PARADOX SQL LINK

CONNECTING TO MICROSOFT AND SYBASE SQL SERVER

BORLAND

Paradox SQL Link

Version 4.0

Connecting to Microsoft and SYBASE SQL Server

Borland International 1800 Green Hills Road
P.O. Box 660001, Scotts Valley, CA 95067-0001, USA

Printed in the U.S.A.

10 9 8 7 6 5 4 3 2 1 R1

CONTENTS

Connecting to Microsoft and SYBASE SQL Server 1

Installation 1
System requirements 2
Server requirements 2
Client requirements 2
Installation steps 5
Information you need 5
Starting Paradox 6
Selecting a Microsoft or SYBASE SQL Server connection 7
Connection parameters 8
Troubleshooting common connection problems 9
Network and server access privileges . . 9
SQL Server system tables 10
Stored procedures 11
Triggers 12
Defaults 13
Rules 13
The RAISERROR command 14
SQL Link and SQL Server views 15
SQL Server data types 15
Query restrictions 17
Error handling 17
SQL Server locking 18
Support for the Tandem SQL Server Gateway 20
Additional reference tables for Microsoft and SYBASE SQL Servers 20

TABLES and FIGURES

Tables

1 Server and software required . . . 2
2 Client requirements 2
3 Sample Net-Library interfaces . . . 2
4 Connection parameters 8
5 SQL Server system tables 11
6 Paradox support of SQL Server defaults 13
7 Paradox validity checks and SQL Server features 14
8 SQL Server to Paradox data type mapping 16
9 Paradox to SQL Server data type mapping 17
10 SQL Server locks 18
11 SQL Server locking for Paradox operations 19
12 General information 20
13 Field-naming rules 21
14 Paradox-SQL Server query locks . . 21

Figures

1 System architecture for Microsoft SQL Server 3
2 System architecture for SYBASE SQL Server 3
3 SYBASE SQL Server architecture for UNIX 4
4 SYBASE SQL Server architecture (VMS) 4
5 SQL Server permission hierarchy . . 10

Connecting to Microsoft and SYBASE SQL Server

This addendum to the Paradox SQL Link *User's Guide* shows how to use Paradox SQL Link to connect to Microsoft SQL Server and SYBASE SQL Server. It also discusses the Tandem SQL Server Gateway. You should read the Paradox SQL Link *User's Guide* in addition to this addendum.

Installation

This section describes the hardware and software required to run SQL Link and tells you how to install SQL Link on your hard disk or network.

This section assumes the following:

- Your network is already installed and running.
- Your database server is already installed and running.
- Paradox version 4.0 is already installed on either the shared disk of your network file server or the local hard disk of a standalone workstation.
- You have sufficient access rights (also called privileges) to the Paradox system files directory to add new files to it.

See your Paradox documentation for information on installing and configuring Paradox.

System requirements

The following sections describe the hardware and software you need to run SQL Link. The diagrams following this section show various system architectures for connecting SQL Link to SQL Server.

Server requirements

To run SQL Link, you need the database server product and its associated network software installed and running.

Table 1 Server and software required

Vendor	Server Product	Network Software
Microsoft	Microsoft SQL Server 1.0 or later	Network software supporting Named Pipes protocol (Microsoft LAN Manager, 3Com 3+Open, or Novell NetWare Requestor for OS/2)
Sybase	SYBASE SQL Server 4.0 or later	Network software supporting the TCP/IP (UNIX), DECnet (VMS), or Pathworks (VMS) protocol

See your Microsoft or SYBASE manuals for more information on specific requirements.

Client requirements

Table 2 lists the requirements for your client (the computer running SQL Link), which must be 100% IBM compatible.

Table 2 Client requirements

Component	Requirement
Microprocessor	80286 or higher
RAM	A minimum of 2MB
DOS version	As required by your network
Disk space	Approximately 1MB (for the SQL Link program itself)
Paradox	Version 4.0

You also need the appropriate Net-Library interface for your server. Table 3 gives examples.

Table 3 Sample Net-Library interfaces

Network software	Example of a Net-Library interface
LAN Manager or NetWare	DBNMPIPE
FTP TCP/IP	DBFTPTCP
DEC DECnet	DBDECDEC

Understanding the system architecture you're working on can help you solve common connection errors. All components of your system must be running before your remote workstation can connect to a server.

