

DbSender

A Direct Technologies SMS Service Client Interface

Document Version 1.0

Date: 18-July-2011

Abstract:

This document explains the working of the DbSender application which is one of client interfaces offered by Direct Technologies to communicate with their SMS Service. The target audience for this document are the clients of the Direct Technologies SMS Service.

Working:

The DbSender is a stand-alone application written in the Java Programming Language, it sends sms messages defined in a database table over to the SMS Service server using a custom protocol. The application runs entirely on the client-side. The application uses two database tables one to pick up sms messages from and the other to push sms messages into that have been successfully sent. The names of the two tables are SendSms (listing sms to send) and SentSms (listing sms sent). Basically all the DbSender application does is to query sms messages from the SendSms table, send them and, if successful, insert them into the SentSms table as an acknowledgement.

Features:

The DbSender is a simple yet powerful application it has several features that make it critical as a client interface.

- **No custom programming needed:** It bypasses the requirement for the clients to write a custom program to interface with the SMS Service. While another client interface the Web Application also fulfills this requirement, it cannot handle a steady stream of sms traffic being generated at periodic intervals rendering itself not much useful in such a use case. For most of the corporate clients this is exactly how the sms traffic is generated and this stresses the requirement for such an interface. DbSender scans the database table for any sms traffic and sends it over to the server as soon as it finds any.
- **Database Source:** Since it uses a database table to source sms messages from it is especially useful for large corporations where sms traffic is generated as a response to end-user behaviour. For example a Bank customer withdrawing money from an ATM which triggers an sms to be sent, or an electronic funds transfer or a credit card transaction or a stock price movement which might cross a set threshold which might trigger an alert sms to be sent. To handle such cases triggers could be set up that pick appropriate data from various tables and simply drop it into the SendSms table so that the DbSender application sends this information as a sms to the SMS Service server for delivery to the mobile subscriber.
- **Secure:** The data stored in the database of most corporations is sensitive and very critical for them to be allowed to be accessed by a third party. The DbSender use of the database can be restricted to only defined and necessary operations using the database management systems security techniques. For most of the database management systems this means creating a user account with restrictive access privileges so that the user account's usage can be controlled as required.
The DbSender requires 'Select' & 'Delete' privileges on the SentSms table in order to select sms messages and later delete them to avoid duplication of sending and 'Insert' privilege on the SendSms table to insert sent sms.
- **Extensible:** Even though the DbSender comes with several standard implementations to

interface with most of the common Database Management Systems, there might still be cases where a client might feel the need to extend the applications capabilities, for the following reasons:

1. The DbSender does not provide a standard implementation for the particular dbms used by a client.
2. The DbSender does provide a standard implementation for the client's system, but a modification is required to the functionality provided in the standard implementation.
3. A company policy restrict's the use of any third party application to communicate directly with the company database.
4. A client wants send smses from a source other than a database, for example a file or some other proprietary data source.

In such cases the DbSender provides a way for the clients to define custom interfaces whereby the data is handled completely by the client written custom program and the DbSender is used only to transport messages over to the SMS Service server.

Read next section on 'Custom DataFeeders' for more.

Custom DataFeeders:

The DbSender's architecture is such that it requests sms data from DataFeeders which then feed the data pulled from a defined source. The DataFeeder is a generic interface implemented by all standard implementations. So when the source is a MySQL server the DbSender relies on a MySQLDataFeeder for feed it the sms data, an OracleDataFeeder in the case of a Oracle database and so on. But the DbSender is also written in a way that it does not need to know what the source of the data is or how the data is pulled from the source as far as a DataFeeder obeys the contract of the generic DataFeeder interface. So a client can write a custom data feeder that follows the DataFeeder contract and feed the data to the DbSender from a custom source. The only thing a client needs to do after implementing the custom feeder is to let the application know that he wishes to use a custom feeder and he's done. This can be done using the DbSender configuration file. So long as the contract's been obeyed the DbSender does not even need to know where the data comes from.

Taking advantage of this technique a client can not only write a feeder for a database system for which the DbSender does not provide a standard feeder implementation, he can also write a feeder that pulls data from a completely different source such as a file.

Further Information:

Information about the DataFeeder interface and other required details is mentioned in the DbSender API. The DbSender API alongwith an example of a sample custom implemented DataFeeder can be provided upon request.