# SQL Report

**Report Generator for HP ELOQUENCE**

## ODBC Driver

**Editions:**

First Edition, August 1996

# Preface

This manual describes the installation and administration of the **SQL/R** ODBC driver.

The manual is divided into the following chapters:

| | |
|---|---|
| Chapter 1 | **Introduction**<br>Introduction and general information about **SQL/R** ODBC. |
| Chapter 2 | **Server installation and configuration**<br>This chapter covers the installation, configuration and administration of the ODBC server. |
| Chapter 3 | **Client installation and configuration**<br>This chapter covers the installation and configuration of the **SQL/R** ODBC driver on the client system. |
| Chapter 4 | **Driver Reference**<br>This chapter contains driver reference information. |

### Typographical Conventions

Unless otherwise stated, the following symbolic conventions are used in this manual:

| | |
|---|---|
| `Computer Font` | Computer font indicates commands, keywords, options, literals, source codes, system outputs and path names. |
| [ ] | The symbol [ ] indicates a key on a computer keyboard or a sign on the screen ("button") that can be activated using a computer mouse in order to activate the function enclosed by the symbol. For example, [CTRL] indicates the Control key while [Continue] is a ("button") on the screen, which can be activated using a mouse in order to continue to the next step. |
| [CTRL]-[char] | The symbol [CTRL]-[char] indicates a control function. For example [CTRL]-[Y] means that the control key and `Y` on the keyboard should be pressed simultaneously. |
| *italics* | Within syntax statements, a word in italics represents a dummy argument that has to be replaced with an actual parameter at run time. In the following example, *filename* has to be substituted by the name of the file to be used in the application:<br>`lp` *filename* |
| [ ] | Square brackets within syntax statements enclose optional parameters. In the next example, `TEMP` indicates that $-ddev$ is optional:<br>`lp [-d`*dev*`]` *filename* |

## Additional Reading

Further in this manual, the reader is referred to the following literature:

**ODBC 2.0, Programmers Reference and SDK Guide**

The definitive guide to the ODBC standard. It includes a complete description of the ODBC API, ODBC programming examples (in C) and covers the ODBC SDK.

Microsoft ODBC 2.0 programmer's reference and SDK guide : for Microsoft Windows and Windows NT. Microsoft Press, ISBN 1-55615-658-8

**HP-UX (online) Documentation**

References of the form `services`(4) refer to the given topic or item (here `services`) contained in the indicated section (here 4) of the HP-UX reference manual. It is also possible to obtain this documentation online using the command `man`, whereby in the case of `services`(4) the user should enter the following statement:

```
man 4 services
```

**SQL Introduction**

ODBC is built on SQL. ODBC specifies the mechanism, how SQL statements and data are passed between the database and the client program in a database independent way. Most ODBC client programs even provide a way of passing “raw” SQL to the database. While a SQL tutorial is out of scope for this manual, you should have a look at a SQL introduction to make best use of the **SQL/R** ODBC.

# Contents

# 1

# General Information

## 1.1 The SQL/R ODBC Driver

The **SQL/R** ODBC driver is an extension to **SQL/R** which gives you online access to your HP Eloquence databases from any ODBC compliant application.

This eliminates the need of redundant data and allows the end user to easily and quickly obtain any necessary data from your company database.

## 1.2 Introduction to ODBC

Corporations typically have applications and data residing on diverse platforms and database management systems due to historical, strategic or technological reasons. The Microsoft Open database connectivity (ODBC) is a de facto standard interface for accessing data in a heterogeneous environment of relational and non-relational database management systems.

ODBC is based on a specification, which was developed by a consortium of over 40 companies (members of the SQL Access Group and others), and has broad support from application and database vendors.

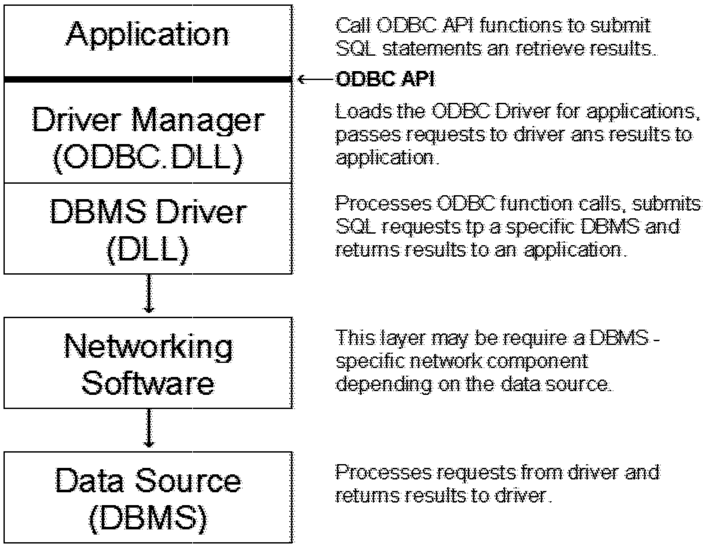
ODBC provides an open, vendor-neutral way of accessing data stored in a variety of proprietary personal computers, minicomputers and mainframe databases. With ODBC, an application can concurrently access, view, and modify data from multiple, diverse databases. ODBC provides many significant benefits to developers and end users by providing an open, standard way to access data:

- ODBC allows corporations to continue to use their existing database management systems, while enabling access to their data by common applications.
- ODBC benefits users as more end-user applications connect to additional data sources, making the vast volumes of corporate data more readily available.
- ODBC allows users to access data in more than one data storage location (for example, more than one server) from within a single application.

- ODBC allows users to access data in more than one type of DBMS (such as DB2, Oracle, HP Eloquence, and dBASE) from within a single application.
- ODBC provides a standard, open, and vendor-neutral API.
- ODBC allows corporations and software vendors to protect their investments in existing DBMSs.
- ODBC is based upon the SQL Access Group (SAG) Call Level Interface (CLI) and provides a standard SQL language based upon ANSI standards.

## 1.3 How ODBC Works

ODBC defines an API. Each application uses the same code, as defined by the API specification, to talk to many types of data sources through DBMS-specific drivers. A Driver Manager sits between the applications and the drivers. In Windows, the Driver Manager and the drivers are implemented as dynamic-link libraries (DLLs).




introa

The application calls ODBC functions to connect to a data source, send and receive data, and disconnect.

The Driver Manager provides information to an application such as a list of available data sources; loads drivers dynamically as they are needed; and provides argument and state transition checking.

The driver, developed separately from the application, sits between the application and the network. The driver processes ODBC function calls, manages all exchanges between an application and a specific DBMS, and may translate the standard SQL syntax into the native SQL of the target data source. All SQL translations are the responsibility of the driver developer.

Applications are not limited to communicating through one driver. A single application can make multiple connections, each through a different driver, or multiple connections to similar sources through a single driver.

To access a new DBMS, a user or an administrator installs a driver for the DBMS. The user does not need a different version of the application to access the new DBMS. This is a tremendous benefit for end users, as well as providing significant savings for IT organizations in support and development costs.

## 1.4 Summary

ODBC defines a standard SQL grammar and set of function calls that are based upon the SAG CLI specification, called the core grammar and core functions, respectively. If an application developer chooses only to use the core functionality, they need not write any additional code to check for specific capabilities of a driver.