Figure 1 shows the Microsoft SQL Server architecture on LAN Manager.

Figure 1 System architecture for Microsoft SQL Server

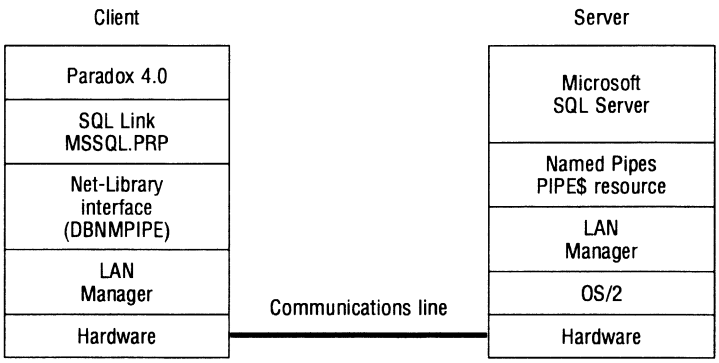


Figure 2 shows the SYBASE SQL Server architecture on Novell NetWare.

Figure 2 System architecture for SYBASE SQL Server

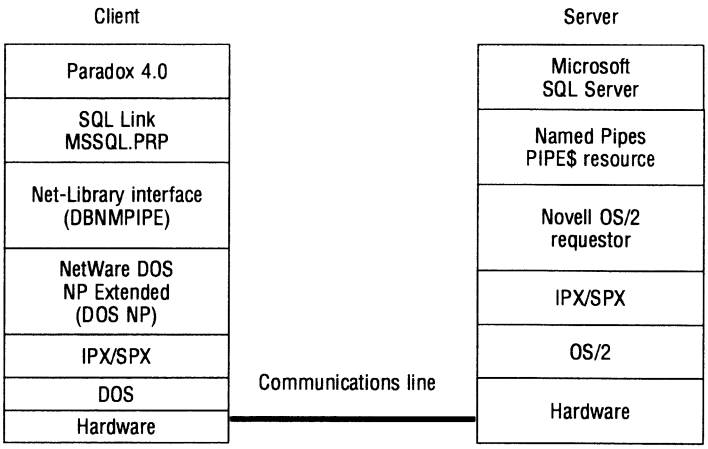


Figure 3 shows the SYBASE SQL Server architecture for UNIX.

Figure 3 SYBASE SQL Server architecture for UNIX

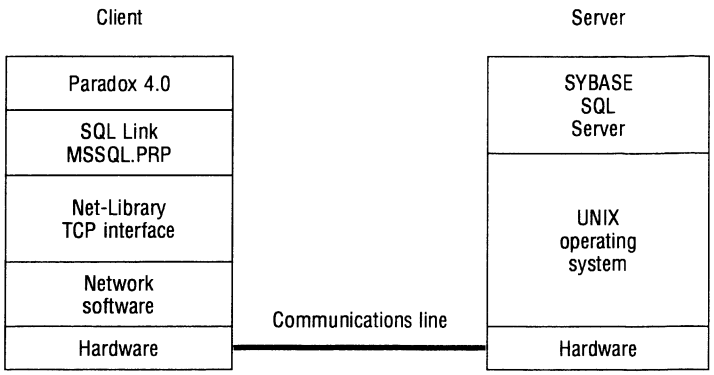
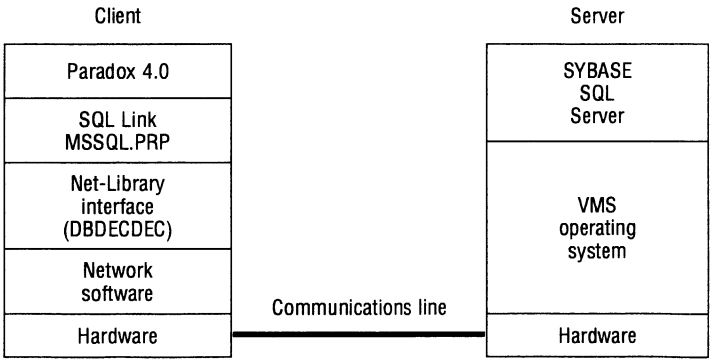


Figure 4 shows the SYBASE SQL Server architecture for VMS.

