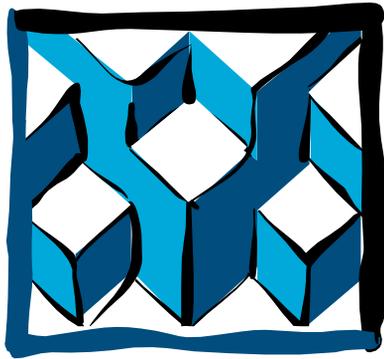


Using SBClient



SBClient[®]

A 4GL Tool Set for Rapid Application
Development and Guitization

Ardent
Software, Inc.

All rights to this publication are reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, without prior written permission from Ardent Software, Inc. If you are a licensed user of this product, Ardent Software, Inc. grants you a limited, nontransferable license to reproduce this particular document. This is limited to photocopy reproduction of a Ardent Software-supplied master copy. Copies must include all pages contained in the master copy (including this page) and must be intended strictly for internal distribution and use by the licensee, its employees, and assigns. Reproduced copies may not be transferred, in whole or in part, to any party outside the licensee's place(s) of business. The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy this software on magnetic tape, disk, or any other medium for any purpose other than the licensee's personal use.

Copyright © 2000 Ardent Software, Inc.
All Rights Reserved
SBClient Release 5
Documentation DSC-5002
Rev. 02032000

Ardent Software, Inc., reserves the right to make changes to this document and the software described herein at any time and without notice. Ardent Software, Inc. makes no warranty, express or implied, other than those contained in the terms and conditions of sale, and in no case is Ardent Software, Inc. liable for more than the license fee or purchase price of this product.

Ardent Software, Inc.
50 Washington Street
Westboro, Massachusetts USA 01581-1021
Telephone: (508) 366-3888
Facsimile: (508) 366-3669
World Wide Web: www.ardentsoftware.com

Principal Technical Writer
Tony Watkin

Contributors and Technical Reviewers
Mark Chazan, and Mike Goddard

Ardent Software is a trademark of Ardent Software Inc. uniVerse, UniData, SB+, SBOPEN, SBClient, DATA/C++, TERMULATOR, SBDesktop, ObjectCall, EasyX, DataStage, ESL, UniVerse Call Interface, UniVerse Objects, uV/Net, Retrieve, PI/open and Prime INFORMATION are trademarks of Ardent Software, Inc. TERMiTE is a trademark of Pixel Innovations, Ltd. Pick is a registered trademark of Pick Systems. Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation in the United States and or other countries. UNIX is a registered trademark in the United States and other countries, licensed through X/Open Company. All other products, standards, or company names mentioned may be trademarks or registered trademarks of their respective owners.

Contents



| | |
|---|-----------|
| Chapter 1 - Introduction to SBClient | 15 |
| The SBClient Office | 17 |
| Active Areas | 18 |
| The SBClient Menu Bar | 20 |
| Creating a Simple Session | 21 |
| Opening a Session | 22 |
| Opening Additional Sessions | 23 |
| Closing a Session | 24 |
| The Session Window | 27 |
| Resizing the Session Window | 31 |
| Saving the SBClient Desktop | 32 |
| Session Views | 32 |
| Licensing SBClient | 35 |
| Register SBClient | 35 |
| Registration Details | 36 |
| Chapter 2 - Configuring a Session | 39 |
| Communication Setup | 41 |

| | |
|-------------------------------------|-----|
| Serial Communication Setup | 41 |
| Network Communication Setup | 46 |
| Modem Communication Setup | 53 |
| Terminal Setup | 66 |
| Terminal Configuration | 67 |
| Printing | 86 |
| File Transfer Setup | 89 |
| Troubleshooting File Transfer | 93 |
| Font Setup | 95 |
| Troubleshooting Fonts | 96 |
| Color Setup | 98 |
| Keyboard Setup | 100 |
| Toolbar Setup | 102 |
| Modifying a Toolbar | 102 |
| Creating a New Toolbar | 105 |
| Status Line Setup | 107 |
| Nationality Setup | 109 |
| Script Setup | 112 |
| Miscellaneous Setup | 115 |
| GUI Parameters Setup | 118 |
| Smart Query Definitions | 121 |
| Client Based Definitions | 121 |
| Host Based Definitions | 121 |
| Environment Variables | 122 |
| SQLator Settings | 124 |

| | |
|--|------------|
| Saving and Sharing Session Configurations | 125 |
| Importing Configuration Settings | 126 |
| Exporting Configuration Settings | 128 |
| Changing a Session Configuration | 129 |
| Creating a New Session Configuration | 129 |
| Saving a Session Configuration | 130 |
| Deleting a Session Configuration | 133 |
| Portable Session Configurations | 133 |
| Chapter 3 - Host Utilities Installation | 137 |
| Host Library Install | 138 |
| Patching SB+ Server | 142 |
| Post Release 4 of SB+ Server | 142 |
| Prior to Release 4 of SB+ Server | 143 |
| Chapter 4 - Data Transfer | 145 |
| Data File Transfer | 147 |
| Standard File Transfer | 148 |
| Windows Clipboard | 149 |
| Dynamic Data Exchange | 150 |
| OLE | 151 |
| VBScript | 153 |
| Data Transfer (Host to PC) | 154 |
| Defining a Data Transfer (Host to PC) | 155 |
| Running a Data Transfer (Host to PC) | 163 |
| Saving a Data Transfer Definition | 164 |
| Load Host to PC Definition | 165 |

| | |
|---|------------|
| Deleting a Data Transfer Definition | 165 |
| Data Transfer (PC to Host) | 167 |
| Defining a Data Transfer (PC to Host) | 167 |
| Defining Host Record Ids | 168 |
| Defining the Fields | 169 |
| Defining the Transfer Format | 170 |
| Running a Data Transfer (PC to Host) | 171 |
| Saving a Data Transfer (PC to Host) | 172 |
| Load PC to Host Definition | 174 |
| Deleting a Transfer (PC to Host) | 174 |
| Downloading and Uploading Files | 175 |
| Using the Clipboard | 177 |
| Chapter 5 - Smart Query..... | 183 |
| Building a New Query | 185 |
| Defining the Selection Criteria | 185 |
| Defining the Sort Criteria | 189 |
| Defining the Display Fields | 190 |
| Creating Headers and Footers | 191 |
| Selecting Additional Options | 192 |
| Directing Your Output | 192 |
| Recalling a Stored Query | 193 |
| Saving a Query | 195 |
| Running a Query | 196 |
| Deleting a Query | 197 |

| | |
|--|------------|
| Chapter 6 - Creating Scripts | 199 |
| Trigger Sequences | 201 |
| Auto Script Learner | 204 |
| Creating Login and Logout Scripts | 206 |
| Logout Scripts | 213 |
| Creating Advanced Scripts | 214 |
| Macro Syntax | 222 |
| Control sequences | 223 |
| Methods | 224 |
| Functions | 224 |
| Internal Macro Functions | 225 |
| Chapter 7 - Troubleshooting | 243 |
| Help Menu | 243 |
| Log Options | 245 |
| Support Report Screen | 247 |
| Possible Problems/Solutions | 249 |
| Cannot connect to the host computer | 249 |
| A GUI Form Hangs | 249 |
| Losing Characters in Screen Display | 250 |
| Extraneous Characters in a Session Window | 250 |
| Appendix A - Internationalization Issues | 251 |
| Explanation of Code pages, character sets, OEM and ANSI. | 252 |
| Unicode | 253 |
| SBClient Handling of Charsets, Fonts, Code Pages and Printing | 254 |

| | |
|--|------------|
| Appendix B - SBClient Defaults | 269 |
| Communication Defaults | 270 |
| Serial | 270 |
| Network | 271 |
| Modem | 271 |
| Terminal Emulation Defaults | 274 |
| Other Defaults | 279 |
| Appendix C - Default Key Mappings | 283 |
| Appendix D - ASCII Code Table | 293 |
| Appendix E - Character Mapping and Translation Tables | 297 |
| Appendix F - SBOPEN.INI Initialization File | 301 |
| Appendix G - Command Line Options | 305 |
| Appendix H - Error Codes | 307 |

Preface



Introduction to Ardent Product Documentation

This preface contains useful information about Ardent's conventions for documenting command syntax and system output, as well as providing explanations of the specially designed icons that indicate important notes, tips, and warnings in the documentation.

Features of Ardent Manuals

Ardent strives to produce useful, high quality documentation while maintaining a consistent look and feel. It will help you to know the following conventions which are common to this and all Ardent manuals. This section explains the following:

- conventions
- elements of syntax statements
- notes, warnings, and tips
- screen captures

Conventions

All Ardent manuals depict command syntax according to industry standards. The following table lists the syntax conventions:

| Convention | Description |
|---------------------------|--|
| bold courier font | Bold courier font indicates required commands that you must enter in the manner shown. You must enter all punctuation marks that appear in bold, unless otherwise indicated. |
| <code>courier font</code> | Courier font indicates system output, such as prompt signs, responses to commands, and program code. Courier font is also used to depict paths to directories or files. |
| bold | Bold font indicates the name of an element on the screen, such as a menu, a form, or a prompt. Also indicates the names of files or directories. |
| <i>lower case italics</i> | Lower case italics is used for syntactical expression of user-supplied words, variables, and expressions |
| <i>xx</i> | Italic lower-case xx indicates a placeholder for a system id. For example, <i>xx</i> CONTROL. |

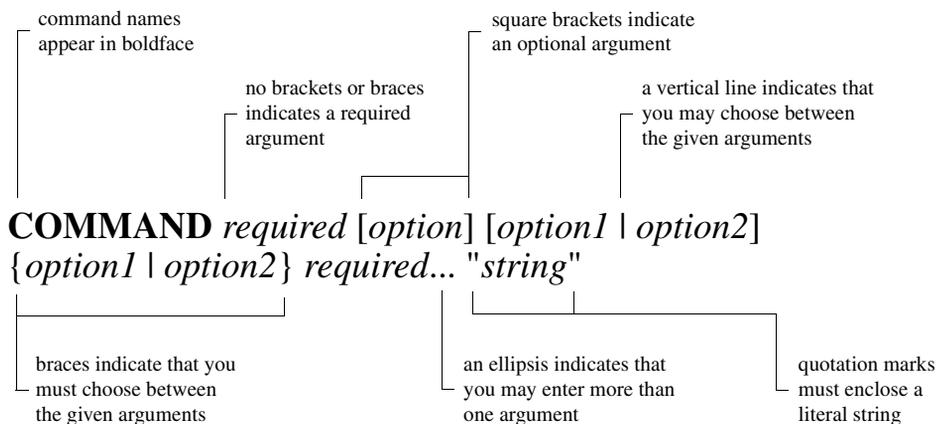
Conventions

| Convention | Description |
|------------|---|
| > | The > character separates each option in a menu hierarchy. For example, “choose Setup > Miscellaneous ”, means choose the Setup menu then the Miscellaneous option. |
| [] | Square brackets indicate an item that is optional. You may include a number of items enclosed in brackets in a Unidata command or function. |
| { } | Braces indicate that you must choose one of the items separated by the vertical bar (). |
| | Vertical bar separates option arguments from which you may choose. |
| ... | An ellipsis (...) after an argument indicates that you may use more than one argument on a single command line. |

Conventions (continued)

Elements of Syntax Statements

in some cases commands may be entered as part of paragraphs. The syntax statement includes the command name, required arguments, and options that you can use with the command. Italics represents a variable that you can replace with any valid option. The following figure illustrates the elements of a syntax statement:



Important Notes, Warnings, and Tips

You will notice several different icons throughout this manual. These icons draw attention to important information about the product.

Note

A **Note** icon indicates important additional information on the subject.

Warning

A **Warning** icon alerts you to the danger of deleting or corrupting data.

Reminder

A **Reminder** icon marks information that is discussed in detail in another chapter or another Ardent manual.

Tip

A **Tip** icon denotes shortcuts, commands, or procedures that may help you use Ardent products more efficiently.

Chapter 1 - Introduction to SBClient



SBClient allows you to create a logical connection between your PC and a host system. This connection is called a session. See Making a Connection on how to make a physical connection.

To open a session you must choose a session configuration. This is the set of configuration parameters required to create a successful connection with a host system and (if appropriate) establish communication with an application running on that host.

Note



Do not confuse a session, which is a transient connection between two systems, and a session configuration, which is the set of parameters used to create the connection.

You may run multiple sessions concurrently on your PC. For example, you could have:

- a session (emulating a VT220 terminal) connecting you to a host application,
- a session connecting you to the same host so that you can transfer files to or from your PC, and
- a session connecting you to a different host running a database inquiry application.

Session configurations may already have been created by your system administrator, or you may create your own session configurations. You may create and store (with a unique name) as many different session configurations as you require.

Each session runs in its own window, and is identified by the name of its session configuration. This name may be followed by a colon and an arbitrary number (for example, Default Network:3). This number differentiates between sessions with the same name. You may only have one open session at a time per serial physical connection, but you may have more than one open session with a network physical connection.

A session is deemed the “active session” when its window has focus (it is the topmost window and has control of the cursor). It is this session that is acted upon by any mouse or keyboard activity, or menu option selection.