With core functionality, an application can:

- Establish a connection with a data source, execute SQL statements, and retrieve results.
- Receive standard error messages.
- Provide a standard login interface to the end user.
- Use a standard set of data types defined by ODBC.
- Use a standard SQL grammar defined by ODBC.

ODBC also defines an extended SQL grammar and set of extended functions to provide application developers with a standard way to exploit advanced capabilities of a DBMS.

Finally, ODBC supports the use of DBMS-specific SQL grammar, allowing applications to exploit the capabilities of a particular DBMS.

## 1.5 Prerequisites

The following are prerequisites to the installation of the **SQL/R** ODBC driver:

Prerequisites for the Server system (HP 9000):

- HP 9000 Series 800 or 700
- HP-UX Release 9.0 or later
- HP-UX ARPA Services (TCP/IP network service)
- HP ELOQUENCE Version A.03.10 or later
- Approx. 4 MB available memory in the `/opt` file system
- DDS drive
- **SQL/R** Version A.01.50 or later

Prerequisites for the client system:

- PC with Windows 3.11, Windows for Workgroups 3.11, Windows 95 or Windows NT
- TCP/IP network protocol stack compliant with the `WINSOCK 1.1` standard.

The **SQL/R** product is only available on 60m DDS tapes (server software) and 3.5" (1.44 MB) floppy disk (client software).

## 1.6 License

Each server (HP 9000) must have an individual license for **SQL/R** and **SQL/R** ODBC. The **SQL/R** ODBC server can only be executed on the system for which the license was obtained.

The client (PC) software can be installed on an any number of systems within one company or institution.

Please refer to the enclosed license agreement for specific details about the conditions for the use of **SQL/R** ODBC.

**SQL/R** is available as an evaluation copy for a nominal fee. It provides the full functionality and comes with all product material. The evaluation copy is valid for one month.

## 1.7 SQL/R on the Internet

Actual information about **SQL/R** is available on the World Wide Web at the following URL

**`http://www.msede.com/public/sqlr/`**

# 2

# Server installation and configuration

This chapter describes the installation and administration of **SQL/R** ODBC on the server system (HP 9000).

It covers the following topics:

- Installing **SQL/R** ODBC on the server
- Configuring the server
- System configuration
- Administration tasks

## 2.1 Installing SQL/R ODBC on the Server

The **SQL/R** ODBC server software is included with the **SQL/R** software distribution. If you installed the **SQL/R** software release A.01.50 or above, the server software should already be present on your system. Please refer to the **SQL/R** installation documentation for further information on how to install the **SQL/R** software.

In order to use **SQL/R** ODBC, you need a license key. You should either have received a temporary license key along with the **SQL/R** software, or you can request your license key from your distributor or Marxmeier Software Entwicklung GmbH using the included fax form.

To install the ODBC license key, perform the following steps:

1. Login as root.
2. Add the license key to your license file using your favorite editor program. The location of the license file depends on the revision of the HP-UX operating system:

   **HP-UX 9.x** `/opt/sqlr/etc/licence`
   **HP-UX 10.x** `/etc/opt/sqlr/licence`

   You can use the `/opt/sqlr/etc/chklic` utility program to check your license file.

**Please note:** The license key must by typed exactly as on the license sheet.

## 2.2 Configuring the Server

This section covers the following topics:

- Configuring the server
- Server Security
- Network options
- Server Options
- Server Logging

### 2.2.1 Configuring the server

The **SQL/R** ODBC server is configured by the configuration file `odbc.cfg`. The location of the file `odbc.cfg` depends on the revision of the HP-UX operating system:

**HP-UX 9.x** `/opt/sqlr/etc/odbc.cfg`
**HP-UX 10.x** `/etc/opt/sqlr/odbc.cfg`

The configuration file is read once by the **SQL/R** ODBC server when it is started.

The `odbc.cfg` config file has sections and configuration items. Each section begins with a title which is enclosed in square brackets (`[...]`). Currently only the section `[config]` is used.

Configuration items contain information in the form of single lines of text.

Lines starting with a hash character (`#`) are comments.

### 2.2.2 Server Security

We recommend that you create a special login and group for the purpose of running the **SQL/R** ODBC server, for example "`sqlr`".

If you decide to use an existing login or group to run the server, you should *not* choose one with is used for administrative tasks, such as "`bin`", "`daemon`" etc. While the ODBC server provides its own authorization scheme and we are not aware of a security problem in the ODBC server, it is generally a bad idea to have complex programs run with special permissions.

The **SQL/R** ODBC server by default uses the access rights of the user, who started the server. When the **SQL/R** ODBC server is started with root capabilities it will switch to the login and group as specified in the configuration file.

| Entry | Description |
|---|---|
| `User` | The login name (or number) of the system account to run the server as. This option is ignored, if the server is not started with root privileges. A configuration value is required, if the server is started with root privileges. |
| `Group` | The group name (or number) of the system group to run the server as. This option is ignored, if the server is not started with root privileges. A configuration value is required, if the server is started with root privileges. |
| `Security` | This configuration item specifies the server security policy. This defines how user names and passwords are validated. The following options are valid:<br>`Security = odbc`<br>The server will validate passwords and user names using `odbc.user`. This is the default. |

For example:

```
User = user
Group = group
Security = odbc
```

Where *user* is the system account and *group* is the system group name the server should run with.

### 2.2.3 Network options

| Entry | Description |
|---|---|
| `Service` | The service name (as defined in /etc/services) or the port number, where the server should listen for requests. The default value is `sqlrodbc`. |
| `UseKeepAlive` | Numeric flag if the KEEP ALIVE socket option should be used. Valid choices are 1 (yes) and 0 (no). The default value is 1.<br>If this option is active, the server will check after a system defined period of inactivity if the client is still alive. |

For example:

```
Service = sqlrodbc
UseKeepAlive = 1
```

### 2.2.4 Server options

| Entry | Description |
|---|---|
| `dsnFile` | The path and name of the odbc.dsn file. This file is used to define all accessible data bases.<br>The default value depends on your HP-UX revision:<br>**HP-UX 9.x** `/opt/sqlr/etc/odbc.dsn`<br>**HP-UX 10.x** `/etc/opt/sqlr/odbc.dsn`<br>Please refer to section 2.4.2 for more information about the file `odbc.dsn`. |

| Entry | Description |
|---|---|
| `userFile` | The path and name of the `odbc.user` file. This file is used to define all users which are allowed to use the **SQL/R** ODBC server.<br>The default value depends on your HP-UX revision:<br>**HP-UX 9.x** `/opt/sqlr/etc/odbc.user`<br>**HP-UX 10.x** `/etc/opt/sqlr/odbc.user`<br>Please refer to section 2.4.1 for more information about the file `odbc.user`. |
| `tmpDir` | Path where tmp files should be written to. If not defined, the system default location will be used. Please refer to tempnam(3) for more information.<br>Please note, that **SQL/R** ODBC can take up considerable amount of temporary disc space with large data bases. To avoid any impact on your normal usage (due to a possible file system overflow), you should choose a distinct filesystem from your data base.<br>**Please note:** If temporary files are created on the same filesystem as your data base, this may cause a data base failure if the filesystem fills up. |
| `panic` | This option defines the action which should be taken when a fatal error is encountered. The following options are valid:<br>`Panic = exit`<br>Terminate the process. This is the default.<br>`Panic = dump`<br>Create a core dump. |

| Entry | Description |
|---|---|
| `Lang` | This configuration option defines the locale, the server should use. The default value is “C”. |
| `Messages` | This configuration option defines the language for server messages. This value defaults to Lang. |

For example:

```
dsnFile = /etc/opt/sqlr/odbc.dsn
userFile = /etc/opt/sqlr/odbc.user
tmpDir = /tmp
Panic = exit
Lang = C
Messages = C
```

### 2.2.5 Server logging

The **SQL/R** ODBC server can use the system syslog or a logfile to record selected events.