Figure 4 SYBASE SQL Server architecture (VMS)



See your Microsoft or SYBASE manuals for information on other hardware platforms, or for more information on specific requirements.

SQL Link must be installed in the Paradox system files directory (usually PDOX40). This directory can be on the local hard disk of a workstation, or on the shared disk of a network file server. Although you can name this directory differently, the SQL Link INSTALL program uses PDOX40 as the default. The Paradox driver for SQL Server (MSSQL.PRP) must be stored in this directory as well.

Installation steps

These installation instructions tell you how to install the SQL Link files, SQL Setup, and the SQL sample application into your Paradox system files directory.

If you have a PARADOX.DSQ file

If you have already defined a list of customized connections (a PARADOX.DSQ file), you must update the connection list each time you install a new SQL Link connection. Follow the instructions in Chapter 7 of the SQL Link *User's Guide* for adding a new connection.

The SQL Link Installation Disk contains an installation program, called INSTALL.EXE, that verifies your hardware setup and copies the SQL Link program files into your Paradox system files directory. Before you install SQL Link, make a backup copy of the SQL Link system disks, and keep the copy in a secure place.

To run INSTALL,

1. Insert the SQL Link Installation Disk into drive A. At the DOS prompt, type

   a:install

   and press *Enter*. (If you use drive B, type **b:install** and press *Enter*.) You'll see the introductory screen for installation.

2. Press *Enter* again to continue, and follow the prompts.

3. Choose the sample application option if you want to copy the SQL sample application to your hard disk. To install the sample application on your server, run the *Sqlinst* script from within Paradox. You will need CREATE TABLE and CREATE INDEX privileges on your server to complete this operation. For more information on the sample application, see Chapter 8 of the SQL Link *User's Guide*.

Note At any time during installation, you can press *Esc* to exit installation and return to DOS.

Information you need

Before starting SQL Link, make sure you have the information you need to log on to your server. Also, read the READSQL.SS file for additional information. To read this file, make the Paradox system directory the current directory and type

readme readsql.ss

To print this file from the Paradox system directory, type

print readsql.ss

Before starting Paradox

Be sure to load the appropriate network support driver before starting Paradox. Here's an example using the Named Pipes protocol:

```
cd\pdox40
dbnmpipe
```

This driver is a TSR (terminate-and-stay resident) program so you must load it before starting Paradox. To remove the driver from memory, use the ENDDBLIB command from the DOS prompt.

Starting Paradox

To start SQL Link, make the Paradox system files directory the current directory, and start Paradox with the usual command:

```
paradox
```

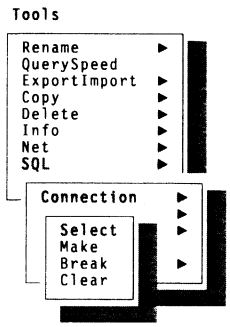
Without SQL

If you want to run Paradox without using its SQL capabilities, start Paradox with the following command:

```
paradox -sql off
```

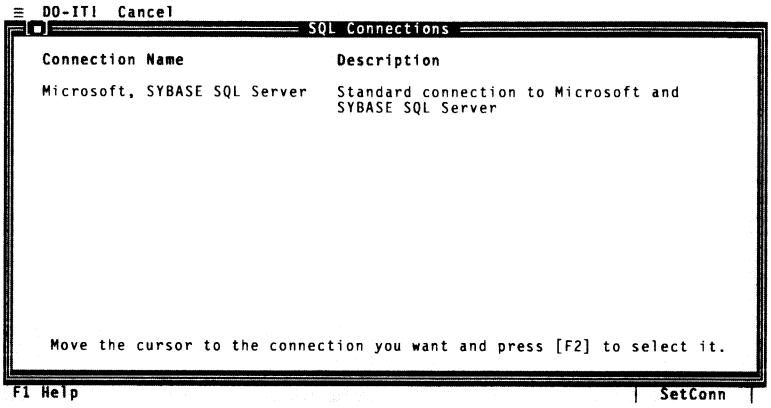
Selecting a Microsoft or SYBASE SQL Server connection