Note

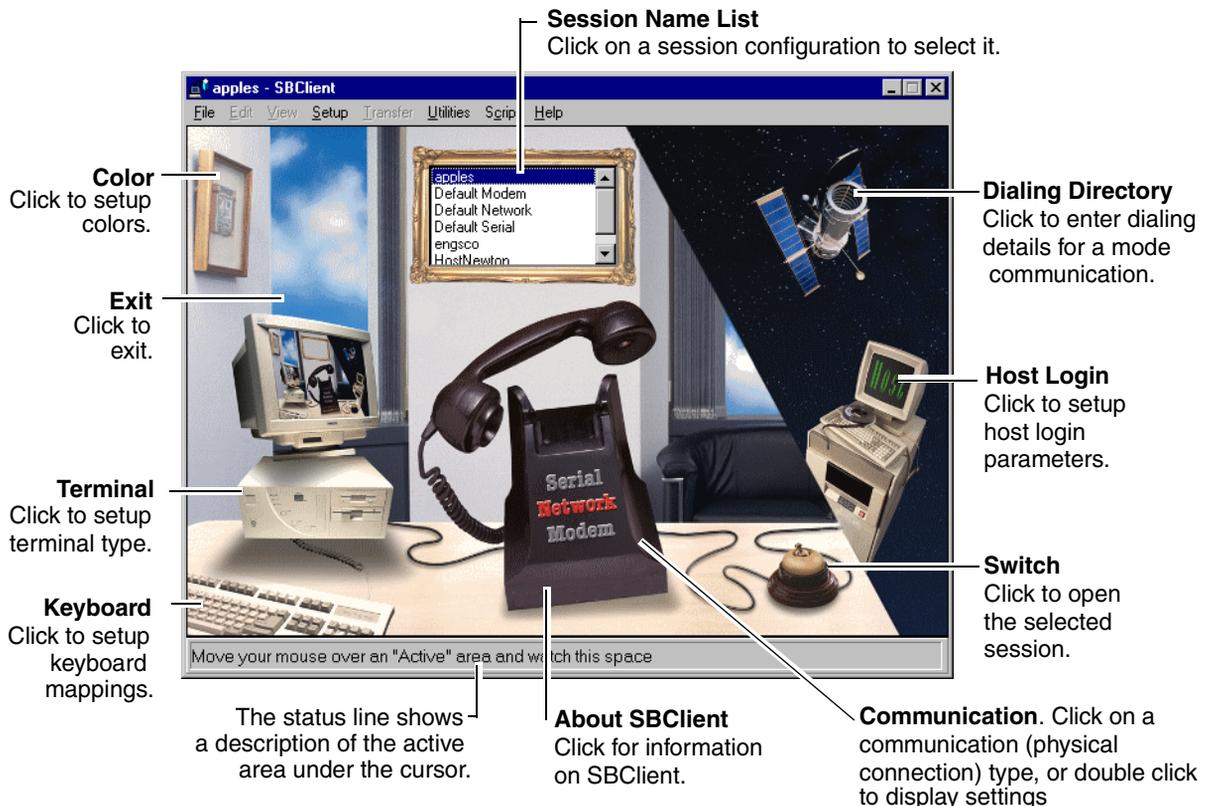


A session that has focus is called the active session, to distinguish it from any other open sessions; although all open sessions represent live connections. You may have many open sessions but only one session may be active at a time.

When you close a session, the logical connection between your PC and host is broken.

The SBClient Office

When SBClient is launched it presents a graphical environment known as the SBClient Office is presented. This is shown below:



The SBClient Office represents a new, unopened session. SBClient allows you to easily maintain your session configurations, and open a session. The SBClient Office is displayed when you:

- first invoke SBClient.
- choose **File > New Session**. A separate SBClient Office is displayed for each new session you create.
- click the **New Session** button on the dialog displayed when you exit the active session.

Note



If you have previously used the **File > Save Desktop** option to save details of your open sessions, the SBClient Office will not appear when you initially invoke SBClient. The saved session windows will be displayed instead.

The SBClient Office is made up of a montage of images incorporating active areas. An active area performs the same function as a Windows button. You click on an active area to perform an operation just as you would click on a button.

The active areas on the SBClient Office provide you with an alternative means of selecting commonly used options from the SBClient menu bar. For example, clicking on the PC keyboard displays the PC Keyboard Setup dialog (as if you had selected **Setup > Keyboard**).

As you move your mouse pointer around the SBClient Office, you will see your standard pointer change to a small hand. This indicates you are in an active area. A description of the active area is displayed in the status line at the base of the SBClient Office.

Active Areas

Active areas are divided into two parts on the SBClient Office, with PC active areas on the left and host active areas on the right. Active areas are:

Session Name List

This is located in the picture frame above the telephone. It provides a list of your preconfigured session configuration names. Click on a session configuration name to select that configuration. Double clicking on a session configuration name will open a session with that configuration (as if you had clicked the Switch active area, or you had selected **File > Open Session**).

Communication

This is located on the telephone. Communication types (based on the means of physical connection: serial, network or modem) are shown embossed on the front of the phone, with the communication type of the current session highlighted. Click on an unselected communication type to select it. Click on the telephone or double-click on a communication type to display the **Communication Setup** dialog (as if you had selected **Setup > Communication**).

Terminal

This is located on the PC to the left of the telephone. It displays the **Terminal Setup** dialog (as if you had selected **Setup > Terminal**). This dialog displays a selection list of the types of terminal that SBClient emulates, with the terminal type for the current session highlighted.

Keyboard

This is located on the PC keyboard. It displays the **PC Keyboard Setup** dialog (as if you had selected **Setup > Keyboard**).

Color

This is located on the picture above the PC. It displays the **Color Setup** dialog (as if you had selected **Setup > Color**).

Dialing Directory

This is located on the satellite. It displays the **Dialing Directory** dialog, where you can maintain details of modem phone numbers (as if you had selected **Setup > Dialing Directory**). This is only active if **Modem** is selected as the communication type.

About SBClient

This is located on the base of the telephone, and displays information about your version of SBClient (as if you had selected **Help > About SBClient**).

Switch

This is located on the switch to the right of the telephone. It opens a session using the currently selected session configuration (as if you had double clicked on the session list, or selected **File > Open Session**).

Exit

This is located on the window above the PC. It allows you to exit SBClient (as if you had selected **File > Exit All**).

The SBClient Menu Bar

This provides a range of options for maintaining your session configurations, opening sessions and writing scripts. Some of these options may also be invoked by selecting the appropriate active area. Options that are available only from the session window menu bar (displayed when you open a session) are grayed out.

You can choose an option using your mouse, the arrow keys (when the menu is activated using the mouse or the Alt key) or by pressing Alt and typing the letter underscored in the option and any suboptions. SBClient displays a short help prompt on the status line for each option that you highlight.

Creating a Simple Session

From the Office screen select either **Default Serial**, **Default Network** or **Default Modem** depending on the type of session you wish to configure.

If you select **Default Network** or **Default Modem** then select **Setup > Communication**. The screen that is displayed will reflect the defaults applicable to the communication type. For network communications you must enter the host name or IP address of the host. For modem communications enter the telephone number of the host to be dialied. Then click the **OK** button.

Select **File > Save As** then enter a name to save the session.

Assuming the defaults are correct for your hardware you are now ready to open a session with the host. If the defaults are different or if you experience difficulties when opening a session consult [“Chapter 2 - Configuring a Session”](#) for detailed description of all session parameters.

Opening a Session

This topic assumes that one or more session configurations have already been set up for your system.

To open a session from the SBClient Office:

1. Select a configuration from the Session Name List.

You can change the configuration settings for this session by clicking on the appropriate active area or by choosing the appropriate option from the Setup menu. Any changes you make will apply only to the current session unless you save those changes.

2. When you are satisfied with your session configuration click on the Switch active area. This will open the session. You may also open a session by double clicking on the session name in the Session Name List, or by choosing one of the following:

File > Open Session > Classic Session

File > Open Session > Document (SBC) Session

SBClient opens the session using its current configuration settings. If you have selected a modem connection and a telephone number, SBClient connects to the modem, dials the number, then establishes a connection to the remote host. If you have selected a network connection, SBClient establishes a connection over the local area network to the specified host. If you have selected a serial connection, SBClient establishes a connection to the host through the serial port.

When the session has been successfully opened (a logical connection to the host system has been established), SBClient displays a session window for that particular session. The full range of menu options are available from the session window.



Tip Once you have opened a session you cannot get back to the SBClient Office for that session. The SBClient Office is only displayed if you choose **File > New Session**, or if you close a session and click the New Session button.

Opening Additional Sessions

SBClient allows you to open more than one session concurrently. You can open additional sessions that use the same configuration as your active session, or that use a different configuration. Each additional session is displayed in its own session window.

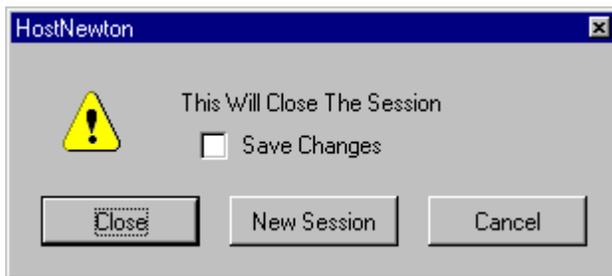
To open a new session using the same configuration as your active session:

- From the session window menu, choose **File > Duplicate Session**. SBClient opens another session using the active session's configuration. SBClient uses the same session name as that duplicated, but appends a different number to it, so that you can differentiate between sessions.

Closing a Session

To close an open session from the session window:

1. Select the session window for the session you wish to close, if there is more than one session open.
2. Choose **File > Close Session**. This displays the following dialog:

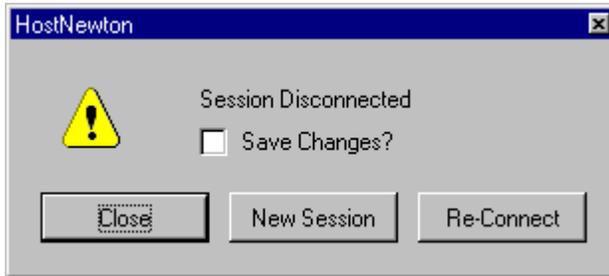


If you have made changes to the session configuration and have not saved them, SBClient displays an additional **Save Changes** checkbox. Select this to save changes made to the session configuration.

3. Click one of the displayed buttons:
 - **Close** button closes your connection to the host system and returns you to the previous active session, if you have more than one session open. If you are closing the only open session, SBClient exits.
 - **New Session** button closes the active session and displays the SBClient Office, where you may open a new session or create a new session configuration.
 - **Cancel** button. Returns you to the session.

To close an open session from within the session:

1. Enter the appropriate exit or quit command in the session area.
2. This displays the following dialog:



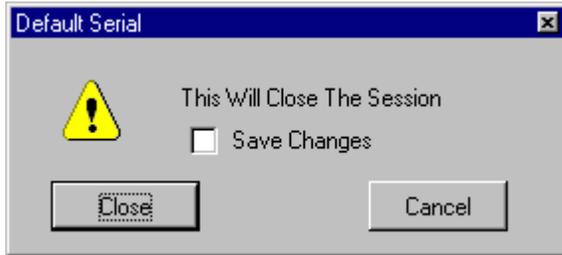
If you have made changes to the session configuration and have not saved them, SBClient displays an additional **Save Changes** checkbox. Select this to save changes made to the session configuration.

3. Click one of the displayed buttons:
 - **Close** button closes your connection to the host system and returns you to the previous active session, if you have more than one session open. If you are closing the only open session, SBClient exits.
 - **New Session** button closes the active session and displays the SBClient Office, where you may select a new session or create a new session configuration.
 - **Re-Connect** button. Reconnects you to the session.

To close an unopened session from the SBClient Office:

1. Choose **File > Close Session**.

If you have made changes to the current session configuration and have not saved them, SBClient displays a dialog with a **Save Changes** checkbox:

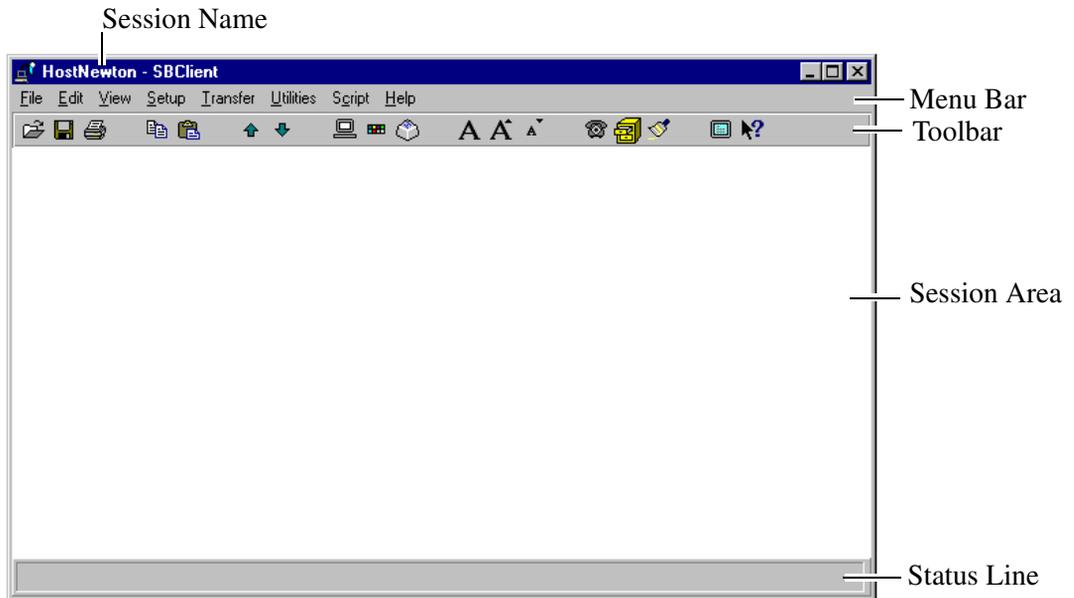


If you select the **Save Changes** checkbox, any existing session configurations will be replaced.

2. Click the **Close** button. SBClient returns you to the session window for the previous active session, if you have more than one session open. If you are closing the only open session, SBClient exits.

The Session Window

A new session window is displayed for each session you open. The session window is made up of the following elements:



Session Name

The name of the open session. This name may be followed by a colon and a number (eg UniData on Unix Host:2). The number increments for each new session so that you can differentiate between sessions with the same name.

Session Area

Displays user input to and host output from the host computer. The session area is the equivalent of a terminal screen.

Menu Bar

Provides the full range of SBClient menu options.