Each message has an associated origin and a severity. The `LogFlags` configuration item defines which messages should be output or suppressed.

The following message origins are available:

| Flag | Description |
|---|---|
| `*` | Applies to all message origins |
| `N` | Network communication |
| `P` | Client/server protocol |
| `T` | Cursor management |
| `S` | Statement management and execution |
| `C` | Configuration subsystem |
| `K` | SQL Kernel |
| `Y` | Syntax parser |
| `I` | Isam subsystem |
| `O` | Query optimizer |

The following severities are available:

| Severity | Description |
|---|---|
| `0` | Error messages |
| `1` | Informational |
| `2` | Debug |
| `3` | Verbose debug |

When logging to the syslog daemon, you can define a syslog facility (syslog channel) which is used by the syslog daemon to route the message to the appropriate destination and a syslog ident which will prepend each message in the syslog.

When using syslog, the following priorities are mapped:

| Severity | Syslog Priority |
|---|---|
| `0` | LOG_ERR |
| `1` | LOG_NOTICE |
| `2` | LOG_DEBUG |
| `3` | LOG_DEBUG |

Please refer to syslogd(1M) for more information.

| Entry | Description |
|---|---|
| `LogFile` | |

This defines where log messages are written to. This configuration value either specifies a path and file name or one of the keywords below:

`LogFile = console`

log messages are written to the console

`LogFile = syslog`

log messages will be sent to the syslog daemon

`LogFile = default`

the log will be written to a default location:

**HP-UX 9.x** `/usr/adm/sqlrodbc.log`
**HP-UX 10.x** `/var/adm/sqlrodbc.log`

It is necessary to start the server with root capabilities in order to use this option.

The default value is " syslog ".

| Entry | Description |
|---|---|
| `SysIdent` | syslog identifier. The default value is " sqlrodbc ". See syslogd(1M) for more information. |
| `SysFacility` | syslog facility (USER/DAEMON/LOCAL0..LOCAL7). The default value is " USER ". See syslogd(1M) for more information. |
| `LogFlags` | This configuration item defines which messages will be logged. Default LogFlags are " *0 ". |

For example:

```
LogFile = syslog
SysIdent = sqlr
SysFacility = USER
LogFlags = "*0"

LogFile = /tmp/sqlrodbc.log
LogFlags = "*0S1Y1"
```

### 2.2.6 Example configuration file

Most configuration options need not be defined explicitly, because the defaults are normally sufficient. The following example contains a complete server configuration file:

```
[config]
User = user
Group = group
tmpDir = /tmp
LogFile = default
LogFlags = "*0"
```

Where *user* is the system account and *group* is the system group name the server should run with.

## 2.3 System configuration

This section covers the following topics:

- Configuring the TCP/IP Transport
- Configuring the syslog daemon
- Starting the server
- Server commandline arguments
- Automatic Server Start

### 2.3.1 Configuring the TCP/IP Transport

The server and clients are connected using the TCP/IP protocol. To enable an access to the network through TCP/IP, each service name must be allocated a particular unique port number. This can be archived by adding the following lines to the file `/etc/services` of the server:

```
sqlrodbc  8003/tcp  # SQL/R ODBC
```

This statement consists of the following parts:

- *sqlrodbc* is the service name
- *8003* is the number (address) of a particular unique port
- *tcp* is the protocol name
- the text after the '#' symbol is a comment

When selecting a port number, please note the following:

- The used port number should *not* be less than 1024.
- Use a port number that is *not* being used for any other service.
- The port numbers between 5800 and 7500 are already allocated to X-Window System and should *not* be used.

Additional information is available in the HP-UX online help in `services`(4).

**Please note:** The port number **must** be the same on both client and server system.

### 2.3.2 Configuring the syslog daemon

When logging to the syslog daemon you may want to redirect the **SQL/R** messages to a distinct file. To accomplish this, you can change the `/etc/syslog.conf`.

If you add the following line to the syslog configuration file `/etc/syslog.conf`, all messages for the `local0` facility are written to the file `/var/adm/syslog/odbc.log`.

```
local0.* /var/adm/syslog/odbc.log
```

**Please note:** When you change the syslog configuration file, you must send a `HUP` signal to the `syslogd` process to cause it to reread the configuration file.

For example:

```
$ kill -HUP ‘cat /etc/syslog.pid‘
```

Please refer to syslogd(1m) for more information.

#### Starting the server

The **SQL/R** ODBC server is either invoked by the Internet daemon `inetd` each time a connection to the sqlrodbc service is made, or alternatively it can run as a daemon which executes continuously, handling requests.

If run as a daemon, the server is started either manually or automatically at system startup time.

A standalone server (daemon) has slight performance enhancements during connection because the setup code will only be executed once. If you use **SQL/R** ODBC regulary, we recommend to run the server as as daemon.

### 2.3.3 Server commandline arguments

The **SQL/R** ODBC server configuration is defined by the server configuration file `odbc.cfg`. The commandline switches can be used to override the configured settings.

```
usage: sqlrodbcd [options] [transport]
options:
 -help       = show usage (this list)
 -d          = debug mode
 -i          = called from inetd
 -l name     = log file name (or console/syslog/default)
 -f          = run in foreground
 -s name     = service name (tcp/ip transport)
 -F facility = syslog facility (USER/DAEMON/LOCAL0..LOCAL7)
 -I ident    = syslog identifier
transport:
 TCPIP       = TCP/IP sockets
```

The following options are available:

| Option | Description |
|---|---|
| `-help` | This option causes the server to show an overview about commandline usage. |
| `-d` | This option can be used to specify different `LogFlags`, for example to track down a problem.<br>Overrides the `LogFlags` settings. |
| `-i` | If the ODBC server is started by `inetd` this option must be present. |
| `-l` *name* | This option can be used to specify different `LogFile`, for example to track down a problem.<br>Overrides the `LogFile` settings. |

| Option | Description |
|---|---|
| `-f[f]` | The presence of this option causes the server to run in the foreground. Normally the server will run in background. |
| `-s` *service* | Overrides the `Service` settings. |
| `-F` *facility* | Overrides the `SysFacility` settings. |
| `-I` *facility* | Overrides the `SysIdent` settings. |

The only transport name supported is `TCPIP`.

Please refer to the server configuration section for more information.

