THOROUGHBRED®
SOFTWARE INTERNATIONAL, INC.

TS DATASERVER™ FOR ORACLE

# Reference Manual

**Version 8.9.1**

**Thoroughbred Software International, Inc.**

7 Cedar Grove Lane, Suite 30 • Somerset, NJ 08873

115 N University Drive, Suite C • Nacogdoches, TX 75961

Telephone: 732-560-1377 • 800-524-0430 • **www.tbred.com**

Published by:
Thoroughbred Software International, Inc.
7 Cedar Grove Lane, Suite 30
Somerset, New Jersey 08873

Copyright © 2026 by Thoroughbred Software International, Inc.

All rights reserved. No part of the contents of this document
may be reproduced or transmitted in any form or by any means
without the written permission of the publisher.

Document Number: TR8.9.1M901

The Thoroughbred logo, Swash logo, and Solution-IV Accounting logo, OPENWORKSHOP, THOROUGHBRED, VIP FOR DICTIONARY-IV, VIP, VIPImage, DICTIONARY-IV, and SOLUTION-IV are registered trademarks of Thoroughbred Software International, Inc.

Thoroughbred Basic, TS Environment, T-WEB, Script-IV, Report-IV, Query-IV, Source-IV, TS Network DataServer, TS ODBC DataServer, TS ODBC R/W DataServer, TS DataServer for Oracle, TS XML DataServer, TS DataServer for MySQL,TS DataServer for MariaDB, TS DataServer for MS SQL Server, GWW Gateway for Windows, Report-IV to PDF, TS ReportServer, TS WebServer, TbredComm, Thoroughbred Web Terminal, WorkStation Manager, FormsCreator, T-RemoteControl, Solution-IV Accounting, Solution-IV Reprographics, Solution-IV ezRepro, Solution-IV RTS, and DataSafeGuard are trademarks of Thoroughbred Software International, Inc.

Other names, products and services mentioned are the trademarks or registered trademarks of their respective vendors or organizations.

# INTRODUCTION

TS DataServer for Oracle is an integral part of the Thoroughbred Software network solution. With TS Environment, server data can be made available across platforms for use in Dictionary-IV, Report-IV, Query-IV, Script-IV, and Thoroughbred Basic applications.

*Operating System Support:* UNIX, Linux, and Windows
For specific information, please contact your Thoroughbred Sales Representative.

This guide first describes the Client side of TS DataServer for Oracle, which is provided with every TS Environment installation. It then provides specific information for accessing Oracle servers.

Your applications may require special privileges to access the database. You should consider the following issues:

- Setting privileges to access the database.
- Designing or modifying site procedures.
- Setting procedures for server transaction processing.

## Requirements

To run TS DataServer for Oracle 10, tsoracle10, your system must meet the following requirements:

- Network communications implemented with TCP/IP protocol
- TS Environment 8.5.2+
- Oracle version 10 or higher

To run TS DataServer for Oracle 11, tsoracle11, your system must meet the following requirements:

- Network communications implemented with TCP/IP protocol
- TS Environment 8.5.2+
- Oracle version 11 or higher

To run TS DataServer for Oracle 19, tsoracle19, your system must meet the following requirements:

- Network communications implemented with TCP/IP protocol
- TS Environment 8.8.3+
- Oracle version 19 or higher

Once the installation is complete. Server access is enabled by the addition of a number of configuration items. For more information see SERVER.MAP and IPLINPUT.

## TS DataServer for Oracle Components

The TS DataServer for Oracle consists of three components:

- A Client enabled TS Environment,
- A Startup Program, and


Copyright © 2026 Thoroughbred Software International, Inc.

- A Server Process (tsoracle##) for each connection.

### *Client Enabled TS Environment*

The client side is enabled in the standard TS Environment. When configured for data server access, it detects requests for data that reside on the server. When this event occurs, it sends the I/O request to the server for processing. The server responds with the result set and processing continues.

### *Startup Program*

The Startup Program (tsoracle##) creates Server Processes for clients as needed. When the Startup Program detects a client attempting to connect, it tries to create a Server Process for the client. If a Server Process is created successfully, the client and server can perform client/server communications.

**NOTE:** A server system must have the Startup Program running before clients attempt to connect to it.

### *Server Process*

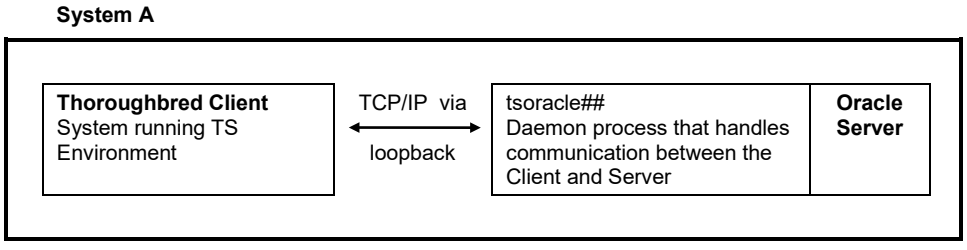
The Server Process (a child tsoracle##) is triggered by requests from the client. Once a request is received, it is processed and the results returned to the client.

## System Architecture

The following diagrams outline the core options for configuring the components of the Thoroughbred client.

- Client and server on one system
- Client on one system - server on another

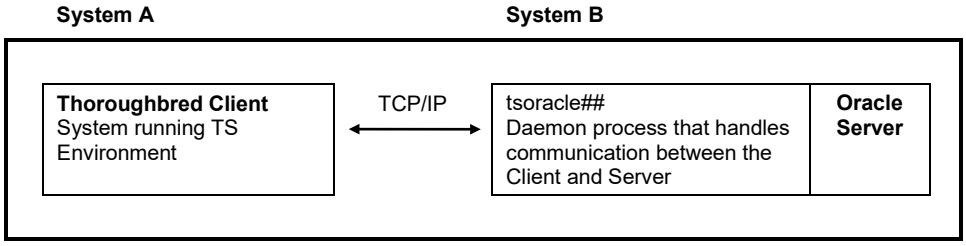
*Client and Server Co-exist on One System*




In the layout described by the above figure, the information is sent to the Server Process through the loopback TCP/IP address.


Copyright © 2026 Thoroughbred Software International, Inc.

*Client on System A and Server on System B*




In the above figure the information is sent over the network to the server system. Once the requests reach the Server Process it creates SQL statements that are interpreted by the server system. The results are then passed back to the client through the Server Process.


Copyright © 2026 Thoroughbred Software International, Inc.

# CLIENT AND SERVER ENVIRONMENT

To set up TS DataServer for Oracle software the following files must be created or modified:

- SERVER.MAP
- IPLINPUT
- tsor.ini

## SERVER.MAP

The SERVER.MAP text file establishes a relationship between a 2-character server ID used by Thoroughbred and the server system. This file is a simple text file. The system searches the current directory first. If the file does not exist there, the system defaults to the **/usr/lib/basic** directory.

**NOTE:** For UNIX Clients, this file should be placed in the **/usr/lib/basic** directory.

Thoroughbred uses the Server ID to reference the server system. Each line in the file represents an entry for one Server.

The syntax for an entry is:

```
server-id:[host-name or TCP/IP address]:TCP/IP port :database
```

server-id — The server ID is a 2-character alphanumeric string used within Basic to reference a particular server system.

TCP/IP address — The TCP/IP address is either the TCP/IP address or the host name for the server system. For installations where the Thoroughbred client and database server environments reside on the same physical system, this field can contain the loopback address or host name.

