



jODBC – Service and SQL Catalog

`-W bufferSize`
Set write buffer size (in bytes)

jbase_agent as a Service

The service options allow jbase_agent to be installed and started as a Win32 Service on Windows platforms, and as a daemon on Linux platforms.

1. Windows:

```
jbase_agent install Service Name {options}
jbase_agent start Service Name
jbase_agent stop Service Name
jbase_agent remove -n Service Name
```

ServiceName is the Name of the service as it appears in Windows services
options are command options as in the jbase_agent section

Note that jbase_agent Win32 services will send logger messages to the Windows Event Log. Changing the tracing level (option `-L`) and logging to a file (option `-F`) is not applicable to a jbase_agent service and only error messages will be logged.

2. Linux:

```
jbase_agent start <options>
jbase_agent stop
```

Enabling SSL Encryption

jAgent may be configured to use SSL encryption. To activate SSL encryption it is enough to provide a valid x509 certificate and a private key (See command line options).

The following example describes how to create a self-signed certificate with OpenSSL on Windows:

1. Create a RSA private key:

```
C:...>openssl genrsa -out key.pem 1024
Loading 'screen' into random state - done
Generating RSA private key, 1024 bit long modulus
.+++++
....+++++
e is 65537 (0x10001)
```

2. Create a certificate (“signing”) request using the previously created private key:

```
C:...>openssl req -config c:\programs\openssl\openssl.cnf -new -key key.pem -out req.pem
Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '!', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MyCompanyName
Organizational Unit Name (eg, section) []:
```

Common Name (eg, YOUR name) []:MyCompanyName
 Email Address []:
 Please enter the following 'extra' attributes to be sent with your certificate request
 A challenge password []:
 An optional company name []

At this stage the certificate request should be sent to a Certification Authority, or proceed to the next step to self-sign the certificate e.g. if used for testing purposes.

1. Create self-signed certificate:
 C:\...>openssl req -config c:\programs\openssl\openssl.cnf -x509 -key key.pem -in req.pem -out cert.pem -days 365
 Loading 'screen' into random state - done
2. Start jAgent in SSL mode:
 C:\programs\openssl\bin>jbase_agent -c cert.pem -k key.pem

Authentication modes:

jbase_agent supports three different authentication modes

1. none - Does not verify any user credentials
2. user - Verifies user credentials against a JAGENT_USER file. See Setup user authentication
3. account - Account / Password will authentication against JEDIFILENAME_SYSTEM file

Setup user authentication

1. Create the jAgent user file
 \$ CREATE-FILE JAGENT_USER 1 53
 [417] File JAGENT_USER]D created , type = J4
 [417] File JAGENT_USER created , type = J4
2. Create the user and set the password
 \$ jbase_agent adduser test
 \$ jbase_agent passwd test newpassword
3. Now start jbase_agent to authentication with user authentication.
 \$ jbase_agent -A user

Setup account authentication

The environment should be configured with JEDIFILENAME_SYSTEM set to a valid jBASE SYSTEM file that contains the jBASE accounts you wish to connect to.

1. Start jbase_agent to authentication with account authentication
 \$ jbase_agent -A account
- 1.
2. When using Account authentication, the password in the ODBC connection dialog is the password for the jBASE account and not the password of the connecting user.

Examples:

- Start jbase_agent on default port (20002):
 jbase_agent

- Start jbase_agent on port 20003 and configure tracing to display errors only
jbase_agent -p 20003 -L 6
- Start jbase_agent with SSL encryption with certificate mycert.cer and private key mypk.pk on Windows
jbase_agent -c c:\certs\mycert.cer -k c:\certs\mypk.pk
- Install and start jbase_agent as a Windows service with default service name (jBASE jAgent Server) on default port
jbase_agent install
jbase_agent start
- Install and start jbase_agent as a Windows Service with service name MyAgent1, port 20003, and user authentication
jbase_agent install -n MyjAgent1 -p 20003 -A user
jbase_agent start
- Starting jAgent as a Unix daemon on port 20003 and redirecting logging messages to text files.
jbase_agent start -p 20003 -F

sqlcatman {options} {-f *filename*} {-e *dictname*}

tablename

The sqlcatman command is used to add and maintain SQL tablenamees in a catalog.

This catalog is used by SQLSELECT and the jBASE ODBC Datasource if specified under Adv. Options as a UserCatalog:

There can be multiple SQL catalogs and the catalog to be altered must first be defined by the environment variable JSQLFILENAME_CATALOG. The first entry added to the catalog will create the file as a JR type file or you can create the file ahead as a JR type file. E.g. CREATE-FILE CATALOG1 TYPE=JR then export/SET JSQLFILENAME_CATALOG=CATALOG1

options

- a Add catalog entry
- d (limited use. Allows a DICT file to be added as a SQL table)
Add dict portion of catalog entry
- r Delete catalog entry
- h Help screen
- f *filename*
jBASE hashed file, directory file or STUB file data portion.
- e *dictname*
File for jBASE DICT formatted items that are to be used with this tablename. This is usually a

subset or altered version of the files actual dictionary which is more SQL standard in naming and format.

tablename

The name of the table as used by SQLSELECT or the ODBC connection.

tablename = name of SQL table

filename = name of jBASE file (if omitted same as tablename)

dictname = name of jBASE dict file to use, (if omitted default for filename)

Examples:

For all examples, you must already have JSQLFILENAME_CATALOG set or SET JQSQFILENAME=CATALOG1 (Windows) or export JSQLFILENAME_CATALOG=CATALOG1 (Linux) where CATALOG1 is the name of this SQL catalog.

- Add a new catalog entry with default data and dictionary files
sqlcatman -a CUSTOMER
- Create a new catalog entry with default data and dictionary, but alternate catalog name. This is aimed at non-standard SQL file names such as those with periods in the name. This would mean SQLSELECT or ODBC connector would refer to the CUSTOMER.HISTORY file and DICT items as the CUST_HIST table.
sqlcatman -a -f CUSTOMER.HISTORY CUST_HIST
- Alternative dictionary with a file, this can help limit which dictionary items users can see, or you may have a non-SQL standard DICT items in the actual dictionary.
sqlcatman -a -f CUSTOMER -e CUSTOMER_SQL CUST_SQL
- Remove a table from the catalog
sqlcatman -r CUSTSQL