Introduction

Welcome to the Butler SQL User’s Guide. This guide covers all aspects of configuring and maintaining a Butler SQL server. You should have Butler SQL installed using the instructions contained in the Getting Started Guide before reading the sections of this guide.

The four separate sections of this guide describe how to use Butler SQL, ButlerHosts, and the ODBC Setup control panel to

- create and manage ports,
- configure Butler SQL,
- connect client applications to Butler SQL, and
- backup and restore data.

The various options which Butler SQL provides will be discussed and demonstrated in each section of this guide. Instructions appear in **bold** type for quick reference; also, the instructions will be accompanied by notes, informing you of system specifics and recommended methods that will help you get the full benefit of Butler SQL.

Our intended audience

This guide is designed to introduce Butler SQL, DAL/DAM, and ODBC configuration to individuals who are unfamiliar with the functions and operations of a Butler SQL server. Those individuals who have never used client-server software should consult an introductory text to familiarize themselves with the general purpose and usage of such applications.

About Butler SQL databases

A Butler SQL database contains one or more tables. Each table has a name and contains specific columns and rows. A cell is the intersection of a row and a column.

These columns list the properties of the data that may be contained in a database. Columns in a table are made up of data items of a single type. For example, in an employee table, the employee name, occupation, Social Security number, and home telephone number would each be separate columns. Every table must have at least one column and each column must be assigned a unique name in the table.

A row is a collection of values which are related. A row in an employee table could consist of the employee name, employee number, home address, and Social Security number.

Databases on a server can be used by many clients. When a client application is connected to a Butler SQL server, the client can access the tables of the databases located on the server.

Physical connections to the server are made through ports. Ports are used in order to allow clients to connect to a server using different protocols, such as program linking or TCP/IP.

Client applications connect to Butler SQL using DAL/DAM or ODBC. On the client machine, DAL/DAM connections are configured using the ButlerHosts application; ODBC connections are configured using the ODBC Setup control panel.

Chapter 1 • Connecting to Butler SQL

DAL/DAM hosts and ODBC data sources created on client machines connect to ports on Butler SQL. This section describes how to create and manage Butler SQL ports on the server machine.

If you want to follow the steps described in this chapter, you should be at the Macintosh running Butler SQL.

Port Types

There are three port types used in communicating to Butler SQL:

- Program Link ports,
- Communications Toolbox (CTB) ports, and
- TCP/IP ports.

Each of these port types differ in the protocol they use for allowing communication with the server. When creating a port, select a port that uses the most appropriate protocol for your environment or application. For example, if you need to connect to Butler SQL from Windows or from client machines connected over the internet, use a TCP/IP port.

You can create as many ports as you like. For example, you may want to create a separate port for each application that uses Butler SQL. However, when creating ports, keep in mind that each port does use some of Butler SQL's available memory. See the "Memory Allocation & Buffers" section on page 65 for more information on how Butler SQL uses its memory.

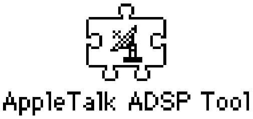
Program Link Ports

Program Link ports allow Macintosh applications to communicate with Butler SQL servers using the program linking feature of System 7. Clients must be running Macintosh System 7 or greater to access a Butler SQL server via this type of port.

You can use program linking to connect to a Butler SQL server running on the same machine or on another machine connected to your AppleTalk network.

Communications Toolbox (CTB) Ports

The Communications Toolbox is an integrated part of the Macintosh System 7 software and an optional part of System 6. It allows client applications to communicate with Butler SQL using the ADSP tool. (Currently, Butler SQL supports the AppleTalk ADSP tool only.) The Butler SQL installer places the AppleTalk ADSP tool in the Extensions folder of your System Folder.



Note

Communications Toolbox support is provided for Macintosh System 6 and 7 users connecting to Butler SQL servers using the Data Access Manager (DAM) only. Communications Toolbox ports are not available for ODBC connections. At the time of writing, support for the Communications Toolbox was uncertain in future Apple System Software releases. Unless your application requires the functionality offered by the AppleTalk ADSP tool, you should consider using Program Linking or TCP/IP ports.

TCP/IP Ports

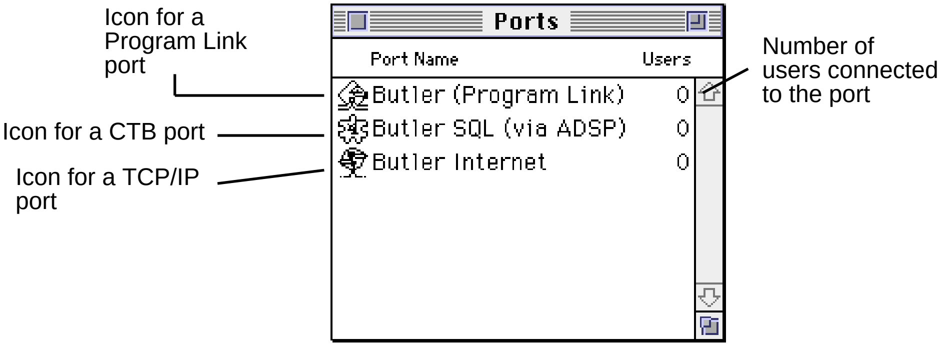
TCP/IP (Transmission Control Protocol / Internet Protocol) is a platform-independent networking protocol. Using TCP/IP ports, both Macintosh and Windows client applications can connect to Butler SQL servers. In addition, you can use TCP/IP ports to connect to a Butler SQL server over the Internet.

Where Ports are Stored

Each port is stored as a separate file in the Ports folder in the Butler Preferences Folder. When Butler SQL is started, it looks in this folder and tries to install each port it finds there. If no ports are found, a default Program Link port is created.

The Ports Window

Installed ports are displayed in Butler's Ports window. To open the Ports window, choose Show Ports from the Options menu.



Ports

Port Name	Users
Butler (Program Link)	0
Butler SQL (via ADSP)	0
Butler Internet	0

Icon for a Program Link port
Icon for a CTB port
Icon for a TCP/IP port
Number of users connected to the port

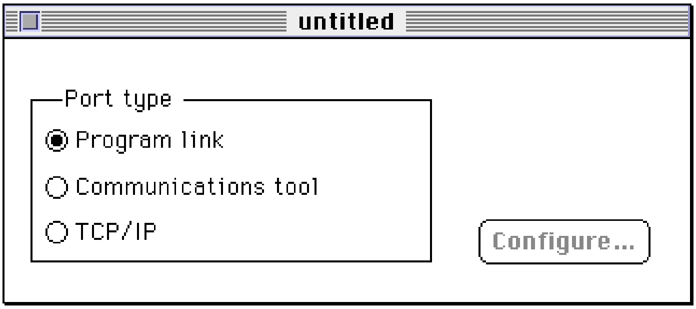
The icons of enabled ports are drawn normally. Disabled port icons are drawn with a slash through them.

Creating a Port

To create a new port:

1. **Choose New Port from the File menu.**

   An untitled port type window appears.



Port type

● Program link
○ Communications tool
○ TCP/IP

2. **Select the Port type.**

   Communications Tool and TCP/IP ports require configuration. See the “Configuring A Communications Toolbox Port” section on page 7 and the “Configuring a TCP/IP Port” section on page 7 for more information.

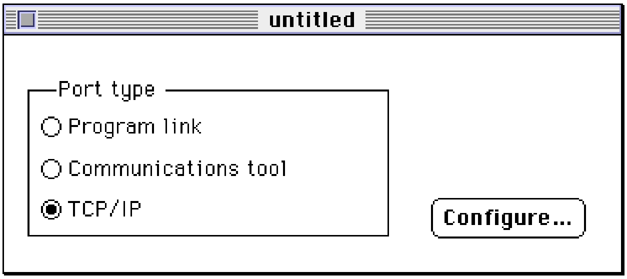
3. **Choose Save from the Edit menu.**

   Once configured, new ports must be installed as described in the “Installing a Port” section on page 10 before they can be available for client connections.

Configuring a TCP/IP Port

TCP/IP ports can be used by ODBC applications to connect to Butler SQL.

To configure a TCP/IP port for Butler, perform the following steps:



Port type
○ Program link
○ Communications tool
● TCP/IP

1. **Select TCP/IP in the Port Type window.**

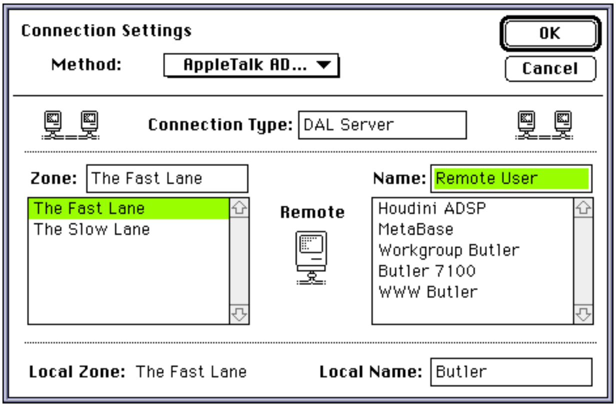
   Butler SQL assigns TCP/IP ports the port number 8682. Normally, you do not need to change this. However, if it conflicts with another TCP/IP service running on the same machine, click the Configure... button and enter a different port number.

Configuring A Communications Toolbox Port

If you selected "Communications tool" in the port type window, continue with the following steps.

1. **Click Configure.**

The Connections Settings dialog box appears.



Connection Settings

Method: AppleTalk AD...

Connection Type: DAL Server

Zone: The Fast Lane

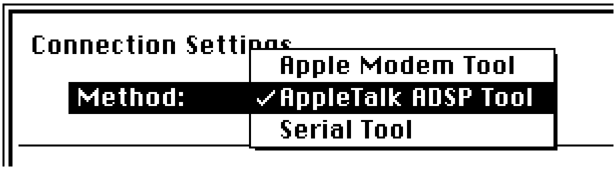
Name: Remote User

Local Zone: The Fast Lane

Local Name: Butler

2. **Choose the AppleTalk ADSP tool from the Method pop-up menu.**

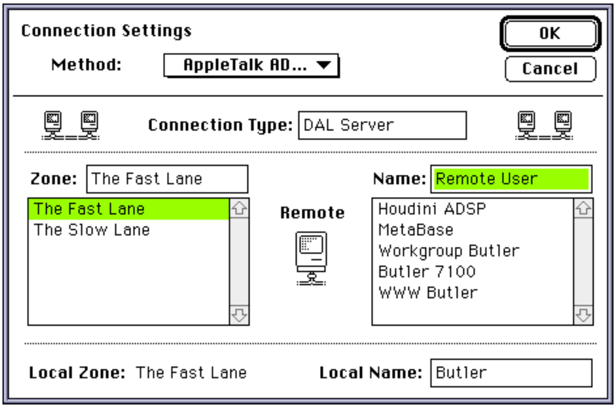
The names of all installed CTB tools are listed in the Method pop-up menu.



Connection Settings

Method: ✓AppleTalk ADSP Tool

After choosing a tool, the Connection Settings dialog box changes to display the tool’s configuration options.



Connection Settings

Method: AppleTalk AD...

Connection Type: DAL Server

Zone: The Fast Lane

The Fast Lane
The Slow Lane

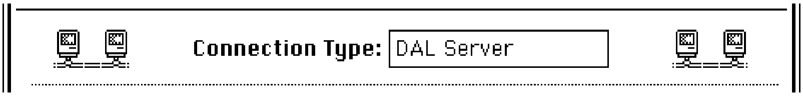
Name: Remote User

Houdini ADSP
MetaBase
Workgroup Butler
Butler 7100
WWW Butler

Local Zone: The Fast Lane

Local Name: Butler

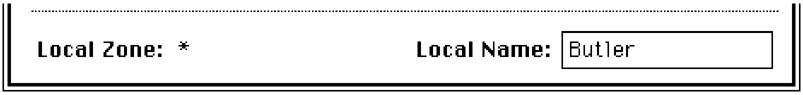
3. **Type `DAL Server` in the Connection Type field.**



Connection Type: DAL Server

4. **In the Local Name field, type the name of this port as you want it to appear to clients configuring a host connection.**

   If you will have more than one Butler server with ADSP ports, it helps to make this name meaningful, for example: Sales Database Server. If you have more than one Butler server in the same zone, make sure that each server's Local Name is different.



Local Zone: *

Local Name: Butler

Note

The Zone and Name lists in the AppleTalk ADSP Tool Connection Settings dialog box are used for configuring client connections only. You do not need to select from these lists to configure a port.

5. **Click the OK button.**

   The Connections Settings dialog box closes.

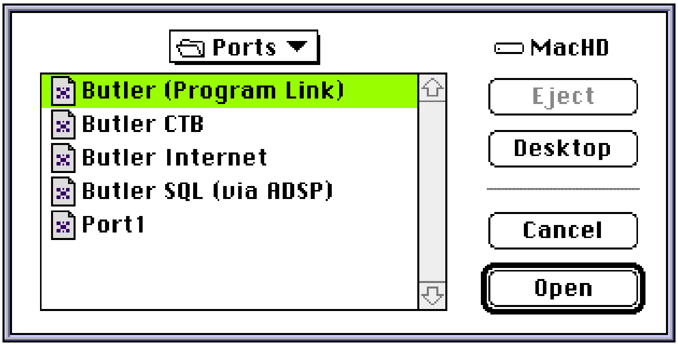
Installing a Port

Installing a port makes it available for client connections to the server. When you start Butler SQL, any existing ports are installed automatically.

To install a port:

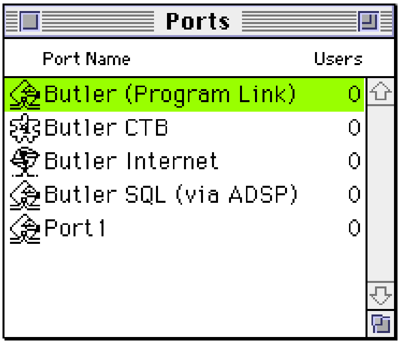
1. **Choose Install Port... from the File menu.**

   A dialog box appears for selection of the port file to install.



2. **Select a port file and click Open.**

   The newly-installed port appears in the Ports window.



Ports

Port Name	Users
Butler (Program Link)	0
Butler CTB	0
Butler Internet	0
Butler SQL (via ADSP)	0
Port 1	0

Opening a Port

Ports which are not installed, that is, those that do not appear in the Ports window, can be opened to display or change their settings.