TCP/IP port — The TCP/IP port number is an optional parameter. It is used to override the default port number (5673), in the event of a conflict with another process using TCP/IP. The transport protocol layer of the TCP/IP process uses the port number to deliver the packet data to the appropriate application.

On UNIX systems the **/etc/services** file lists the port numbers used on a system. (This list may be a subset of the actual port numbers used on the system.) In the event of a conflict on a particular system add the port number to the SERVER.MAP file and execute the server process using the port number as an argument. For more information see Server Process in the following section.

database — This is the name of the database where Basic will access tables.

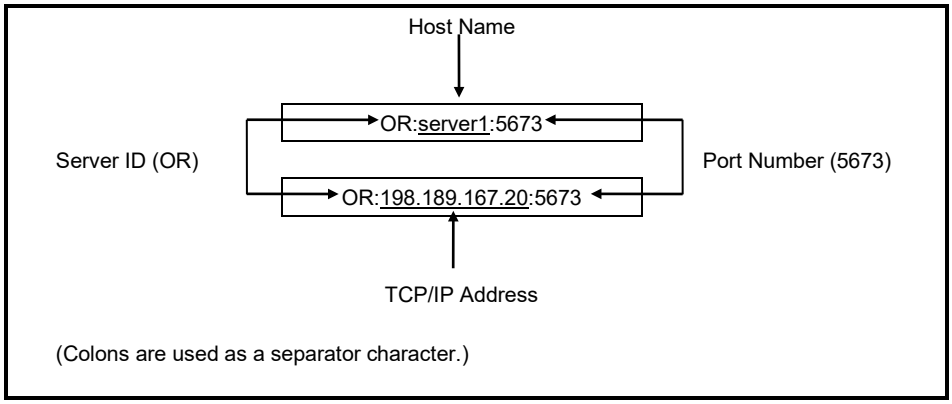
This parameter is optional only if connection identifiers are being used on the DEV line in an IPL file.


Copyright © 2026 Thoroughbred Software International, Inc.

*Example*

The fields in a SERVER.MAP entry are separated by a colon ( : ).The following are examples of valid SERVER.MAP entries:

*Server ID or TCP/IP Address*




# IPLINPUT

Once you establish the server ID you can make the required entries for the IPLINPUT file.

## DEV

Each DEV line corresponds to a directory on the Server. The syntax for a DEV line is:

```
DEV device-name, 1,,server-flag,,logon-cache,case-sensitivity,server-
id:log/pswd[@database]
```

| | |
|---|---|
| device-name | is device ID used to reference logical disk directories. Valid values range from D0 through D9, DA through DZ, and Da through Dz. |
| server-flag | indicates that this will be a client to the configured server system. The value for the Oracle server is 1. |
| logon-cache | configures a value for the maximum number of active logins to the Oracle database. For more information see Logon Cache in the Performance Tuning section. |
| case-sensitivity | indicates whether or not to perform a case sensitive comparison of the key value returned from the SQL data server during a READ, KEY= operation. The default is to perform a case sensitive comparison (blank or 0). To have Basic perform a case insensitive comparison, set this flag to 1. |
| server-id | is the two-character alphanumeric code assigned to this server. The server ID on the DEV line must match an entry in the SERVER.MAP file. |
| log/pswd | is the Oracle login or password used to access the database. The id and password are optional. If omitted, the system prompts for the information when the client is executed. |


Copyright © 2026 Thoroughbred Software International, Inc.

database is a connection identifier for a database. The server ID on the DEV line must match an entry in the SERVER.MAP file.

### PRM

If multiple Oracle servers are configured and the login/password are the same for each, the PRM ORACLE-LOGIN may be used as a shortcut to defining the login/password on each DEV line.

**PRM ORACLE-LOGIN=login/password**

login/password is the Oracle login and password used to connect to the database. Only the login and slash ( / ) character are required. A prompt for the password will appear when the client connects.

**PRM CREATWDBATTR**

This parameter will create new tables using the attributes from the system dictionary. Mandatory fields (Entry Type 2 and 3 and the first key field) will be created with a NOT NULL constraint. In addition any fixed length fields (Entry Type 1 or 3) will be created with the CHAR data type so the field always contains data.

**PRM ORA_NVLNULLS**
**PRM ORA_DONTWRITENULLS**

These parameters control how the system orders records when the key field contains nulls. For more information see Null Processing.

**PRM SQL_NO_SEPARATORS**

This parameter controls how short data values are sent to/from the SQL DataServer. If enabled, a short data value will be sent with trailing spaces. If disabled, a short data value will be sent with a trailing field separator, $8A$.

**PRM SQL_NUMERIC_NULLS**

This parameter controls how numeric null values are stored in the database. When the data for a numeric field is sent over from a Thoroughbred Basic application as all spaces, if enabled, then this data will be written to the database as a null value, otherwise, it will be written to the database as a 0 value.

**PRM UNIQUE-KEYS**

This parameter is used in the client Basic IPLINPUT file. Treats all secondary sorts unique by adding the primary sort to the end of all the secondary sorts when the various READ statements are translated to their equivalent SELECT statements on the server.

## tsor.ini

The tsor.ini contains control parameters for the tsoracle## server process. The tsmysql.ini file is located in the /usr/lib/basic directory.

**CLIENT-SYSTEM-INFO**


Copyright © 2026 Thoroughbred Software International, Inc.

This flag will enable process and host information to be logged. For more information see Using the Debugging Facility.

**COMMIT-COUNT=#**

This entry is used to specify the commit count. The maximum commit count is 65535. If the specified commit count is greater than the maximum, the maximum value will be used. The default commit count is 1.

The commit count can be overridden by SET CTC in Basic.

**DEBUG-LEVEL=#**

This entry is used to specify the debugging level. For more information see Using the Debugging Facility.

**DEBUG-FLAGS=#**

This entry is used to enable or disable additional information that can be written into the debug log file. The Flag values are:

| Flag Number | Description |
|---|---|
| 1 | Disable logging of the process ID number. |
| 2 | Enable logging of a time stamp. |
| 4 | Enable logging of parameter values. |
| 8 | Enable logging of fetched values. |
| 16 | Enable logging of I/O to/from the SQL DataServer. |
| 32 | Enable logging of connection information for the SQL DataServer. |

**DEBUG-WIDTH=#**

The entry is used to specify the width of the output in the /usr/lib/basic/tsmysql.debug file. The default width is 80.

**LDA-CACHE=#**

The entry is used to specify the logon cache count. For more information see Logon Cache.

**NO-OPENLOCK**

The flag will disable the use of the TS_TABLE_LOCK table.

**UNIQUE-SORTS**

This flag will enable tables to be created with unique secondary keys.

Starting with release level 8.9.0, a *tsor.ini.dflt* file will be included in the master file. All control parameters in this *tsor.ini.dflt* file are disabled.


Copyright © 2026 Thoroughbred Software International, Inc.

## Setup Example

This environment consists of two Oracle Servers on two different host systems.

Existing server-name TCP/IP address information:

```
acct:125.32.1.1
dev:125.32.1.10
```

The SERVER.MAP file may be set up in one of the following ways:

- Using host names
  O1:acct
  O2:dev

- Using TCP/IP addresses
  O1:123.31.1.1
  O2:125.32.1.10

- Using a combination of host names and TCP/IP addresses
  O1:125.32.1.1
  O2:dev

In this example the login information to the Oracle database is as follows:

acct: login id = mary
password = mjs

dev: login id = kurt
password = ktc

The IPLINPUT should reflect the following:

```
DEV D8,1,,1,,,,O1:mary/mjs

DEV D9,1,,1,,,,O2:kurt/ktc
```

After Basic is executed, there will now be a child tsoracle## process on each of the server systems (acct and dev).


