATE);
```

As with normal broadcast Intents, you can package additional information within the Intent using the extras Bundle. These extras will be included in the Intent when it's re-broadcast on the remote device.

```
intent.putExtra("long", String.valueOf(location.getLatitude()));
intent.putExtra("lat", String.valueOf(location.getLatitude()));
intent.putExtra("target", "Sarah Conner");
intent.putExtra("sender", gTalk.getUsername());
```

*Only String extras are currently supported in the OTA Intent broadcast mechanism. Non-string extras will be disregarded before transmission and won't be available on the target device.*

Send the message using the `sendDataMessage` method, passing in the target username and the Intent to broadcast. The `sendDataMessage` is available on IM Session or Chat Session objects, as shown below:

```
String username = "T1000@sky.net";
// Send to target user.
imSession.sendDataMessage(username, intent);
// Send to all chat room participants.
chatSession.sendDataMessage(intent);
```

## **Receiving Data Messages**

To listen for data messages, register a Broadcast Receiver that filters on the action String included in a transmitted Intent.

GTalk data messages are processed as normal broadcast Intents, so they have no sender information associated when they're received by a Broadcast Receiver. If you require such metadata, you should include them in the extras Bundle of the source Intent as was done in the code shown in the previous section.

The following skeleton code shows how to register a simple Broadcast Receiver implementation that can handle the Intent transmitted in the previous example:

```
BroadcastReceiver otaGTalkIntentReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context _context, Intent _intent) {
        if (_intent.getAction().equals(ACTION_OTA_ELIMINATE)) {
            String sender = _intent.getStringExtra("sender");
            String target = _intent.getStringExtra("target");
            String lat = _intent.getStringExtra("lat");
            String lng = _intent.getStringExtra("long");
            Location location = new Location(LocationManager.GPS_PROVIDER);
            location.setLatitude(Double.parseDouble(lat));
            location.setLongitude(Double.parseDouble(lng));
            // TODO: Do something with the data transmitted.
        }
    }
};
IntentFilter filter = new IntentFilter(ACTION_OTA_ELIMINATE);
registerReceiver(otaGTalkIntentReceiver, filter);
```