To open a port:

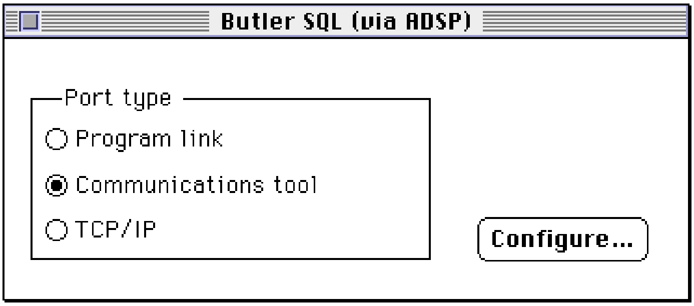
1. **Choose Open Port... from the File menu.**

   A directory dialog box appears for selection of the port file.

2. **Select a port file and click Open.**

   The port type window appears with the information for the port you selected. If you selected a port file that is installed, Butler will display an alert to let you know that the port is already open.

You can now change the port type and the configuration of the port.



Butler SQL (via ADSP)

Port type

○ Program link

● Communications tool

○ TCP/IP

Editing an Installed Port

Editing a port allows you to change its type (Program Link, CTB, or TCP/IP) and, in the case of CTB ports, to choose and configure the particular CTB tool.

To edit an installed port:

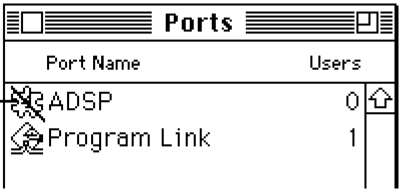
1. **Make sure there are no clients using the port you want to edit.**

   The Ports window indicates the number of users connected to a given port.

2. **Select the port in the Ports window.**

3. **Choose Disable Port from the Options menu.**

   The selected port's icon is redrawn with a slash through it, indicating that the port is disabled.



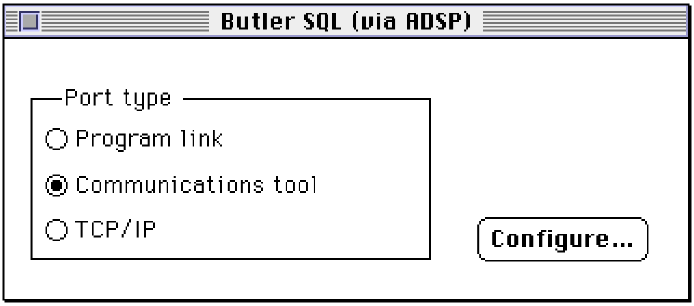
Port Name	Users
ADSP	0
Program Link	1

Slash indicates disabled port

If there are users connected to the port when you disable it, Butler SQL will ask if you want to log off the users. If you proceed, the users connected to the port will be disconnected.

4. **Choose Edit Port from the Options menu.**

   The port type window appears.



Port type
○ Program link
● Communications tool
○ TCP/IP

You can also edit a port by double-clicking the disabled port in the Ports window.

5. **Set the port type by selecting the "Program link", "Communications tool", or "TCP/IP" radio button.**

   If you select "Program link," there is no further configuration necessary. For the CTB and TCP/IP port configurations, see "Configuring A Communications Toolbox Port" section on page 7 or "Configuring a TCP/IP Port" section on page 7.

6. **Choose Save from the File menu.**

Butler SQL saves the port with its new settings.

7. **Close the Ports type window.**

8. **Select the port in the Ports window.**

9. **Choose Enable Port from the Options menu.**

   The port is enabled.

Note

An installed Program Link port will be available for client connections only if your server's system software has been configured to allow program linking by users. Your system software documentation describes how to do this. You must also ensure that the "Allow program linking" option in the Butler Preferences dialog box is enabled. (See "Allowing program linking" section on page 53.)

Removing a port

Removing a port prevents client applications from connecting to the server using the port.

To remove a port:

1. **Make sure there are no clients connected using the port.**

   Check the users column in the ports window. If the number is not zero, you should warn clients using the port to disconnect before proceeding.

Important

Removing a port with active users is dangerous, especially if transaction processing is disabled. The result could be data corruption.

2. **In the Ports window, select the port to remove.**

3. **Choose Remove Port from the File menu.**

   If there are clients using the port, you are given a warning and the option to cancel your request. Click Cancel to leave the port installed or click OK to disconnect clients using the port.

   Removing a port removes the port's name in the Ports window.

Disabling a port

Disabling an installed port makes it unavailable to clients. Disable a port to prevent clients from connecting to the Butler SQL server or if you need to edit the port's settings.

To disable a port:

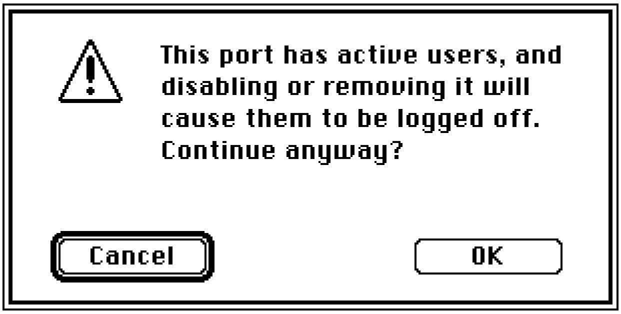
1. **Make sure there are no clients using the port you want to disable.**

   Check the users column in the ports window. If the number is not zero, you should warn clients using the port to disconnect before proceeding.

2. **In the Ports window, select the port to disable.**

3. **Choose Disable Port from the Options menu.**

   If there are clients using the port, you are given a warning and the option to cancel your request. Click Cancel to leave the port enabled; click OK to disconnect clients using the port.

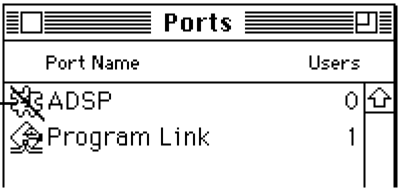


This port has active users, and disabling or removing it will cause them to be logged off. Continue anyway?

Important

Disabling a port with active users is dangerous, especially if transaction processing is disabled. The result could be data corruption.

The Ports window displays the disabled port with a slash through its icon.



Port Name	Users
ADSP	0
Program Link	1

Slash indicates disabled port

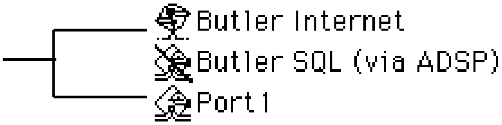
Enabling a port

Enabling a disabled port makes it available for client connections.

To enable a port:

1. **Select a disabled port in the Ports window.**
2. **Choose Enable Port from the Options menu.**

The slash is removed from the port's icon and the port is again available for client connections.



enabled ports

Chapter 2 • Client Connections to Butler SQL

Client applications can connect to Butler SQL using DAL/DAM and ODBC. This section describes how you use ButlerHosts to configure a DAL/DAM connection, and the ODBC Setup control panel to configure an ODBC connection.

These configurations are performed on client machines and not the machine running Butler SQL, unless the client application will run on the same machine as the server.

Choosing a Connection Method

Client applications can connect to Butler SQL using three different API’s (Application Program Interface): ODBC, DAL, or DAM. If you are using an existing application, your application’s documentation will describe which API it uses to communicate with database servers. If you are creating your own application, refer to the Butler SQL Programmer’s Guide for technical information on each API.

Macintosh Connections

Macintosh applications can be programmed to connect to Butler SQL using any or all of the following API’s:

DAL - Data Access Language - Refers to three different things: the manager, SQL dialect, and the driver. DAL connections use the “Butler DAL” database extension to connect to Butler SQL or Apple's 'DAL' database extension to connect to other DAL servers. This connection is configured using the ButlerHosts application and by editing the DAL Preferences file.

DAM - Data Access Manager - Refers to the manager and driver components with support for the DAL/SQL dialect. This is more robust than pure DAL and is preferred over the pure DAL connection whenever possible. A DAM connection uses the Butler Access database extension to connect to Butler SQL. This connection is configured using the ButlerHosts application.

ODBC - Open Database Connectivity - This is Microsoft's offering for cross-platform database connectivity. This is the best choice for cross-platform development and offers better performance than DAM. An ODBC connection is configured using the “ODBC Setup” control panel. The “Butler SQL ODBC Driver”, a shared library, is used by ODBC and your client application to communicate with the Butler SQL server.

When presented with options for connection from a Macintosh, you should consider them in this order: ODBC, DAM, DAL.

Each connection method also allows different protocols to be used:

DAL -- Program Linking, ADSP

DAM -- Program Linking, ADSP

ODBC -- TCP/IP, Program Linking

Of the three communication protocols to choose, Program Linking appears to be the fastest, whereas ADSP is considered more secure. TCP/IP allows connections via the internet as well as local or wide area networks, but is currently the slowest of the three methods.

Microsoft Windows Connections

If you are connecting from a Microsoft Windows application, your choices are simpler. Windows clients can only connect to Butler SQL using ODBC via TCP/IP.

Butler supports Windows applications running under Windows NT, 95, and 3.1 with the 32-bit extensions.

About ODBC connections

An ODBC connection is called a data source. A data source consists of a:

- name,
- database name,
- protocol (link type), and
- protocol configuration information.

ODBC data sources are configured using the ODBC Setup control panel. The following sections describe this configuration on Macintosh and Microsoft Windows computers.

It is assumed that you are familiar with, and have already set up, a Butler SQL server. If you have not, then please do so before you configure ODBC on client machines.

ODBC on Macintosh

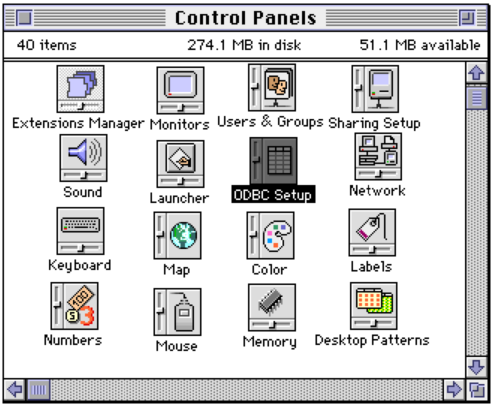
In order to set up ODBC Data Sources for Macintosh, you must have ODBC installed on your Macintosh system. See the Butler SQL Getting Started guide for assistance on the installation of ODBC.

Creating a Data Source

To set up an ODBC data source on a Macintosh:

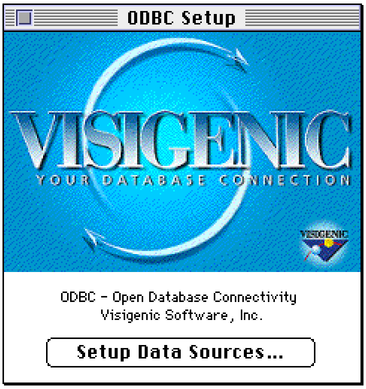
1. **Open the Control Panels folder in your System Folder.**

A window appears with the ODBC Setup control panel listed.



2. **Double-click on the ODBC Setup control panel icon to open it.**

   The ODBC Setup window appears.

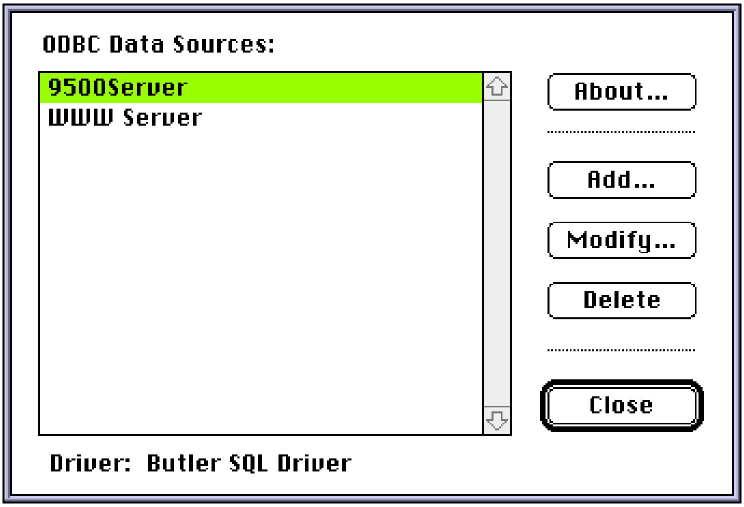


Note

If you have a Macintosh with a PowerPC processor, you may have two ODBC Setup control panels: ODBC Setup and ODBC Setup PPC. Currently, Butler SQL data sources can be created with the ODBC Setup control panel only.

3. **Click Setup Data Sources....**

   The ODBC Data Sources dialog box appears.



ODBC Data Sources:

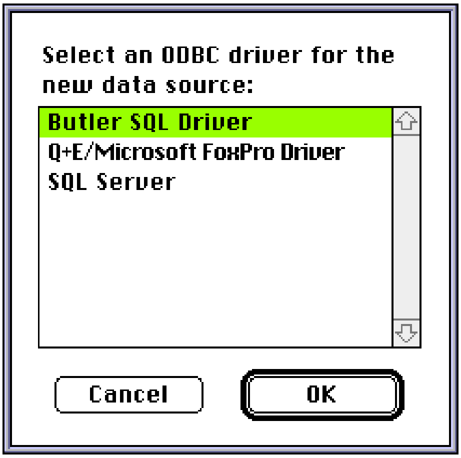
9500Server
WWW Server

Driver: Butler SQL Driver

The ODBC Data Sources dialog box lists any existing data sources created on your machine.

4. **Click Add....**

The select ODBC driver dialog box appears.



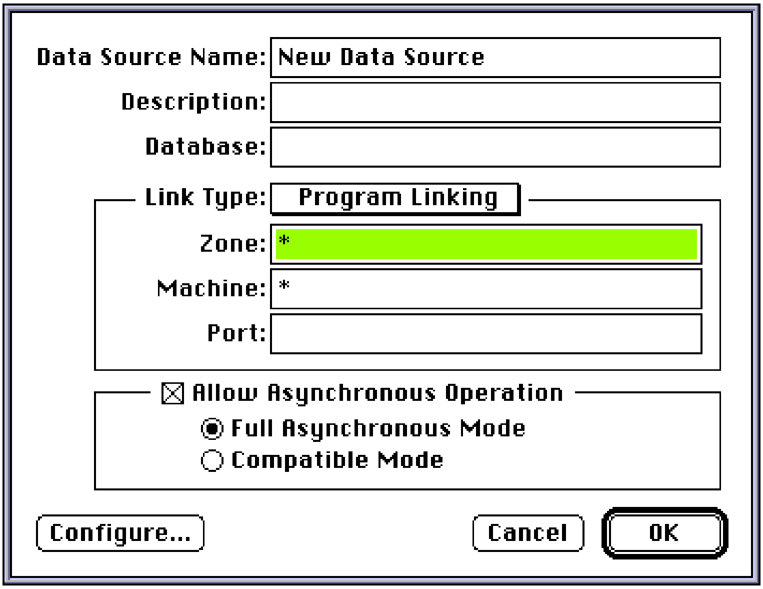
Select an ODBC driver for the new data source:

Butler SQL Driver
Q+E/Microsoft FoxPro Driver
SQL Server

This dialog box lists the ODBC drivers that are installed on your Macintosh. ODBC uses a driver to communicate with a particular brand of database. When you install Butler SQL, the Butler SQL ODBC driver is also installed on your machine.