Copyright © 2026 Thoroughbred Software International, Inc.

## Start the Server Process (tsoracle##)

The tsoracle## is a system daemon that is installed on the server system. Each time a Basic client executes, a child tsoracle## process is started. It should be added to the appropriate startup procedure, but it must be executed subsequent to the execution of the database server program. Verify that the account that executes the startup procedure has set the ORACLE_HOME and ORACLE_SID shell values.

The user that starts tsoracle## must include the following in the autostart file:

**HP UX (.profile):**

```
SHLIB_PATH=$ORACLE_HOME/lib
or
SHLIB_PATH=SHLIB_PATH:$ORACLE_HOME/lib
```

**Linux (.bash_profile):**

```
LD_LIBRARY_PATH=$ORACLE_HOME/lib
or
LD_LIBRARY_PATH=LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

**Other Operating Systems:**

Consult with your Oracle DBA for the correct shell variable.

This program is executed with the following syntax:

```
tsoracle## [-dx][port-num][-s]
```

-dx is the debugging level. For more information see Using the Debugging Facility.

port-num is the TCP/IP port number for the process.

-s causes process and host information to be logged. For more information see Using the Debugging Facility.

The tsoracle## process requires **/usr/lib/basic** to exist on the system where it is executed. Once it is started, execute Thoroughbred Basic using the IPLINPUT file that contains the reference to the Server IDs.


Copyright © 2026 Thoroughbred Software International, Inc.

# Start the Server Process (tsoracle##) as a Service

## Installing the TS DataServer for Oracle Service

To install the TS DataServer for Oracle service, the *loadservice.sh* script will copy service file from the TSORACLE## directory to the appropriate system directory or the service files can be loaded manually.

To install the TS DataServer for Oracle service via the script, run the following as root:

```
# cd /usr/lib/basic/TSORACLE##
# ./loadservice.sh
```

To manually install the TS DataServer for Oracle service:

```
# cp /usr/lib/basic/TSORACLE##/tsoracle.systemd /etc/systemd/system/tsoracle.service
# systemctl daemon-reload
```

The service file defaults to running the TS DataServer for Oracle from the /usr/lib/basic directory, listening on port 5673. If the TS DataServer for Oracle has been installed into a different directory or needs to listen on a different port, change the tsoracle.systemd file before installing the service.

These service files are available starting with release level 8.9.0.

## Managing the Service

To start the TS DataServer for Oracle service:

```
# systemctl start tsoracle
```

To have the TS DataServer for Oracle service start automatically when the system boots:

```
# systemctl enable tsoracle
```

To stop the TS DataServer for Oracle service:

```
# systemctl stop tsoracle
```

To prevent the TS DataServer for Oracle service from starting automatically when the system boots:

```
# systemctl disable tsoracle
```


Copyright © 2026 Thoroughbred Software International, Inc.

# DICTIONARY-IV FORMATS AND LINKS

You must have a Format and Link defined to access an Oracle Server table.

## Format Definition

Although no changes are required to existing formats in order to use the client capabilities of the environment, you should be aware of the following information.

**NOTE:** The data element names must not be Oracle reserved words. For more information see the appropriate Oracle documentation.

### *File Type*

When the Oracle tale is created, the field types are translated as follows:

```
Alphanumeric:  VARCHAR2 or CHAR
Numeric:       NUMBER
DATE:          DATE
```

The VARCHAR2 data types and CHAR allow only the characters above hex 20 to be included in the data. Any character below hex 20 is converted to a space.

The CHAR type will be used when the Basic PRM CREATWDBATTR is configured and the data element is set as a fixed length field either mandatory or optional (Entry Type 1 or 3).

When records are written to this table, any field of the CHAR type will be assigned a value of at least one space. This will ensure that the column is not null. For more information see PRM CREATWDBATTR in the IPLINPUT topic.

### *Entry Type*

All mandatory fields (Entry Type 2 and 3) will be set as NOT NULL fields in the Oracle tables when the Basic PRM CREATWDBATTR is set.

### *Data Elements – Naming*

Oracle only allows alphanumeric characters, the currency symbol ($), an underscore ( _ ), and a pound sign (#) in column names. The first character of the column must start with a letter. Any deviation from this convention results in table creation failure.

tsoracle## program translates the hyphen ( - ) that is acceptable in Thoroughbred Dictionary-IV Formats to an underscore ( _ ).

The use of Oracle reserved words (DATE, NUMBER, etc.) also results in table creation failure. For more information see your appropriate Oracle reference material.


Copyright © 2026 Thoroughbred Software International, Inc.

### Data Elements - Maximum Number

Although Thoroughbred Dictionary-IV supports the definition of 255 data elements, Oracle enforces a maximum number of 254 columns (one column corresponds to one data element).

Oracle does not support multiple occurrences for a single data element. For more information Multiple Occurrences in the Dictionary-IV Developer Guide.

The system creates separate columns in the Oracle table from a multiple occurrence field. The following is the naming convention for multiple occurrences.

**`data-element-name_seq#`**

seq# is the range from one to the number of defined occurrences. The total number of data element names plus each occurrence must not exceed 254. This is important to remember when moving an existing file into the server environment.

### Binary Numeric Fields

Fields defined as binary in the Format are converted to decimal values and stored in ORACLE NUMBER fields. The type of binary field determines the length of the field.

The following table displays the formulas for calculating the field width in the Oracle table.

*Formulas for Calculating Field Widths*

| Numeric Type | Field Width | Formula for Oracle Width<br>L = Field Width |
|---|---|---|
| 3,5 | 1 - 3 | (L*2) + 1 + 1 |
| | 4 - 5 | (L + 1)*2 = 1 |
| | 6 - 8 | (L + 1)*2 + 1 + 1 |
| 4 | 1 - 2 | (L*2) + 1 |
| | 3 - 4 | (L + 1)*2 |
| | 5 - 7 | (L + 1)*2 + 1 |
| | 8 | (L + 2)*2 |
| 6 | All | (L*2) + 1 |
| 7 | 10 | 16 |
| | Odd Prec. | (((L*10) - 15)*2)/10 |
| | Even Prec> | (((L*10) - 5)*2)/10 - 1 |
| 8 | All | 8 |
| 9 | All | 14 |
| A | All | (L*2) - 1 + 1 |
| B | All | (L*2) - 1 |
| C | All | (L*2) |
| D | All | L + 1 |
| E,F | All | L |

In all cases: if the value contains a decimal, add one to the field width.


Copyright © 2026 Thoroughbred Software International, Inc.

### *SQL Dates*

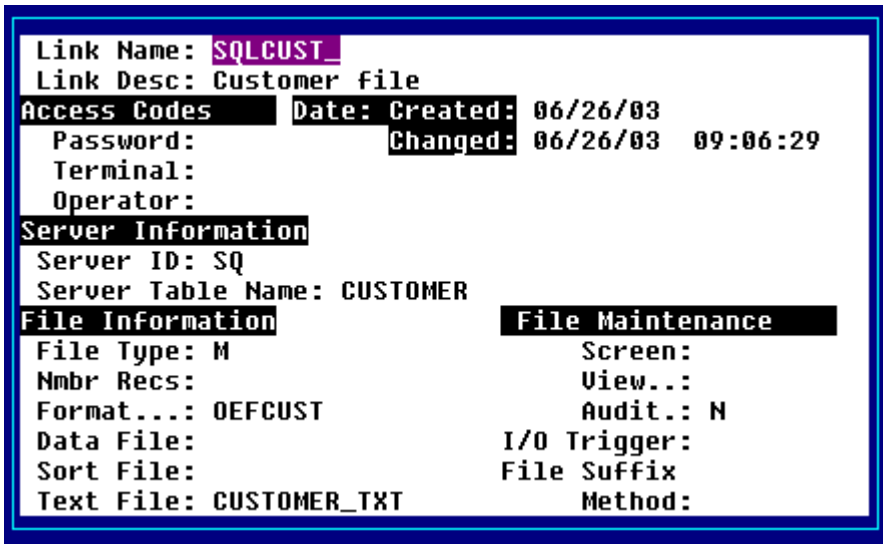
The system automatically translates dates stored in either of the SQL formats into the standard Oracle date storage format as you update records.

## Link Definition

The Link definition contains the configuration parameters necessary to describe a server table to Dictionary-IV. To access the Oracle Server you must enter the following information in the Link definition:

- File Type: M
- Server ID
- Table Name

Below is a sample Link definition screen:




Server ID
The server ID is a unique identifier that references the host via the definition in the SERVER.MAP file. It will be validated against the list of server devices configured in the IPLINPUT file. Although no ID will be disallowed, a warning will be displayed.

Table Name
The Table Name is the name used to create a table on the server.

Text File Name
The Text File Name in this example appears as CSSERFL1.TXT. In Oracle it is converted to CSSERFL1_TXT.

Both the table and text file names must conform to Oracle table naming conventions. For more information see the appropriate Oracle document to review the restrictions.


Copyright © 2026 Thoroughbred Software International, Inc.

Links must be defined as a file type M (MSORT). While the physical file is not stored as an MSORT file, it will be processed as one in Dictionary-IV for sorts and text field storage. For more information see Using Sorts and Text Fields for Oracle Data.

For Dictionary-IV to detect that a Link references a server-based table, File Type, Server ID, and Table Name must all contain valid information. If they do not a local file will be used, referenced by the name entered in the Data File field.

The system supports file suffixes. The file suffix character ( @ ) must be placed in the Table Name field for server files, rather than the Data File field. Where text fields are configured and a file suffix is used, the text file name must include the suffix character. For more information refer to the Dictionary-IV User Guide.

Since the @ is a Dictionary-IV convention, access to tables on other Oracle servers via SQL*NET (TABLE@OTHER_SERVER) can only be accomplished by configuring a synonym on the Oracle server referenced by the Server ID.

Sort definition occurs in the same way for server Links as for local Links and should be saved and updated with **F8** processing. Even though no physical rebuild takes place, this will properly update the Link header. For more information see the Dictionary-IV Developer Guide.

If sort definitions exist in the Link and the table is recreated, the indexes will be defined automatically on the server as part of the create process.

Under the current Dictionary-IV methodology, first build a Link header and create the table. Modify the Link header to define the sorts. Erase and recreate the table. The indexes for all sorts are now defined on the server.

Once the Link header is configured, multiple and single record maintenance, CONNECTs, reports, queries, and recompiled scripts can access server tables. For more information about modifying Basic programs to support access to server tables see Using Basic.

While the data and sort file names may be populated during the Link header creation, neither is referenced during server access. However, the data file name is used to generate the text file name for text field storage. Therefore, be sure to either enter the data file name for the automatic generation of a text file name or manually enter the field.


Copyright © 2026 Thoroughbred Software International, Inc.

# ORACLE TABLES

The following describes how to process Oracle tables from the TS Environment.

## Accessing Oracle Tables

The following describes how to create Oracle tables and then access these tables from the TS Environment. For more information specific to Script-IV and Dictionary-IV see Script-IV/Dictionary-IV Tables.

### *Using Basic*

Existing server files are accessed using the following syntax:

```
OPEN(CH,OPT="LINK")"link-name"
```

"LINK" specifies that the file to be opened on this channel is found in the system dictionary in the link-name record.

"link-name" specifies a valid link-name containing the necessary server references.

New server files may be created from the Basic environment as follows:

```
OPEN(CH,OPT="LINK|CREATE")"link-name"
```

"CREATE" is a parameter that causes Basic to generate the server commands necessary to build the file. The Oracle table definition is based on the format listed in the Link Definition. If the table already exists on the Oracle server, the CREATE is ignored and the table opened.

A special OPEN syntax is used for access to text field data. This syntax is used internally by the Dictionary-IV system for accessing text field data. For more information see the Text Fields for Oracle Data section later in this manual.

```
OPEN(CH,OPT="LINK|TEXT")"link-name"
```

"TEXT" specifies that the text file from the link is to be opened.

If multiple SQL servers are configured, existing files on different servers can be accessed using the following syntax:

```
OPEN(CH,OPT="LINK|SID='server-id'")"link-name"
```

"SID=" is a parameter that causes Basic to access the specified server instead of the server ID defined in the link definition.

"server-id" specifies the id of one of the configured SQL servers.


Copyright © 2026 Thoroughbred Software International, Inc.

Server files with names different from the names defined in a link may be accessed using the following syntax:

```
OPEN(CH,OPT="LINK|TABLE-NAME='table-name'|TEXT-FILE='text-table-name'")"link-name"
```

| | |
|---|---|
| "TABLE-NAME=" | is a parameter that causes Basic to use the specified table name instead of the one defined in the link definition. |
| "table-name" | specifies the table name. |
| "TEXT-FILE=" | is a parameter that causes Basic to use the specified text field table name instead of the one defined in the link definition. |
| "text-table-name" | specifies the text field table name. |

The following code fragments show two ways of using the new OPEN syntax to access files. The examples assume a Link definition with information in the Server ID and Table Name fields and a file type of M.

If no server information is present, the OPEN(OPT="LINK") will access the local file referenced by the Data File name. In version 8.3.0 a Link opened this way expected the Format to contain field separators and the data file to be a fixed field/fixed length record. In this version the Format and data are required to match. In addition the OPT="LINK" supports local files with or without field separators.

*Example 1*

```
OPEN/READ server file into IOLIST (variables)

10       IOLIST CUST_CODE$,CUST_NAME$,ADDRESS$

100      CH=UNT;
         OPEN(CH,OPT="LINK") "UTCUST"
         READ(CH,KEY="0001")IOL=10
```

This example assumes a Format containing field separators.

*Example 2*

```
OPEN/READ server file into FORMAT

10       CH=UNT;
         OPEN(CH,OPT="LINK") "UTCUST";
         FORMAT INCLUDE #UTCUST;
         READ(CH,KEY="0001")#UTCUST
```

This example can be used on any format with or without field separators.


Copyright © 2026 Thoroughbred Software International, Inc.

*Example 3*

```
OPEN/READ server file into record variable

10        CH=UNT;
          OPEN(CH,OPT="LINK") "UTCUST";
          READRECORD(CH,KEY="CUST0001) REC$
```

This example shows access using a Format without field separators. The data in REC$ could be used to populate the format data area.

Under certain circumstances an error 167 may be returned when writing to a Link opened with OPT="LINK". When employing a Format directly, or a Format that contains field separators, the system assigns that data to the data element name.

If this data is invalid for the particular data element name (invalid value for Yes/No fields, blank mandatory fields, or invalid data in a date field) an error 167 occurs. Printing the DNE system variable at the instance of the error will display the data element name for which the assignment failed.

## *OPT="LINK" and Local Files*

The OPT="LINK" parameter now allows you to use it with local files. The same application code that accesses server tables can be used to access local files. Once you establish a Link to reference a server table, clear the Server ID and you can use this Link to reference a local file.

This is applicable to 3GL code. For Script-IV, changes to the Link header require you to recompile your script.

ERASE
Server tables are deleted from within the Basic environment by specifying the Link name and OPT="LINK".