You must connect to your server before you can access a remote table. Regardless of what network you're using, you select a Microsoft or SYBASE connection as described here. To select a server connection,



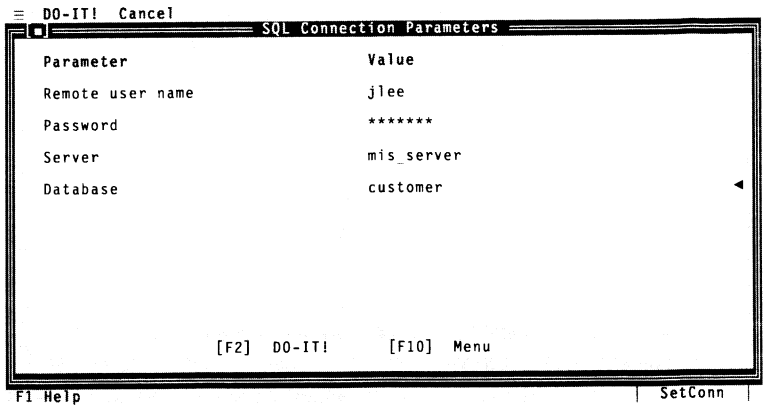
1. Choose Tools | SQL | Connection | Select from the Paradox Main menu.

   SQL Link displays a list of available server connections. Only those products that you install (or customize with SQL Setup) are displayed on this screen.



Note If you don't see a SQL choice on the Tools menu, there may be a problem with your SQL Link installation. Check the Paradox system files directory for PARADOX.OV3 and any file with a .PRP extension. If you don't see these files, reinstall SQL Link by following the installation steps described earlier in this addendum.

2. Select the server connection you want, then press *F2* Do_It!. SQL Link displays the parameters for the Microsoft/SYBASE connection.



3. Type the appropriate parameter values and press *F2* Do_It!. For information on the parameters, see the following section, "Connection parameters."

Connection parameters

The Paradox SQL Connections screen for Microsoft and SYBASE SQL Servers has four parameters, as shown in Table 4.

Table 4 Connection parameters

Parameter	Description	Example
Remote user name	Your authorization ID on the server. The authorization ID is assigned by the database administrator.	jlee
Password*	Your password on the server. The password is assigned by the database administrator. If omitted, defaults to NULL.	trapper
Server	The Workstation Name of your server.	mis_server
Database*	Defines the location where tables will be created. If omitted, you will connect to your default database.	accounts

* Optional parameter

Microsoft and SYBASE SQL Servers can be installed as case-sensitive; in this case, the user name, password, and database are case-sensitive.

For more information about selecting server connections and providing the connection parameters your server requires, see Chapter 5 of the SQL Link *User's Guide*.

Troubleshooting common connection problems

If you run into problems when you first use SQL Link to establish a SQL Server connection, try to isolate the problem using these steps:

1. Use your Microsoft or SYBASE tools to verify the connection at each layer. For Microsoft SQL Server, use the DOS System Administrator's Facility (SAF.EXE) to verify the connection. For SYBASE, use DBPING.EXE to check the network connection and ISQL.EXE to verify the server connection.

   Enter a valid query using either tool to verify your connection. For example,

   select @@version

   This query shows the server version number and thus verifies that you are connected to the server.

2. If you have file and print services, verify that the network layer is functioning by trying to share files and print jobs to the spooler.

3. Use hardware diagnostics to make sure your network interface card is working properly.

For more information on your vendor-supplied diagnostic tools, see your server documentation.

Network and server access privileges

SQL Link fully supports network and database server security. In addition, Paradox passwords can be used to enhance server security. See the Paradox *User's Guide* for information on Paradox security.

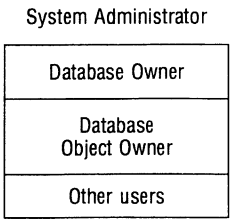
To access a SQL Server, you need a valid user name and password in addition to two types of access privileges: network and database server.

Network Your network administrator (the person in charge of managing the network) can help you obtain privileges on your network.

Database server You need a valid user name and password on your SQL Server database server. Your system administrator (the person in charge of managing your database server) can help you obtain privileges on your server.