5. **Select Butler SQL Driver and click OK.**

The ODBC data source configuration dialog box appears.



Data Source Name: New Data Source

Description:

Database:

Link Type: Program Linking

Zone: *

Machine: *

Port:

☒ Allow Asynchronous Operation

● Full Asynchronous Mode

○ Compatible Mode

6. **Enter a name for this data source. Press Tab.**

   Client applications use this name to identify the data source they want to use for their database connection.

7. **Optionally, enter a description into the Description field. Press Tab.**

   The description is a field you can use to record the purpose or usage of this data source.

8. **Enter the name of the database you want to use. Press Tab.**

   The database name must match the name of a database installed in the Public Databases folder on the Butler SQL server.

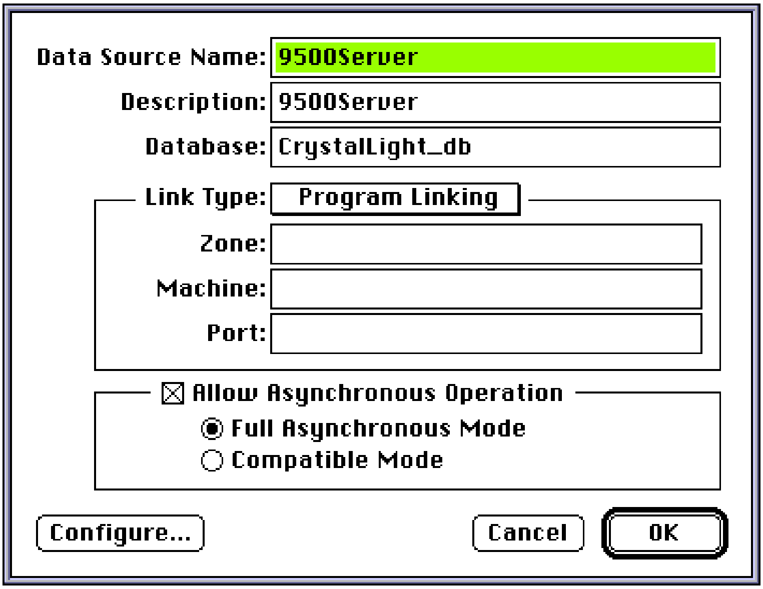
9. **Choose the Link Type from the Link Type pop-up menu.**

   You can use Program Linking or TCP/IP to connect to a Butler SQL server. In order to connect to Butler SQL from a client application, you must create a port on the server of the same type as the selected link type.

After choosing a link type, the other fields in the link type area change to allow configuration of the link type.

10. **If you are creating a TCP/IP link type, skip to step 17.**

To configure a Program Linking link type, you need to identify the Butler SQL server's network zone, machine name, and port name.



Data Source Name: 9500Server

Description: 9500Server

Database: CrystalLight_db

Link Type: Program Linking

Zone:

Machine:

Port:

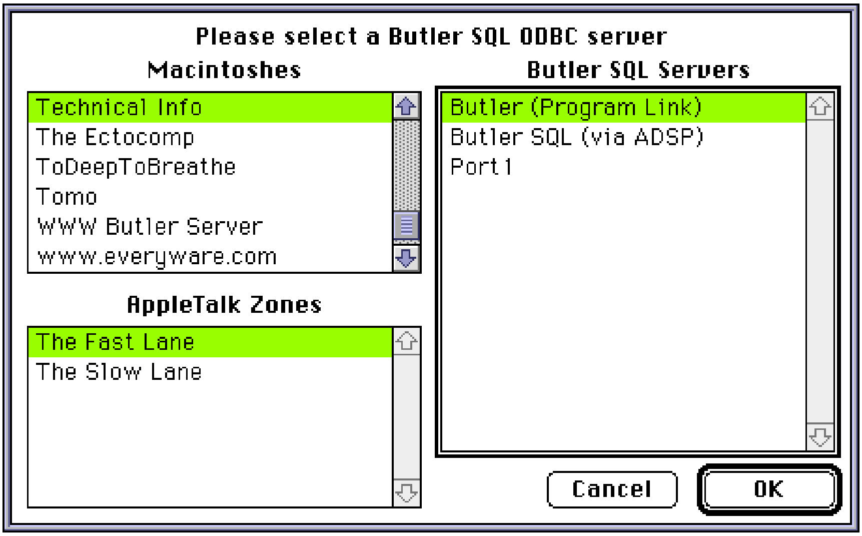
☒ Allow Asynchronous Operation

● Full Asynchronous Mode

○ Compatible Mode

You can enter these manually, or you can select these from a network browsing dialog box.

11. **To select the zone, machine name, and port name from the network browsing dialog box, click Configure....**



If your network does not have any zones, the AppleTalk Zones list will not be displayed.

12. **Select the network zone containing the Butler server to which you wish to connect.**

    The Macintoshes list changes to display the Macintosh computers connected to the selected network zone.

13. **Select the server's Macintosh name from the Macintoshes list.**

    The Butler SQL Servers list changes to list the installed and enabled Butler SQL ports on the selected Macintosh.

Note

If you don't see the server's name in the list, the machine may not be running or program linking may not be turned on. Make sure the machine is on, that a Program Link port is installed, and that program linking is enabled (both in Butler and the system software).

14. **In the Butler SQL Servers list, select the name of the Butler server's Program Link port.**

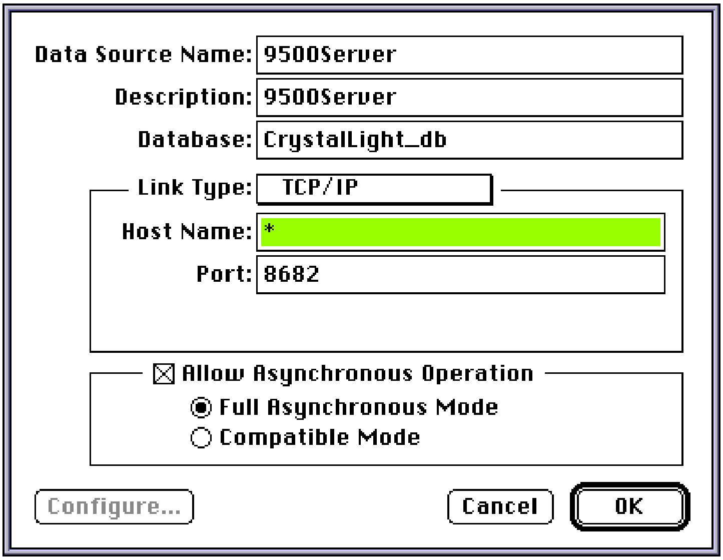
15. **Click OK.**

    The selected zone, Macintosh, and port are entered into the Link Type fields in the ODBC configuration dialog box.

    If you are connected to a network with only one zone, an asterisk ("*") will be entered into the zone field.

16. **To continue with the configuration of the data source, skip to step 19.**

To configure a TCP/IP link type, you need to enter the TCP/IP address of the Butler SQL server Macintosh and the port number assigned to the TCP/IP port in Butler SQL.



Data Source Name: 9500Server

Description: 9500Server

Database: CrystalLight_db

Link Type: TCP/IP

Host Name: *

Port: 8682

☒ Allow Asynchronous Operation

● Full Asynchronous Mode

○ Compatible Mode

17. **Enter the TCP/IP address of the server into the Host Name field. Press Tab.**

18. **If necessary, change the default port number.**

    The port number must be the same as the port number assigned to the TCP/IP port on the Butler SQL server. Normally, you do not need to change the default port number.

19. **Click OK to save the data source.**

    If you are finished modifying data sources, click the Close button in the ODBC Setup control panel window. This data source is now available to client applications.

Note

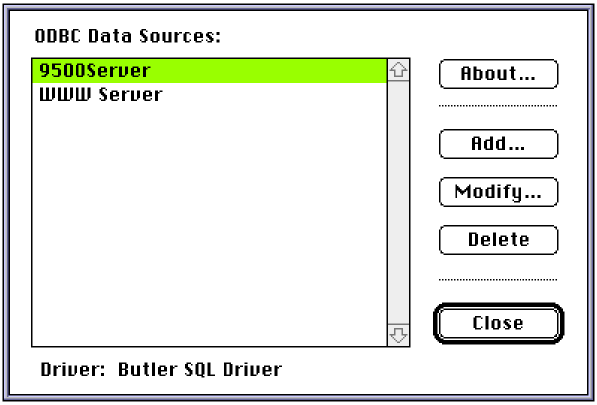
The values displayed in the Port field and the Allow Asynchronous Operation settings are defaults which do not normally need to be changed. The default values are Allow Asynchronous Operation and Full Asynchronous Mode.

Editing a Data Source

When editing a data source, you can change any of the details in the ODBC Data Source configuration dialog box. You cannot change the database driver that a data source uses.

To edit a data source:

1. **Select the desired data source in the ODBC Data Sources dialog box.**


ODBC Data Sources:

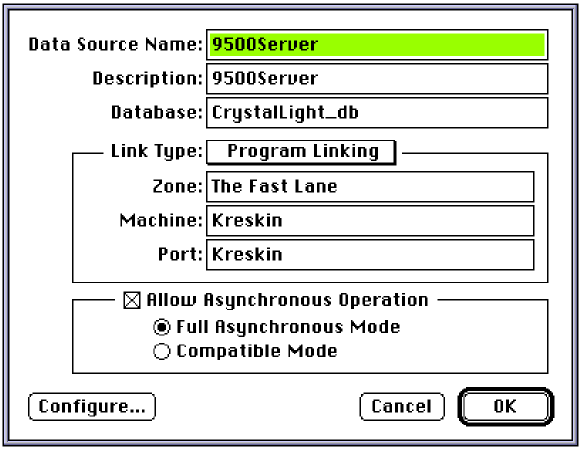
9500Server

WWW Server

Driver: Butler SQL Driver

2. **Click Modify....**

The ODBC Setup configuration dialog box appears. You can also edit a data source by double-clicking the data source in the ODBC Data Sources dialog box.


Data Source Name: 9500Server

Description: 9500Server

Database: CrystalLight_db

Link Type: Program Linking

Zone: The Fast Lane

Machine: Kreskin

Port: Kreskin

☒ Allow Asynchronous Operation

● Full Asynchronous Mode

○ Compatible Mode

3. **Make the desired changes to the data source.**

You can change any of the fields displayed in the dialog box. Refer to the "ODBC data sources are configured using the ODBC Setup control panel. The following sections describe this configuration on Macintosh and Microsoft Windows computers.” section on page 21 for a description of each field in this dialog box.

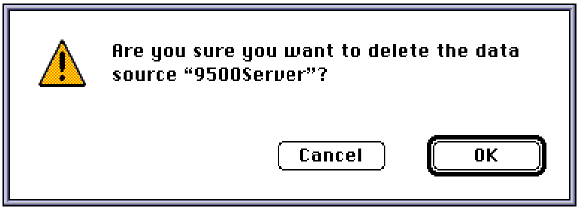
4. **Click OK to save the changes to the data source.**

Deleting a Data Source

To delete a Data Source:

1. **Select a Data Source in the ODBC Data Sources dialog box.**
2. **Click Delete.**

   A confirmation alert box appears.



Are you sure you want to delete the data source “9500Server”?

3. **Click OK to delete the data source.**

   The data source is removed from the ODBC Data Sources dialog box.

   If you are finished modifying data sources, click the Close button in the ODBC Setup control panel window.

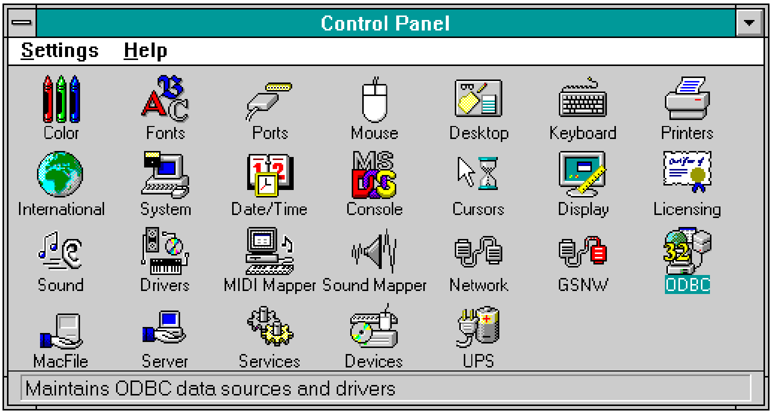
ODBC on Windows

Butler SQL allows ODBC Data Sources to be configured in a Windows system environment. In order to set up ODBC Data Sources for Windows, you must have ODBC installed on your Windows system.

Creating a Data Source

To create a new ODBC Data Source in Windows:

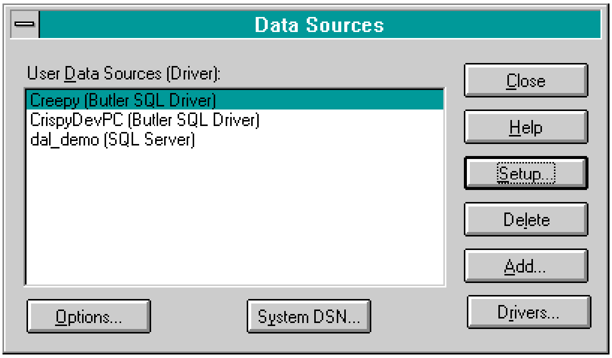
1. **Open the Control Panels directory.**



Maintains ODBC data sources and drivers

2. **Select and open the ODBC control panel.**

   The Data Sources window appears.