### 2.3.4 Automatic Server Start

If run as a daemon, you can configure `sqlrodbcd` to start automatically using one of the methods below:

#### HP-UX 9.x

You can start the network server using `/etc/rc`. In this case modify your `/etc/rc` start script by adding the following lines either to the section `localrc` or at the end of the file:

```
/opt/sqlr/bin/sqlrodbcd
```

#### HP-UX 10.x

In order to startup/shutdown the **SQL/R** ODBC server automatically with the operating system, simply set `START_SQLRODBCD=1` in the **SQL/R** startup configuration file `/etc/rc.config.d/sqlr`.

In order to startup the **SQL/R** ODBC server manually, you can enter the following command:

```
/sbin/init.d/sqlr start
```

In order to shutdown the **SQL/R** ODBC server manually, you can enter the following command:

```
/sbin/init.d/sqlr stop
```

### 2.3.5 Server Start with inetd

You can start the `sqlrodbcd` server using the `inetd` daemon. In this case you must change the configuration file `/etc/inetd.conf` by adding the following line to `inetd.conf`:

```
sqlrodbc stream tcp nowait user /opt/sqlr/bin/sqlrodbcd sqlrodbcd -i
```

This statement consists of the following parts:

- *sqlrodbc* is the service name
- *stream* is the socket type
- *tcp* is the protocol name
- *nowait* is the recommended wait option
- *user* is the login name
- option `-i` represents the `inetd` option

Additional information is available in the HP-UX online help under `inetd.conf`(4).

**Please note:** If you configure to handle **SQL/R** ODBC connections, you cannot start the server manually. Both configuration are mutually exclusive.

## 2.4 Administration tasks

This section covers the following topics:

- User administration
- Data source administration

### 2.4.1 User administration

The **SQL/R** ODBC server provides its own authentification scheme. This makes it possible to allow data base access without the need to have a system account for each individual user. All users which are known to the server must be defined in the user configuration file `odbc.user`.

The default location of the file `odbc.user` depends on the revision of the HP-UX operating system:

**HP-UX 9.x** `/opt/sqlr/etc/odbc.user`
**HP-UX 10.x** `/etc/opt/sqlr/odbc.user`

The file name and location can be customized with the `userFile` configuration item in the server config file.

#### File format

The user configuration file `odbc.user` is text file and can be maintained with a text editor such as `vi`. Empty lines and lines starting with a hash character (`#`) are ignored.

The file is built of sections and configuration items. Each section begins with a title which is enclosed in square brackets (`[...]`). Configuration items are local to a section. Section names and configuration items are not case sensitive.

Each user has its own section. The user id is used as the section title.

| Entry | Description |
|---|---|
| `Name` | The full user name. This item is currently not used by the server. |
| `Password` | The user password. This password must be provided by the ODBC client along with the user name in order to be authorized as a legitimate user. |

#### Example

The following example defines two ODBC users. The user id “ sample ” with the password “ secret ” and the user “ demo ” without a password.

```
[sample]
Name = A sample ODBC user
Password = secret

[demo]
Name = Demo user
```

**Please note:** As unencrypted passwords are defined in the user configuration file, you may want to restrict the access to the ODBC server. You can chown it to the *sqlr* account and chmod it to 400 (`chmod a=,u=r odbc.user`). If the server is started as `root`, you can also chown it to root and chmod it to 400 (`chmod a=,u=r odbc.user`).

### 2.4.2 Data source administration

The **SQL/R** ODBC server operates on "data sources" instead of data bases. A data source consists of one or multiple data bases along with common information such as access rights and repository information.

The data sources known to the ODBC server are defined in the `odbc.sdn`.

The default location of the file `odbc.dsn` depends on the revision of the HP-UX operating system:

**HP-UX 9.x** `/opt/sqlr/etc/odbc.dsn`
**HP-UX 10.x** `/etc/opt/sqlr/odbc.dsn`

The file name and location can be customized with the `dsnFile` configuration item in the server config file.

#### File format

The data source configuration file `odbc.dsn` is text file and can be maintained with a text editor such as `vi`. Empty lines and lines starting with a hash character (#) are ignored.

The file is built of sections and configuration items. Each section begins with a title which is enclosed in square brackets (`[...]`). Configuration items are local to a section. Section names and configuration items are not case sensitive.

Each data source is a different section. The section name is not case sensitive. Besides that, the data source name on the the client must be written exactly as the one on the server.

| Entry | Description |
|---|---|
| `Database` | The path and name of the data base. |
| `Password` | The data base password. This is an optional item and may be omitted or empty if no password is required to access the data base.<br>This is a data base specific configuration item and is associated with the data base previously defined with theitem `Database`. |

| Entry | Description |
|---|---|
| `Alias` | Alternate name for the data base. This is an optional item and may be omitted.<br>This is a data base specific configuration item and is associated with the data base previously defined with the item `Database`.<br>When an alias name is defined, it will be provided to the ODBC client instead of the original data base name. The data base name (or alias) can be used by the client to qualify a table in a specific data base.<br>With an alias name you can provide a logical name instead of a data base name (for example "ACCOUNTS" instead of "SGLD25". Some clients (for example MS Access) also save some structural information of the data base locally. When you change the data base name later on (for example due to a new data base revision), some reports may fail. |
| `Repository` | The path and name of a repository file. Repository files are associated with the data source and are processed after all data bases have been opened successfully. Multiple repository files are possible. This is an optional item and may be omitted.<br>A repository file contains **SQL/R** statements such as `FIELD` and can be used to privide additional information about logical field contents and format. |
| `UserList` | List of comma separated user id. Only users included in this list are authorized to access the data source. The user id must have been defined in odbc.user. If a data source has no UserList entry it is not accessible by anyone. Multiple UserList entries are possible.<br>The user id "`public`" is predefined by the ODBC server and must not be defined in the user configuration file. When included in the `UserList`, the user id "public" can be used to provide anonymous access to database ressources. |

### Example

The following example defines the data sources "SQL/R Example". The data source contains one data base without a password. The alias name "SAMPLE" has been defined for the data base "DB". The data source has an associated repository file. Only the user id public and demo are allowed to access the data source.

```
[SQL/R Example]
Database   = /opt/sqlr/share/db.g/db
Password   =
Alias      = SAMPLE
Repository = /opt/sqlr/share/db.g/db.rep
UserList   = public,demo
```

The following example defines the data source “Accounts”. It consists of two data bases (SGL1, SGL2). Each data base has a password and an associated alias name.

```
[Accounts]
Database   = /path/to/db/sgl1
Password   = ALL
Alias      = GL1
Database   = /path/to/db/sgl2
Password   = ALL
Alias      = GL2
Repository = /path/to/file/gl1.rep
Repository = /path/to/file/gl2.rep
Repository = /path/to/file/gl.rep
UserList   = sample
UserList   = gl,stat
```

**Please note:** As unencrypted data base passwords are contained in the data source configuration file, you may want to restrict the access to the ODBC server. As unencrypted passwords are defined in the user configuration file, you may want to restrict the access to the ODBC server. You can chown it to the *sqlr* account and chmod it to 400 (`chmod a=,u=r odbc.dsn`). If the server is started as `root`, you can also chown it to root and chmod it to 400 (`chmod a=,u=r odbc.dsn`).

# 3

# Client installation and configuration

This chapter describes the installation of **SQL/R** ODBC on the client system (PC).

It covers the following topics:

- Installing **SQL/R** ODBC on the client
- Configuring the client system
- Administration of data sources

## 3.1 Installing SQL/R ODBC on the Client

The **SQL/R** ODBC client software is not copy protected and can be installed on any number of PCs within one company or institution.