Toolbar

Provides shortcut buttons for commonly used menu options. You can toggle the display of the toolbar. SBClient's default toolbar buttons are:

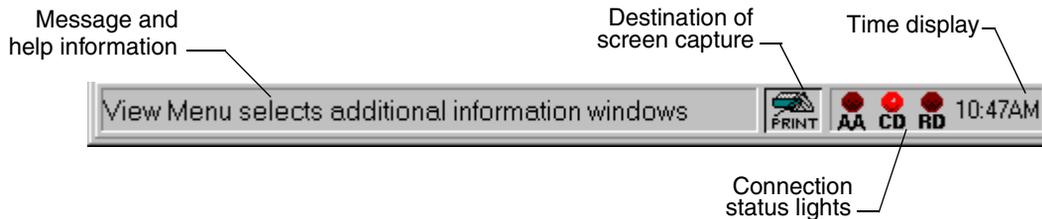
| To do This | Use This Option | Use This Button |
|---------------------------------|------------------|--|
| Copy to clipboard | Edit > Copy |  |
| Paste from clipboard | Edit > Paste |  |
| Scroll back a page | |  |
| Scroll forward a page | |  |
| Define terminal characteristics | Setup > Terminal |  |
| Define colors | Setup > Color |  |
| Define keyboard | Setup > Keyboard |  |

Session Commands

| To do This | Use This Option | Use This Button |
|---------------------------|---------------------------|---|
| Choose font | Setup > Font |  |
| Next font size up | |  |
| Next font size down | |  |
| Display Dialing Directory | Setup > Dialing Directory |  |
| Launch the File Manager | Utilities > File Manager |  |
| Launch the Form Painter | Utilities > Form Painter |  |
| Use the full screen | View > Full Screen |  |
| Switch on pointer help | |  |

Session Commands

Status Line



The status line displays:

- SBClient messages and help information.
- the destination for your screen capture. This is defined using **Edit > Capture To**, and is indicated on the status line by the following icons:
 - Clipboard 
 - Printer 
 - File 
- status lights and time information. This is defined using **Setup > Status Line**.

You can toggle the display of the status line.

Command Line



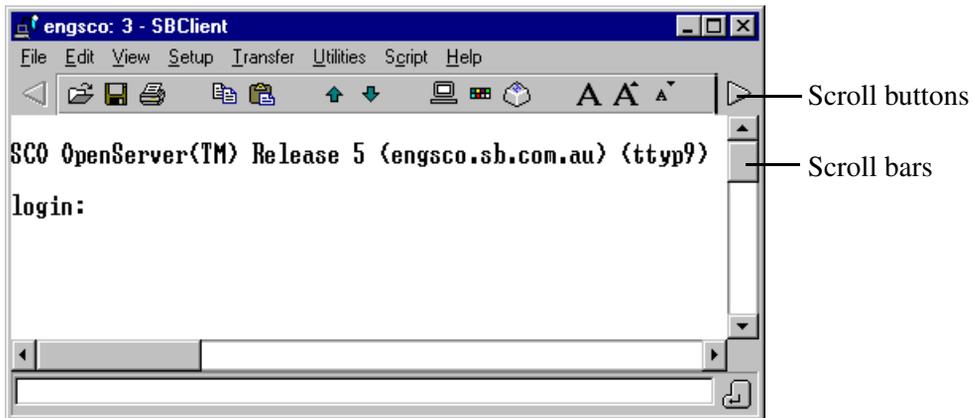
The command line is displayed instead of the status line if toggled on from the **View** menu.

You can use the command line to enter commands or responses that you would normally type in the session area.

To send an entry in the command line to the host computer, press Enter.

To recall a command from the displayed list, double-click the desired command or highlight it and press Enter.

Resizing the Session Window



When you resize a session window, the following adjustments are made:

- If you reduce the width of the session window so that the menu cannot be displayed on a single line, SBClient wraps the menu onto subsequent lines.
- If you reduce the width of the session window so that the toolbar cannot be displayed on a single line, SBClient adds left and right scroll buttons.
- If the Resize Font checkbox in the Miscellaneous Setup dialog is checked (the default) the font is rescaled to fit the resized session window. There are a limited number of sizes for each font supplied; therefore, the closest size that fits the resized window is selected. SBClient does not allow the window to be resized greater than the width of the screen.
- If the Scroll Bars checkbox in the Miscellaneous Setup dialog is checked (the Resize Font checkbox and Scroll Bars checkbox are mutually exclusive), scroll bars are added to the session window if it is resized smaller than the displayed text. If you resize the session window so that it is larger than the session window size for the current font, the bitmap specified in the Tile BMP box (in the Miscellaneous Setup dialog) is used to fill the extra space around the text.

Saving the SBClient Desktop

Use **File > Save Desktop** to save the positions and sizes of open session windows.

The next invocation of SBClient will automatically open the session windows you have saved. The SBClient Office will not be displayed.

To reset the desktop, (so that sessions are not automatically invoked, and the SBClient Office is displayed the next time you invoke SBClient), ensure there are no open sessions, display the SBClient Office and choose **File > Save Desktop**.



If you have a session active then you will need to select **File > New Session** to display the SBClient Office screen from which you can then select **File > Save Desktop**, having first closed all previous session windows.

Session Views

The **View** menu allows you to change the presentation of your session window. Its options are:

Toolbar

Toggles the display of the toolbar on the session window. A tick indicates that the toolbar is displayed.

Command Line

Toggles the display of the command line on the session window. A tick indicates that the command line is displayed.

Status Line

Toggles the display of the status line on the session window. A tick indicates that the status line is displayed.

Warning

You can display either the command line or the status line at the bottom of the session window, but not both at the same time.

Clipboard Viewer

Opens a separate window, showing the contents of the clipboard.

This allows you to view information on the clipboard when using the Edit options to copy and paste information.

Note

For this option to function the Clipboard Viewer needs to have been installed in the Windows environment.

Full Screen

Uses the entire PC screen to display the session area.

When you choose this option, your session area fills the entire screen (hiding your normal Windows display). You can press Alt+Enter to return to the original Windows display, or Alt+Tab to switch between Windows tasks.

You can access the SBClient menus in full screen mode by pressing the Alt key then the down arrow key, or by entering the keyboard shortcut. For example, pressing Alt+F displays SBClient's File menu.

The full screen mode continues to display the characters in the same font family used prior to switching modes as long as there is a screen font size large enough for an entire screen (TrueType fonts can display at any size — this limitation only applies to screen fonts). If there is no screen font size large enough, SBClient displays the character in the System Builder font, which is installed with SBClient.

SBClient remembers full screen mode. If you exit a session in full screen mode, the next time you open the session, SBClient will display it in full screen mode.

Note

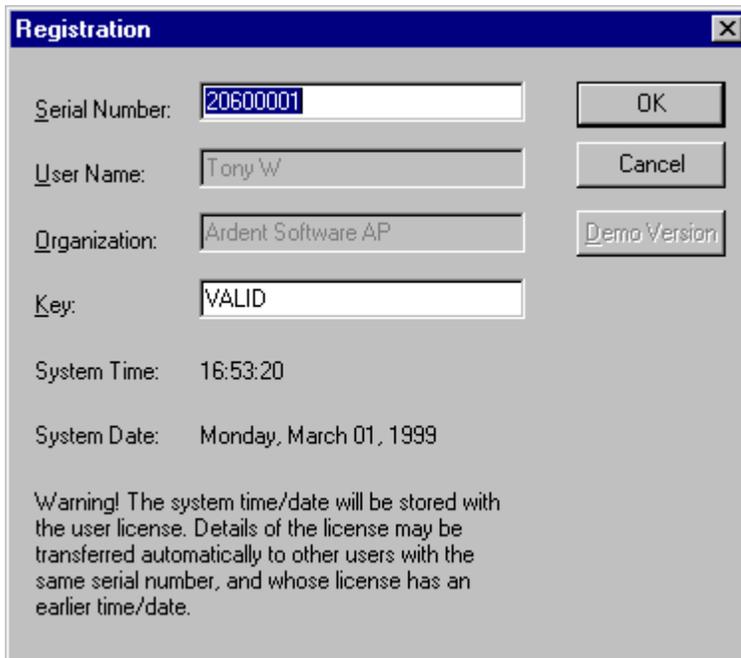


This option is available only if you have selected **Resize Font** as the Resize Behavior mode in the Miscellaneous Setup dialog.

Licensing SBClient

Register SBClient

This can be used to change registration details for this copy of SBClient



The image shows a Windows-style dialog box titled "Registration". It contains several input fields and buttons. The "Serial Number" field is highlighted with a blue selection bar and contains the text "20600001". The "User Name" field contains "Tony W". The "Organization" field contains "Ardent Software AP". The "Key" field contains "VALID". Below the input fields, the "System Time" is displayed as "16:53:20" and the "System Date" as "Monday, March 01, 1999". At the bottom of the dialog, there is a warning message: "Warning! The system time/date will be stored with the user license. Details of the license may be transferred automatically to other users with the same serial number, and whose license has an earlier time/date." On the right side of the dialog, there are three buttons: "OK", "Cancel", and "Demo Version".

Parameters are:

Serial Number

This is the serial number which will have been issued to you at the time of purchase.

User Name

This is the name of the authorised user of your SBClient installation.

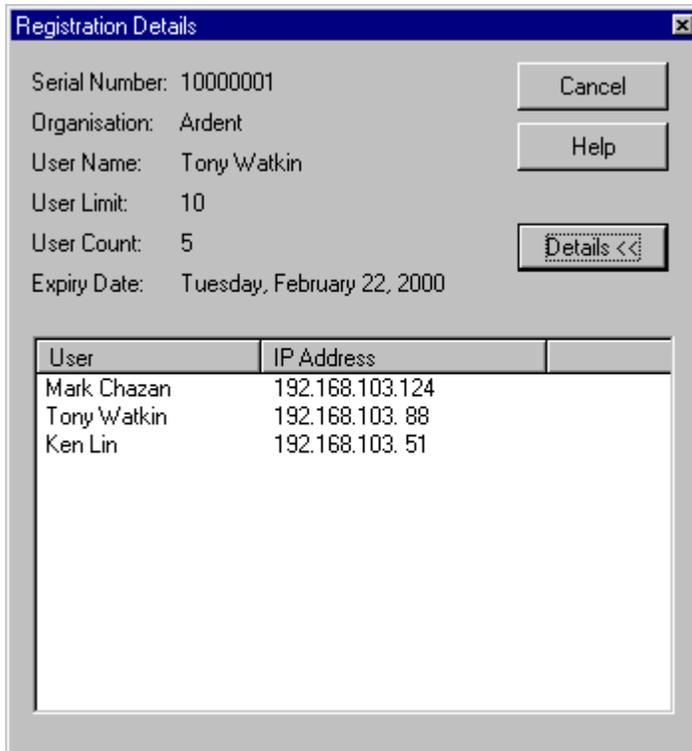
Organization

This is the name of the corporation/organization that has purchased SBClient.

Key

This is the authorization code which is required to validate your copy of SBClient at the time of software installation.

Registration Details



The Registration Details screen (viewed using the **Help > Registration License Details** option) provides details of the license for this copy of SBClient. These details includes:

- The product serial number.
- The organization to which the product is registered to.
- The user whose computer the product is registered on.
- The user limit for your license.

- The number of licenses currently in use on your network.
- An expiry date for the license, if any.

Click the Details button for information on current users of SBClient. This provides a list of users and their IP address.

Chapter 2 - Configuring a Session



When SBClient is first installed it provides a set of default serial, network and modem configurations. If you are running SBClient on a network, your system administrator may have added to or modified these default configurations

You can use the Setup menu to change the existing configurations to meet your particular communication requirements. If you make changes to an existing session configuration, you can: discard them at the end of the session, permanently save them as the new settings for the session configuration, or permanently save them as a new session configuration.

Note



To configure a session correctly you need some knowledge of the host computer, your connection to the host and the terminal you are emulating. If you are unsure of any parameter (or if you have difficulties in operating a session after configuration), consult the appropriate hardware, network or terminal documentation and check your configuration, or consult your system administrator.

The **Setup** menu is available from both the SBClient Office and the session window (some of the Setup options are also represented as active areas on the SBClient Office).

Options are:

- **Communication.** Allows you to configure the communication method used by the session to connect your PC with the host computer. The communication method chosen depends on the type of physical connection used: serial, network or modem.
- **Terminal.** Allows you to configure the session to emulate the characteristics of a particular terminal type (eg VT220).

- **File Transfer.** Allows you to configure some overriding flow control options.
- **Font.** Allows you to select initial font to be used during a session.
- **Color.** Allows you to associate color with video attributes such as foreground and background, reverse video etc.
- **Keyboard.** Allows you to reprogram any key on the keyboard to send a different sequence to the host. This allows you to emulate many different types of keyboard.
- **Toolbar.** Allows you to customize the toolbar located at the top of your session window. For example, you could create buttons for commonly used menu options, host commands or DOS/Windows programs.
- **Status Line.** Allows you to select the items to be displayed on the status line located at the bottom of your session window.
- **Nationality.** Allows you to define the character set to be used during a session.
- **Script.** Allows you to specify additional script data substitution parameters, to be used for all scripts run from a session.
- **Miscellaneous.** Allows you to define additional settings such as the characteristics of your cursor and window resize behavior.
- **Dialing Directory.** Allows you to specify telephone numbers for use when opening a session using a modem.
- **GUI Parameters.** Allows you to define various GUI parameters when running in GUI mode.
- **Smart Query Definitions.** Smart Query definitions can be stored on either the client or the host. You can configure Smart Query to use client or host definitions through the Smart Query Config screen located on the Setup menu. By default, Smart Query stores definitions on the client.
- **Environment Variables.** The **Environment Settings** dialog allows you to view the SBClient system variables (you cannot edit these values) and to view and edit the SBClient user variables.
- **SQLator Settings dialog.** This allows you to view and set SQLator variables. This option is only present if you have Visual Atlas installed on your computer.

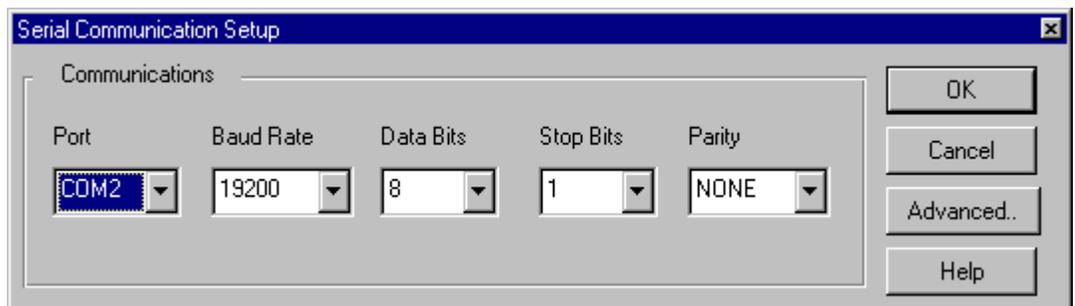
Communication Setup

The **Communication** option on the **Setup** menu allows you to configure the communication method used by the session to connect your PC to the host computer.