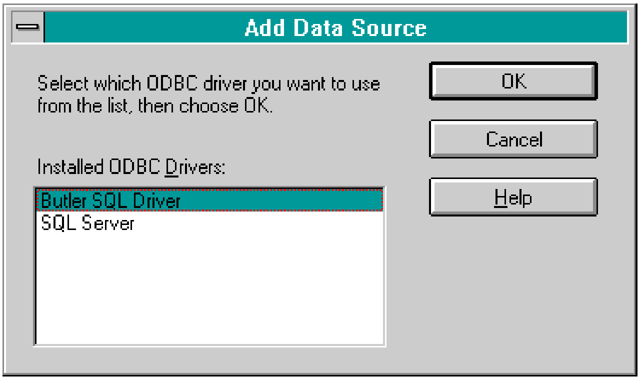
User Data Sources (Driver):

Creepy (Butler SQL Driver)
CrispyDevPC (Butler SQL Driver)
dal_demo (SQL Server)

Listed in the Data Sources window are the names of the existing Data Sources.

3. **Click Add....**

   The Add Data Source window appears.



Add Data Source

Select which ODBC driver you want to use from the list, then choose OK.

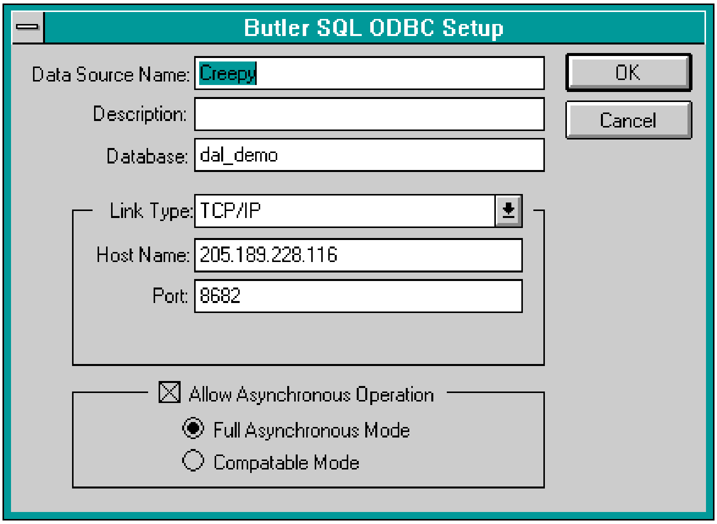
Installed ODBC Drivers:

Butler SQL Driver
SQL Server

The Add Data Source window lists all of the ODBC drivers installed on the machine.

4. **Select the Butler SQL Driver, click OK.**

The Butler SQL ODBC Setup window appears.



Butler SQL ODBC Setup

Data Source Name: Creepy

Description:

Database: dal_demo

Link Type: TCP/IP

Host Name: 205.189.228.116

Port: 8682

☒ Allow Asynchronous Operation

● Full Asynchronous Mode

○ Compatable Mode

5. **Enter a name for this data source. Press Tab.**

   Client applications use this name to identify the data source they want to use for their database connection.

6. **Optionally, enter a description into the Description field. Press Tab.**

   The description is a field you can use to record the purpose or usage of this data source.

7. **Enter the name of the database you want to use. Press Tab.**

   The database must match the name of a database installed in the Public Databases folder on the Butler SQL server.

   The Butler SQL Driver supports TCP/IP connections only. TCP/IP is selected for you in the Link Type field.

8. **In the Host Name field, enter the TCP/IP address of the server.**

   The values displayed in the Port field and the Allow Asynchronous Operation settings are defaults which do not have to be changed.

Important

If you are going to be using a specific data source with Microsoft Access, turn the “Asynchronous Operation Allowed” checkbox off. For use with FoxPro or BrioQuery, turn the “Asynchronous Operation Allowed” checkbox on and ensure that the “Compatibility Mode” radio button is selected.

9. **Click OK.**

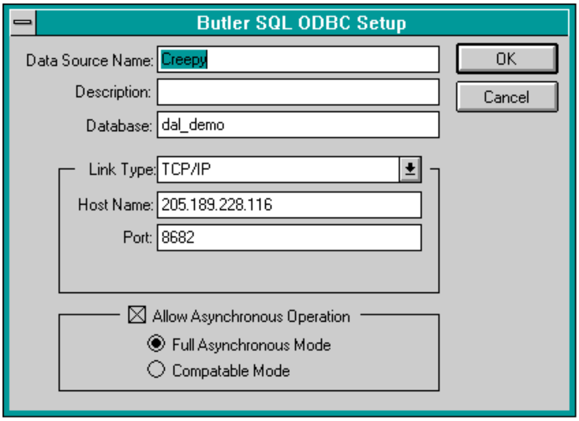
   The new ODBC data source is set up.

Editing a Data Source

To edit an existing data source:

1. **Select a data source from the list in the data sources window.**
2. **Click Setup....**

   The Butler SQL ODBC Setup window appears.



Butler SQL ODBC Setup

Data Source Name: Creepy

Description:

Database: dal_demo

Link Type: TCP/IP

Host Name: 205.189.228.116

Port: 8682

☒ Allow Asynchronous Operation

● Full Asynchronous Mode

○ Compatable Mode

3. **Make the desired changes to the setup of the data source.**

4. **Click OK.**

   The data source is changed to the new settings you have specified.

Deleting a Data Source

To delete a data source:

1. **Select a data source from the list of data sources in the Data Sources window.**
2. **Click Delete.**

   A warning dialog box appears, asking if you want to delete the selected data source.
3. **Click Yes.**

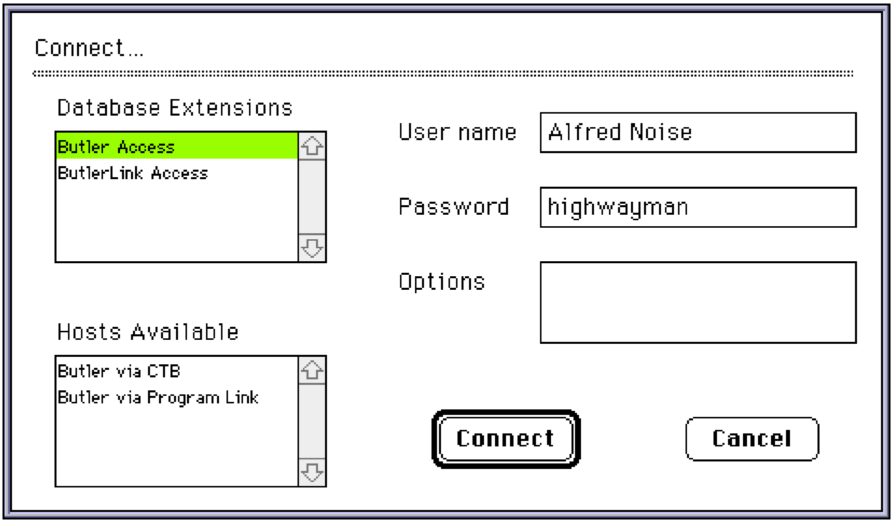
   The data source is removed from the Data Sources window.

Configuring a DAM Connection

Client connections to Butler SQL servers using DAM are made using hosts. A host separates the details of how to make a physical connection to a server (e.g. using program linking to connect to a specific Butler SQL port on a given Macintosh in a given network zone) from client applications. To connect to a server, a client application need only specify a host name. The ButlerAccess database extension then takes care of looking up this host name to determine how to connect to the server and where to find the server on your network.

If the Butler SQL server is moved to a different Macintosh or if the Macintosh is moved to a different network zone, you will only need to update the details of the given host, instead of each application you use to connect to the server. Once a host is updated, all client applications that use the host will know how to locate the server in its new location automatically.

You can create many different hosts on your machine, enabling applications to connect to different servers or to the same server using different protocols (program linking vs. an ADSP connection). Most client applications display a list of hosts when you log on, so you can select the host you want to use. For example, the log on dialog box displayed by the ButlerClient application (included with Butler SQL) displays a list of the hosts defined on the user's machine.



Connect...

Database Extensions

- Butler Access
- ButlerLink Access

Hosts Available

- Butler via CTB
- Butler via Program Link

User name: Alfred Noise

Password: highwayman

Options:

Hosts are created and configured on client machines using the ButlerHosts application.

About ButlerHosts

The ButlerHosts application is used to create and define hosts that use program linking or the Communications Toolbox to connect to a Butler SQL server. A host consists of a name as well as configuration information. For program link hosts, you must identify the Butler Port, Server Machine, and Server Zone. For Communications Toolbox hosts, you must configure the AppleTalk ADSP tool.

ButlerHosts stores the list of hosts and their configuration inside the ButlerHosts Data file. ButlerHosts creates this file for you automatically and saves it into your Preferences folder. You can also create and save hosts into other files; although these hosts will not be recognized by the Butler-Access database extension and will not be available to client applications.

If you have many users accessing your Butler SQL server, instead of using ButlerHosts to configure hosts on each machine individually, you can just distribute a prepared ButlerHosts Data file to each user and instruct them to place this file into their Preferences folder.

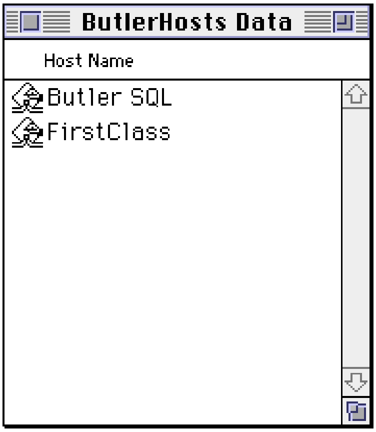
Creating a Hosts File

ButlerHosts creates and opens a new ButlerHosts Data file if none exists. If you are creating different files for different clients though, choose New from the File menu to create a new hosts data file.

When you save the file after adding host entries as described in the following sections, you will be allowed to save the file in any location and with any name. Remember though, only the hosts data file named "ButlerHosts Data" located in the client's Preferences folder will be used.

Opening a Hosts File

The active ButlerHosts Data file is opened automatically upon launching ButlerHosts.



To open a different hosts data file:

1. **Choose Open from the File menu.**

   A directory dialog box appears.

2. **Select the hosts file you want to view or edit.**

3. **Click Open.**

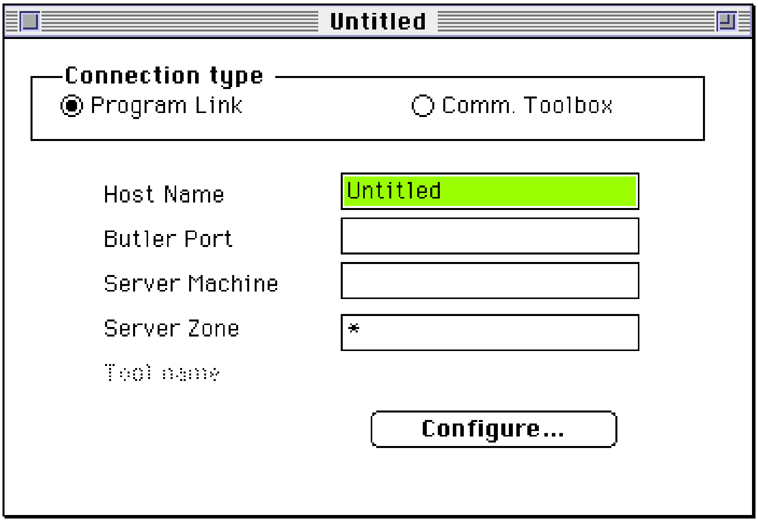
Creating a Host

To create a host:

1. **As described in the preceding two sections, open or create the hosts data file into which you want to add an entry.**

2. **Choose Create Host from the Edit menu.**

   A host configuration window opens for the new host.



Untitled

Connection type

● Program Link ○ Comm. Toolbox

Host Name: Untitled

Butler Port:

Server Machine:

Server Zone: *

Tool name

3. **Select the desired connection type radio button.**

4. **In the Host Name field, type a name for the new host.**

   This is the name that will appear in a list of hosts when initiating a connection with a SQL client program.

5. **Configure the host connection as described in the following sections.**

   See "Configuring a Program Link Host Connection" section on page 41 or "Configuring a CTB Host Connection" section on page 44 for more information.

6. **Close the host configuration window.**

7. **Choose Save from the File menu.**

   ButlerHosts saves the list of hosts and their configuration information to the host data file.

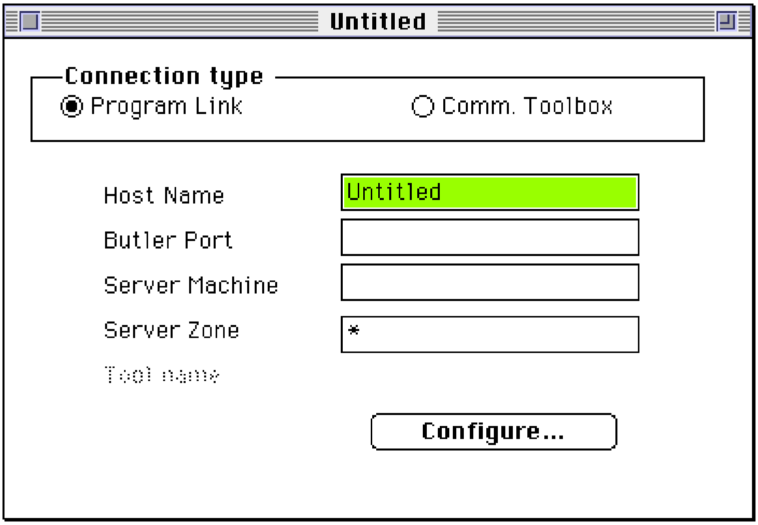
Configuring a Program Link Host Connection

Note

Only clients using System 7 are able to configure and use Program Link connections. If the machine running ButlerHosts is using System 6, the Program Link option is disabled.