Figure 5 shows the SQL Server permission hierarchy:

Figure 5 SQL Server permission hierarchy



If you are the system administrator, you operate outside the permission system and always have all privileges. In SQL Link, you can grant privileges using SQL...ENDSQL or UseSQL. Before granting privileges, you must add the user to the SQL Server with the SAF or with the stored procedures *sp_addlogin* and *sp_adduser*.

If you are not the system administrator, you must explicitly be granted object privileges (SELECT, INSERT, UPDATE, DELETE, and EXECUTE) on specific tables before accessing SQL Server data. If you are not a database owner or system administrator, you must be granted command privileges to create new objects (tables, views, and so on). Command privileges include CREATE TABLE, CREATE VIEW, CREATE DEFAULT, and CREATE PROCEDURE. If you don't have proper access permissions, Paradox returns the message **Insufficient privileges to complete operation**.

If you are the database owner, you have permission to use only those objects you create; you must be granted privileges to use objects created by others, even in databases that you own.

Once you create a table, you are its owner. You will need to grant privileges to others if they are to use your table.

See your server documentation for more on privileges.

SQL Server system tables

SQL Server includes a special set of tables called *system tables*, which provide information about production tables. You can access the system tables from Paradox with SQL...ENDSQL or UseSQL. You can also create replicas for system tables with SQL Setup and then query them with SQL Link just as you would any other table. System tables can answer questions about topics such as server privileges and remote table structures. For instance, if you receive the **Replica is out**

of date error, you can check the structure of the table in question by querying the system tables or by using the *sp_help* stored procedure.

The following table lists SQL Server system tables you might want to access through SQL Link.

Table 5 SQL Server system tables

Table name	Use
syscolumns	Lists each column in a table or view.
sysdepends	Lists each procedure, view, or table referenced by a procedure, view, or trigger.
sysindexes	Lists each clustered index, nonclustered index, and table with no index.
syslogins	Lists each valid user account.
sysobjects	Lists each object for the database.
sysprotects	Lists user permission information.
sysusers	Lists each user and group for the database.
syscolumns	Lists each column in a table or view.

In addition to the system tables, SQL Server provides *system procedures* to simplify querying and updating the system tables. System procedures are stored procedures (described in the next section) that operate on system tables.

Stored procedures

SQL Server's *stored (system) procedures* support many tasks (such as adding new users, creating primary keys, and reporting information on locks). The *sp_helpxxx* procedures in particular can help you troubleshoot problems by giving you information on objects in the database.

If you're using the Microsoft SQL Server, you can use the *sp_helpsql* procedure, which lists the syntax of TRANSACT-SQL commands. You can also create a replica for the *Helpsql* table (a system table stored in the Master database) and query its contents for help on SQL syntax.

You can run stored procedures in SQL Link with SQL...ENDSQL or UseSQL. If a stored procedure returns multiple answer sets, however you can view only the first answer in Paradox.

If you write your own stored procedures, use the RAISERROR command, not the PRINT command, to communicate errors to the user. RAISERROR calls Paradox's error handler and sets SQLERRORMESSAGE() and SQLERRORCODE(). See the section on RAISERROR later in this addendum for details.

Triggers

SQL Server *triggers* are special stored procedures. Triggers respond to a certain action. For example, a trigger could be used to update a record in another table when you delete a record from the current table. Following are some notes on using triggers with SQL Link, along with some examples of useful triggers.

- The definition and existence of a trigger are transparent to SQL Link.
- Use RAISERROR rather than PRINT in your stored procedures including triggers. RAISERROR returns an error number and message to Paradox, which you can use to determine if an operation succeeds.
- You can create and drop triggers with SQL...ENDSQL or UseSQL.

Getting the syntax of a trigger

To check the syntax of an existing trigger, first use SQL...ENDSQL to find the trigger name:

```
SQL select name, owner from sysobjects where type = "TR" ENDSQL
view "answer"
```

Then use

```
SQL sp_helptext "trigger name" ENDSQL view "answer"
```

to see the trigger code in a Paradox table.