```
ERASE"UTCUST",OPT="LINK"
```

This deletes the table from the server referenced in the Link UTCUST. It also deletes the text field table if any text fields were defined. Specifying an OPT="" erases local files.

INITFILE
Server tables may be cleared of all data by using the INITFILE directive. This directive is useful for clearing a table without the extra step of performing an erase.

```
INITFILE "UTCUST",OPT="LINK"
```

This clears the table on the server referenced in the Link UTCUST. It also clears the text field table if any fields are defined. Specifying an OPT="" clears local files.

RENAME
The rename function allows a table name to be changed on the server from the client process.

```
RENAME "UTCUST","UTTEMP",OPT="LINK"
```


Copyright © 2026 Thoroughbred Software International, Inc.

This renames the table referenced in Link UTCUST to the table UTTEMP. No changes are made to the original Link. Specifying an OPT="" performs the rename on local files.

FID Function
When the FID is executed on a channel where a Link referencing a server table is opened, the resulting string contains (in addition to the standard FID information) the following:

Byte 1 Value is "S". Indicates that this is a channel opened to a link referencing a server table.

Byte 2 - 3 The two character Server ID from the Link.

The bytes that normally contain the file name now contain the Link name.

TCB Function
Error 150 has been added to the Basic error list to reflect errors resulting from server operations. To return the specific server error, an argument has been added to the TCB( ) function. The TCB(20) now returns the server error in the same way that the TCB(3) returns the operating system error. The value returned for the TCB(20) can be interpreted by referring to the server error guide.

DUMP Directive
Several of the DUMP options have been changed to be aware of server access.

DUMP CHANNELS now display the link name when a link has been OPENed with OPT="LINK".

DUMP IPLDEVS will specify that devices configured as a server will return "Access using server xx."

XFD Function
The following describes the current values returned for server tables:

Option = 0

| *Bytes* | *Description* |
|---|---|
| 1-2 | Unused |
| 3-10 | N/A |
| 11-35 | Unused |
| 36-38 | N/A |
| 38-41 | N/A |
| 42-47 | Unused |
| 48 | "N" |
| 49-53 | Unused |
| 54-59 | N/A |
| 60-65 | N/A |
| 66-71 | N/A |
| 72-85 | Unused |
| 86+ | N/A |


Copyright © 2026 Thoroughbred Software International, Inc.

Option = 1
As documented in the Thoroughbred Basic Reference Manual.

Option = 3

| *Bytes* | *Description* |
|---|---|
| 1 | $05$ |
| 2 | $83$ |
| 3 | Number of sorts (binary) |
| 4 | Current sort number (binary) |
| 5-12 | "NNY" (bytes 8-12 Unused) |
| 13-14 | N/A |
| 15-16 | COBOL only |
| 17-42 | Unused |
| 45+ | Sort definition parameters (For more information see the Thoroughbred Basic Reference Manual.) |

Option = 4
As documented in the Thoroughbred Basic Reference Manual.

Option = 6

| *Bytes* | *Description* |
|---|---|
| 1 | $06$ |
| 2 | $00$ |
| 3-6 | $00000000$ |
| 7-8 | Bytes per record (binary) |
| 9+ | Link name |

Option = 10

Returns -1

FST Function
This function is not supported for server tables.

IND Function
The value returned by the IND function is the ORACLE ROWID. Unlike local files this value does not start at 0 for the first record and continue incrementally. That is, reading the table sequentially by index number will produce unexpected results. However, reading specific records by a determined index value is supported.

**NOTE:** The string following OPT= can be a variable in any of the commands where it is specified. The value does not need to be hard-coded into the program.

## *Using Sorts*

Sorts, also known as indexes, are handled in substantially the same manner for server tables as they are for local files. Created in the Link definition, they can be used for access in any operation that currently supports sorts. As with MSORT, if the Link references an existing file, the link definition can be updated to include sorts defined in the file without rebuilding the file. The differences occur in the way that sorting is applied to the server table.


Copyright © 2026 Thoroughbred Software International, Inc.

Indexes do not need to physically exist in the server environment to use Link sorts anywhere in Dictionary-IV. As long as the sort definition is established in the Link definition, Dictionary-IV can use it to access the data file. Basic will expand the sort statement defined in the Link into the appropriate field names, rather than passing only the sort number.

Oracle sort statements will refer to these field (column) names, not index names, when defining an ORDER BY statement. So, if the index does not exist, Oracle performs the sorting operation in-line. However, if an Oracle index exists for the columns in the ORDER BY statement, the index is used to access the data file. However, the Indexes will be used only if one of the columns has been defined with a NOT NULL constraint. For more information see the Performance Tuning section later in this manual.

If sort definitions exist in the Link and the table is recreated, the indexes will be defined automatically on the server as part of the create process.

Under the current Dictionary-IV methodology first build a Link header and create the table. Modify the Link header to define the sorts. Erase and recreate the table. The indexes for all sorts are now defined on the server.

Oracle does not support sorts on substrings of fields.

The sorts (indexes) are created using the following naming convention:

**"I_" + *table-name* + " _ " + *Dictionary-IV sort number***

Example:

For sort number 2 on a Link with a table-name defined as TSISERVER, the index would be created as "I_TSISERVER_2".

If there are sorts defined in the Link that refer to fields that contain blank data, Oracle places these records at the end of an ascending sort and at the beginning of a descending sort. To modify this behavior see Null Processing.

Adding and deleting sorts is not supported from within the Basic client (ADDSORT/REMSORT). These functions are reserved for the database administrator.

### *Locking Records*

The locking facilities available with Oracle provide the developer with resources to ensure data consistency. Many of the features designed for these goals occur automatically. For more information refer to the Oracle Concepts Manual.

This section explains how these locking mechanisms correlate to Thoroughbred directives and how to relate locking that occurs in the Thoroughbred application to that, which occurs in an Oracle application.

The following table summarizes the relationship of Thoroughbred directives to Oracle SQL statements and the level of locking associated with each.


Copyright © 2026 Thoroughbred Software International, Inc.

SQL Statements/Thoroughbred Directives

| Thoroughbred Directive | SQL Statement | Type of Lock | Lock Modes Permitted (see legend below chart for description) | | | | |
|---|---|---|---|---|---|---|---|
| | | | RS | RX | S | SRX | X |
| READ, FIND, PREAD, and all key functions | SELECT...<br>FROM table... | none | Y | Y | Y | Y | Y |
| WRITE, if new | INSERT INTO table... | RX | Y | Y | N | N | N |
| WRITE, if exist | UPDATE table... | RX | Y* | Y* | N | N | N |
| ERASE, INITFILE | DELETE FROM table... | RX | Y* | Y* | N | N | N |
| EXTRACT, WRITE | SELECT...<br>FROM table...<br>FOR UPDATE OF... | RS | Y* | Y* | Y* | Y* | N |
| OPEN, UNLOCK | LOCK TABLE table IN ROW SHARE MODE | RS | Y | Y | Y | Y | N |
| | LOCK TABLE table IN ROW EXCLUSIVE MODE | RX | Y | Y | N | N | N |
| | LOCK TABLE table IN SHARE MODE | S | Y | N | Y | N | N |
| | LOCK TABLE table IN SHARE ROW EXCLUSIVE MODE | SRX | Y | N | N | N | N |
| LOCK | LOCK TABLE table IN EXCLUSIVE MODE | X | N | N | N | N | N |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **RS** | Row Share | **RX** | Row Exclusive | **S** | Share | **SRX** | Share Row Exclusive | **X** | Exclusive |

* Valid as long as no conflicting records are involved.

For more information about the WRITE statement see the Generated SQL Code section later in this manual.

The Lock Modes Permitted column indicates what type of lock may be placed on the file (table) or record (row) while the particular directive is in effect. For example, if a record is EXTRACTed in Thoroughbred, exclusive (X) locks by another client or Oracle task are not allowed and produce an error. All other lock types are valid as long as a different row is locked.

Since the query-type functions (READ, FIND, etc.) do not generate a lock, it will not fail under any locking mode the table or record may currently have. This means that READing locked records will not require PRM READONLY to access tables or rows locked by either an Oracle task or another Thoroughbred client process.


Copyright © 2026 Thoroughbred Software International, Inc.

### Null Processing

By default, null key values will be returned to the Thoroughbred Basic application following all other records for ascending sorts and before all records for descending sorts. This is an Oracle convention and cannot be changed. However, two options exist to alter this behavior. Both methods involve the modification of the null data so that the sorting mechanism treats the null key value as non-null data. This is more consistent with a Thoroughbred Basic application where null keys appear at the top of an ascending sort and at the bottom of a descending sort.

In the first method, the null data is logically changed. This is accomplished by using the ORACLE NVL function in the ORDER BY clause of the SELECT statement. Each column listed in the ORDER BY clause is passed through this function which will map null values to the value supplied to the function. In this case, the value is a single space. This value will then be passed to the sorting process so that these records will sort at the appropriate end of the result set. While this method makes no modification to the physical data, it may cause sorts to perform poorly for composite indexes. To activate this method, specify the PRM ORA_NVLNULLS in the IPLINPUT file.

| |
|---|
| For single column indexes that allow null values, performance may not be an issue since this index would not be utilized. |

In the second method, the data is changed physically. Whenever a null character field is written or updated to an Oracle table, it will be changed to a single blank. All subsequent accesses with indexes will return the data with the blank fields sorted appropriately. This method will allow the index to be used for queries so that the result set is returned efficiently, but the data is modified. It will be important for other applications to maintain the space in the data. To activate this method, specify the PRM ORA_DONTWRITENULLS in the IPLINPUT file.

As an alternative to the above methods, the table could be created so that columns used as indexes can never contain null values. This will happen if the field is defined in Dictionary-IV as a fixed length type, either optional or mandatory (Entry Type 1 or 3). This attribute will ensure that the data for that column will not be null.

For access to existing tables, all of the options are available, however, care must be taken if new data is written to a table when the ORA_DONTWRITENULLS parameter is specified. This will create a situation where only the new records with null keys appear in the proper sorted order.

### CMASK

The Thoroughbred Basic client will always output data to Oracle tables as if CMASK is set to ".=.". Oracle will accept this regardless of its setting for the numeric character masking. However, when reading data, no assumptions are made and the masking must match on both the client and server systems.


Copyright © 2026 Thoroughbred Software International, Inc.

## Processing Tables from Oracle

To access existing Oracle tables you must build a Format and Link with the appropriate table information to be derived from the Oracle system.

The following Oracle constructs have no Dictionary-IV equivalent:

- LONG data type
- VARCHAR2 fields greater than 255 bytes in length
- NUMBER fields exceeding Thoroughbred's maximum of 14 significant digits cannot be supported. The maximum length of this column type in Oracle is 40 where 38 are significant. The concepts of negative scale and scale greater than precision are not supported.
- Column names exceeding 20 characters in length.

You have the ability to access tables containing the above characteristics for reading and updating. By defining a format that contains a partial list of the fields defined in the table, data can be read and updated without affecting the fields not included in the format.

Numeric columns that Thoroughbred will support are defined as a precision of 5 and scale of 2 with a length of 7.2 for numeric types 0 and 2. It must be defined with a length of 6.2 for type 1. Oracle does not include the minus ( - ) sign in column length.

### *Text Fields for Oracle Data*

Dictionary-IV treats server tables internally as MSORT for I/O processing. Therefore, text field data associated with servers must be stored in a separate file as is the current convention with MSORT type files. This file, referenced as the text file in the Link header, must be stored on the server with the main table to maintain data integrity, to provide access to all clients, and for backing up.

If the Format defines one or more text fields then a separate table for the text data is automatically generated when the table for the record data is created. The record data is stored in the table referenced by the Table Name field and the text field data is stored in a table referenced by the Text File field. If present, a dot ( . ) in the file name is converted to an underscore
( _ ).

The definition of the text table is derived from the record data and a number of preset fields that are used internally by Dictionary-IV. This does not appear as a Dictionary-IV Format but exists as an Oracle table. A logical Format called IDSV1 has been created to aid in the handling of this information. It is used by the system and is not associated with any specific Link.

The layout is as follows:

```
KEY_FIELD       VARCHAR2 (len=length of primary key + 1 for text id)
CREATE_DATE     DATE
CHANGE_DATE     DATE
TEXT_WIDTH      NUMBER(3)
TEXT_FIELD      LONG
```


Copyright © 2026 Thoroughbred Software International, Inc.

*KEY_FIELD*
Contains the key to the record data plus the text id. This is similar to the current key structure of text field data, except there is no need for a sequence counter since the entire text field is stored in one record. This is possible because the LONG column type has a maximum storage capacity of 2 gigabytes.

*CREATE_DATE*
Specifies the creation date.

*CHANGE_DATE*
Specifies the date of the last change.

*TEXT_WIDTH*
Used by Dictionary-IV for compatibility to local MSORT processing.

*TEXT_FIELD*
Because this text data has been made available as an Oracle table, the data is comprised of purely printable characters (except line terminating $0A$). Therefore, any color or attribute information is stripped prior to storage in the table. Graphic characters are converted to a standard ASCII character representation. Changes made from an Oracle application are available to the Thoroughbred application.

All file level access (create, erase, and clear) performed on the record table is also executed on the text table. However, all record level access requires a distinct OPEN of the text table. This is consistent with the current MSORT text field operation.

During I/O processing a $FF$ is implied in the first byte of the key. This byte does not appear in the data but is required for programming consistency. The byte is required for all key access and is returned in all key functions (KEY, FKY, LKY, and PKY).

The Text field data is automatically maintained when accessed via Dictionary-IV. No changes are required to CONNECTs, Scripts, Reports, or Queries that currently use Text fields.


Copyright © 2026 Thoroughbred Software International, Inc.

# PERFORMANCE TUNING

The following describes how to maximize the throughput using sorts and process cache.

## Sorts

To maximize the throughput for a number of the Basic I/O operations that operate on server tables, you must carefully plan the design of the table. When the client process creates the table in the Oracle database, it does so generically. It makes no assumption about the data to be contained in the record other than what it determines from the Dictionary-IV Format. This is sufficient to describe the record to Oracle but under some conditions, sorting, for example this may not be enough.

Oracle creates result sets that it generates from statements that specify ordering whether or not an index exists for the columns listed in the ORDER BY clause. Without an index this operation is very time consuming for large tables. Within the Basic environment this is multiplied since each access to the table may require an in-line sort. Basic saves the result set from a particular query and uses it if the subsequent query matches the one that generated the result. If it does not match, the result set must be regenerated the next time the query is executed.

The solution is to create an index for the columns to sort and define at least one column as NOT NULL (a mandatory field in Dictionary-IV). This must be done or the index is ignored for ORDER BY processing.

The create process translates mandatory fields and the first column of the primary key into columns with the NOT NULL constraint if PRM CREATWDBATTR is configured. If these columns are part of indices you need no further changes.

Degradation due to indices being ignored becomes evident in:

Basic
using LKY( ), PKY( ), and I/O with sorts

Dictionary-IV
File access with sorts (multi-record maintenance, view maintenance, sort order change) and backward reads in files (multi-record maintenance up arrow and page up)

Script-IV
READ PREVIOUS, LAST TO FIRST, sorts, etc.

## Logon Cache

All access to the Oracle database occurs through one login to the server using multiple cursors. When a LOCK, EXTRACT, or WRITE is performed in Basic a new login is executed to preserve key pointers to the other tables opened on the original login (LOCKs, EXTRACTs, and WRITE will perform commits and will clear all record locks). Once the LOCK, EXTRACT, or WRITE is complete, this login is terminated. The logon cache count parameter in the IPLINPUT file is used to specify the number of login processes to retain.


Copyright © 2026 Thoroughbred Software International, Inc.

For example, if the logon cache count is set to 5, a pool of up to 5 Oracle login processes will remain open. The next LOCK, EXTRACT, or WRITE executed by Basic will attach itself to one of these 5 processes bypassing the overhead to login and establish a new process.

By configuring this value properly, a balanced maximum can be maintained between excessive login processing and the number of active logins. A maximum of 20 logins can be cached.


Copyright © 2026 Thoroughbred Software International, Inc.

# USING THE DEBUGGING FACILITY

## SQL Output

A debugging facility is available from the tsoracle## process. Execute this program with the following syntax to activate the debugger.