To configure a Program Link host connection:

1. Select the Program Link connection type radio button in the Host configuration window.



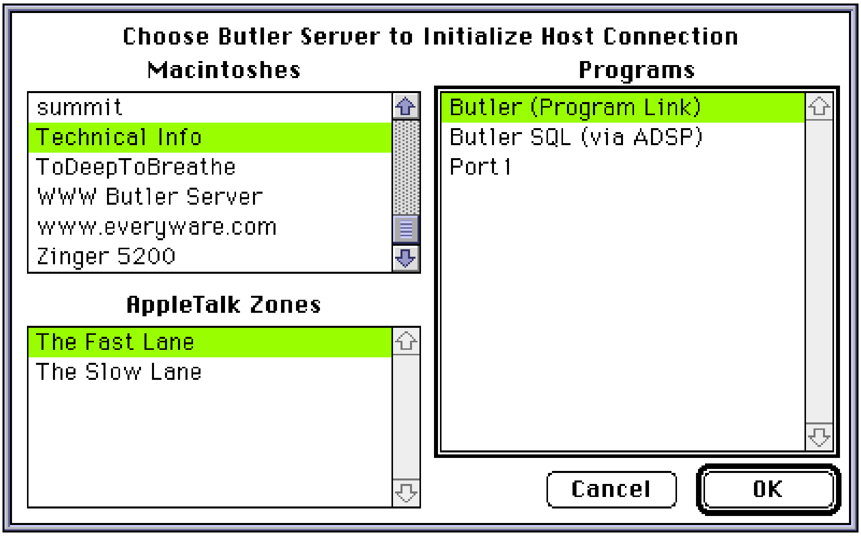
Untitled

Connection type
● Program Link ○ Comm. Toolbox

Host Name: Untitled
Butler Port:
Server Machine:
Server Zone: *
Tool name

2. Click the Configure... button.

If you are on a network with zones, a dialog box similar to the one below will appear.



If your network does not have any zones, the AppleTalk Zones list will not be displayed.

3. **Select the network zone containing the Butler server to which you wish to connect.**

   The Macintoshes list changes to display the Macintosh computers connected to the selected network zone.

4. **Select the server's Macintosh name from the Macintoshes list.**

   The Programs list changes to list the installed and enabled Butler SQL ports on the selected Macintosh.

Note

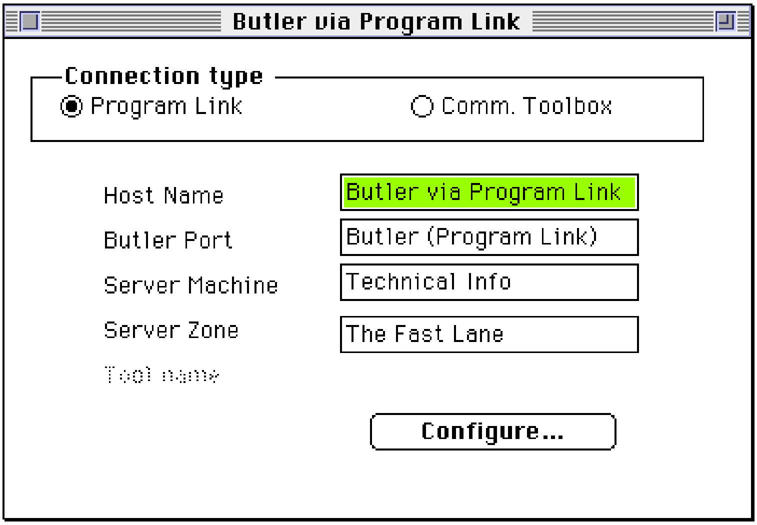
If you don't see the server's name in the list, the machine may not be running or program linking mat not be turned on. Make sure the machine is on, that a Program Link port is installed, and that program linking is enabled (both in Butler and the System software).

5. **In the Programs list, select the name of the Butler server's Program Link port.**

6. **Click OK.**

   The selected zone, Macintosh, and port are entered into the fields in the Host configuration window.

   If you are connected to a network with only one zone, an asterisk ("*") will be entered into the zone field.



Butler via Program Link

Connection type
● Program Link ○ Comm. Toolbox

Host Name: Butler via Program Link
Butler Port: Butler (Program Link)
Server Machine: Technical Info
Server Zone: The Fast Lane
Tool name

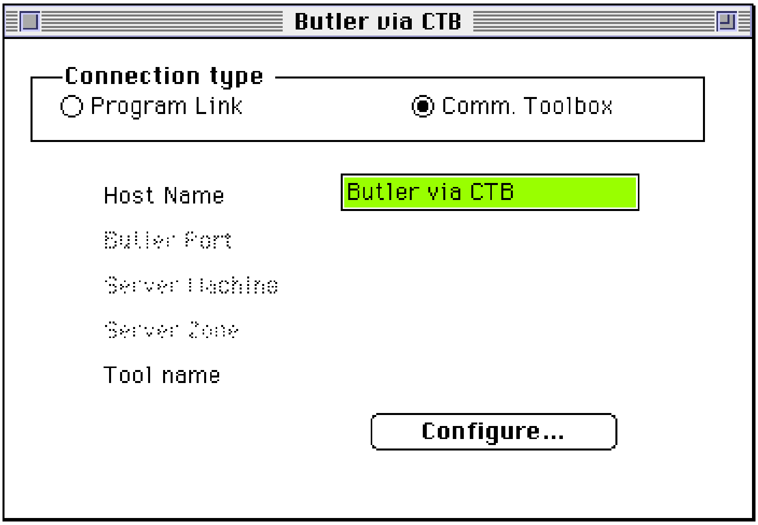
Note

If you are running ButlerHosts on the server Macintosh or if your network has only one zone, you will notice that the Server Zone box in the Program Link configuration window contains an asterisk (‘*’). This indicates that the server is the same zone as the client. This is not a problem unless you want to distribute the ButlerHosts Data file containing this host to clients in other zones. If you want to do this, simply type (exactly) the name of the server’s zone into the Server Zone field.

Configuring a CTB Host Connection

To configure a Communications Toolbox (CTB) host connection:

1. **Select Comm. Toolbox Connection Type radio button in the host configuration window.**



Butler via CTB

Connection type

○ Program Link ● Comm. Toolbox

Host Name: Butler via CTB

Butler Port

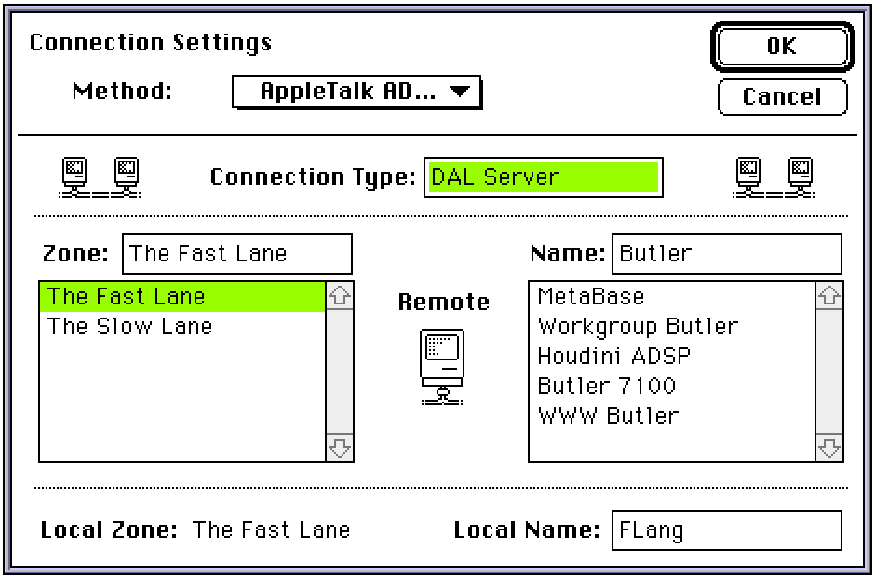
Server Machine

Server Zone

Tool name

2. **Click the Configure... button.**

The Connection Settings dialog box appears.



Connection Settings

Method: AppleTalk AD...

Connection Type: DAL Server

Zone: The Fast Lane

The Fast Lane
The Slow Lane

Remote

Name: Butler

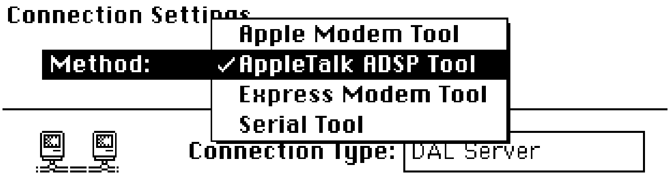
MetaBase
Workgroup Butler
Houdini ADSP
Butler 7100
WWW Butler

Local Zone: The Fast Lane

Local Name: FLang

3. **From the Method pop-up menu, choose the AppleTalk ADSP tool.**

   The menu lists all the Communications ToolBox tools installed in the Extensions folder. Butler SQL supports the AppleTalk ADSP tool only.



Connection Settings

Method: ✓AppleTalk ADSP Tool

Connection Type: DAL Server

The port name is entered in the Name field.

Note

If you don't see the port name in the list, make sure the Butler server is running and has an ADSP port installed.

The Local Name field is not used for client host configuration. You may leave it as it is.

7. **Click OK.**

Editing a Host

To edit an existing host:

1. **Select the host in the host list window.**
2. **Choose Edit Host from the Edit menu.**

   You can also double-click the host to edit it.

   The host configuration window appears.

   You can now enter a new name for the host, change the connection type, or click Configure to change the host configuration.
3. **Close the host configuration window.**
4. **Choose Save from the File menu.**

Removing a Host

To remove a host:

1. **Select the host in the host list window.**
2. **Choose Clear from the Edit menu.**

The selected host is removed from the list.

Configuring a DAL connection

To configure a DAL connection, you must first use ButlerHosts to create a host for each Butler SQL server to which you wish to connect. Refer to the "Configuring a DAM Connection" section on page 36 for information on creating a host. In addition to creating a host, you must enter the name of the host into the DAL Preferences file.

The DAL Preferences file

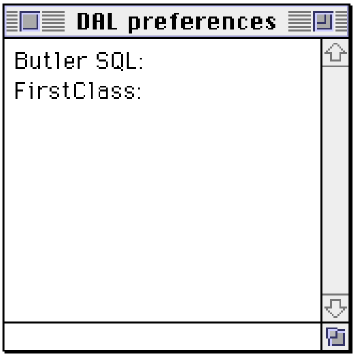
The DAL Preferences file is a text file used to describe a connection to a DAL database server. For Butler SQL servers, this file contains the names of the Butler SQL hosts. The DAL Preferences file is contained in the Preferences folder in your System Folder.

To add a host to the DAL Preferences text file:

1. **Open the DAL Preferences text file.**

   You can find the DAL Preferences file in your Preferences folder. If one does not exist, create a new document in your text editor and save this file into your Preferences folder, naming it "DAL Preferences".

You can use the SimpleText application, included with the Macintosh System Software, to open and update the DAL Preferences folder.



```
Butler SQL:
FirstClass:
```

2. **Type the name of the host.**

   The name must start on a new line and must end with a colon.

3. **Press Return after typing each host name and its trailing colon.**

4. **Choose Save from the File menu.**

   The changes you made are saved to the DAL Preferences file.

Chapter 3 • Configuring Butler SQL

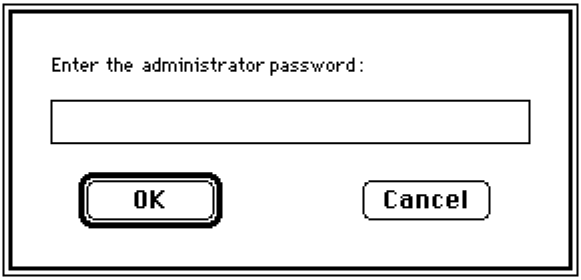
This section describes how to setup Butler SQL in order to match its operation to the requirements of your databases, machine, environment, and usage levels.

This configuration is performed using the Preferences and Tuning windows in Butler SQL, as well as by adjusting the amount of memory allocated to Butler SQL.

Who Can Administrate Butler SQL

By default, there are no restrictions on access to the configuration windows in Butler SQL. If you want only certain users to be able to view or change information in the Preference and Tuning dialog boxes, you can use Butler's access privileges to create and designate administrative users.

If an administrator user has been defined, choosing the Preferences or Tuning items from the Edit menu will display the following dialog box:



Enter the administrator password:

If the correct password of an administrator user is entered, the selected dialog box will open.

Note

We strongly recommend setting up an administrator user to guard against unauthorized changes in the Butler server's configuration.

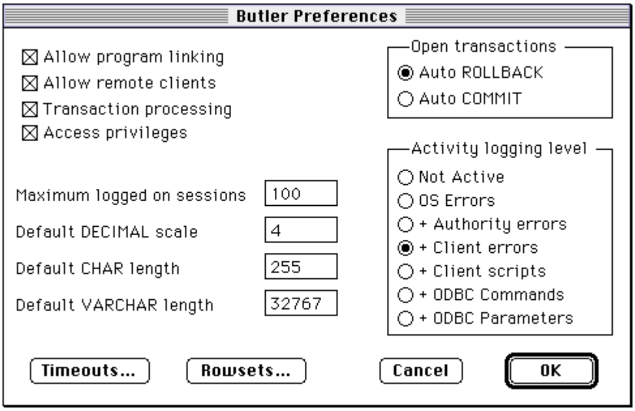
Access privileges are configured using the ButlerTools application. Refer to the "Access Privileges" section of the ButlerTools manual for instructions on defining administrative users.

Butler SQL Preferences

Many different functions of Butler SQL can be configured to suit the specific needs of the databases and users of the database server. These preferences are set in the Preferences dialog box.

To open the Preferences dialog box:

1. Choose Preferences... from the Edit menu.



Butler Preferences

☒ Allow program linking
☒ Allow remote clients
☒ Transaction processing
☒ Access privileges

Maximum logged on sessions	100
Default DECIMAL scale	4
Default CHAR length	255
Default VARCHAR length	32767

Open transactions
● Auto ROLLBACK
○ Auto COMMIT

Activity logging level
○ Not Active
○ OS Errors
○ + Authority errors
● + Client errors
○ + Client scripts
○ + ODBC Commands
○ + ODBC Parameters

Each of the options in the Butler Preferences dialog box are discussed in the following sections.

Allowing program linking

If you have a Program Link port installed, you must have the “Allow program linking” checkbox in the Preferences dialog box selected to allow clients to connect to the port. The default setting is to allow program linking.

As noted in the “Editing a port” section (page 12), you must also configure your server’s system software to allow program linking. (You can also disable individual Program Link ports as described in the section “Disabling a port” section on page 15.)

Allowing remote clients

The “Allow remote clients” checkbox in the Butler Preferences dialog box determines whether remote clients (as distinguished from a client on the server Macintosh) will be able to connect to the Butler server.

A side-effect of disabling remote access to the Butler server is that only program linking connections are possible. Communications Toolbox and TCP/IP ports will be disabled automatically. This is necessary because it is not possible for the server to determine whether a TCP/IP or CTB connection originates from a local or remote client.