**SQL/R** ODBC provides a 16 bit and 32 bit version of the driver. You should install the 16 bit driver on **Windows 3.x** and the 32 bit driver on **Windows 95** and **Windows NT**. If you use 16 bit application software on a 32 bit platform (for example MS Access 2.0 with Windows NT), it will make use of the 32 bit driver.

To install the **SQL/R** ODBC client perform the following steps:

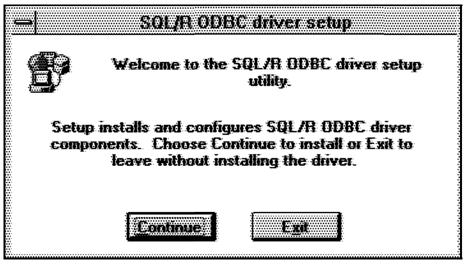
- Start WINDOWS. If WINDOWS is already active, stop all the current active WINDOWS applications.
- If you received the client software on a single floppy disk, there is a distinct subdirectory for the 16 bit and the 32 bit driver.

  To install the 16 bit version of the driver execute the `SETUP.EXE` program in the **SETUP16** directory on the floppy disk. To install the 32 bit driver execute the `SETUP.EXE` program in the **SETUP32** directory on the floppy disk.
- If you received the client software as a self extracting archive, execute the appropriate archive file.

  The 16 bit archive should be named like `ODBC16.EXE` and the 32 bit archive like `ODBC32.EXE`. After executing the archive file, a Windows dialog appears, which lets you choose a temporary directory where the files should be unpacked and if the setup program should be started automatically.

  Select the `Unzip` button to unpack the files from the archive and run the SETUP program. Select the `Close` button to cancel the installation.

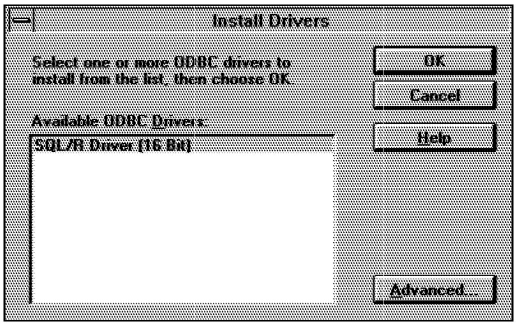
After short time the welcome dialog of the installation program appears on the screen.



odbca

Select the `Continue` push button to continue with the installation process.

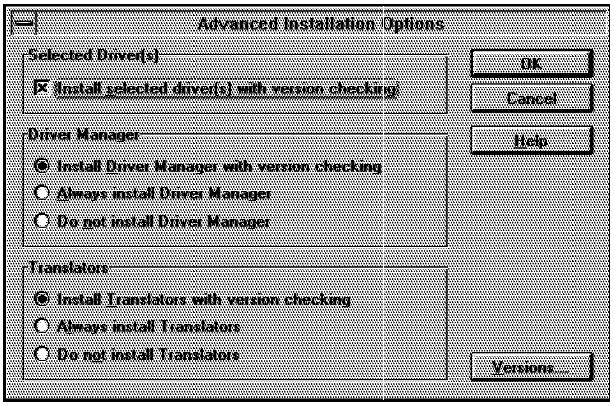
The following dialog box displays a list of available ODBC drivers which can be installed by the `SETUP` program.




odbcb

Please select the **SQL/R** driver and choose the button `OK` to continue the default installation process.

Custom installation options can be set by selecting the `Advanced...` push button. It is normally not necessary to change the default installation options.



odbcc

The following installation options are available:

- `Selected Driver(s)`

  If the check box is selected, version checking is turned on (the default). You are asked to confirm the installation of the **SQL/R** driver, if it has the same or earlier version number than the driver currently installed. If you are installing the **SQL/R** driver for the first time, this option has no effect. If the check box is cleared, version checking is turned off. The **SQL/R** driver is installed, regardless of its version number.

- `Driver Manager`

  These option buttons specify whether the ODBC Driver Manager is installed.

  - Install Driver Manager with version checking
    The ODBC Driver Manager will only be installed, if it is is the same or newer version than the existing Driver Manager.
  - Always install Driver Manager
    The Driver Manager is installed regardless of the version number
  - Do not install Driver Manager
    The Driver Manager is not installed even if it is the same as or newer than the existing Driver Manager.

- `Translators`

  The **SQL/R** driver does not support translators. You can ignore these options.

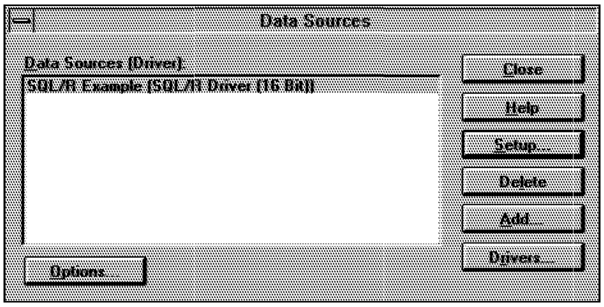
- `Versions`

  Choose this button to open a dialog that displays version information about the ODBC components.

Close the dialog by selecting the `OK` or `Cancel` push button. In the main window, please select the **SQL/R** driver and choose the button `OK` to continue the installation process.

After the files have been copied, the ODBC Driver Manager (a Microsoft ODBC component) dialog appears on the screen. It can be started again later by executing the ODBC Driver Manager from the Windows Control Panel.

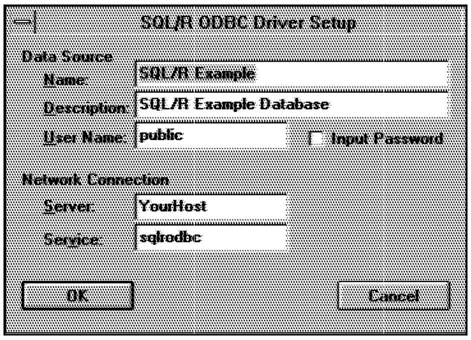
A full description regarding the administration of ODBC drivers and especially data sources can be found at the end of this chapter (section 3.4).




odbcd

Now you can define a new or change existing data source. The installation program installs an example data source named "**SQL/R** Example" which uses the **SQL/R** example database.

In order to use the example data source, you must complete its definition. Press the `Setup...` push button to open the **SQL/R** ODBC driver setup dialog.




odbce

The hardcopy shows the default definition of the "**SQL/R** Example" data source.

Please enter the server host name in the field "Server". Press the `OK` button to return to the previous dialog. Press the button `Close` to close the dialog and end the installation process.

## 3.2 Configuring the Client System

In order to open a TCP/IP network connection to the server system, you must define the server name in your `HOSTS` file and the service name `SERVICES` file.

The location of both files depends on your TCP/IP protocol stack. If you are using the Microsoft TCP/IP protocol stack, you can find these files at the following location:

| | |
|---|---|
| MS Windows 3.x, WfW 3.11 | `\WINDOWS` |
| MS Windows 95 | `\WINDOWS` |
| MS Windows NT | `\WINNT\DRIVERS\ETC` |

Please enter a line like the following into your `HOSTS` file:

```
192.168.1.1       server name
```

Replace the *IP address* and the *server name* by the appropriate values. Please contact your system administrator to obtain the server name and IP address.