When you select this option, SBClient displays a configuration dialog specific to the type of communication used in the current session:

- The configuration of your serial port (ie COM1, COM2, etc).
- The configuration for your host network connection.
- The initialization commands and dialing options for your modem, as well as the configuration of the serial port the modem will be using.

Serial Communication Setup



The **Serial Communication Setup** dialog allows you to change the communication settings for your serial port.

Parameters are:

Port

The serial port you are using to connect to the host. COM1 through COM9 may be selected.

Baud Rate

The baud rate for transmission. When you set the baud rate, ensure the port on the host computer connected to your PC is set to the same baud rate. 300 through to 115200 may be selected. Consult your system administrator or the appropriate host documentation for more information.

Data Bits

The number of data bits together with the number of stop bits (see Stop Bits) determines how a character is encoded. In addition, the number of data bits is usually set together with a given parity type (see Parity). 4 through to 8 may be selected. Normally you would choose 8 data bits per character. If you are using parity, the link may only support 7 data bits.

Stop Bits

The number of stop bits, together with the number of data bits (see Data Bits) determines how a character is encoded. 1, 1.5 and 2 stop bits may be selected. Normally you would set Stop Bits to 1 (the default), regardless of the number of data bits.

Parity

Parity is one of several methods the communication program uses for detecting transmission errors. Normally you would choose no parity (None). If the communication link can only be made with parity, then select the parity method required: Odd, Even, Mark or Space.

Parity, data bits and stop bits are normally set using either of the following combinations:

- 8 data bits, 1 stop bit, no parity
- 7 data bits, 1 stop bit, even parity

Consult your system administrator or the appropriate host documentation for more information.

Advanced Serial Configuration

The screenshot shows the 'Advanced Serial Configuration' dialog box. It features a title bar with a close button. The main area is divided into several sections:

- Flow Control:** Contains three checkboxes: 'XON/XOFF' (unchecked), 'RTS/CTS' (checked), and 'DTR/DSR' (unchecked). To the right of these are two text input fields: 'XON Char' with the value '17' and 'XOFF Char' with the value '19'.
- Break Duration (ms):** A text input field containing the value '100'.
- Time Out (ms):** A section containing two text input fields: 'CTS' with the value '0' and 'DSR' with the value '0'.
- Send CR on Connect:** A checkbox that is currently unchecked.

On the right side of the dialog, there are three buttons: 'OK', 'Cancel', and 'Help'.

Parameters are:

Flow Control

SBClient uses flow control to indicate to the host whether it is ready to receive data. Without the correct flow control, data sent to or received from the host can be lost.

For software flow control, select the XON/XOFF checkbox. Ensure the host is set correspondingly on the connected port. You also need to set:

- **XON Char.** A decimal number specifying an ASCII character that, when sent, indicates to the host or PC that it can resume sending data. The XON character is sent after an XOFF has been previously sent. The default value is 17 (Ctrl-Q).
- **XOFF Char.** A decimal number specifying an ASCII character that, when sent, indicates to the host or PC that the buffer is full and that it should stop sending data. The default value is 19 (Ctrl-S).

For hardware flow control, select:

- RTS/CTS checkbox (Request To Send/Clear To Send) or
- DTR/DSR checkbox (Data Terminal Ready/Data Set Ready)

You will probably be using RTS/CTS flow control if you are using a modem.

Break Duration (ms)

The duration (in milliseconds) of the break signal that is used by SBClient to send interrupts to the host software. The default value is 100 milliseconds.

Time Out (ms)

The duration (in milliseconds) of the loss of each of the CTS (Clear To Send) and DSR (Data Set Ready) signals before a timeout occurs. The default value is 0 for both signals.

Send CR on Connect

Sends a CR (Carriage Return) character on connection. After a connection has been established, some host computers will not respond with a login request until the user presses the Enter key to generate a CR character. Select this option to generate the CR automatically on connection.

Troubleshooting Serial Connections

There are two types of connectors (called D-connectors) used with serial cards.

One type has 25 pins (called a DB-25 connector) and the other type has 9 pins (called a DB-9). You may have both on your PC or host computer. D-connectors come in two genders: male and female. Usually, PCs have male connectors, while terminals and modems have female connectors, and host computers have any combination of gender. A common configuration in PCs is a 9-pin connector for COM1, and a 25-pin connector for COM2, if fitted. Do not confuse the normally female DB-25 connector with your PC's parallel printer port connector.

Computers require some means of flow control (also called handshaking) when talking to each other. Otherwise, one computer could send data faster than the other was capable of processing and characters would be lost. Flow control can be handled by software or hardware. Software flow control using XON/XOFF handshaking is the simplest form of handshaking and requires only 3 pin connections.

The cabling required to make a connection will usually be similar to that used for connecting an ASCII terminal (there is no standard cable that is suitable for all types of serial connection). ASCII terminals generally use pins 2 (RD), 3 (TD), and 7 (GND) while most serial cards on PCs use pins

2 (RD), 3 (TD), and 5 (GND) or 7 (GND). You may need a gender changer and possibly a 25- or 9-pin converter to adapt the cable from the host to the PC's serial port connector. The following table illustrates standard pin numbering for male 9-pin and 25-pin serial connectors.

| PC | Host | PC | Host |
|----------|--------|----------|--------|
| 9-pin | 25-pin | 25-pin | 25-pin |
| 2 RD < | 3 TD | 2RD < | 3 RD |
| 3 TD > | 2 RD | 3 TD > | 2 RD |
| 5 GND <> | 7 GND | 7 GND <> | 7 GND |

Serial Pin Assignments

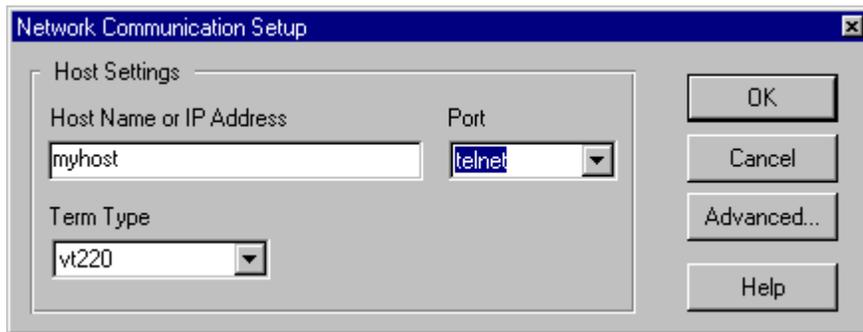
If none of these configurations work for you, try swapping pins 2 and 3 on one end of the cable. You may also try using a null modem cable or null modem adaptor. A null modem cable is a short connector box or cable containing the twists necessary for two computers to exchange information. Consult your serial card documentation for information on pin numbers and the corresponding connections.

The use of a line tester or breakout box might help resolve problems in making a physical connection. A breakout box lets you use short wires to connect specific pins. A line tester is a straight-through connector box (plug-to-socket) which has LEDs highlighting the active pins. Pins 2 and 3 flash as data is sent and received from the host computer.

Even though you may not understand the lights in detail, you can quickly identify a wrongly cabled port by checking whether the LEDs for pins 2 and 3 are lit, and by comparing the result using a known working port. The flashing lights of the line tester are also useful for checking that data is being exchanged between computers during a data transfer.

For further information you should refer to the ANSI RS232C standard.

Network Communication Setup



The **Telnet Setup** dialog allows you to change the connection settings to your network host machine.

Parameters are:

Host Name or IP Address

This is either the name of the host machine or the IP address of the host machine.

Port

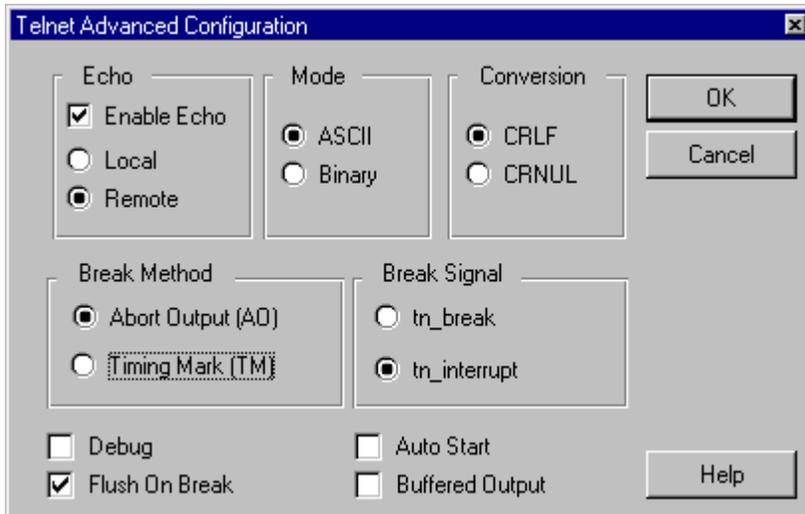
This is the port to be used for connection. A number of protocol settings are available to choose from e.g. Telnet, SMTP, WWW etc.

Term Type

This is the terminal emulation mode that SBClient will use. A number of terminal settings are available to choose from e.g. VT52, VT100 etc.

Configure Windows Sockets (TCP/IP)

The **Windows Sockets (TCP/IP) Configuration** dialog is presented when for the **Default Network** option you select **Setup > Communication** and click on the **Advanced** button.



Mode

Determines the format in which data is transferred. Options are:

- ASCII (used to transmit 7 bit ASCII characters). ASCII mode translates end of line sequences (LF and CR) so that you end up with the one required by your operating system. For example, Unix only has one LF at the end of each line whereas DOS requires an LF and CR pair at the end of each line. This is the mode to use when transferring text files.
- Binary (used to transmit 8 bit ASCII characters). Binary mode transfers data exactly. This is the mode to use for archive or program file transfers.

Conversion

When operating in ASCII mode, the host computer converts End Of Line (EOL) to either:

- CRLF. Carriage Return and Line Feed.
- CRNUL. Carriage Return and NUL.

The choice depends on the host computer.

Echo

Determines the way in which characters are echoed back to the PC screen. This operates if the Enable Echo parameter is selected. Options are:

- **Enable Echo.** Echoes characters back to the screen. The Enable Echo checkbox needs to be selected in order for the Echo parameter below to operate.
- **Local.** In Local Echo mode, the terminal sends your keystrokes to the remote host and displays them automatically on the screen. The host does not echo the characters back to the PC screen. Telnet is responsible for the display of characters on the screen.
- **Remote.** If your session is in Remote Echo mode, every keystroke you type makes a round trip from your PC to the remote host and back to your PC screen. Network congestion or too many network connections between PCs and the remote host can cause this exchange to be time-consuming and consequently may slow down typing.

To speed up your display, use the Local Echo mode (if Local Echo mode works with the host application you use).

Flush On Break

Clears the host output data from the interrupted process when a break signal is sent to the host.

Buffered Output

Allows data to be accumulated on the host where it is generated until a complete line of data is ready for transmission, or until some locally defined explicit signal to transmit data occurs. This reduces network traffic.

This parameter should be configured only when Telnet is operating in Local Echo and ASCII modes.

Break Method

A break signal may be sent to the host when the user believes a process is in a continuous loop, or when an unwanted process has been inadvertently started. This parameter determines the consequences of sending a break signal.

Options are:

- **Abort Output.** Allows the process that is generating output to run to completion but without sending the output to the user's terminal.
- **Timing Mark.** Discards all output locally until a Will/Won't Timing Mark command is received from the Telnet server on the host.

Break Signal

Determines whether the **Utilities > Send Break** option sends a break (tn_break) or interrupt (tn_interrupt) signal.

The host software determines the effect of each of these signals. Consult your host documentation for more information.

Debug

Allows Telnet to display commands and options during login and during interactions with the host (for example, during an interrupt process).

Auto Start

When the Auto Start checkbox is disabled, Telnet waits for host negotiations before it responds with any messages.

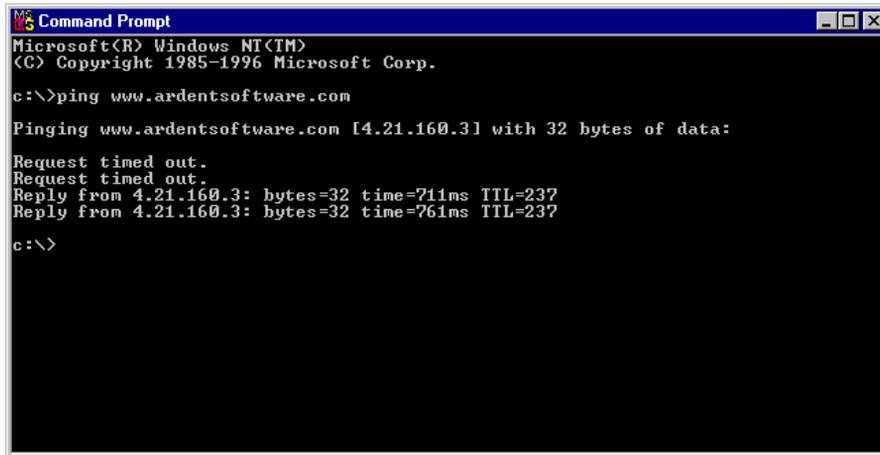
Troubleshooting Network Communications

Windows Sockets

Ping hostname. Ping is a tool that helps to verify IP-level connectivity. The ping command can be used to send an Internet Control Message Protocol (ICMP) echo request to a hostname or IP address. You should first try pinging the IP address of the host to see if it will respond, as this is the simplest test. If that succeeds, then try pinging the hostname. Ping uses name resolution to resolve the name to an address; therefore, if pinging by address succeeds but pinging by name fails, the problem lies in name resolution, not network connectivity.

Use ping -? to see what command-line options are available. Ping allows you to specify the size of packets to use, how many to send, whether to record the route used, what Time To Live (TTL) value to use, and so on.

The following example demonstrates the use of ping.:



```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

c:\>ping www.ardentsoftware.com

Pinging www.ardentsoftware.com [4.21.160.3] with 32 bytes of data:

Request timed out.
Request timed out.
Reply from 4.21.160.3: bytes=32 time=711ms TTL=237
Reply from 4.21.160.3: bytes=32 time=761ms TTL=237

c:\>
```

Note



If your ping time to your target host is >200ms then you should not use SB+ Server in GUI mode because the network connection will not be fast enough to support the GUI mode.