Transaction processing

The Transaction Processing checkbox is used to enable the transaction processing capabilities of Butler SQL. Transaction processing promotes data integrity by ensuring that a group of SQL statements modifying a database are either carried out completely or not at all. For example, consider this SQL program fragment which successively updates invoice, inventory, and customer receivables information:

```
INSERT INTO line_items
(quantity, prod_code, price, cust_code, invoice_num)
    VALUES(2, 29873, 379.99, 98076, 386091);
UPDATE inventory SET on_hand = on_hand - 2
    WHERE inv_code = 29873;
UPDATE customers SET receivables = receivables + 379.99 * 2
    WHERE cust_num = 98076;
```

Without transaction processing, if processing were interrupted during the execution of this fragment, the database might decrease inventory without increasing receivables.

Transaction processing prevents partial updating of a database. In the above example, enabling transaction processing would ensure that an interruption in processing would leave the database as it was before the fragment was executed. Let the Butler SQL server know that you are finished the transaction and want your changes written to the database by including this statement at the end of the transaction:

```
COMMIT;
```

Note

To prevent large transaction processing journals, Butler SQL does not do transaction processing for updates including large-item data types, such as LONGCHAR, VARBIN, PICTURE, SOUND, DOCUMENT, ICON, and MOVIE. Additionally, Butler SQL does not include changes to the database schema (using the ALTER command) as part of transaction processing.

See the entries for COMMIT and ROLLBACK in the section “SQL Statements” of the SQL Reference Guide for more information on transaction processing.

Note

Enabling transaction processing does entail some performance disadvantages. Database update operations are approximately 10–15% slower than with transaction processing disabled.

Access privileges

The Access Privileges checkbox determines whether access privileges are used. If this checkbox is not selected, any user will have full access to any and all databases residing on the server. If this checkbox is selected, then access is restricted based on the settings defined using the ButlerTools application.

The ButlerTools application provides advanced capabilities for defining the access individual users and groups have to the Butler SQL server, Butler SQL databases, and the tables within databases. See the ButlerTools manual for information on defining access privileges.

Auto ROLLBACK and COMMIT

When transaction processing is enabled you have the option of having open transactions rolled back or committed automatically. A transaction is left open when one of the following occurs:

- a client closes a database without closing a transaction,
- a client with an open transaction disconnects (normally or due to a system or network problem), or
- the Butler SQL server closes down (normally or due to a system or network problem) with open client transactions.

Setting the Auto COMMIT option in Butler SQL will cause any changes made in the open transactions to be saved upon any of the above occurrences (or upon Butler's next launch in the case of the server quitting unexpectedly). Setting the Auto ROLLBACK option causes any changes made in the open transactions to be discarded.

Maximum sessions logged on

The Maximum logged on sessions field is used to limit the number of concurrent sessions or users that Butler SQL will accept. Use this feature if Butler is running on a slow machine. By limiting the number of connections, you will ensure that the users which connect receive the best performance possible.

DECIMAL scale

The Default Decimal Scale field is used to specify the number of decimal places assigned to DECIMAL type columns or variables when no specific quantity is assigned.

This default applies to columns created or changed by sending CREATE TABLE or ALTER statements to Butler SQL, as well as in any variables used in DAL procedures you create.

CHAR length

The Default CHAR length field is used to specify the length of characters allowed for CHAR type columns or variables when no specific length is assigned.

This default applies to columns created or changed by sending CREATE TABLE or ALTER statements to Butler SQL, as well as in any DAL procedures you create.

VARCHAR length

The Default VARCHAR length field is used to specify the maximum number of characters allowed for VARCHAR type columns or variables when no specific length is assigned.

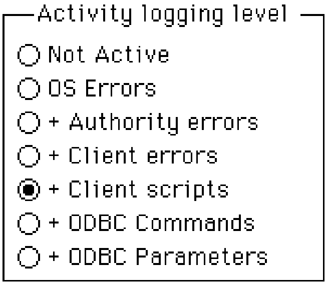
This default applies to columns created or changed by sending CREATE TABLE or ALTER statements to Butler SQL, as well as in any variables in DAL procedures you create.

Activity Logging Level Options

Butler SQL allows you to log server activity to a text file. The activity logging level options allow you to indicate the level of detail you want to include in the activity logs.

Setting the activity logging level

There are seven activity logging levels available. The levels are cumulative; that is, higher levels include all entries logged by lower levels.



Activity logging level

○ Not Active
○ OS Errors
○ + Authority errors
○ + Client errors
● + Client scripts
○ + ODBC Commands
○ + ODBC Parameters

Note

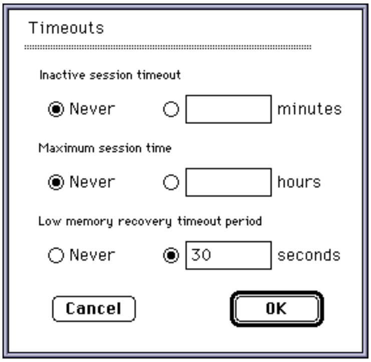
Logging activity affects the performance of Butler SQL especially if a high level of detail is selected. Additionally, if your Butler SQL server has a limited amount of disk space, you may want check your log files on a periodic basis to ensure that ample disk space is available for the normal operation of the server.

Activity logs are saved into the Activity Logs folder inside the Butler Preferences folder. (See “Activity Logs” section on page 82 for more information on the Activity Logs folder.) Butler SQL creates a new activity log each day the Butler SQL server is opened. If the server remains open for more than one day, then the logging information is saved to the previous day’s log file.

You should only use the ODBC Commands or ODBC Parameters logging levels for brief periods. These levels write a significant amount of information to the activity logs which will adversely affect the operation of your Butler SQL server.

Timeouts

Butler allows you to specify values that control the maximum time an inactive client (a connection that has no communication with the Butler SQL server) can be connected without sending a query and the absolute maximum client connection time. These timeout options are useful for ensuring that clients who have unexpectedly disconnected from the Butler server are removed from the Users window, and the memory used by their session is made available for other Butler SQL users.



Timeouts

Inactive session timeout

● Never ○ minutes

Maximum session time

● Never ○ hours

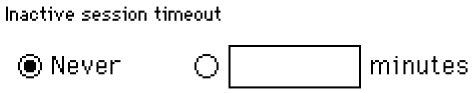
Low memory recovery timeout period

○ Never ● 30 seconds

Inactive session timeout

The inactive session timeout value controls the length of time a user is permitted to remain connected to the Butler SQL server without sending any commands to the server. A client who remains inactive beyond the specified length of time is disconnected automatically.

To allow inactive clients to remain connected for an indefinite period, select Never in the “Inactive session timeout” area. To limit the length of inactive sessions, type a value (in minutes) into the field.



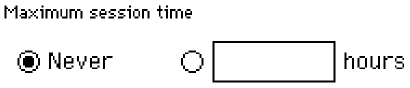
Inactive session timeout

● Never ○ minutes

Maximum session time

The maximum session time value controls the maximum time period of a client can be connected to Butler in a single session. A client who remains connected beyond the specified length of time is disconnected automatically. If a client is in the midst of sending or receiving data from the Butler server, they will be disconnected upon the completion of the transaction, that is, when any processing associated with their session is finished.

To allow clients to remain connected for an indefinite period, select Never in the “Maximum session time” area. To limit the length of client sessions, type a value (in hours) into the field.

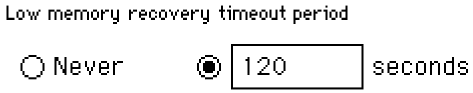


Maximum session time

● Never ○ hours

Low memory recovery timeout

If Butler SQL does not have enough memory to hold a rowset resulting from a client data selection, the settings for the “Low memory recovery timeout period” determine what happens.



Low memory recovery timeout period

○ Never ● 120 seconds

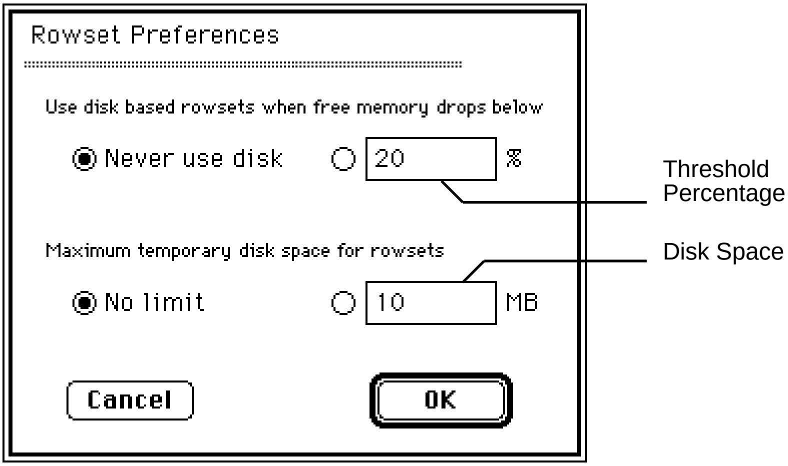
To have Butler SQL wait indefinitely for memory to become available, select Never. The client session that initiated the request will be paused until enough memory is available. Otherwise, type a value (in seconds) into the field. If the time specified elapses and Butler still doesn’t have enough memory, an error will be returned to the client.

Rowsets

Butler has the ability to store client data selections on disk while they are in use. Normally, the size of rowsets selected by clients is limited by the amount of memory available to the Butler server application; with disk-based rowsets, the size is limited only by the amount of hard disk space you choose to allocate for this purpose.

Note

The rowsets are stored on the Butler server’s start-up volume. There is no way to specify a different volume.



Rowset Preferences

Use disk based rowsets when free memory drops below

● Never use disk ○ 20 %

Maximum temporary disk space for rowsets

● No limit ○ 10 MB

The Threshold Percentage field is used to set the percentage of application memory available to Butler SQL below which you want disk-based rowsets to be used. If, when attempting to load a client's data selection into memory, Butler's available memory drops below this threshold, the rowset will be stored on disk instead.

Note

Writing and reading rowsets from disk is slower than accessing them in memory. If you set this value too high, the performance of Butler SQL will be affected.

Select “No limit” if you want the maximum storage space allocated to rowsets to be limited only by available disk space. Otherwise, select the radio button to the left of the Disk Space field and type in the maximum storage space (in megabytes) you want allocated for rowset storage.

Tuning Butler SQL

Butler has several settings for optimizing server performance. The defaults for these settings will work well for most sites. If you decide to change any of the tuning settings, make sure you read this section carefully.

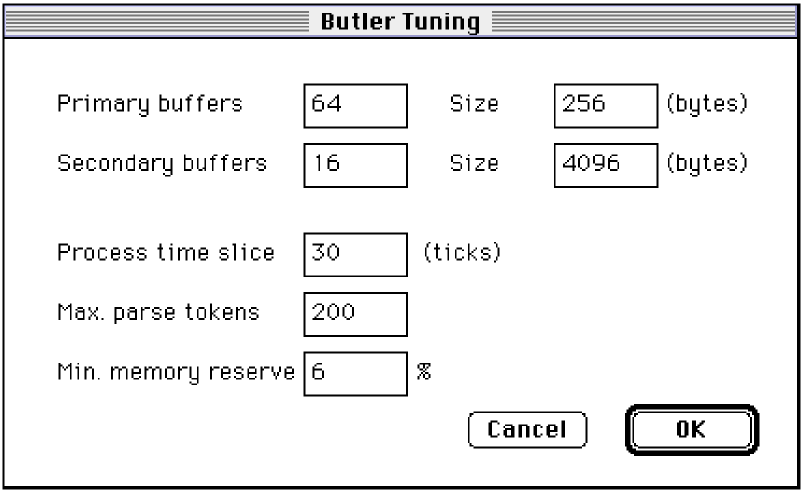
Note

Changes to the Tuning settings take effect only upon the next launch of the Butler application.

To open the Tuning dialog box:

1. **Choose Tuning... from the Edit menu.**

   The tuning dialog box appears.



Butler Tuning

Primary buffers 64 Size 256 (bytes)

Secondary buffers 16 Size 4096 (bytes)

Process time slice 30 (ticks)

Max. parse tokens 200

Min. memory reserve 6 %

Each of the tuning options is discussed in the following sections.

Process time slice

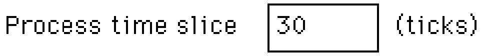
The Butler server handles all client requests sequentially. The process time slice setting determines how much time is devoted to each client during their turn or "slice" of processing time. The default process time slice value is 30 ticks (a tick is approximately one-sixtieth of a second).

A higher value will allow each client to retrieve more data at once, but will also increase the time between successive slices of processing time (because all other clients also have more processing time). If the value is set too high, clients will notice a greater delay in receiving data from Butler, particularly if there are a large number of active sessions on the server.

A lower value will cause Butler to spend less time on each client's requests, but will decrease the amount of time between each client's slices of processing time.

To change the process time slice setting:

1. **In the "Process time slice" field, type the new value, in ticks.**



Process time slice 30 (ticks)

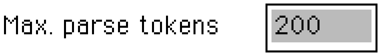
Maximum parse tokens

You can set the maximum number of tokens, or words, Butler can process at once. Increasing this value speeds execution of DAL statements, but at the expense of decreasing the memory available for processing. Decreasing the value provides more memory for processing but can slow execution of large DAL programs (over 200 tokens).

We recommend keeping the maximum parse tokens value above 100 at all times. The default is 200.

To change the maximum parse tokens setting:

1. **In the “Max. parse tokens” field, type the new value.**



Max. parse tokens 200

Minimum memory reserve

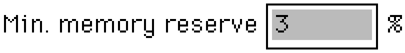
You can change the percentage of application memory Butler reserves for processing. The remaining percentage is available for holding data resulting from client queries, variables, query fragments, and for port buffers. The default percentage of memory reserved for processing is three. We do not recommend setting it below this value unless you have increased the total memory allocated to Butler from its Get Info window in the Finder.

Increasing the minimum memory reserve value may improve processing speed, but will reduce the amount of data a client can select at one time (or, if you have the disk-based rowsets option turned on, the size of data selections that can be handled without storing the data to disk before sending it to the client).

Decreasing this value may slow processing of client requests, but will allow clients to select larger amounts of data (or, with disk-based rowsets, larger RAM-based client data selections).

