

mod_db

About

mod_db implements an API and dialplan interface to a database backend. The database can either be in sqlite or ODBC. It also provides support for group dialing and provides database backed limit interface.

Requirements

ODBC must be configured to use ODBC resources (configure with --enable-core-odbc-support).

Installing

mod_db is compiled and loaded by default. Modify db.conf to enable ODBC.

Concepts

Realm and key are arbitrary strings. Consider realm as a container for keys.

Dialplan Application

Database

Insert a value into the database:

```
<action application="db" data="insert/realm/key/value" />
```



mod_db manages 'realm/key' as a UNIQUE tuple (2-tuple). So if you 'insert' two, or more, times the same 'realm/key' tuple, then only one row, of 'realm/key/value', will be in the persistent database. The one-only stored row is the row resulting from the last successful 'insert/realm/key/value' method (last-in). Method 'select/realm/key' returns this last-in, one-only resulting row.

As of 2013-05-08, newly created schemas enforce this uniqueness with an index. Previous versions only maintain it internally with a SQL delete + insert.

Delete an entry from the database:

```
<action application="db" data="delete/realm/key" />
```

Retrieve a value from the database:

```
<action application="set" data="var=${db(select/realm/key)}" />
```

Use as a condition:

```
<condition field="${db(select/realm/key)}" expression="^value$" />
```

Use as a condition checking if item exists:

```
<condition field="${db(exists/realm/key)}" expression="^true$" />
```

Group

Group allows one to define multiple endpoints for a given destination (group).

Insert a group entry:

```
<action application="group" data="insert:groupname:sipurl"/>
```

Delete a group entry:

```
<action application="group" data="delete:groupname:sipurl"/>
```

Select a group entry:

```
<action application="set" data="api_result=${group(call:groupname)}/>
```

API

Database

```
db insert/realm/key/value  
db delete/realm/key  
db select/realm/key  
db exists/realm/key
```

Group

```
group insert:groupname:sipurl  
group delete:groupname:sipurl  
group call:groupname[:order]
```

Limit

The db backend for limit provides a persistent datastore for limit data. If using ODBC, this backend can also be shared across a cluster of FreeSWITCH™ instances.



This backend does NOT support rate based limits.