By default, ping waits 1 second for each response to be returned before timing out. If the remote system being pinged is across a high-delay link such as a WAN or the Internet, responses could take longer to be returned. The -w (wait) switch can be used to specify a longer time-out.

Tracert. This is a route tracing utility. Tracert uses the IP TTL field and ICMP error messages to determine the route from one host to another through a network. Sample output from the tracert command is shown in the following graphic:



```
C:\>tracert mail.sb.com.au

Tracing route to mail.sb.com.au [203.14.103.99]
over a maximum of 30 hops:

  1  <10 ms    <10 ms    <10 ms    192.168.103.1
  2  <10 ms    <10 ms    <10 ms    sb00.sb.com.au [203.14.103.99]

Trace complete.
```

Nslookup. This was added in Windows NT 4.0 and is a very useful tool for troubleshooting DNS problems, such as host name resolution. When you start nslookup, it shows the hostname and IP address of the DNS server that is configured for the local system, and then displays a command prompt. If you type a question mark (?), nslookup shows the different commands that are available.

To look up the IP address of a host using the DNS, type the host name and press the Enter key. Nslookup will default to using the DNS server that is configured for the computer it is running on, but you can focus it on a different DNS server by typing server name (where name is the host name of the server you want to use for future lookups).

When you use Nslookup, you should be aware of the domain name devolution method. If you type in just a hostname and press the Enter key, nslookup appends the domain suffix of the computer (such as ardentsoftware.com) to the hostname before it queries the DNS. If the name is not found, then the domain suffix is "devolved" by one label (in this case, ardentsoftware is removed and the suffix becomes .com), and the query is repeated. If a fully qualified domain name is typed in (as

indicated by a trailing dot) then the DNS server is only queried for that name and no devolution is performed. To look up a hostname that is completely outside of your domain, you must type in a fully qualified name.

```
C:\>nslookup
Default Server: sb00.sb.com.au
Address: 203.14.103.99

> cerberos
Server: sb00.sb.com.au
Address: 203.14.103.99

Non-authoritative answer:
Name: sb13.sb.com.au
Address: 203.14.103.67
Aliases: cerberos.sb.com.au

> www.ardentsoftware.com
Server: sb00.sb.com.au
Address: 203.14.103.99

Non-authoritative answer:
Name: www.ardentsoftware.com
Address: 4.21.160.3

>
```

IPConfig is a command-line utility that prints out the TCP/IP-related configuration of a host. When used with the /all switch, it produces a detailed configuration report for all interfaces,

including any configured serial ports (RAS). Output can be redirected to a file and pasted into other documents.

Windows NT IP Configuration

```

Host Name . . . . . : gumby.sb.com.au
DNS Servers . . . . . : 203.14.103.99
                    192.168.103.1
                    192.168.103.233
Node Type . . . . . : Broadcast
NetBIOS Scope ID. . . . . :
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
NetBIOS Resolution Uses DNS : No
    
```

Ethernet adapter E100B1:

```

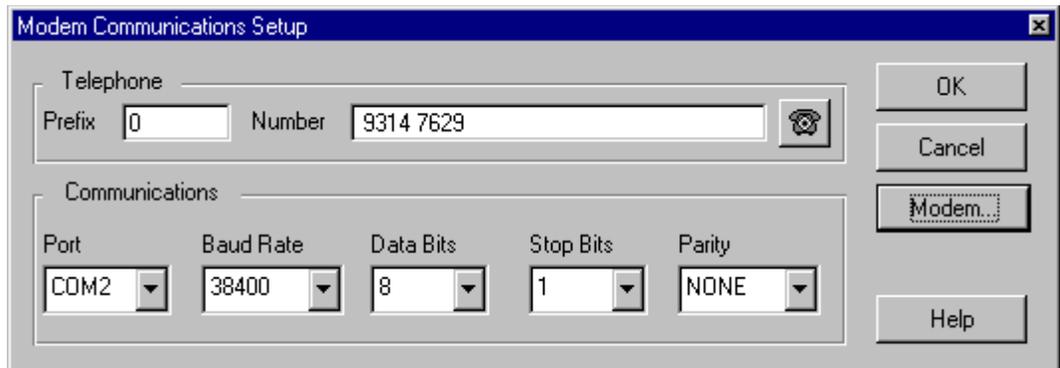
Description . . . . . : Intel EtherExpress PRO PCI Adapter
Physical Address. . . . . : 00-A0-C9-B7-58-9D
DHCP Enabled. . . . . : No
IP Address. . . . . : 192.168.103.118
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.103.1
    
```

Ethernet adapter NdisWan5:

```

Description . . . . . : NdisWan Adapter
Physical Address. . . . . : 00-00-00-00-00-00
DHCP Enabled. . . . . : No
IP Address. . . . . : 0.0.0.0
Subnet Mask . . . . . : 0.0.0.0
Default Gateway . . . . . :
    
```

Modem Communication Setup



The **Modem Communication Setup** dialog allows you to change the communication settings for your serial port.

Parameters are:

Port

The serial port you are using to connect to the host. COM1 through COM9 may be selected.

Baud Rate

The baud rate for transmission. When you set the baud rate, ensure the port on the host computer connected to your PC is set to the same baud rate. 300 through to 115200 may be selected. Consult your system administrator or the appropriate host documentation for more information.

Data Bits

The number of data bits together with the number of stop bits (see Stop Bits) determines how a character is encoded. In addition, the number of data bits is usually set together with a given parity type (see Parity). 4 through to 8 may be selected. Normally you would choose 8 data bits per character. If you are using parity, the link may only support 7 data bits.

Stop Bits

The number of stop bits, together with the number of data bits (see Data Bits) determines how a character is encoded. 1, 1.5 and 2 stop bits may be selected. Normally you would set Stop Bits to 1 (the default), regardless of the number of data bits.

Parity

Parity is one of several methods the communication program uses for detecting transmission errors. Normally you would choose no parity (None). If the communication link can only be made with parity, then select the parity method required: Odd, Even, Mark or Space.

Parity, data bits and stop bits are normally set using either of the following combinations:

- 8 data bits, 1 stop bit, no parity
- 7 data bits, 1 stop bit, even parity

Consult your system administrator or the appropriate host documentation for more information.

Modem Configuration



The Modem button on the **Modem Communications Setup** dialog displays the **Modem Configuration** dialog. This allows you to change the communication settings of the serial port you use to connect to your modem.

The dialog displays the current modem settings assigned to your session. The term (Current Config) prefixes the modem model name on which the current settings are based. This differentiates your settings from the permanent settings in the modem database. The modem database contains the default settings for each modem that SBClient supports and is unaffected by any changes you make in this dialog. For example, if you select the model name **GENERIC HIGH SPEED** and change a setting, SBClient retains the original **GENERIC HIGH SPEED** settings in the Model list and assigns the name (Current Config) **GENERIC HIGH SPEED** to your new settings. This helps you identify which modem you used for the base settings. The original **GENERIC HIGH SPEED** modem settings are unaffected.

If you want to save your changes, you should save your session configuration. If you do not save your changes, they apply only for the duration of your session. Your changes do not affect any other user's modem settings.

Note



You should make changes to these modem settings only if you understand modem operations and are familiar with the specific characteristics of your modem. Consult your modem documentation for an explanation of the settings displayed in this dialog.

Parameters are:

Model

This is the list of modems that SBClient maintains in its modem database. Select the modem that most closely resembles the settings you require from this list. The selected modem appears in the Model text box, prefaced with the term (Current Config).

Speaker Mode

You can turn the modem speaker permanently off (Off), permanently on (On) or on until a connection has been established (Until Connected).

Speaker Volume

You can adjust the modem speaker volume to Low, Medium or High. This parameter may not work with some modems.

Initialization String

The initialization string is a series of commands used to initialize and configure your modem. SBClient sends this command string to the modem before establishing a connection.

Reset String

The command(s) used to reset your modem to its default settings.

Hangup String

The control character and command sequence required to instruct your modem to hang up.

Connect String

The message sent by the modem to your PC indicating a connection has been established.

No Connect Strings

The messages (separated by commas) that your modem sends to your PC, indicating a connection attempt has failed.

Pulse Dial

Uses pulse instead of tone dialing.

Warning

National telecommunications authorities may impose restrictions on the use of pulse dialing.

Detect Busy Tone

Checks for a busy signal from the dialled number.

Detect Dial Tone

Does not attempt to dial a phone number until a dial tone is detected.

Dial Retries

The number of times SBClient redials a number (or set of numbers) before aborting the connection attempt.

Retry Every

The number of seconds SBClient waits before redialing a number.

Wait For Carrier

The number of seconds SBClient waits before deciding a connection has failed.

Warning

National telecommunications authorities impose restrictions on the number of times a phone number may be redialed. Check the regulations for your country before altering the setting of Dial Retries, Retry Every or Wait for Carrier.

Auto Answer

The modem will automatically answer if the phone rings.

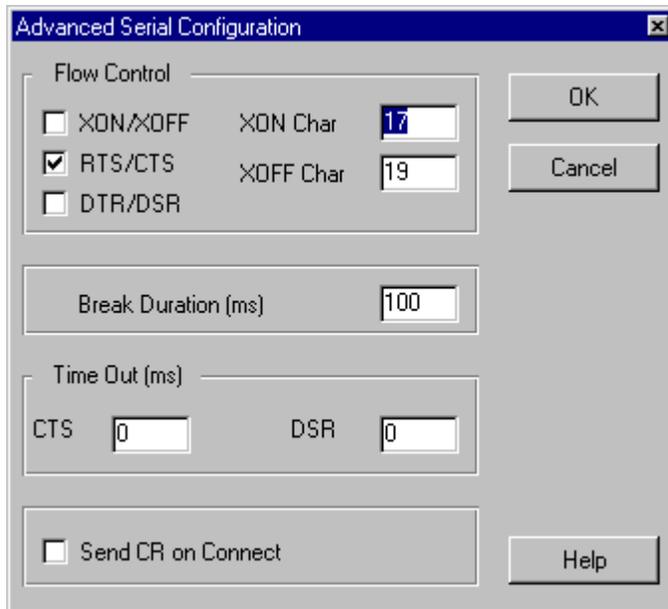
Answer After Rings

The number of rings the modem waits before answering an incoming call.

Drop DTR to Hangup

Choose this option if your modem requires the DTR signal to be lowered to make it hang up.

Advanced Serial Configuration



Parameters are:

Flow Control

SBClient uses flow control to indicate to the host whether it is ready to receive data. Without the correct flow control, data sent to or received from the host can be lost.

For software flow control, select the XON/XOFF checkbox. Ensure the host is set correspondingly on the connected port. You also need to set:

- XON Char. A decimal number specifying an ASCII character that, when sent, indicates to the host or PC that it can resume sending data. The XON character is sent after an XOFF has been previously sent. The default value is 17 (Ctrl-Q).
- XOFF Char. A decimal number specifying an ASCII character that, when sent, indicates to the host or PC that the buffer is full and that it should stop sending data. The default value is 19 (Ctrl-S).

For hardware flow control, select:

- RTS/CTS checkbox (Request To Send/Clear To Send) or
- DTR/DSR checkbox (Data Terminal Ready/Data Set Ready)

You will probably be using RTS/CTS flow control if you are using a modem.

Break Duration (ms)

The duration (in milliseconds) of the break signal that is used by SBClient to send interrupts to the host software. The default value is 100 milliseconds.

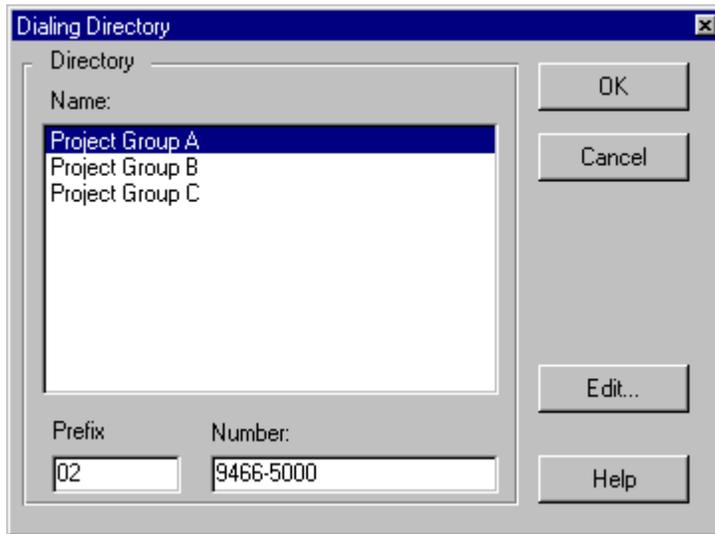
Time Out (ms)

The duration (in milliseconds) of the loss of each of the CTS (Clear To Send) and DSR (Data Set Ready) signals before a timeout occurs. The default value is 0 for both signals.

Send CR on Connect

Sends a CR (Carriage Return) character on connection. After a connection has been established, some host computers will not respond with a login request until the user presses the Enter key to generate a CR character. Select this option to generate the CR automatically on connection.

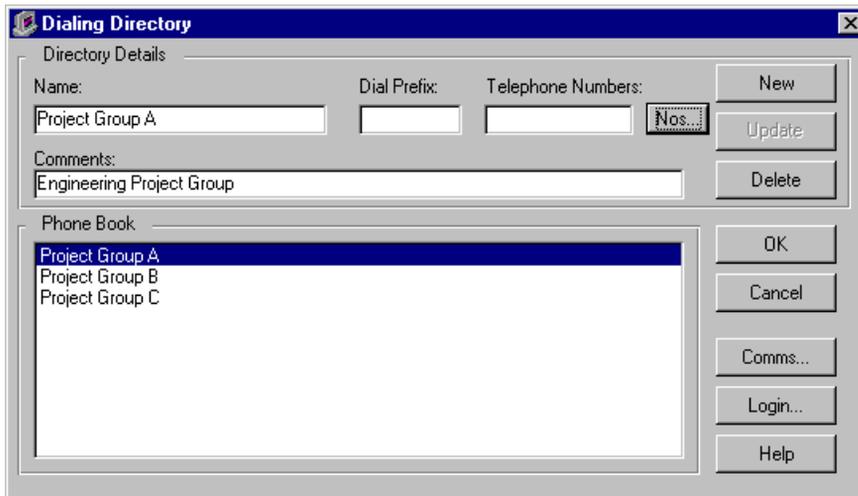
Dialing Directory



The **Dialing Directory** dialog is displayed when you click on the **Modem Communications Setup** dialog, or if you have a modem connection and choose **File > Dial a Number**.