To change the minimum memory reserve value:

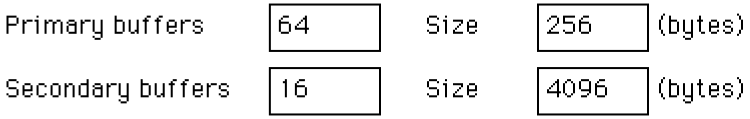
1. **Type the new value into the “Min. memory reserve” field.**



Min. memory reserve 3 %

Buffers settings

The Buffers settings are available in the Butler Tuning dialog box. If you are transferring data items less than 32K between clients and the Butler server, we do not recommend changing them. The default values are shown here.



Primary buffers	64	Size	256	(bytes)
Secondary buffers	16	Size	4096	(bytes)

The number of primary buffers only needs to be changed if you expect a large number of concurrent users. Set this value to 105 if you expect 100 concurrent users, 205 if you expect 200 concurrent users.

If you use large data items, you may have to modify the secondary buffers settings for improved performance.

The secondary buffers are used for temporary storage while receiving binary data from clients. If the size and number of secondary buffers is insufficient to hold a received item, Butler resorts to a slower method of retrieving the data. If you find that large data items (like pictures or sounds) are taking a long time to send to the server, try doubling both the size and number of secondary buffers. (After changing the values, you must quit and relaunch Butler for the new values to take effect.) We do not recommend setting the size of the secondary buffers larger than 64K.

Memory Allocation & Buffers

The following guidelines will assist you in determining the amount of memory you need to allocate Butler SQL. When setting up Butler SQL it is best to tune your settings based on the memory allocation and the number of users that will be accessing the server.

The default setting for Butler memory allocation assumes several users, one or two installed ports, and no stored procedure files on the server.

To determine the amount of memory you need to allocate Butler SQL, you need to review:

- the number of ports installed on the server,
- the number of concurrent users or sessions you expect, and
- the amount of memory used by stored procedures.

The next sections review the impact these items have on memory usage.

Port Memory

One of the attributes that affects Butler SQL's memory usage is the number of installed ports. Each installed port requires approximately 100K of RAM. To calculate the exact amount, use the following formula:

port memory = 4K + (# of primary buffers * primary buffer size) + (# of secondary buffers * secondary buffer size).

Using the default buffer settings, the memory required for each port is 74K (4K + (64 * 25K) + (16 * 4K)).

Session Memory

The following table outlines the memory allocation required for different levels of concurrent sessions or users. These settings assume that you have two ports installed and are using a simple set of stored procedures (msad$procedure file). If you have additional ports installed or a complex set of stored procedures, you will need to adjust these memory settings accordingly.

# Users	RAM (68K Macintosh)	RAM (PowerPC)	# Primary Buffers
1	3500K	4600K	64
5	4000K	5100K	64
10	4500K	6000K	64
20	5500K	7000K	64
50	9000K	11000K	64
100	14000K	16000K	105
200	23000k	25000K	205

If the amount of RAM sessions that can be assigned to Butler is limited, you should set the number of maximum users accordingly. For example if you can use the Butler Preferences dialog box to set the maximum number of sessions to 20, you should also leave some system memory free as Butler can make use of it when required.

Stored Procedures

If your procedures contain a large number of variables, or use variables that require a large amount of memory, you will need to adjust Butler SQL's memory allocation.

Other Memory Considerations

The amount of memory required by a Butler server is affected by many variables. Here are some additional, important points to be aware of:

- Each connection requires a minimum of 10K of memory, even if there is no activity associated with the connection.
- Performing a SELECT with a FOR EXTRACT update mode requires that the server have enough memory to hold all of the data retrieved (slightly more, actually). See “Rowsets” section on page 60 for an explanation of rowsets and the SQL Reference guide for information on the SELECT statements.
- Printing of data using the DAL print command requires memory for each item printed but not yet retrieved by the client.
- Every data item sent to the Butler server requires enough free memory to hold two to three times the size of the item. Much of this must be contiguous (i.e., all in one block). If you have clients dealing regularly with 100K PICTURE items, for example, it would be a good idea to set Butler’s application memory size so that at least 400K is always available to each of them. The extra memory makes it more likely that the required amount will be available in a contiguous block.

Record Locking and Memory Usage

Record locking ensures that data being modified by one user is not simultaneously updated by another user. Butler SQL has numerous options for ensuring that conflicting client data-modification operations are handled properly. (See the entries for the OPEN DATABASE, OPEN TABLE, and SELECT statements in “SQL Statements” in the SQL Reference guide for more information.)

Butler’s record locking mechanism provides locks as required, in 1000-lock increments. Each block of 1000 locks requires 32K of memory. The only limitation on the number of possible simultaneous locks is the amount of available contiguous memory in Butler SQL.

To avoid memory problems arising from having a large number of locks active, periodically perform a COMMIT. Commit releases all locks, regardless of whether transaction processing is enabled.

If a user is disconnected for any reason, Butler automatically releases all locks associated with that user.

Defensive Programming

When developing an application, it is best to try to anticipate memory problems so that your application can behave correctly. You can determine the amount of free memory in a couple of ways. First, the memory indicator in the Butler application Resources window gives you a rough idea of how much memory is available. Second, there is the **$freemem( )** system function which returns the exact amount, in bytes, of free server memory. See the SQL Reference guide for information on the $freemem() system function.

It is a good idea, if possible, to use the **$freemem( )** function before sending large items to the server. That way, you can check for and prevent out-of-memory errors rather than have to deal with them after they occur. If the Butler server does run out of memory while trying to fulfill a client's request, it will return a CENOMEM error (–10112).

See also "Rowsets" section on page 60 and "Low memory recovery timeout" section on page 60 for more information on low-memory condition handling.

Chapter 4 • Backing Up and Database Recovery

The best insurance you can have for recovering from problems with your database is a reliable backup. Butler SQL includes several methods you can use to backup your database. This chapter describes what you need to do to make backup copies of your database, transaction journals, and activity logs, as well as the steps you need to take in the unlikely event you need to recover your database because of hardware problems or a damaged database.

Preparing Your Backup Strategy

Before you implement your Butler SQL-based solution, you need to investigate and plan your backup strategy. This includes, but is not limited to:

- selection of the backup media (e.g. DAT tape or optical disk) and backup software you want to use
- use of separate hard disks for databases, activity logs and transaction journals
- investigating RAID (redundant array of inexpensive disks) drives and disk mirroring software
- securing onsite and offsite locations for your backup media
- frequency you require your backups to be made
- plans to replace your server should a hardware problem occur

The monthly computer magazines often run articles describing backup software and hardware, RAID drives, and disk mirroring options. Review these articles to assist you in determining your needs for backup hardware and software.

Backing Up Your Database

Butler SQL includes support for online backups using its Backup command. The Backup command is necessary because it is not possible to make reliable copies of files that are in use. For example, you do not want to backup a database that includes partially completed transactions.

Butler SQL's Backup command:

- waits until all current transactions are completed,
- makes a copy of the one or all of the databases installed on the server, and
- detaches the database journal and activity logs.

While the backup is in progress no users will be allowed to update the affected databases or begin a database transaction. Users will, however, be allowed to view data in the databases. Once the Backup command has completed normal access to the server will be restored. At this point you can then use your backup software to copy the files to archival media. Refer to the SQL Reference Guide for a detailed description of the Backup command.

The backup command can be issued from any application that allows you to send SQL to a Butler server. For example, you could use AppleScript to write an applet that connects to your server, issues the backup command, then disconnects. Here is a simple AppleScript script that will backup a database on a server:

```
-- AppleScript script to backup a database

-- You must have the Butler ODBC osax installed to run this script

SQLAllocEnv -- allocate ODBC environment

set myConnection to 0

set myStatement to 0
```

```
-- connect to the server

try

  -- replace "MyDataSource", "Administrator", "AdminPswd" with your
data source name,

  -- user name, and password, respectively

 set myConnection to SQLConnect "MyDataSource" user "Administrator"
password "AdminPswd"

on error message number errNum

 if (errNum ≠ -128) then

  display dialog "Error: " & message & " " & errNum as text

 end if

end try

-- create and send the Backup statement to backup the Cars_db database

if myConnection ≠ 0 then

 set myStatement to SQLAllocStatement myConnection

 SQLExecDirect "BACKUP DATABASE 'Cars_db' TO 'MyHD:Backup
Folder' WITH JOURNAL, ACTIVITYLOG" statement myStatement

 SQLDisconnect myConnection

end if

SQLFreeEnv -- free the ODBC environment
```

Backing up manually

You can also backup your database manually. Before quitting Butler, ensure that all users have logged off the server, then use Butler's Detach Journal command (on Butler SQL's Options menu) to detach the transaction journal and activity logs. Quit Butler, then backup the desired databases, and the detached activity log and journal files. When completed, restart your Butler server.

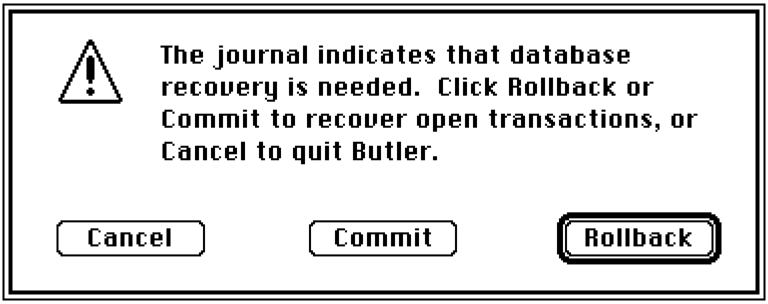
Recovering your Database

If you your database becomes damaged, or if you have a hardware failure, you may need to restore your database from your backups. If you think your database is damaged, use ButlerTools' Verify command to confirm the status of the database. If ButlerTools is not able to recover your database, you will need to use your backups.

After you restore your database from your backup media, you will need to issue the RESTORE DATABASE command to process transactions recorded from the time of the backup to the time of the database failure. The RESTORE DATABASE command uses the journal to reissue each Insert, Update, and Delete command, allowing you to reconstruct your database. Note that any commands issued that change the structure of the database, such as CREATE TABLE, CREATE INDEX, or ALTER TABLE, are not included in the transaction journal, so they will not be execute as part of the restore operation. To avoid this problem be sure to backup your database after changing your database's schema. For more information on the RESTORE DATABASE command refer to the SQL Reference Guide

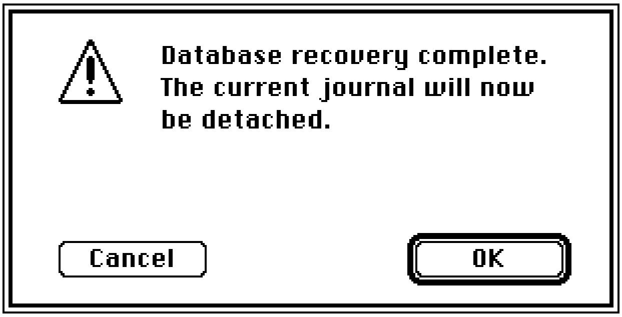
Recovering from a system error or power failure

If Butler SQL is interrupted while there are open transactions, the next time Butler is launched it will prompt you to commit or rollback the outstanding transactions.



The journal indicates that database recovery is needed. Click Rollback or Commit to recover open transactions, or Cancel to quit Butler.

When Butler has complete the commit or rollback it will prompt you to detach the journal file.



Database recovery complete. The current journal will now be detached.

Detaching the journal creates a new journal file for logging transactions. If you prefer, you can click the Cancel button to continue to use the current journal to log transactions.

Note

You can also detach the current journal at any time by choosing Detach Journal from the Options menu. We recommend you do this occasionally, especially if your Butler server is very busy. Database recovery time is proportional to the size of the journal.

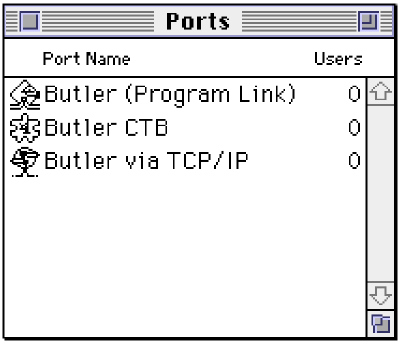
Chapter 5 • Butler SQL Status Windows

Ports, Users, Resources, and Tables

Butler SQL provides four windows where information on the server's users, open tables, and resource usage are displayed. These windows are updated regularly to allow for constant supervision of the Butler SQL servers. Each of these windows is discussed in the following sections.

The Ports Window

The Ports window lists the installed ports. Ports enable client applications to communicate with the Butler SQL server. The Ports window is displayed by choosing Show Ports from the Options menu.

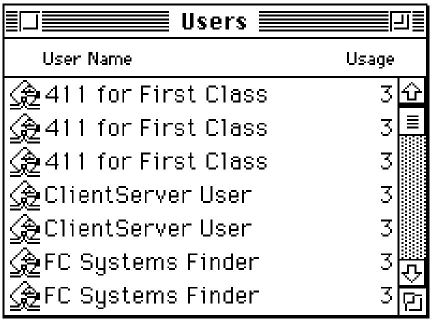


Port Name	Users
Butler (Program Link)	0
Butler CTB	0
Butler via TCP/IP	0

The Ports window is described in detail in the "Chapter 1 • Connecting to Butler SQL" section on page 3.

The Users window

The Users window is displayed by choosing Show Users from the Options menu. The window shows the names of the currently connected users and an indication of how active each user is.



User Name	Usage
411 for First Class	3
411 for First Class	3
411 for First Class	3
ClientServer User	3
ClientServer User	3
FC Systems Finder	3
FC Systems Finder	3

The name shown initially for a user is the one the user uses to connect. If no name is specified, “Guest” is displayed. The name shown changes to reflect any name entered as a parameter to a statement executed by the user. For example, if a client executes the statement:

```
OPEN RMR DATABASE 'DAL_Demo' AS USER 'John';
```

then his name in the Users window will change to “John”.

The Usage column shows how many send and execute commands (Butler SQL functions used by client applications) have been performed by each client.

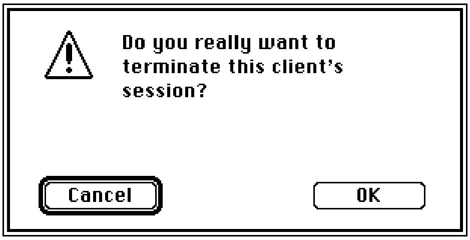
Disconnecting a user

It may be necessary occasionally to disconnect a user from the server. Perhaps a user has left their session open, but you know that the session is not being used. Or alternatively, a user’s client application may have quit abnormally.