A trigger that deletes a row

The following trigger deletes corresponding customer invoices in the *Invoicer* table when a customer is deleted from the *Customer* table. This trigger, created in a PAL script, is permanently associated with the *Customer* table until the trigger is dropped (using DROP TRIGGER). The *Customer* table contains columns for Custid and Name, and the *Invoicer* table contains columns for Invno and Custid. The *Inserted* and *Deleted* tables temporarily hold values inserted and deleted from a SQL table.

```
SQL
   create trigger Outinvoice on Customer
      for delete as
      begin
         delete Invoicer
            from Invoicer, Deleted
            where Invoicer.Custid = Deleted.Custid
      end
ENDSQL
```

A trigger that archives deleted values

You can create a trigger for a single table as well. This trigger fills an archive table with updated values when you attempt to delete an entry. If you choose Tools | More | Empty | Table from Paradox, all values in *School* are deleted, then transferred to an archive table

(*Arcschool*). *Arcschool* must already exist. This transaction occurs on a SQL Server, and is transparent to SQL Link.

```
SQL
   create trigger Schoolt on School
      for delete as
      begin
         insert into Arcschool(Name,Enrolled,Graduated)
         select Name, Enrolled, getdate() from Deleted
      end
ENDSQL
```

Defaults

On SQL Server, a *default* defines what value to enter in a field if the user doesn't make an explicit entry. Table 6 summarizes SQL Link support of defaults in various SQL Link operations that add data to the database.

Table 6 Paradox support of SQL Server defaults

Feature	Summary of Paradox support
Tools|More|Add	Paradox specifies that blank rows are NULL. If a column is defined as NOT NULL and you try to insert a blank row, an error results and the value is stored in a *Problems* table.
Creating Paradox picture strings and Paradox defaults on DataEntry	Paradox picture strings are supported but you must use *F2* Do_It! to save your defined validity checks. If you cancel your DataEntry session prior to saving, your validity checks are lost. You can also create defaults or pictures on an *Answer* table and use Tools|Copy|JustFamily to copy the family to the replica. You can use pictures to enforce formatting rules as well as default values.
INSERT QUERIES	Empty columns receive the default value. In this case, however, if the user specifies Null, the field will contain a NULL value. Columns with the Paradox keyword BLANK receive a NULL value.

See your SQL Server manuals for more information on creating and using defaults.

Rules

Rules enforce integrity constraints beyond those implied by the column's data type. When bound to columns, rules provide validity checks during data entry.

In DataEntry mode, rows violating an established server rule are stored in a *Problems* table. Processing is the same as it is for key violations. You can use *sp_helptext* to display the syntax of rules.

The following table summarizes Paradox's validity checks and the SQL Server features that correspond to the Paradox features.

Note Paradox's validity checks operate at the field level, while SQL Server's features operate at the table or row level.

Table 7 Paradox validity checks and SQL Server features

Paradox validity check	SQL Server feature
LowValue	Rule or trigger
HighValue	Rule or trigger
Default	Default
TableLookup	Foreign key constraint to check values against a primary key in another table
Picture	Rule to force the values to comply with a particular format*
Required	Not Null constraint

* Paradox completes the fill-in, SQL Server does not. Validity checks are local to Paradox; the presence of a ValCheck attribute on a field does not imply the presence of a rule on the remote table. Creating a validity check on a replica does not create a new rule on the remote table.

The RAISERROR command

You can use the RAISERROR command in a TRANSACT-SQL statement to return a user-defined error. When you use RAISERROR, SQL Link sets the ERRORCODE() to 1000, returns the user-defined error number in SQLERRORCODE(), and returns the user-defined error message in SQLERRORMESSAGE().

If a user-defined error occurs while adding a record to the table, the row where the flag was raised is returned in the *Problems* table, but the Paradox error handler is not called. (Invoking the error handler during this operation would roll back the entire transaction.) To see the error message, place the values from the *Problems* table into an INSERT query and process each row individually.

Here's an example of RAISERROR, using the *Customer* and *Invoicer* tables from earlier examples.

```
create trigger checkvalues on Invoicer
for insert as
  if (select count(*)
      from customer, inserted
      where customer.custid = inserted.custid) = 0
  begin
    raiserror 20005 "You must enter customer ID number."
  end
```

The trigger in the previous example forces the user to enter a value in the *Invoicer* table, which has a key that matches the Custid field in the *Customer* table. The user sees an error message if there is no corresponding Custid in the *Customer* table.