You can select the telephone number you wish to dial from the list of saved numbers. The telephone number associated with the selected name is displayed in the Prefix and Number text boxes.

The **Edit** button displays the **Dialing Directory Maintenance** dialog, allowing you to add or edit modem phone numbers.



To add new numbers to the dialing directory:

1. Click the **New** button. The **Directory Details**' text boxes are cleared.
2. In the **Directory Details** text boxes, enter the new name, prefix, telephone number and any additional comments.
3. Click the **Add** button. The entry is added to the directory.

To change an existing number in the dialing directory:

1. Highlight the name in the **Phone Book** list. The details of the selected entry are displayed in the **Directory Details** text boxes.
2. Change the details.
3. Click the **Update** button to save the changes. The Update button becomes active when you click on any of the **Directory Details** text boxes.

To specify additional numbers in the dialing directory:

Each entry in the **Phone Book** list can have up to ten additional numbers associated with it. These are used in cases where the service provider or bulletin board has more than one access line. If you specify additional numbers, SBClient will attempt to get through to the service provider by trying each number in turn until a connection is made.

1. Highlight the name in the Phone Book list.
2. Click the **Nos** button. This displays the **Additional Telephone Numbers** dialog:

| Primary Number | |
|----------------|------------------|
| Prefix | Telephone Number |
| 02 | 9724-6813 |

| Additional Numbers | | |
|--------------------|--------|------------------|
| | Prefix | Telephone Number |
| 1. | 33 | 9749-9469 |
| 2. | 52 | 3245-9899 |
| 3. | 366 | 6547-2345 |
| 4. | | |
| 5. | | |
| 6. | | |
| 7. | | |
| 8. | | |
| 9. | | |
| 10. | | |

3. Enter any additional numbers and their prefixes.
4. Click the **OK** button.

To change communication or login settings for an entry in the dialing directory:

You can associate specific communication and login configuration settings to a selected entry in the dialing directory. These settings may be different to the settings you have selected using the Communication and Login options from the Setup menu.

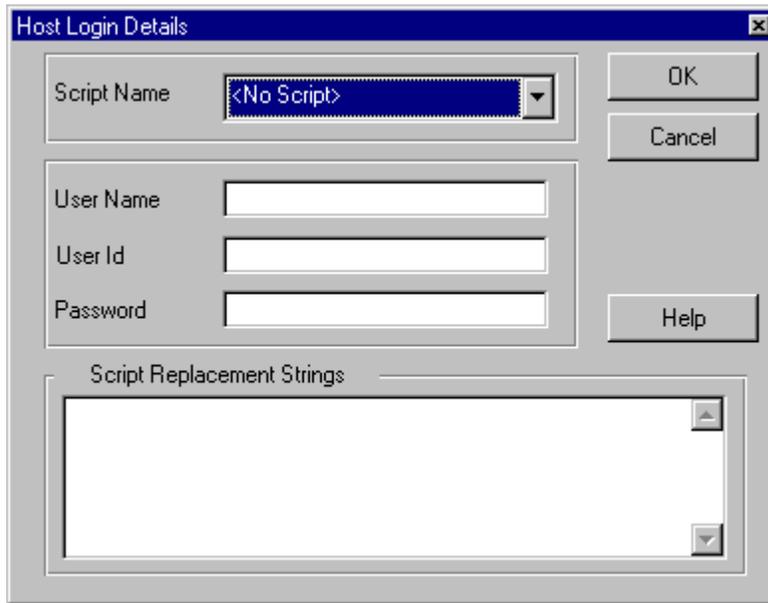
1. Highlight the name in the Phone Book list.
2. To change communication settings, click the **Comms** button to display the **Communication Parameters** dialog.



To change login settings, click the **Login** button to display the Host Login Details dialog.

3. Any changes you make to either of these dialogs apply only to the selected phone book entry. They override the settings defined in the **Communications** or **Script** options in the **Setup** menu for that phone book entry.

Host Login Details



The **Host Login Details** dialog accessed via the **Login** button is used to change the login details for an entry in the dialing directory.

Parameters are:

Script Name

A selection list of existing login scripts.

User Name

The user name that your host is expecting when you log in.

User Id

The user id that your host is expecting when you log in.

Password

The password that your host is expecting when you log in. The characters you type are displayed on your screen as asterisks.

Script Replacement Strings

Statements assigning values to variables used by your login script. For example, if your login script expects values for the variables `faxid` and `sb+servername`, you might enter the following replacement strings:

```
faxid=+6123874784
```

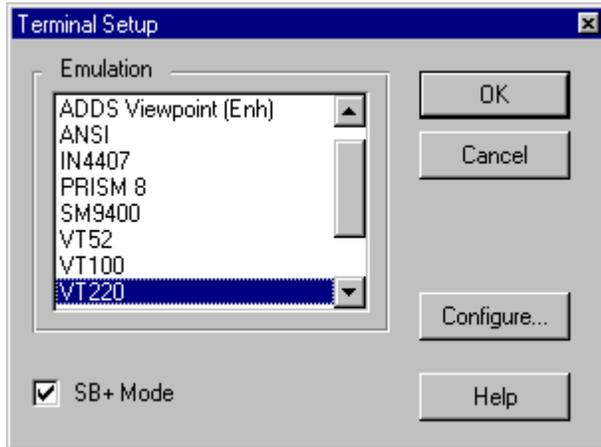
```
sb+servername=sb
```

Troubleshooting Modem Communications

When troubleshooting problems with modem communications you should try the following:

- connect to the modem using a serial session. Use a Hayes command such as `ATDT` to establish if you have a dial tone.
- try dialing the number using Hayes commands e.g. `ATDT 09466 5190`.
- dial the modem number using a normal telephone. This will tell you if the other end is answering.
- try connecting using the Windows Hyperterminal application.
- refer to your modem documentation for further diagnostic tests.

Terminal Setup



The **Terminal** option on the **Setup** menu allows you to configure your session to emulate the characteristics of a particular terminal type. The **Configure** button displays the configuration dialog for the terminal type you have selected.

Parameters are:

Emulation

The type of terminal emulation you wish to use.

Emulations include:

- ADDS ViewPoint (and enhanced ADDS ViewPoint)
- ANSI
- IN4407
- PRISM 8
- SM9400
- VT52, VT100 and VT220
- Wyse 50, 60 and 120

SB+ Mode

Select this if you are using SB+ Server on the host. SB+ mode enables additional escape sequences to be detected, and changes default key mapping assignments for some emulations. This allows you to use the extra features supported by the SB+ Server environment.

Terminal Configuration

The following terminals can be configured using the **Configure** button from the Terminal Setup dialog:

ADDS ViewPoint (and Enhanced ADDS ViewPoint)



Enhanced ADDS ViewPoint is the same as ADD ViewPoint but recognizes some additional escape sequences. If your application requires additions to the standard emulation then use enhanced emulation instead.

The ADDS VP Configuration dialog is displayed when you configure the terminal type as ADDS ViewPoint.

Refer to your ADDS terminal user guide for an explanation of the features available with your ADDS terminal.

Parameters are:

Smooth Scroll

Text is scrolled more smoothly, but more slowly.

Cursor Home at Bottom

Causes the cursor to return to the bottom of the screen rather than the top when you press the Home key.

Auto Scroll

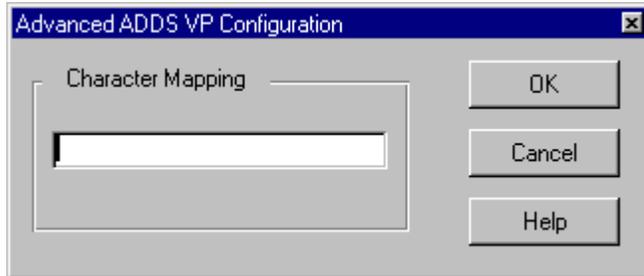
Defines the role of the cursor in scrolling. If selected, data scrolls up one line when the cursor moves past the last line of the page. Otherwise the cursor returns to the top of the same page when the cursor moves past the last line of the page.

Auto Wrap

Turns on automatic line-wrap. This causes lines of text wider than the screen width to automatically wrap to the next line.

Advanced ADDS VP Configuration

This sub dialog is displayed by selecting **Setup > Terminal > ADDS Viewpoint > Configure > Advanced**.

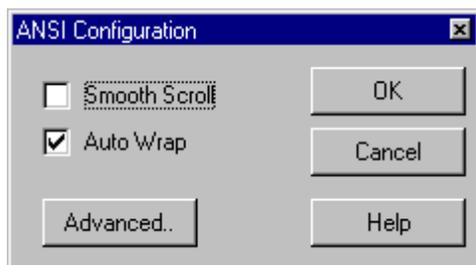


Parameters are:

Character Mapping

Specifies a DOS file pathname used to define character mappings in SBClient. This is useful if you are using multinational characters. You do not have to define these mapping specifications for normal use. These character mapping files are stored in the SBClient installation directory.

ANSI



The ANSI Configuration dialog is displayed when you configure the terminal type as ANSI.

Consult the appropriate ANSI terminal documentation for an explanation of the features available with your ANSI terminal.

Parameters are:

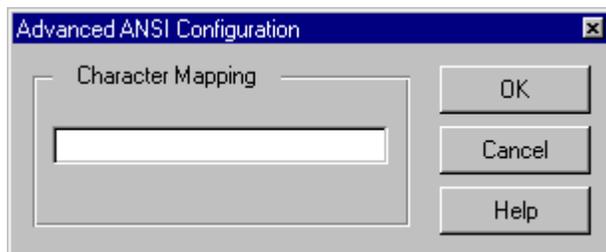
Smooth Scroll

Text is scrolled more smoothly, but more slowly.

Auto Wrap

Turns on automatic line-wrap. This causes lines of text wider than the screen width to automatically wrap to the next line.

Advanced ANSI

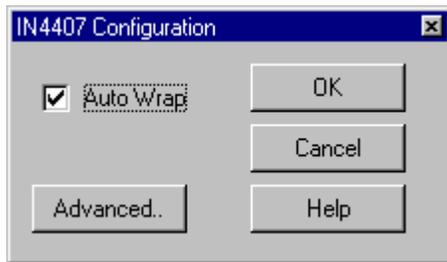


Parameters are:

Character Mapping

Specifies a DOS file pathname used to define character mappings in SBClient. This is useful if you are using multinational characters. You do not have to define these mapping specifications for normal use. These character mapping files are stored in the SBClient installation directory.

IN4407



The IN4407 Configuration dialog is displayed when you select IN4407 as the terminal type.

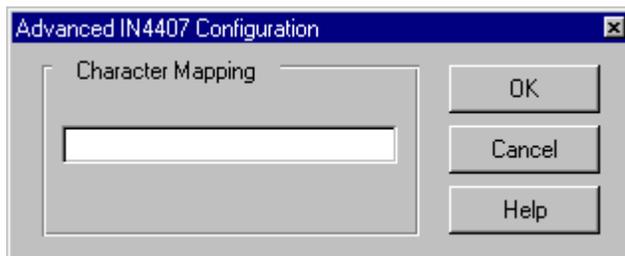
Consult your IN4407 terminal documentation for an explanation of the features available with your IN4407 terminal.

Parameters are:

Auto Wrap

Turns on automatic line-wrap. This causes lines of text wider than the screen width to automatically wrap to the next line.

Advanced IN4407

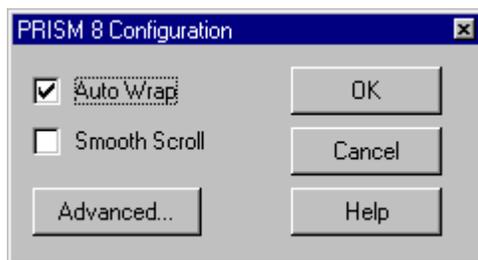


Parameters are:

Character Mapping

Specifies a DOS file pathname used to define character mappings in SBClient. This is useful if you are using multinational characters. You do not have to define these mapping specifications for normal use. These character mapping files are stored in the SBClient installation directory.

PRISM 8



The PRISM 8 Configuration dialog is displayed when you select PRISM 8 as the terminal type.

Consult your PRISM 8 terminal documentation for an explanation of the features available with your PRISM 8 terminal.

Parameters are:

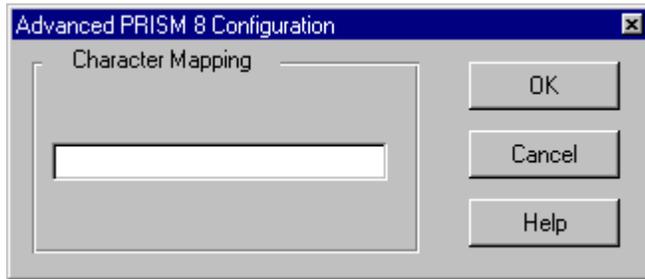
Auto Wrap

Turns on automatic line-wrap. This causes lines of text wider than the screen width to automatically wrap to the next line.

Smooth Scroll

Text is scrolled more smoothly, but more slowly.

Advanced PRISM 8

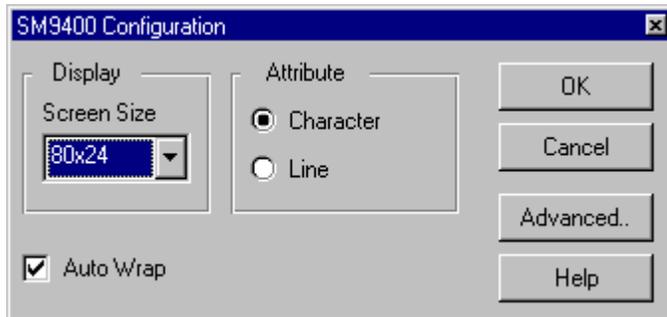


Parameters are:

Character Mapping

Specifies a DOS file pathname used to define character mappings in SBClient. This is useful if you are using multinational characters. You do not have to define these mapping specifications for normal use. These character mapping files are stored in the SBClient installation directory.

