

Guide for SQL developers

This paper provides information for software developers about how to configure an SQL-SMS connection in the Ozeki NG Gateway to send and receive SMS messages using a database server (All major database servers are supported: Oracle, MSSQL, MySQL, Sybase, Informatix, etc.) If you set up this functionality, you will be able to send an SMS message by simply executing an SQL Insert statement in your database server. This paper gives you all the necessary information to achieve this. You can learn how to add an SQL user in the configuration file and how to configure sending and receiving. We also provide links to explanations of different database server connection settings.

The Ozeki NG SMS Gateway supports database connectivity. Database connectivity allows you to send or receive SMS messages using simple SQL queries. The way this works is illustrated in Figure 1.

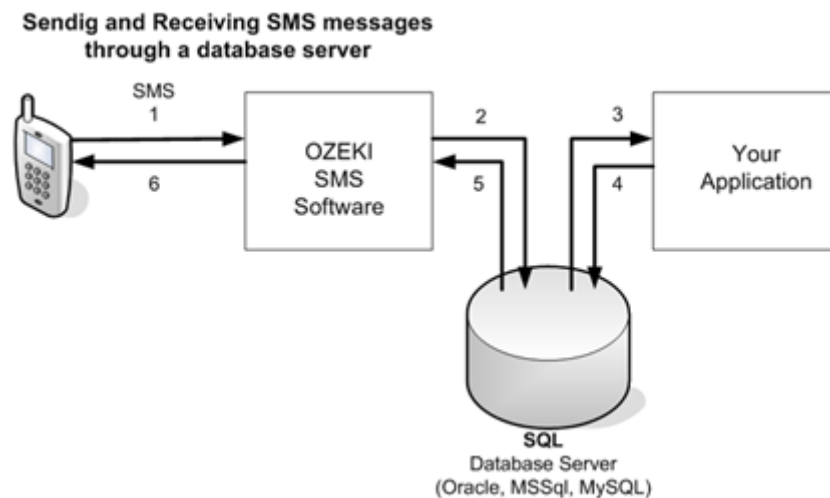


Figure 1 - Database connectivity

If you want to send and receive SMS messages through a database server, you need an SQL user in the OZEKI NG SMS Gateway.

The SQL user is a virtual user in the SMS Gateway that can communicate with a specified SQL server. The Ozeki NG SMS Gateway supports Oracle, MS SQL, MySql, Postgres, Sybase and any other relation database server that has an ADO or an ODBC driver.

To add an SQL user, you have to configure database connectivity. To do this, you need to specify the SQL user directives and include the insert scripts in the configuration file.

1.) Example of creating an SQL user

```
<USER>
```

```
Username sql1  
Type SQL
```

```
Database "Driver={MySQL ODBC 3.51  
Driver};Server=localhost;Database=sms;User=root;Password=;Option=4"
```

```
SQLSelect "select oz_id,oz_receiver,oz_msg,oz_operator,oz_msgtype,oz_sender,oz_reference from  
ozekimessageout where  
oz_status='send'"  
SQLUpdateDownloaded "update ozekimessageout set oz_status='sent' where oz_id in ($idsq)"  
SQLUpdateTransmitted "update ozekimessageout set  
oz_status='transmitted',oz_senntime='$senttime',oz_reference='$reference' where oz_id='$id'"  
SQLUpdateDelivered "update ozekimessageout set oz_status='received',oz_receivedtime='$receivedtime'"
```

```
where oz_reference='$reference'"
SQLUpdateNotSent "update ozekimessageout set oz_status='deleted',oz_errormsg='$errormsg' where oz_
id='$id'"
SQLInsert "insert into ozekimessagein
(oz_operator,oz_sender,oz_receiver,oz_msg,oz_senttime,oz_receivedtime,oz_msgtype)
values ('$operator','$sender','$receiver','$msg','$senttime','$receivedtime','$type')"
```



```
SQLValueReplaceFrom1 '
SQLValueReplaceTo1 "
```



```
SQLValueReplaceFrom2 "
SQLValueReplaceTo2 ""
```



```
Pollinterval 10
```



```
DateFormat yyyy-mm-dd hh:nn:ss
```



```
LogMessages On
LogSelects On
LogUpdates On
</USER>
```

2.) SQL user directives

Username

The username is a unique text, so you cannot use the same username for two users.

Username <text>

Password

The password of the users is an unencrypted password text. You should use a plain text password, as encrypted passwords will not work. You ought to use at least 8 characters including characters representing a combination of numbers and letters (both uppercase and lowercase).

Password <text>

Type

The type defines the user and its connection type: SQL

Type SQL

Database

It is the database connection string which describes the data source.

Database <text>

SQLSelect

The SQLSelect template describes the way of selecting outgoing messages.

SQLSelect "select id,receiver,msg,operator,msgtype,sender,reference from ozekimessageout where status='send'"

SQLUpdate

The Update Downloaded SQL template describes the way of marking outgoing messages as sent messages.

SQLUpdateDownloaded "update ozekimessageout set status='sent' where id in (\$idsq)"

SQLUpdateTransmitted

The SQLUpdateTransmitted template describes the way of marking sent messages as transmitted messages.

SQLUpdateTransmitted "update ozekimessageout set status='transmitted',senttime='\$senttime',reference='\$reference' where d='\$id'"

SQLUpdateDelivered

The SQLUpdateDelivered template describes the way of marking transmitted messages as delivered messages.

SQLUpdateDelivered "update ozekimessageout set status='received',receivedtime='\$receivedtime' where reference='\$reference'"

SQLUpdateNotSent

The SQLUpdateNotSent template describes the way of marking sent messages as not sent messages.

SQLUpdateNotSent "update ozekimessageout set status='deleted',errmsg='\$errmsg' where id='\$id'"

SQLInsert

The SQLInsert template describes the way of inserting messages.

SQLInsert "insert into ozekimessagein (operator,sender,receiver,msg,senttime,receivedtime,msgtype) values ('\$operator','\$sender','\$receiver','\$msg','\$senttime','\$receivedtime','\$type')"

SQLValueReplaceFrom1

The SQLValueReplaceFrom1 describes what to replace.

SQLValueReplaceFrom1 <text> Default '

SQLValueReplaceTo1

The SQLValueReplaceTo1 describes what to replace with.

SQLValueReplaceTo1 <text> Default "

SQLValueReplaceFrom2

The SQLValueReplaceFrom2 describes what to replace.

SQLValueReplaceFrom2 <text> Default "

SQLValueReplaceTo2

The SQLValueReplaceTo2 describes what to replace with.

SQLValueReplaceTo2 <text> Default ""

Pollinterval

The poll interval defines the periodic SQL message fetching in seconds.

Pollinterval <integer> Default 10

DateFormat

The DateFormat directive defines the date representation.

DateFormat yyyy-mm-dd hh:nn:ss

LogMessages

The LogMessages directive allows you to log message traffic.

LogMessages On

LogSelects

The LogSelects allows you to log all SQL selects.

LogSelects On

LogUpdates

The LogUpdates directive allows you to log all SQL updates.

LogUpdates On

3.) SQL Insert scripts and database connection strings for various database engines

Every SQL user account means a database connection between the Ozeki NG - SMS Gateway and the database server.

We explain the following database server connections:

2.3.1) [MySQL](#)

2.3.2) [MS SQL](#)

2.3.3) [ORACLE](#)

2.3.4) [Sybase / SQLAnywhere](#)

2.3.5) [PostgreSQL](#)