To disconnect a user:

1. **Select a user in the Users window.**
2. **Choose Disconnect User from the Options menu.**

   A warning dialog box appears.



Do you really want to terminate this client's session?

3. **Click OK.**

   The user is disconnected and removed from the Users window.

Note

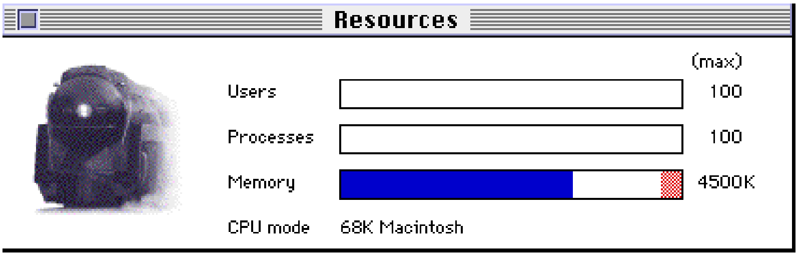
You can also use the Timeouts options in the Preferences window to have Butler SQL remove inactive users automatically. Refer to the "Timeouts" section on page 58.

The Resources window

The Resources window displays in graphical format:

- the number of users connected,
- the number of current processes (the number of clients currently executing procedures on the server), and
- the amount of memory used and available to be used.

To display the Resources window, choose Show Resources from the Options menu.



Resources

(max)

Users 100

Processes 100

Memory 4500K

CPU mode 68K Macintosh

The number of users (sessions), executing processes, and memory usage are indicated by a black (blue on color monitors) bar in each area.

The memory indicator bar also has a shaded (red, on color monitors) area at the end. This shows the amount of memory reserved for processing as set in the “Minimum memory reserve” field in Butler’s Tuning dialog box (see “Minimum memory reserve” section on page 64). When the memory used reaches this threshold, all client processing is suspended for the amount of time specified in the “Low memory recovery timeout period” (see “Low memory recovery timeout” section on page 60). At the end of this period, processing continues normally if enough memory has been freed.

If memory is still unavailable, Butler locates the user that is using the most server memory, then clears all pending data (data which has been printed, but not retrieved by the client) as well as of the client’s cursors. A no memory available error is then returned to the client.

The Tables window

This window displays the names of the currently open tables and indicates how many users have each table open.

To view the Tables window, choose Show Tables from the Options menu.

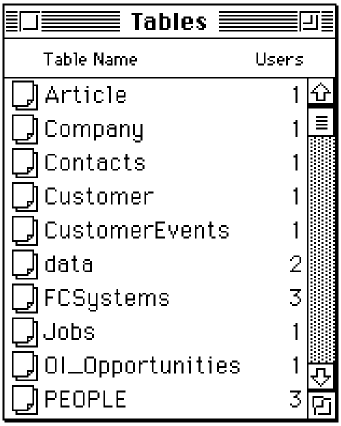


Table Name	Users
Article	1
Company	1
Contacts	1
Customer	1
CustomerEvents	1
data	2
FCSystems	3
Jobs	1
OI_Opportunities	1
PEOPLE	3

A table is opened explicitly when a client executes an OPEN TABLE statement (see "SQL Statements" in the SQL Reference guide for information on this statement) or implicitly when a client executes any statement requiring access to a table (SELECT, UPDATE, etc.). Tables remain open until closed with the CLOSE TABLE statement, until the database containing the table is closed with the CLOSE DATABASE statement, or until the user logs off of Butler SQL.

Butler Preferences Folder

Butler SQL uses a special folder, the Butler Preferences Folder, to store all of the configuration, logging, and port information as well as databases and transaction journal files. The Butler Preferences Folder is installed into the server's Preferences folder by the Butler SQL installer. If, when launching Butler SQL, the Butler Preferences Folder cannot be found, a new folder will be created automatically.

Note

You may want to create an alias to the Butler Preferences Folder and keep this alias on your desktop, in the Apple Menu Items folder, or in the same folder as the Butler SQL application. This will make it easier to look at or change the files in the Butler Preferences Folder.

The Butler Preferences Folder contains a number of folders Butler uses to manage ports, activity logs, transaction journals, procedures, and databases. The following sections describe these folders.

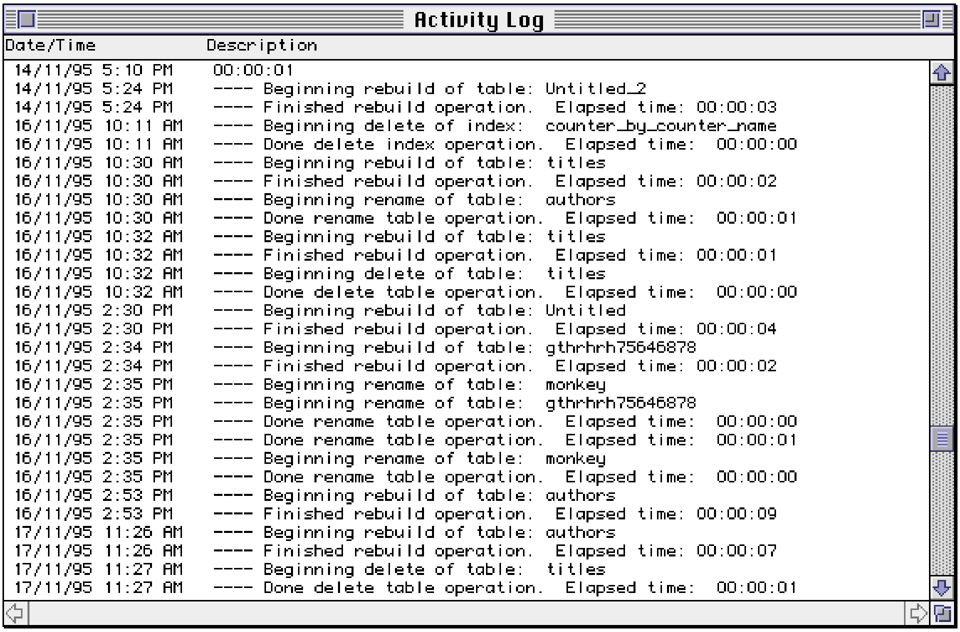
Activity Logs

Butler SQL activity logs allow you to view a log of server activity. Log entries can include:

- Client connections and disconnections
- SQL statements executed
- SQL errors
- operating system errors
- access privilege errors
- journal functions
- ODBC commands
- ODBC parameters

You set the level of detail you want logged using the Butler Preferences dialog box. (See the "Setting the activity logging level" section on page 57.) Each log entry also includes the date and time of the event as well as the user name associated with the session that triggered the event. Butler creates a new log when it is launched for the first time each day.

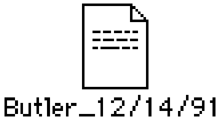
Here is an example of a Butler activity log:



```
Date/Time           Description
14/11/95 5:10 PM    00:00:01
14/11/95 5:24 PM    ---- Beginning rebuild of table: Untitled_2
14/11/95 5:24 PM    ---- Finished rebuild operation.  Elapsed time: 00:00:03
16/11/95 10:11 AM   ---- Beginning delete of index:  counter_by_counter_name
16/11/95 10:11 AM   ---- Done delete index operation.  Elapsed time:  00:00:00
16/11/95 10:30 AM   ---- Beginning rebuild of table: titles
16/11/95 10:30 AM   ---- Finished rebuild operation.  Elapsed time: 00:00:02
16/11/95 10:30 AM   ---- Beginning rename of table:  authors
16/11/95 10:30 AM   ---- Done rename table operation.  Elapsed time:  00:00:01
16/11/95 10:32 AM   ---- Beginning rebuild of table: titles
16/11/95 10:32 AM   ---- Finished rebuild operation.  Elapsed time: 00:00:01
16/11/95 10:32 AM   ---- Beginning delete of table:  titles
16/11/95 10:32 AM   ---- Done delete table operation.  Elapsed time:  00:00:00
16/11/95 2:30 PM    ---- Beginning rebuild of table: Untitled
16/11/95 2:30 PM    ---- Finished rebuild operation.  Elapsed time: 00:00:04
16/11/95 2:34 PM    ---- Beginning rebuild of table: gthrhrh75646878
16/11/95 2:34 PM    ---- Finished rebuild operation.  Elapsed time: 00:00:02
16/11/95 2:35 PM    ---- Beginning rename of table:  monkey
16/11/95 2:35 PM    ---- Beginning rename of table:  gthrhrh75646878
16/11/95 2:35 PM    ---- Done rename table operation.  Elapsed time:  00:00:00
16/11/95 2:35 PM    ---- Done rename table operation.  Elapsed time:  00:00:01
16/11/95 2:35 PM    ---- Beginning rename of table:  monkey
16/11/95 2:35 PM    ---- Done rename table operation.  Elapsed time:  00:00:00
16/11/95 2:53 PM    ---- Beginning rebuild of table: authors
16/11/95 2:53 PM    ---- Finished rebuild operation.  Elapsed time: 00:00:09
17/11/95 11:26 AM   ---- Beginning rebuild of table: authors
17/11/95 11:26 AM   ---- Finished rebuild operation.  Elapsed time: 00:00:07
17/11/95 11:27 AM   ---- Beginning delete of table:  titles
17/11/95 11:27 AM   ---- Done delete table operation.  Elapsed time:  00:00:01
```

Viewing activity logs

Activity Logs are plain-text files. Each log file's name includes the log file's creation date. You can open a log file with any text editor or word processor.



Butler_12/14/91

If you double-click a log file your Macintosh may try to open it using the TeachText or SimpleText application. If the log file has been in use for a long time or if you are logging a high level of detail, the log file may be too big to open using these applications. Try opening the log files from a different text editor or from your word processor.

Ports Folder

Each port you create is stored as a separate file in the Ports folder in the Butler Preferences Folder. When Butler SQL is started, it looks in this folder and tries to install each port it finds. If no ports are found, Butler SQL will create and install a Program Link port. (See “Chapter 1 • Connecting to Butler SQL” section on page 3 for details on creating and installing ports.)

Procedures Folder

Instead of sending commands to Butler SQL from client applications one at a time, you can group these commands into procedures and place them onto the server. This capability, known as stored procedures, also improves the performance of Butler SQL, as less network traffic is required (the procedure code is stored on the server, not sent to the server from client applications). Additionally, Butler SQL can parse the procedures (translate the procedures from the text version that you enter into a format that can be executed by the server) ahead of time, in order that the procedure is ready to be used when you need it.

Butler SQL supports several types of procedure files.

- global procedures, procedures that are available to all Butler SQL users and sessions. Global procedures must be stored in a file named “msad$procedures”.
- user or application procedures, procedures that are loaded automatically when the first user with a given name connects. User procedures are shared by all users that log on to Butler SQL with the given name. User procedures must be stored in a file named “<username>$procedures”, with <username> replaced by the name used to log on to Butler SQL.

- on demand procedures, procedures that are loaded explicitly by client applications. These procedures are available to any user, but must be loaded (using the execute file command) prior to their use. On demand procedure files can be in any file that has a valid Macintosh file name.

For more information on writing and using stored procedures, refer to the Butler SQL Programmer's Guide.

Public Databases Folder

The default location for Butler SQL databases is in the Public Databases folder in the Butler Preferences Folder. Any database in this location (or any database with an alias in this location) can be opened by a client simply by executing the standard OPEN DATABASE statement without specifying a location (provided, of course, the client has the appropriate access privileges). If you want to use a database located elsewhere on the server machine, you must specify the complete path to the database when you use it.

Note

If you place an alias to a Butler SQL database into the Public Databases folder, be sure that the name of the alias file is the same as the name of the database file. Butler SQL will not be able to use the database if the names are different.

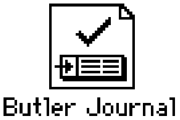
You can copy files into the Public Databases folder at any time, making them available to Butler SQL users. You do not need to restart the server in order to make new databases available.

Note

If you rename an existing database, you should quit and restart the server, particularly if the database has been used while the server was running. Additionally, if you are using access privileges, you may need to use ButlerTools to reset access to the database, as access privileges are lost as a result of changing a database's name.

Database Journals

Butler keeps track of transactions in journal files stored in the Database Journals folder (found in the Butler Preferences Folder).



The current journal file is called Butler Journal. Other journal files have the date each was detached (see below) appended to their names.

Journals allow Butler SQL to:

- undo executed statements when the ROLLBACK command is issued and
- rebuild a database using the log of transactions.

For more information on database journals, see the "Chapter 4 • Backing Up and Database Recovery" section on page 71.

Index

Disclaimer of Warranty and Limited Warranty on Media

EveryWare will replace defective distribution media or manuals at no charge, provided you return the item to be replaced with proof of purchase to EveryWare during the 90-day period after purchase. ALL IMPLIED WARRANTIES ON THE MEDIA AND MANUALS, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF THE ORIGINAL RETAIL PURCHASE OF THIS PRODUCT.

EVERYWARE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH BUTLER, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, BUTLER IS LICENSED "AS IS"AND YOU THE PURCHASER ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE.

EVERYWARE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH BUTLER, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, BUTLER IS LICENSED "AS IS"AND YOU THE PURCHASER ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE.

IN NO EVENT WILL EVERYWARE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN BUTLER OR ITS DOCUMENTION, even if advised of the possibility of such damages. In particular, EveryWare shall have no liability for any programs or data stored using Butler on data storage devices of any description, including the cost of recovering such programs or data.

THE WARRANTY AND REMIDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No EveryWare Development distributor, dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty. Some states do not allow exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Apple Documentation Disclaimer

IN NO EVENT WILL APPLE, ITS DIRECTORS, OFFICES, EMPLOYEES OR AGENTS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE APPLE DOCUMENTATION EVEN IF APPLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Copyright Notice

The Butler manual, program design, and design concepts are copyrighted, with all rights being subject to the limitations and restrictions imposed by the copyright laws of the United States of America and Canada. Under the copyright laws, this manual may not be copied, in whole or part, including translation to another language or format, without the express written consent of EveryWare Development Corporation.

2.0 2-96

Publishing Tools

This manual was created on a Macintosh using FrameMaker from Frame Technology Corporation.

The Butler SQL software, ButlerClient, ButlerHosts, ButlerTools, ButlerLink Access, Tango, the documentation, and associated materials are © 1992–1996 EveryWare Development Corp. All rights reserved. Butler SQL is a trademark of EveryWare Development Corp. All other trademarks mentioned are the property of their respective owners.