SM9400



The SM9400 Configuration dialog is displayed when you select SM9400 as the terminal type.

Consult the appropriate SM9400 terminal documentation for an explanation of the features available with your SM9400 terminal.

Parameters are:

Screen Size

Defines the dimensions of the character display in columns by rows. Options are:

- 80x24. For normal display.
- 160x24. For condensed display.

Attribute

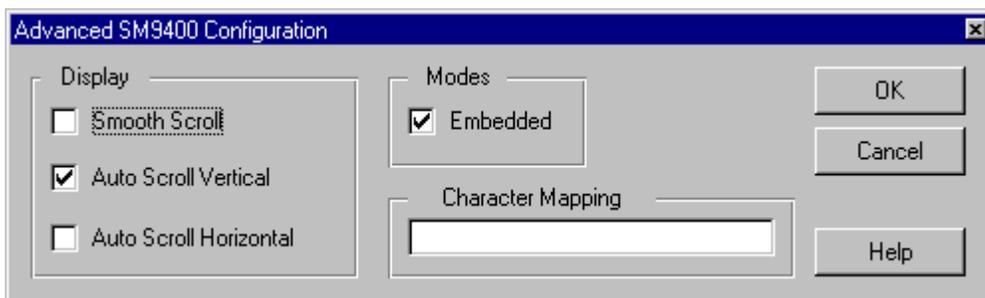
Options are:

- **Character.** A video attribute applies to all subsequent characters until another video attribute is received.
- **Line.** A video attribute applies to the end of the current line.

Auto Wrap

Turns on automatic line-wrap. This causes lines of text wider than the screen width to automatically wrap to the next line.

Advanced SM9400



The **Advanced SM9400 Configuration** dialog is displayed when you click the Advanced button on the SM9400 Configuration dialog. For normal operation, you will not need to configure the advanced options.

Parameters are:

Smooth Scroll

Text is scrolled more smoothly, but more slowly.

Auto Scroll Vertical

Defines the role of the cursor in scrolling down the screen. If selected, data scrolls up one line when the cursor moves past the last line of the page. If not selected, the cursor returns to the top of the same page when the cursor moves past the last line of the page.

Auto Scroll Horizontal

Defines the role of the cursor in scrolling across the screen. If selected, data scrolls left one character when the cursor moves past the right edge of the screen. If not selected, the cursor returns to the beginning of the next line when the cursor moves past the right edge of the screen.

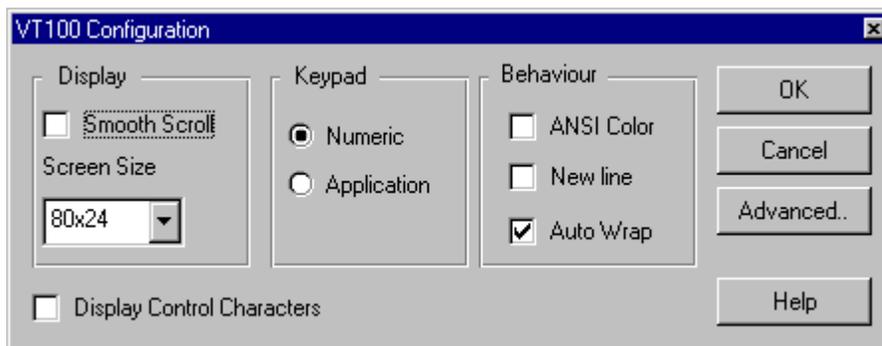
Embedded

Video attributes will occupy extra character positions on the screen. Normally, video attributes are embedded.

Character Mapping

Specifies a DOS file pathname used to define character mappings in SBClient. This is useful if you are using multinational characters. You do not have to define these mapping specifications for normal use. These character mapping files are stored in the SBClient installation directory.

VT52 and VT100



The **VT52 (or VT100) Configuration** dialog is displayed when you select VT52 (or VT100) as the terminal type.

Consult the appropriate terminal documentation for an explanation of the features available with your VT terminal.

Parameters are:

Smooth Scroll

Text is scrolled more smoothly, but more slowly.

Screen Size

Defines the dimensions of the character display in columns by rows. Options are:

- 80x24. For normal display.
- 132x24. For condensed display.

Keypad

Defines the use of the keypad. Options are:

- **Numeric.** For normal numeric keypad operations. The keypad sends ASCII character codes corresponding to the numeric characters on the keys.
- **Application.** For keypad operations for specific applications. The numeric keypad transmits special codes recognized by some host application programs.

ANSI Color

Enables the recognition of ANSI-standard color escape sequences. If the ANSI Color checkbox is set it overrides colors set using **Setup > Colors**.

New Line

Causes the terminal to translate CR (Carriage Return) to CRLF (Carriage Return/Line Feed).

Auto Wrap

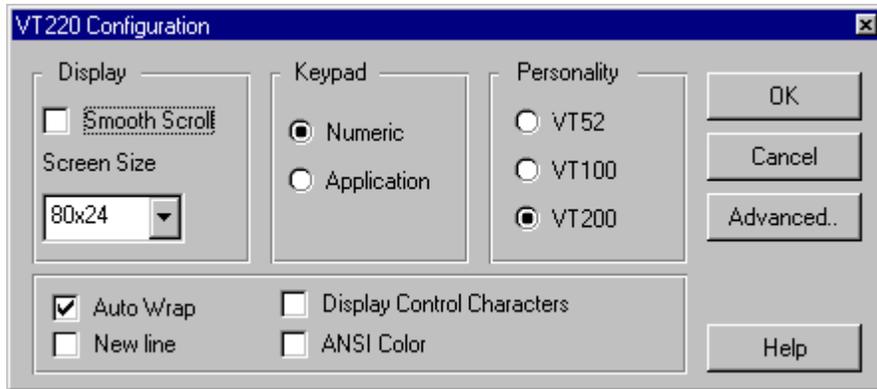
Toggles on automatic line-wrap. This causes lines of text wider than the screen width to automatically wrap to the next line.

Display Control Characters

Allows control characters to be passed. Check this parameter if you are using a non-US English keyboard.

True VT emulations do not allow control characters to be passed, which would cause problems for UK keyboards which require a control character to be passed when a user enters a pound sign.

VT220



The **VT220 Configuration** dialog is displayed when you select VT220 as the terminal type.

Note



A terminal definition called TU.VT220 is provided with SB+ Server Release 3 for use on the host (specifically for use with SBClient running VT220 emulation in SB+ mode).

Consult the appropriate VT220 terminal documentation for an explanation of the features available with your VT220 terminal.

Parameters are:

Smooth Scroll

Text is scrolled more smoothly, but more slowly.

Screen Size

Defines the dimensions of the character display in columns by rows. Options are:

- 80x24. For normal display.
- 132x24. For condensed display.

Keypad

Defines the use of the keypad. Options are:

- **Numeric.** For normal numeric keypad operations. The keypad sends ASCII character codes corresponding to the numeric characters on the keys.
- **Application.** For keypad operations for specific applications. The numeric keypad transmits special codes recognized by some host application programs.

Personality

Defines the terminal personality the PC should exhibit. This is useful where host applications rely upon the features of a specific VT terminal. VT52, VT100 or VT200 may be selected.

Auto Wrap

Turns on automatic line-wrap. This causes lines of text wider than the screen width to automatically wrap to the next line.

New Line

Causes the terminal to translate CR (Carriage Return) to CRLF (Carriage Return/Line Feed).

Display Control Characters

Allows control characters to be passed. Check this parameter if you are using a non-US English keyboard.

True VT220 emulations do not allow control characters to be passed, which would cause problems for UK keyboards which require a control character to be passed when a user enters a pound sign.

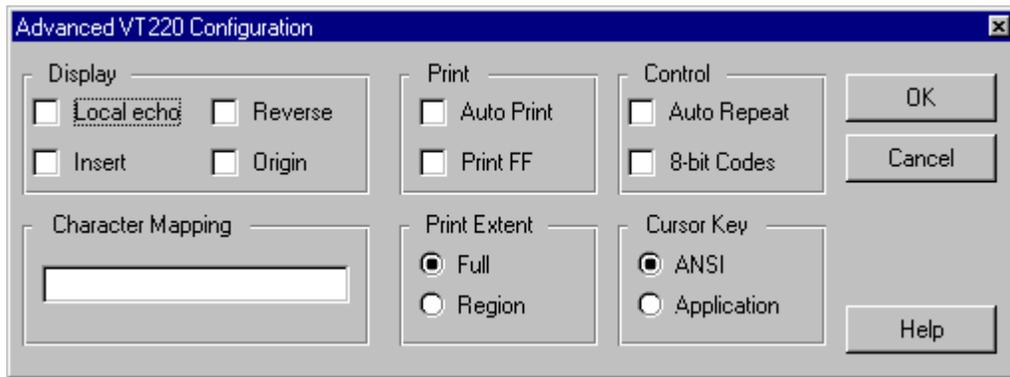
ANSI Color

Enables the recognition of ANSI-standard color escape sequences.

Note

If the ANSI Color checkbox is set it overrides colors set using **Setup > Color**.

Advanced VT220, VT52, VT100



The **Advanced VT220 (or VT 52 or VT100) Configuration** dialog is displayed when you click the **Advanced** button on the **VT220 (or VT52 or VT100) Configuration** dialog. For normal operations, you will not need to configure the advanced options.

Parameters are:

Local Echo

Directs data from the keyboard to the monitor as well as to the host computer. If you do not select this option, the terminal directs data from the keyboard to the host computer only, which may or may not send data back to the monitor.

Insert

Defines character insertion. If selected, the character at the cursor position and any characters on the same line to the right of the cursor move right for each character entered. If not selected, each character entered replaces the existing character at the cursor position.

Reverse

Displays dark text on a light background. On a color monitor, it switches foreground and background colors.

Origin

Causes the cursor to address the active scroll region rather than the top left-hand corner of the session window.

Auto Print

Sends the current display line to the printer when the terminal receives a line feed, form feed, vertical-tab code or during auto-wrap.

Print FF

Requires the terminal to send a form feed character at the end of a printing job.

Auto Repeat

Ensures the character corresponding to the key being pressed continues to be generated until the key is released.

8-bit Codes

Refers to the transmission of control characters from the terminal to the host. Consult the relevant terminal documentation for more information.

Character Mapping

Specifies a DOS file pathname used to define character mappings in SBClient. This is useful if you are using multinational characters. You do not have to define these mapping specifications for normal use. These character mapping files are stored in the SBClient installation directory.

Print Extent

Defines the region of the screen to be printed during a Print Screen operation. Options are:

- **Full.** The entire screen is printed.
- **Region.** Only the scroll region is printed.

Cursor Key

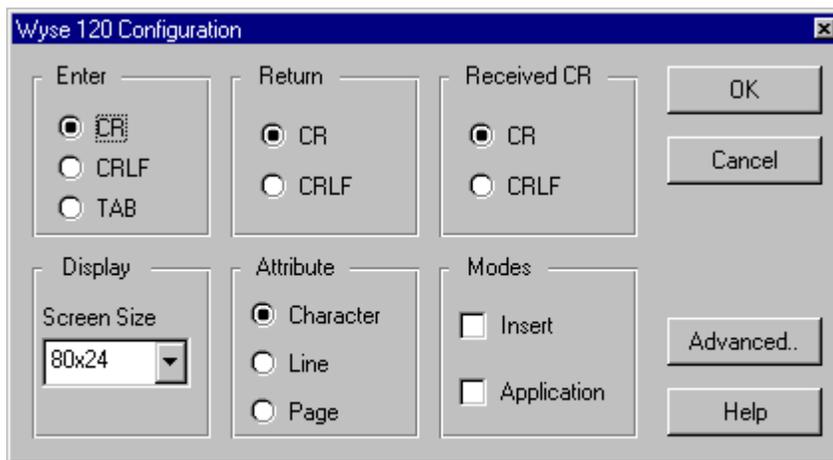
Defines cursor key transmission. Options are:

- **ANSI.** Transmits ANSI cursor control sequences when the keys on the numeric keyboard are used for cursor movement.
- **Application.** For cursor operations during specific applications. The numeric keypad transmits special codes recognized by some host application programs.

Warning 

Some applications send escape sequences to the terminal to change the cursor key from ANSI to Application. This can sometimes cause problems. SBClient can be instructed to ignore these sequences (see TU.APPLICATION.MODE in the “Host Library Reference” section of the online help supplied with SBClient).

Wyse 50, 60 and 120



The **Wyse 50 (or 60 or 120) Configuration** dialog is displayed when you select Wyse 50 (or Wyse 60 or Wyse 120) as the terminal type.

Note 

A terminal definition called TU.WYSE60 is provided with SB+ Server Release 3 for use on the host (specifically for use with SBClient running Wyse 50, 60 or 120 emulation in SB+ mode).

Consult the appropriate Wyse terminal documentation for an explanation of the features available with your Wyse terminal.

Parameters are:

Enter

The ASCII character to be sent when the Enter key is pressed. Options are:

- CR. Carriage Return; the cursor returns to the beginning of the current line.
- CRLF. Carriage Return and Line Feed; the cursor goes to the beginning of the next line.
- TAB. The cursor goes to the next horizontal tab position, if any. (Wyse 60 only).

Return

The ASCII character to be sent when the Return key is pressed. Options are:

- CR. Carriage Return; the cursor returns to the beginning of the current line.
- CRLF. Carriage Return and Line Feed; the cursor goes to the beginning of the next line.
- TAB. The cursor goes to the next horizontal tab position, if any.

Received CR

Determines how the terminal interprets an incoming CR. Options are:

- CR. Carriage Return; the cursor returns to the beginning of the current line.
- CRLF. Carriage Return and Line Feed; the cursor goes to the beginning of the next line.