```
tsoracle10 –dn

or

tsoracle11 -dn
```

n Currently the only debug level is 1. This sends the generated SQL statements to the log file (/usr/lib/basic/tsor.debug). The output in the log file contains a separate line for each generated SQL statement prefixed by the child tsoracle10 process id.

While tsoracle## is active, output for all clients (located on the same machine) is sent to one tsor.debug file.

If the tsor.debug file is copied, output will continue to be placed in the renamed file until the parent tsoracle10 process is terminated. A subsequent execution will use a new tsor.debug.

## Process Identification

A method is available to associate the child tsoracle## process with the client process. Execute this program with the following syntax to activate the debugger:

```
tsoracle10 –s

or

tsoracle11 -s
```

By including the -s argument on the tsoracle## command, each time the client logs into the server, information is sent to tsoracle10.pid/tsoracle11.pid file. The file is located in /usr/lib/basic. The format of the information is as follows:

Line 1 parent tsoracle## process

Line 2 - n [child tsoracle## process][client host name][client basic process]


Copyright © 2026 Thoroughbred Software International, Inc.

# GENERATED SQL CODE

This section shows the resulting SQL syntax for Thoroughbred functions (Italic typeface). They are presented in alphabetical order.

## SQL Syntax for Thoroughbred Functions

### *CLOSE (channel)*

COMMIT
This is performed via a direct OCI call, not as a literal SQL statement.

### *ERASE*

DROP TABLE table-name
If text fields are configured: DROP TABLE text table-name

### *EXTRACT[RECORD] (channel)*

SELECT columns FROM table-name WHERE key-conditions >=
current-key ORDER BY sort-information
Get key from first record or current SELECT
SELECT columns FROM table-name WHERE key-conditions =
current-key ORDER BY sort-information FOR UPDATE NOWAIT

### *EXTRACT[RECORD] (channel,KEY=)*

SELECT columns FROM table-name WHERE key-conditions =
current-key ORDER BY sort-information FOR UPDATE NOWAIT

### *FKY( )*

SELECT columns FROM table-name WHERE key-conditions
ORDER BY sort-information
Get the first record.
Extract the key information from the record.

### *INITFILE*

DELETE FROM table-name
If text fields are configured: DELETE FROM text-table-name


Copyright © 2026 Thoroughbred Software International, Inc.

### KEY( )

If a record is not extracted
SELECT columns FROM table-name WHERE key >
    key of last record read ORDER BY sort-information
Otherwise: Extract the key information from the record.

### LKY( )

SELECT columns FROM table-name ORDER BY reverse-sort
Get the first record.
Extract the key information from the record.

### LOCK (channel)

LOCK TABLE table-name IN EXCLUSIVE MODE NOWAIT

### OPEN (channel,OPT="LINK|CREATE")

CREATE TABLE table-name
If text fields are configured: CREATE TABLE text-table-name
LOCK TABLE table-name IN ROW SHARE MODE NOWAIT
If text fields are configured:
LOCK TABLE text-table-name IN ROW SHARE MODE NOWAIT

For sort0:
CREATE UNIQUE INDEX I_table-name_0

For sortn:
CREATE INDEX I_table-name_n

### OPEN (channel, OPT="LINK")

LOCK TABLE table-name IN ROW SHARE MODE NOWAIT

### OPEN (channel, OPT="LINK|TEXT")

LOCK TABLE text-table-name IN ROW SHARE MODE NOWAIT

### PKY( )

If a record is not extracted
SELECT /*+ INDEX_DESC(table-name) */ columns FROM table-name
    WHERE key-conditions ORDER BY reverse-sort
Otherwise: Extract the key information from the record.


Copyright © 2026 Thoroughbred Software International, Inc.

### *PREAD[RECORD] (channel)*

If a SELECT is in memory get the next record
Else
SELECT /*+ INDEX_DESC(table-name) */columns FROM table-name
    WHERE key-conditions ORDER BY reverse-sort

### *READ[RECORD] (channel,KEY=)*
### *FIND[RECORD](channel,KEY=)*

If a SELECT is in memory get the next record
    Else
    SELECT columns FROM table-name WHERE key-conditions >=
        current key ORDER BY sort-information

### *REMOVE (channel,KEY=)*

SELECT columns FROM table-name WHERE key-conditions
    FOR UPDATE NOWAIT
DELETE FROM table-name WHERE key-conditions

### *RENAME*

RENAME table-name TO new-table-name

### *UNLOCK (channel)*

LOCK TABLE table-name IN ROW SHARE MODE NOWAIT

### *WRITE[RECORD] (channel,KEY=)*

Get key value from the record data
SELECT columns FROM table-name WHERE key-conditions =
    current-key FOR UPDATE NOWAIT
If the record exists:
UPDATE table-name SET column = value,...
Else:
INSERT INTO table-name column ... VALUES (value...)
fi
COMMIT
This is performed via a direct OCI call, not as a literal SQL statement.


Copyright © 2026 Thoroughbred Software International, Inc.

## Multiple Select Statements

The following Basic commands may produce multiple select statements when sequentially accessing tables with multiple key fields defined in the Format:

READ[RECORD]
FIND[RECORD]
PREAD[RECORD]
EXTRACT[RECORD]
KEY( )
PKY( )
FKY( )

## Unsupported Basic Directives

There is no equivalent SQL code for the following Basic directives:

ADD
ADDSORT
FILE
REMSORT

These items are not supported when applied to server tables.


Copyright © 2026 Thoroughbred Software International, Inc.

# TECHNICAL REFERENCE

The following provides information specific to Script-IV, Dictionary-IV, Thoroughbred Basic, and Oracle views.

## Script-IV/Dictionary-IV

The following information is specific to Script-IV and Dictionary-IV

### *DATA-FILE IS, SORT-FILE IS*

This feature will be supported but the object referenced by this command must be a Link rather than a local data file. The Link can then reference either a server table or a local file. The file type set in this Link must be "M".

### *Access Subset of Columns*

By defining a format that contains a subset of the columns that are defined in a table, access to tables that contain unsupported data types (LONG, RAW, etc.) is possible. All of the possible I/O directives are supported with no effect on the unreferenced fields.

## Basic

### Soft Includes

When an OPEN is executed on a server Link, the Format referenced in the Link header is soft included. This means that the Format data area is defined in memory. However, the Format name will not appear in the FMTNL list and will not interfere with any hard includes that take place. The soft include is executed internally as a FORMAT INCLUDE #format-name, OPT=”NONE” so that any data defined in a Format data area at the time of an OPEN of a server Link that references this Format is preserved.

If however a READ or a WRITE is executed using an IOLIST, the Format data area will be modified since the I/O operation uses the Format data area to parse the data into the proper variable values.

### *Enhanced Access to Tables*

Where possible sequential access to tables may provide faster access times during READ operations. This is possible because a particular I/O statement will generate on the server a result set. Subsequent READ operations will simply move through the result set and return the data to Basic. The area where this is most obvious is when a table is sorted by Basic in a way that no index exists in Oracle. The initial READ will force the sort to take place, but all READS that follow will get the data from the result set without the need to sort again. If any other I/O statement had occurred between READs another result set would have been created forcing a re-sort on the next READ statements.


Copyright © 2026 Thoroughbred Software International, Inc.

#### *Reestablish a Connection*

To reestablish a connection, use the following command:

```
ENABLE disk-no,"RECONNECT"
```

Where *disk-no* is the DEV line designation to the Oracle server.

#### *SET CTC and CTC()*

Allow setting and retrieving of COMMIT COUNT from Basic.

#### *PRMs*

If data for the numeric is all spaces and PRM SQL-NUMERIC-NULLS is set and PRM ORA-DONTWRITENULLS is not set, numeric fields will be written to the database as NULL.

## Tables

### *Oracle Views*

Where possible use of an Oracle view as the table in Link definition should be avoided. Views will not support all the SQL statements that are generated by Basic statements. The most common of these will be the SELECT FOR UPDATE which is the EXTRACT directive in Basic. Therefore, it is possible that server errors (ERR=150) will occur accessing Views which will interrupt normal program functioning.

Also, for the View data to be returned in a sorted order, the sort must be explicitly defined in the definition of the view in Oracle. Merely defining key fields will not produce sorted results. A sort must be defined in the link header and then used in all I/O statements.


Copyright © 2026 Thoroughbred Software International, Inc.

# ERROR MESSAGES

The following describes client error messages.

## Client/Server Basic Error Messages

**150**

Explanation: Server system error. For more information refer to TCB(20) to get the server error number.

**171**

Explanation: Undefined Link.

## Thoroughbred Basic Startup Messages

**No match for server id in SERVER.MAP**

Explanation: The two-character server id defined in the IPLINPUT file cannot be located in the SERVER.MAP file.

Action: Verify contents of both files.

**Could not resolve tcp/ip address for Thoroughbred foreign file server**

Explanation: System does not understand the tcp/ip address configured in the SERVER.MAP file.

Action: Verify that the address or host name in the SERVER.MAP file is a valid value for the system. Also verify that the /etc/hosts file is accessible. This file must be readable for all users. On systems that use DNS, be sure the system is configured.

**Invalid Thoroughbred foreign file server type**

Explanation: The system found an invalid value in the server type parameter in the IPLINPUT server device line.

*Action:* Verify that the server device line is defined properly.

**Could not create client-end connection to Thoroughbred foreign file server**

Explanation: This can occur if TCP/IP has not been started or configured on the client system.

Action: Verify that the client machine can be reached over the network.


Copyright © 2026 Thoroughbred Software International, Inc.

**Could not connect to Thoroughbred foreign file server**

Explanation: Thoroughbred Basic could not establish communication to tsoracle##.

Action: Verify that tsoracle## is started.

**Thoroughbred foreign file server could not connect to foreign database**

Explanation: tsoracle## could not establish communication to the server.

Action: Verify that the server processes are started and that the account starting tsoracle## has ORACLE_HOME and ORACLE_SID shell variables set appropriately.

## Oracle Error Messages

The following errors are returned in the TCB(20) when an error 150 occurs. This is a very small set of errors that are included here because they are resolvable from within the Basic environment.

**904**

Explanation: A column is either missing from the Dictionary-IV Format, one of the data element names is misspelled, or a data element contains an invalid character**.**

**917, 936, 1843**

Explanation: Any of these errors can occur at WRITE time if a data record contains bad data. There are either characters in a numeric field or a bad date in the data field.


Copyright © 2026 Thoroughbred Software International, Inc.

# TROUBLESHOOTING

The following provides troubleshooting information specific to the Oracle Server.

## Link Error

Error 0 occurs opening the Link and no locks seem to be active in Basic or from an Oracle application.

Since the Link header is read internally by Basic when opening the Link, this record needs to be accessible at open time. If it is extracted through Dictionary-IV, for example, the error 0 occurs. Setting the PRM READONLY parameter in the IPLINPUT file will clear this situation.

## Server Table Error

Error 12 opening server tables.

Verify that the DEV line for the server where the table resides is present in the appropriate IPLINPUT file.

## tsoracle## will not start

Verify that the /usr/lib/basic directory is present and accessible on the system where tsoracle## is executed.

## Unexpected Sorting Sequence or Errors Accessing Some Oracle Tables

- If a Link header has been configured to access a server table defined on the server as a view, unexpected results ranging from incorrectly sorted data to tsoracle## process failures may result. It is suggested that access to the server be limited to objects defined as tables. Oracle does not support certain operations on views such as SELECT ...FOR UPDATE. This may also occur if a table has no index and the fields in the sort are defined as NULL. This can be resolved by adding an index and/or changing the field constraint to NOT NULL.

- Be sure that the primary key index is present

## Hanging on a WRITE

Under certain circumstances a WRITE may hang if the row is locked in Oracle using a share (S) or share row exclusive (SRX) type lock. This occurs because the INSERT/UPDATE part of the SQL code for a WRITE statement does not support NOWAIT and will wait until the lock is removed by the Oracle process. The share and share row exclusive type locks are not generated by any Basic code, and therefore this situation cannot occur between Basic tasks.


Copyright © 2026 Thoroughbred Software International, Inc.

## Some Access Using Sorts is Very Slow

Sorts that contain references to substrings of fields may not perform as well as sorts on full fields. This is because Oracle has no facility to define indexes on portions of fields. For more information see the Performance Tuning section of this manual.


Copyright © 2026 Thoroughbred Software International, Inc.