SQL Link and SQL Server views

A *view* of a SQL table is like a window that shows a specified portion of one or more tables or views. You can access views just as you do tables by using SQL Setup to create replicas for the views you want to access.

Views provide an additional level of table security and can also be used to hide complex data relationships. A view is simply a table to anyone who accesses it, yet that view could be a subset of columns from several tables.

Using views on wide tables

The maximum Paradox record size for keyed (that is, indexed) tables is 1350 bytes; for unkeyed (unindexed) tables, it is 4000 bytes. To replicate tables with larger record lengths, create a view on a subset of the fields in these tables, then replicate the view.

SQL Server data types

SQL Server to Paradox

When you query remote tables, SQL Link maps SQL Server data types to Paradox data types. Table 8 lists SQL Server data types and their corresponding Paradox data types.

Table 8 SQL Server to Paradox data type mapping

SQL Server	Paradox
varbinary	A4–A254 (query only; returns a hexadecimal value)
timestamp	A18 (query only; returns a hexadecimal value)
varchar(n)	A1–A255*
binary	A4–A254 (query only; returns a hexadecimal value)
char(n)	A1–A255*
bit	A4 (returns a hexadecimal value, 0x00 or 0x01)
real	N
int	N
float	N
tinyint	S
smallint	S
money	$
smallmoney	$
datetime	D
smalldatetime	D
image	A18 (BLANK)
text	A18 (BLANK)

* Data longer than 255 characters is truncated.

Paradox displays timestamp data as A18.

When you create a replica of a table containing a timestamp column, Paradox assigns an A18 to that column. If you try to query a timestamp column, Paradox returns a hexadecimal interpretation of the timestamp value.

Paradox to SQL Server

When you create or copy to a remote table, SQL Link maps Paradox data types to SQL Server data types. Table 9 lists Paradox data types and their corresponding SQL Server data types.

Table 9 Paradox to SQL Server data type mapping

Paradox	SQL Server
A1..A255	varchar(n)
N	float
D	datetime
S	smallint
$	money

Paradox converts user-defined data types.

TRANSACT-SQL lets you name and design your own data types. When you create a replica for a table with a user-defined type, Paradox maps it to a corresponding Paradox type, based on the base type.

Any rules or defaults bound to the data type are supported when you insert data.

Query restrictions

Be aware of these restrictions when querying remote tables:

- SQL Server limits the number of fields you can sort to 16. If you check more than 16 fields while connected to SQL Server, Paradox creates an ORDER BY statement with only the first 16 fields listed.
- Multiline INSERT queries based on only one table are not allowed. (You cannot insert values from a table into that same table.)

Error handling

This section describes error information specific to your server. General error handling for SQL Link is described in Chapter 6 of the SQL Link *User's Guide*.

The *Sysmessages* system table stores all SQL Server error messages.

The following error message is specific to the Microsoft and SYBASE connection. The Paradox error code and a possible solution are included.

Database or disk full (1011)
Either the database or the transaction log is full. See your SQL Server documentation for information on dumping the transaction log.

SQL Server locking

This section briefly describes SQL Server's locking operation. See your SQL Server manuals for more information on server-controlled locking.

Typical database management systems issue locks on a record or set of records while that data is in use. When the lock is released, the data is again available to other users. Locks preserve the integrity of your data, and successful locking mechanisms provide the highest level of concurrent data access.

Paradox relies on your server's locking mechanisms. To trace the server's locks, you need to understand the types of locks SQL Server produces. SQL Server supports shared and exclusive locks, as described in Table 10.

Table 10 SQL Server locks

Lock type	Description
Shared	A lock that allows other shared locks to be placed against the same data. You can hold a shared lock even if there is already a shared lock in place.
Exclusive	A lock that gives exclusive access to data. You cannot place an exclusive lock if there is already a shared or exclusive lock on the data.
Demand	Prevents further shared locks. Indicates that a transaction is next in line to lock a table or page.

Use the *sp_lock* and *sp_who* stored procedures to report information on transactions that currently hold locks. Table 11 shows how SQL Server's locking mechanisms work with common Paradox operations (nonkeyed tables assumed except where noted).