Please enter a line like the following into your `SERVICES` file:

```
sqlrodbc          8003/tcp    # SQL/R ODBC
```

Please contact your system administrator to obtain the port number.

**Please be aware, that the port number must be the same on the client and on the server side.**

## 3.3 Driver options

Additional driver options can be configured in the file SQLRODBC.INI which is located in your Windows directory. **It is normally not necessary to change the default driver options.**

The SQLRODBC.INI file is not installed by the setup program. Instead a configuration template file (SQLRODBC.SAM) is placed in your Windows directory. You can either create a new SQLRODBC.INI file or copy the template file SQLRODBC.SAM to SQLRODBC.INI and modify it using an editor program (such as Notepad).

The configuration file SQLRODBC.INI consists of several sections. Each section begins with a section name which is enclosed in square brackets ([...]). Each section may contain configuration items. The configuration item and the associated value are separated by an equal sign (=). Comment lines start with a semicolon (;).

### 3.3.1 The Config section

| Entry | Meaning |
|---|---|
| Ctl3D | By default SQLRODBC uses the CTL3DV2 or CTL3D32 DLL to achieve a 3 dimensional appearance. You can disable the use of the CTL3D library by setting Ctl3D=0. The default is Ctl3D=1. |
| UseKeepAlive | If you want SQLRODBC to activate the TCP KEEPALIVE option to determine if the server is still alive and well, you can set UseKeepAlive=1. The default is UseKeepAlive=0. |

### 3.3.2 The Debug section

| Entry | Meaning |
|---|---|
| `LogFile` | When a `LogFile` is defined, the **SQL/R** ODBC driver logs debug information to it. The amount and type of log information depends on the `LogMode` (see below). If the `LogFile` is undefined, no logging will be done.<br>`LogFile` is undefined by default (no logging). |
| `LogMode` | LogMode consists of log flags which enables logging for specific execution paths and a logging level, which determines what (actually how much) is logged. If the `LogFile` is undefined, no logging will be done. The default is `LogFile` undefined (no logging) and `LogMode="*0"`.<br><br>Log Flags:<br>* Applies to all log flags<br>A API<br>N Network<br>P Protocol<br><br>Log Level:<br>0 Error Messages<br>1 Informational (“nice to know”)<br>2 Debug<br>3 Verbose debug |

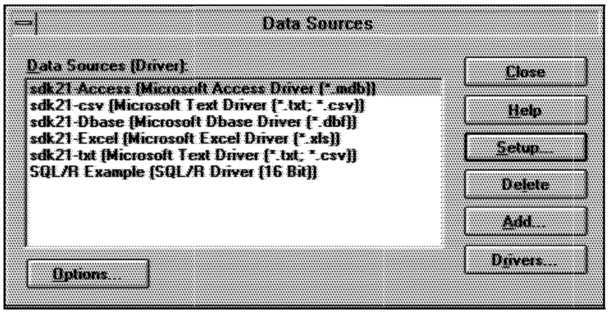
### 3.3.3 The driver configuration template file

```
; sqlrodbc.ini

[Config]

; By default SQLRODBC uses the CTL3DV2 or CTL3D32 DLL to achieve a
; 3 dimensional appearance. You can disable the use of the CTL3D
; library by setting Ctl3D=0. The default is Ctl3D=1.

; Ctl3D=1

; If you want SQLRODBC to activate the TCP KEEPALIVE option to
; determine if the server is still alive and well, you can set
; UseKeepAlive=1. The default is UseKeepAlive=0.

; UseKeepAlive=0


[Debug]

; LogMode consists of log flags which enables logging for specific
; execution paths and a logging level, which determines what
; (actually how many :-) is logged.
; If the LogFile is undefined, no logging will be done.
; The default is LogFile undefined (no logging) and LogMode="*0".
;
; Log Flags:
;  * - Applies to all log flags
;  A - API
;  N - Network
;  P - Protocol
;
; Log Level:
;  0 = Error Messages
;  1 = Informational ("nice to know")
;  2 = Debug
;  3 = Verbose debug

; LogMode="*0"
; LogFile=c:\odbc.log
```

## 3.4 Administration of Data Sources

The ODBC Driver Manager is a Microsoft ODBC component which is responsible for managing ODBC drivers and data sources. If an application wants to connect to an ODBC data source, it actually calls the Driver Manager to in order to access the data source.

The ODBC Driver Manager user interface can be started from the Control Panel to define the data sources you want to access.




adma

The Driver Manager main dialog displays the list of all defined data sources along with the corresponding driver.

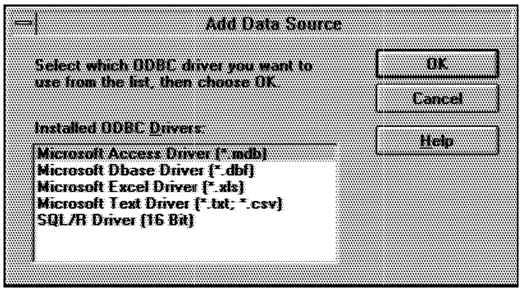
The following choices are available:

| | |
|---|---|
| `Close` | Exit the ODBC Driver Manager. |
| `Help` | Display the online help. |
| `Setup...` | Configure an existing data source. |
| `Delete` | Remove an existing data source. Select the the data source you want to delete from the list and press the `Delete` button. |
| `Add...` | Add a new data source. |
| `Drivers...` | Install or delete an ODBC driver. This can also be used to obtain version information about an ODBC driver. |
| `Options...` | Set Driver Manager options. |

### 3.4.1 Add a new Data Source

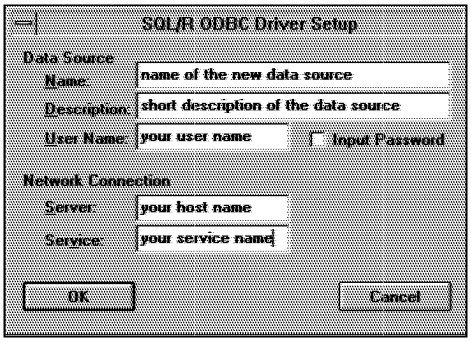
In order to add a new data source, select the `Add...` button in the main dialog. The Driver Manager opens a dialog with the list of available drivers.

Each ODBC driver has a unique setup dialog, so you must first select the driver you want to use.




admc

Please select the **SQL/R** ODBC driver from the list and press the `OK` button. Next, the **SQL/R** ODBC driver setup dialog is displayed.




admd

In order to define the new **SQL/R** data source you must provide the following information:

- `Data source name`

  Enter the name of the new data source. This name will be included into the list of installed data sources.

  **Please note, that this data source must also be configured on the server. The data source name must be written exactly the same on the client and the server**.

- `Description`

  Enter a short description of the data source.

- `User Name`

  Enter your username. This is required by **SQL/R** in order to authorize you to access the data source.

  Please note, that the ODBC user name may be different from your HP-UX login name. Please contact your system administrator to obtain your user name.

- `Server`

  Enter the host name (or IP number) of the server system. The server name must be defined in your `HOSTS` file.

- `Service`

  Enter the service name (or port number) which is used by the server. The default service name is `sqlrodbc`.

  The service name must be defined in your `SERVICES` file. The client and the server must use the same service/port.

- `Input Password`

  If you activate this option, the **SQL/R** ODBC driver will prompt you for a password when you open a connection to this data source.

  Please note, that the ODBC password may be different from your HP-UX login password. Please contact your system administrator to obtain your ODBC password.

Select the `OK` push button to save your input and return to the main dialog. The `Cancel` button discards your input returns you to the main dialog.

The new data source should now be displayed in the list of installed data sources.

### 3.4.2 Configure an existing Data Source

In order to change the properties of an installed data source, select the data source from the list and press the `Setup...` push button.

The driver-specific dialog box will be opened and you can configure the selected data source. Please refer to the previous section (3.4.1) for a description of the **SQL/R** ODBC driver setup dialog.

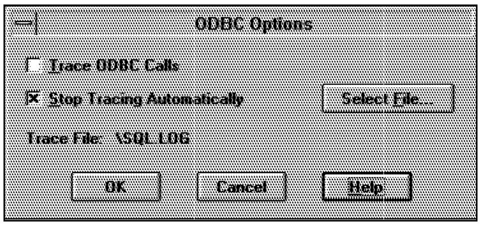
### 3.4.3 Drivers

If you select the `Drivers...` push button in the main dialog, a dialog box with a list of all installed ODBC drivers appears.

You can now either install a new driver, remove an existing driver or display version information about an installed ODBC driver. Please refer to online help for more information.

### 3.4.4 ODBC Options

If you select the `Options...` button in the main dialog, the "ODBC Options" dialog appears.




admb

This dialog can be used to enable tracing of all ODBC calls by the Driver Manager. Please refer to online help for more information.

4

# Driver Reference

This chapter contains reference information about the **SQL/R** ODBC driver and the **SQL/R** ODBC server.

It covers the following topics:

- The list of installed files
- The Repository file

## 4.1 List of Files

### 4.1.1 List of Files on the Server

This section includes a list of all files, which are installed on the Server (HP 9000) respectively relevant to the **SQL/R** ODBC software.

| File/ Directory | Default Directory | Description |
|---|---|---|
| sqlrodbcd | /opt/sqlr/bin | **SQL/R** ODBC server executable |
| odbc.cfg | /opt/sqlr/etc | ODBC server configuration file |
| odbc.dsn | /opt/sqlr/etc | Data source definition |
| odbc.user | /opt/sqlr/etc | User authorization |

Default configuration files are located in the directory /opt/sqlr/newconfig/etc.

### 4.1.2 List of Files on the Client

This section includes a list of all files, which are related to the **SQL/R** ODBC software. The location of the files depends on the version of MS Windows.

**Table 1: Windows 3.x, WfW 3.11, Windows 95**

| Files/ Directory | Default Directory | Description |
|---|---|---|
| SQLRO16.DLL | \WINDOWS\SYSTEM | **SQL/R** ODBC 16 bit DLL |
| SQLRO32.DLL | \WINDOWS\SYSTEM | **SQL/R** ODBC 32 bit DLL |
| SQLRODBC.INI | \WINDOWS | Driver configuration file |
| SQLRODBC.SAM | \WINDOWS | Driver configuration template file |

**Table 2: Windows NT**

| Files/ Directory | Default Directory | Description |
|---|---|---|
| SQLRO16.DLL | \WINNT\SYSTEM | **SQL/R** ODBC 16 bit DLL |
| SQLRO32.DLL | \WINNT\SYSTEM32 | **SQL/R** ODBC 32 bit DLL |
| SQLRODBC.INI | \WINNT | Driver configuration file |
| SQLRODBC.SAM | \WINNT | Driver configuration template file |

## 4.2 The Repository file

A repository file is a text files containing **SQL/R** ODBC statements. **SQL/R** ODBC uses repository files to handle persistent dictionary information which are beyond the concept of the Image data base.

Repository files are associated with a data source and are processed each time the data source is connected to after all data bases have been opened successfully.

The initial **SQL/R** ODBC release only supports the use of `FIELD` statements in a repository files. This will be extended in a subsequent release.

**SQL/R** repository files are intended to define

- field formats (`FIELD`)
- alternate field names (“ field alias ”)
- named formulas ("virtual fields")
- views (`CREATE VIEW`)
- access rights (`GRANT/REVOKE`)

This section specifies a *subset* of the **SQL/R** ODBC language. Please refer to the ODBC Reference for a complete reference.

### 4.2.1 Definition of Terms

**Field (or Item)**

A field is the smallest logical unit of a database. Its contents are not limited to a word or a numeric value, but can consist of several words.

For example: The field Street may hold the value "Martin Luther King Boulevard"

Fields are the equivalent to a single column of a table.

**Array**

An array is a group of fields of the same type (also called elements) that can be referenced with the same name and an index.

**SQL/R** ODBC handles each array element as a distinct field. The field name is build from the base name by appending two underline characters (__) and the index (starting with 1).

For example, when there are 12 values for monthly budget, the month of May is budget__5 and the month of January is budget__1.

**Record (or Entry)**

A record is a collection of fields. Each field in a record has a unique name. Records are stored in tables. Records are the equivalent to a single row of a table.

**Table (or Dataset)**

A table is a collection of records, arranged in columnar form.

**View**

A view is a virtual table consting of selected fields from one or more tables. In the simplest case, a view is a single table. The `CREATE VIEW` statement is used to create a permanent view. The `SELECT` statement defines a temporary view.

**Correlation Name**

A Correlation name is an alias name or pseudonym of a table. It is used to simplify a table reference or to define a distinct table instance.

**Instance**

In cases where a single table is referenced several times in one view (for example a self join), there are multiple instances of the same data. Instances are separated by correlation names.

For example:

```
SELECT A.PartNo, A.PartDesc, B.PartNo, B.PartDesc
FROM Parts A, Parts B
WHERE B.PartNo = A.PartReference;
```

This defines a temporary view containing two different records of the table Parts. They are identified by the correlation name A and B. A and B are instances of the table Parts.

**Alias**

An alias is a pseudonym (alternate name) for a database field and is defined using the `FIELD` statement.

### 4.2.2 Reserved Words

Reserved words are **SQL/R** predefined words with a special meaning. These words are not case sensitive.

| | | | |
|---|---|---|---|
| ABS | DAYOFYEAR | LEFT | REVOKE |
| ACOS | DEFAULT | LENGTH | RIGHT |
| ADD | DELETE | LIKE | RTRIM |
| ALL | DESC | LOCATE | SECOND |
| ALTER | DESCRIBE | LOG | SELECT |
| AND | DISTINCT | LTRIM | SET |
| ANY | DOUBLE | MAX | SIGN |
| AS | DROP | MICROSOFT | SIN |
| ASC | ESCAPE | MIN | SMALLINT |
| ASCII | EXISTS | MINUTE | SQRT |
| ASIN | EXP | MOD | SUBSTRING |
| ATAN | FIELD | MONTH | SUM |
| ATAN2 | FLOAT | NOT | TABLE |
| AVG | FLOOR | NOW | TAN |
| BETWEEN | FN | NULL | TIME |
| BY | FOR | ODBC | TIMESTAMP |
| CASCADE | FOREIGN | OF | TO |
| CEIL | FROM | OJ | UCASE |
| CHAR | GRANT | ON | UNION |
| CHECK | GROUP | OPTION | UNIQUE |
| CONCAT | HAVING | OR | UPDATE |
| CONVERT | HOUR | ORDER | USER |
| COS | IFNULL | OUTER | VALUES |
| COT | IN | PI | VENDOR |
| COUNT | INDEX | PRIMARY | VIEW |
| CREATE | INSERT | PRODUCT | WEEK |
| CURDATE | INTEGER | PUBLIC | WHERE |
| CURRENT | INT | QUARTER | WITH |
| CURTIME | INTO | RAND | XOR |
| DATABASE | IS | REFERENCES | YEAR |
| DATE | JOIN | REPEAT | |
| DAYOFMONTH | KEY | REPLACE | |
| DAYOFWEEK | LCASE | RESTRICT | |

### 4.2.3 Syntax overview

The Repository file is text file and can be maintained with a text editor such as `vi`.