Table 11 SQL Server locking for Paradox operations

Operation	SQL Server lock type	Lock types blocked
CHECK query	None	None
CHECKPLUS query	Shared page lock	Exclusive locks
Tools|More|Add|NewEntries	Exclusive lock	All
INSERT query	Exclusive lock	All
Tools|More|Add|Update	Exclusive lock	All
Tools|More|Empty	Exclusive lock	All
DELETE query	Exclusive lock	All
CHANGETO query	Exclusive lock	All

When AutoCommit is set to No, SQL Server users will be blocked from completing most operations until the user holding the locks commits the pending transaction. PAL developers should frequently commit transactions in a multiuser environment. When AutoCommit is set to No and you issue a CHECK PLUS query, the HOLDLOCK keyword is added to the query sent to the server.

You can nest transactions. For each transaction started, issue a COMMIT. If the number of commits does not equal the number of transactions started, open locks remain on the server. To determine if multiple transactions have been started, use the following SQL Link statement:

```
SQL select @@trancount ENDSQL view "answer"
```

In nested transactions, SQLROLLBACK or Tools | SQL | Transaction | Rollback will roll back to the first SQLSTARTTRANS in the transaction.

Deadlocks and shared locks

Deadlock describes the situation where two or more users cannot complete transactions because each user is holding resources that the other requires. Because SQL Server has multiple levels of locks, deadlocks can occur when two users are sharing a lock and each user requires a more restrictive lock.

You can trap for deadlocks in an error-handling procedure by looking for ERRORCODE()=1002.

Support for the Tandem SQL Server Gateway

Together with the SQL Server Gateway from Tandem, this release of SQL Link provides connectivity to Tandem NonStop SQL data. The SQL Server Gateway runs on Tandem NonStop systems and allows the Tandem system to emulate a SQL Server. The gateway accepts TRANSACT-SQL statements and converts them to NonStop SQL. In the connection parameters screen, select the Microsoft, SYBASE SQL Server connection and specify the user name, password, server name, and database name appropriate for your gateway. The server name is case-sensitive for the gateway.

With the SQL Server Gateway, you can use Paradox to access data stored on a Tandem system. You do not need to transfer the data to a PC, and you do not need to translate, import, or export the data.

Tandem recommends that you use the gateway as a decision-support tool. As such, you should not use it for online transaction processing. Paradox acts as a query tool in this situation, reducing the time required to construct queries on remote data.

If you use Paradox with the SQL Server Gateway, be aware of these restrictions:

- Do not include string manipulations within a SELECT statement.
- Exercise caution when comparing floating-point numbers. Exact comparisons might not return valid results.

Not all SQL Server features are supported by the gateway. For a complete list of gateway support, refer to the Tandem SQL Server Gateway manuals.

Additional reference tables for Microsoft and SYBASE SQL Servers

Table 12 lists general items that you might find helpful in working with SQL Server.

Table 12 General information

Item	Description
Does the server require an explicit SQLSTARTTRANS for multistatement transaction processing?	Yes
Product name/SQL dialect	MSSQL
Case-sensitive for data?	As installed
Case-sensitive for objects?	As installed

Table 13 lists field-naming rules.

Table 13 Field-naming rules

Naming rule	Paradox	Microsoft, SYBASE SQL Server
Max length (characters)	25	30
Valid characters*	All except " , [,], {, }, (,), –, >	All except spaces and –
Must begin with	Any valid character	Letter

* Paradox field names cannot be the symbol # alone.

Note You cannot use reserved words as object names. See the *PAL Reference* and your server manuals for a list of reserved words.

Table 14 shows the types of queries that lock records with AutoCommit set to No.

Table 14 Paradox-SQL Server query locks

Item	Description
Types of queries that lock records with AutoCommit = No	Check Plus
Method used by Paradox for locking	Shared lock

Note All locks are released when you commit or roll back a transaction.

PARADOX®
SQL LINK

BORLAND

Corporate Headquarters: 1800 Green Hills Road. P.O Box 660001. Scotts Valley. CA 95067-0001. (408) 438-5300 Offices in: Australia. Belgium. Canada. Denmark. France. Germany. Hong Kong. Italy. Japan. Korea. Malaysia. Netherlands. New Zealand. Singapore. Spain. Sweden. Taiwan and United Kingdom ▪ Part #23MN-PQS01-40 ▪ BOR 3156