The following rules apply:

- Empty lines and lines starting with a hash character (#) are ignored.
- A statement can span multiple lines and is terminated by a semicolon.
- The semicolon **must** be the last (non space) character in the line.
- Keywords and field names are case insensitive.

### 4.2.4 Constant values

**SQL/R** supports the following constant values:

- **Quoted string**
  String values must be enclosed in either single or double quotes. The string must end with the same quote type. To place a quote in a quoted string, prepend it with a backslash (\).

  For example:

  ```
  "String"
  'String with \' quote'
  ```

- **Numeric value**
  Integer and floating point values are supported. Any value which is either outside the integer domain ($2^{31} - 1$) or contains either a decimal point or an exponent is converted to a floating point value.

  For example:

  ```
  123
  123.45
  3.1415E10
  ```

- **Date value**
  **SQL/R** ODBC supports the Microsoft ODBC extensions. A date constant value must have the form of:

```
{D "yyyy-mm-dd"}
```

Where *yyyy* is a four digit (or two digit, relative to 1900) year, *mm* is a one or two digit month of the year and *dd* is a one or two digit day of the month.

For example:

```
{D "1996-08-07"}
```

- **Time value**
  **SQL/R** ODBC supports the Microsoft ODBC extensions. A time constant value must have the form of:

  ```
  {T "hh:mm:ss"}
  ```

  Where *hh* is a one or two digit hour value, *mm* is a one or two digit minute and *dd* is a one or two digit second.

  For example:

  ```
  {T "14:30:50"}
  ```

- **Timestamp value**
  **SQL/R** ODBC supports the Microsoft ODBC extensions. A timestamp constant value must have the form of:

  ```
  {TS "yyyy-mm-dd hh:mm:ss"}
  ```

  For example:

  ```
  {TS "1996-08-07 14:30:50"}
  ```

All date, time and timestamp values are currently converted internally into seconds.

### 4.2.5 Identifers

An identifier consists of alphabetic, numeric, or underline (_) characters. An identifier must always start with an alphabetic character.

Identifiers are not case sensitive. Therefore the identifiers “Name”, “NAME”, “name” are treated the same.

Identifiers can be used for all expressions such as table and fields names. The only limitation is that no **SQL/R** reserved words are permitted.

Identifiers consisting of a reserved word must be *quoted* by enclosing the identifier between vertical bars (|). For example "|time|" is recognized as an identifier and not a reserved word.

**SQL/R** ODBC handles each array element as a distinct field. The field name is build from the base name by appending two underline characters (_) and the index (starting with 1). For example when there are 12 values for monthly budget, the month of May is budget__5 and the month of January is budget__1.

### 4.2.6 The FIELD Statement

The field statement can be used to define field specific properties like field format or conversion operations.

`FIELD` [ *qual*`.` ] *table*`.`*field*
`CONVERT TO` *data type*
[ `DESCRIBE AS` "*comment*" ] `;`

Where *qual* is an optional qualifier (such as the data base (alias) name), *table* is a table name, *field* is a field name. *Comment* is a quoted string containing a field description.

**Please note:** The field name **must** always be qualified (for example customers.number). Unqualified field names are reserved for use by "virtual fields". The field description is currently unused.

*Data type* is one of the following:

| | |
|---|---|
| `CHAR` [ `(`*wdth*`)` ] | String |
| `SMALLINT` [ `(`*wdth*`)` ] | Integer |
| `INTEGER` [ `(`*wdth*`)` ] | Integer |
| `FLOAT` [ `(`*wdth* [`,`*prec*]`)` ] | Floating Point number |
| `DOUBLE` [ `(`*wdth* [`,`*prec*]`)` ] | Floating Point number |
| `DATE` [ `(`"*fmt*" [`,`*year*]`)` ] | Convert a numeric value into a Date value. |
| `TIME` [ `(`"*fmt*"`)` ] | Convert a numeric value into a Time value. |
| `TIMESTAMP` | Convert a numeric value into a Timestamp value |

The optional *wdth* argument specifies a different column width. The optional *prec* argument specifies a different precision (currently unused). The *fmt* argument specifies a date or time conversion format. The optional *year* argument specifies a base year.

Please refer to the ODBC reference for more information on data types.

Date conversion formats are:

| | |
|---|---|
| `"GREG"` | Convert serial date. If the optional *year* is not present, 1972 will be assumed. |
| `"YMD"` | Convert date from [YY]YYMMDD format. |

Time conversion formats are:

| | |
|---|---|
| `"HM"` | Convert from HHMM |
| `"HMS"` | Convert from HHMMSS |

### For example

```
FIELD Orders.Orderno CONVERT TO CHAR(8);
FIELD Orders.Orderdate CONVERT TO DATE("YMD");
FIELD Lineitem.Delivdate CONVERT TO DATE("YMD");
```

# A

# HP Eloquence Format Numbers

The HP Eloquence format numbers are defined for a database by either the `schema` or the `dbmods` utility. These numbers are then used by HP Eloquence QUERY to evaluate and format data. When **SQL/R** opens a database it translates these numbers to the corresponding format.

The HP Eloquence format numbers are *cumulative* values. For each group or attribute, a code value is added.

| Group | Value | Comments |
|---|---|---|
| **Query Write inhibit** | | |
| No write inhibit (default) | 0 | (ignored) |
| Write inhibit | 1 | (ignored) |
| **Item type** | | |
| Date type | 2 | `DATE`("GREG",1972) |
| Currency | 4 | (ignored) |
| Undefined | 6 | (ignored) |
| **Spacing** | | |
| Default | 0 | (ignored) |
| Comma every 3 digits | 8 | (ignored) |
| **Post decimals** | | |
| Default | 0 | (ignored) |
| FIXED 0 | 16 | (1) |
| FIXED 1 | 32 | (1.0) |
| FIXED 2 | 48 | (1.00) |
| FIXED 4 | 80 | (1.000) |
| FIXED 3 | 64 | (1.0000) |
| FIXED 5 | 96 | (1.00000) |
| FIXED 6 | 112 | (1.000000) |

For the item types `MONEY` and `DATE`, all further entries are ignored.

The number of decimal places (post decimals) are recognized for floating point decimal data types (float, double) only.

For example:

Date = 2
Money = 4
Value with 2 decimal places = 48

# Index