Screen Size

The display screen size, in columns by data lines. The sizes supported are:

- 80 x 24
- 132 x 24
- 80 x 25
- 132 x 25
- 80 x 42
- 132 x 42
- 80 x 43

- 132 x 43

Attribute

Determines how the display attributes are assigned to the terminal. Options are:

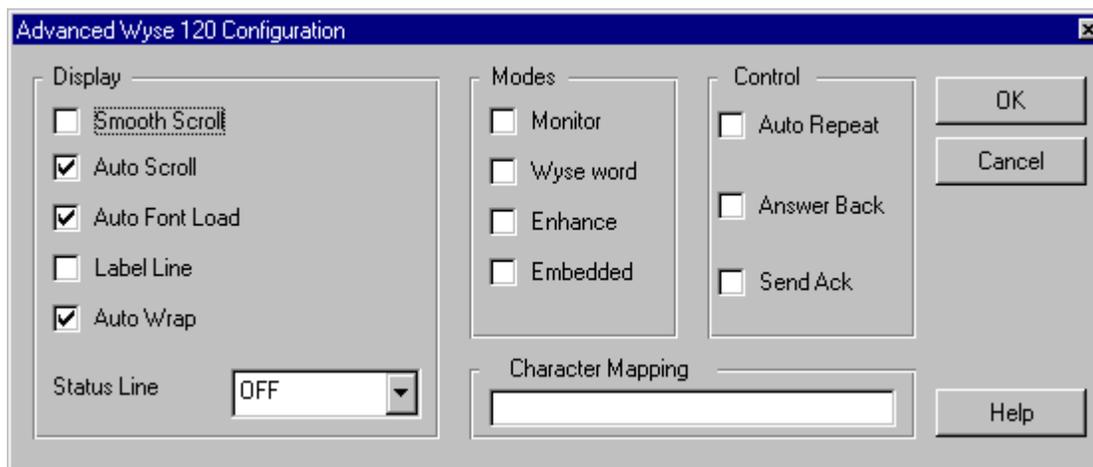
- **Character.** The attributes are assigned as each character is entered.
- **Line.** The attributes are active to the end of the line.
- **Page.** The attributes are active to the end of the page.

Insert

Defines character insertion. If selected, the character at the cursor position and any characters on the same line to the right of the cursor move right for each character entered. If not selected, each character entered replaces the existing character at the cursor position.

Certain keys on the Wyse 60 ASCII keyboard send 8-bit codes that override the existing codes, including key redefinitions and Wyse word codes. If selected, an application program incorporating these unique codes can readily interpret what key has been pressed, regardless of user reprogramming.

Advanced Wyse 50, 60 and 120



The **Advanced Wyse 120 (or 50 or 60)** dialog is displayed when you click the **Advanced** button on the **Wyse 120 (or Wyse 50 or Wyse 60) Configuration** dialog. For normal operation, you will not need to configure the advanced options. Consult the relevant host documentation for more information.

Parameters are:

Smooth Scroll

Text is scrolled more smoothly, but more slowly.

Auto Scroll

Defines the role of the cursor in scrolling down the screen. If selected, data scrolls up one line when the cursor moves past the last line of the page. If not selected, the cursor returns to the top of the same page when the cursor moves past the last line of the page.

Auto Font Load

Determines whether the characters in the range 128 to 254 are interpreted as multinational graphics characters.

Auto Wrap

Turns on automatic line-wrap. This causes lines of text wider than the screen width to automatically wrap to the next line.

Label Line

Displays a label line at the bottom of the screen. The Label Line checkbox appears only on the Advanced Wyse 60 & Advanced Wyse 120 dialogs.

Status Line

Defines the status line of the display. Options are:

- **Off.** The screen displays no status line.
- **Standard.** The screen displays a status line that can be modified by the host program.
- **Extended.** The screen displays a status line with additional features required by some applications.

Monitor

The terminal interprets escape sequences and control codes as normal. If not selected, the terminal displays symbols for escape sequences and control codes without interpreting them.

Wyse word

Terminal keys send WordStar compatible codes. These codes take precedence over any other key codes except application defined key codes. If not selected, terminal keys send standard key codes.

Enhance

Ensures the terminal supports additional features in some non-Wyse type personalities. Consult the appropriate Wyse documentation.

Embedded

Determines whether video attributes occupy character positions on the screen. Selecting this parameter causes the Wyse 60 or Wyse 120 to behave like a Wyse 50 for most applications. Normally, video attributes are hidden (not embedded); that is, they do not take any extra space on the screen. The Embedded checkbox appears only on the Advanced Wyse 60 & Advanced Wyse 120 dialogs.

Auto Repeat

Ensures the character corresponding to the key being pressed continues to be generated until the key is released.

Answer Back

If the terminal receives an ENQ character (Ctrl-E) it will return an answer-back message. The answer-back message is a programmed string of characters. Consult the appropriate Wyse 60 and Wyse 120 documentation.

Send Ack

If the terminal receives an ENQ character (Ctrl-E) it will return an ASCII ACK character if the Answer Back checkbox is not selected.

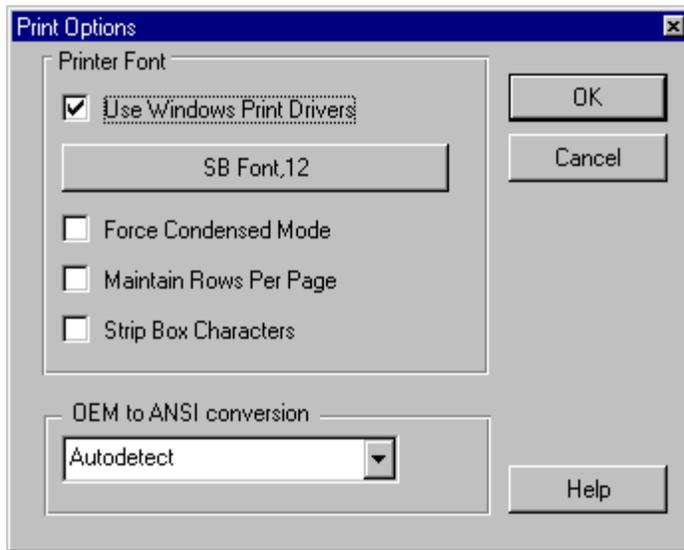
Character Mapping

Specifies a DOS file pathname used to define character mappings in SBClient. This is useful if you are using multinational characters. You do not have to define these mapping specifications for normal use. These character mapping files are stored in the SBClient installation directory.

Printing

The File menu provides the following options for printing the contents of a session window:

- **Print Screen** — this menu has three suboptions:
- **Text** — prints the text contents of the session area in the session window
- **GUI Windows** — prints a screen capture of window in focus
- **GUI Screen** — prints a screen capture of the entire Windows Desktop
- **Print Setup** — this provides standard print options.
- **Print Options** — this provides additional print options:



Parameters are:

Use Windows Printer Drivers

Sends auxiliary output from the host to the appropriate Windows printer driver before passing it to the printer, rather than sending it directly to the printer. Set this option, in conjunction with the Font Selection button set to SB Font, to print graphic box characters.

If this option is not set (pass through mode), the output from the host is sent directly to the printer, allowing escape sequences embedded in the host output to be passed to the printer (no box graphics are printed — these are replaced with “+” and “|” characters).

Note



Not all printer drivers support pass through mode. If you are using pass through mode and experience problems printing then try configuring a generic text printer or alternatively a raw print driver and choose this driver when printing in pass through mode. If you still have difficulty then try another driver within the same family of printer drivers. For example, if one HP LaserJet doesn't work then try a different HP LaserJet printer driver. Pass through mode doesn't need the printer driver to interpret and translate commands for a specific printer model but rather simply requires a driver to faithfully pass the sequences through without alteration. It is therefore quite safe to use a different driver to the printer you are using.

SB Font, 12

This is the font selection button. Press this button to choose the default font that SBClient prints with. You can only use this button when the **Use Windows Print Drivers** option is checked.

It is possible to print using any fixed pitch font that is installed on your computer. The default font is the SB font (TTF). This font allows for full graphic printing. SB font is an OEM font which contains the correct entries for box graphics. Some printers may not be able to handle the SBClient printer font in which case this option must be turned off and graphic characters cannot be printed.

Force Condensed mode (132 Columns)

Prints in 132 column mode. This option can only be set in conjunction with the Use Windows Printer Drivers option.

Maintain Rows Per Page

When you print in 132 column mode, SBClient reduces the size of the font's width and height in order to fit 132 columns on a page. Due to the reduced font height, more rows can fit on a page. To ensure that the number of rows displayed in 132 column mode is the same number that is displayed in 80 column mode, select this option.

Strip Box Characters

Do not print any box characters at all.

OEM to ANSI Conversion

This option allows SBClient to print from an OEM character set on the host to an ANSI character set on the printer. You can specify that SBClient:

- always converts OEM to ANSI character sets
- never converts OEM to ANSI character sets
- automatically detects whether to convert an OEM character set to an ANSI character set, based on the host's character set and the font selected with the **Font Selection** button. See the **Use Windows Print Drivers** option above for more information on the **Font Selection** button.

File Transfer Setup

The screenshot shows the 'File Transfer Setup' dialog box. It is titled 'File Transfer Setup' and has a close button in the top right corner. The dialog is organized into several sections:

- Buffer Value:** 'Input Buffer Size' is set to 'Auto' in a dropdown menu.
- Transfer Type:** Three radio buttons are present: 'FTP', 'Network', and 'SBZ'. The 'SBZ' option is selected.
- FTP Options:** Two text input fields for 'User Name' and 'Password'.
- Network Copy Options:** A text input field for 'Share Name' and a browse button (three dots).
- SBZ Options:**
 - 'Acknowledge All Packets' checkbox is unchecked.
 - 'Timeout' section: 'Auto' checkbox is checked, 'Seconds' checkbox is unchecked.
 - 'Line Protocol' section: '7 Bits' radio button is unselected, '8 Bits' radio button is selected.
- ASCII Options:** Two checkboxes, 'Character Delay' and 'Line Delay', both are unchecked.
- Progress Bar:** Two browse buttons for 'Foreground' and 'Background'. Below them is a progress indicator showing '948/1976'.

On the right side of the dialog, there are three buttons: 'OK', 'Cancel', and 'Help'.

The **File Transfer** option on the Setup menu allows you to configure some overriding flow control options to be used during file transfer operations.

Parameters are:

Input Buffer Size

On some platforms, when data is sent to the host and the data is larger than the input buffer, the system may "hang". In most cases SBClient knows the correct size of the input buffer and will break the data into small enough blocks, however there are some cases where SBClient only finds out the size after the data transfer has already failed. SBClient always tries to resend the data but on some platforms it is too late. If you experience problems when uploading the host code or doing DDE transfers, set the value of the input buffer to an actual value rather than the default of Auto. If the system hangs again, try a smaller value. Keep trying smaller values until the system works correctly.

Transfer Type

Allows you to choose the transfer type. Options are:

- FTP — SBClient uses for file transfers (only available on uniVerse, UniData and D3).
- Network Copy — SBClient uses a Windows Network Copy protocol. This transfer type provides the best performance on Windows NT Servers running UniData or uniVerse. It is only available on these platforms.
- SBZ (SB ZModem) — SBClient uses the ZModem protocol (default for all platforms).

FTP Options

FTP (File Transfer Protocol) is a set of programs for transferring files between systems on the Internet. FTP provides a robust and faster file transfer interface to SBClient. Ardent recommends that you use this facility on UNIX hosts if possible.

The FTP interface in SBClient is supported on the following platforms:

- Advance Pick/D3 on UNIX
- uniVerse on UNIX and Windows NT
- UniData on UNIX and Windows NT

When you use FTP you must have a correct user setup in the **Host Login** dialog. You must also have adequate permissions to gain access to where FTP data is stored. On UNIX this is in the /tmp directory. On Windows NT this is wherever the TEMP environment variable is pointing. If this environment variable is not set then you cannot gain access to the appropriate directory and FTP will fail.

Warning 

TEMP and TMP must be system variables, not user variables.

You must have a valid user name and password to enable FTP transfers.

- **User Name.** The user name that your host is expecting when you log in.
- **Password.** The password that your host is expecting when you log in. The password you enter is displayed on your screen as asterisk (*) characters.

Warning 

FTP services must have a valid password. It cannot be null.

Network Copy Options

This is the preferred file transport protocol to use in the NT environment and provides the best performance for transferring files on NT based systems running UniData or uniVerse.

Share Name. This is the UNC path which SBClient uses to exchange data with SB+ Server. By default, SBClient resolves the:

- Windows NT Server *machinename* (the related Unix term for this is host name).
- Temporary environment variables (TEMP and TMP).

When using the Network Copy transport protocol between an NT Server and an NT Client (the NT client can be either an NT Server or NT Workstation PC) running SBClient, then the NT Server PC will send the full UNC path to the NT client. The UNC path will be that of a file which is to be transferred across the network between the two machines. The UNC path will take the following syntax:

`\\machinename\pathname\filename`

While a Windows NT Server and Client combination will do this automatically, in the case of Windows95/98 Client PCs it will be necessary to identify the pathname for SBClient. The following steps explain how to do this.

To determine the pathname from a Windows95 PC running SBClient:

1. From SBClient, select **Setup > File Transfer**.
2. Click on the **Network** option button in order to choose the Network Copy file transfer protocol
3. Either enter the path directly in the Share Name textbox or click on the ... browse button and navigate to the location of the file to be copied. This will, in effect, specify the pathname portion of the UNC path.

When you use Network Copy, you must have adequate permissions to gain access to where the data is stored. On Windows NT this is wherever the TEMP environment variable is pointing. If this environment variable is not set, you cannot gain access to the appropriate directory and Network Copy will fail.

Note



TEMP and TMP must be system variables, not user variables.

To set system environment variables in Windows NT (version 4):

1. Choose **Start > Settings > Control Panel**
2. Double-click on the System icon
3. Select the **Environment** tab
4. Ensure that the TEMP and TMP variables are defined in the System Variables listbox, NOT the User Variables listbox, e.g. C:\TEMP.

Other than the points raised above, the configuration details are the same for SBClient Client PCs running either Windows NT or Windows95.

Warning



When your machine is logged into Domain A and you are trying to access files on an NT server in Domain B it will be necessary to map a network drive to the UNC path and reconnect it at login.

SBZ Options

- **Acknowledge All Packets.** Ensures that the next packet is not sent until the previous packet is acknowledged. This is not selected by default. This is used for unreliable networks and will slow down data transfer but will be more robust.

- **Timeout.** Enter the timeout value (the number of seconds after which the transfer is aborted if no packets are transferred) in the Seconds prompt. SBClient can automatically timeout when you check the Auto checkbox.
- **Line Protocol.** 8 Bits is selected by default. Older networks may only support 7 Bits.

ASCII Options

- **Character Delay.** The number of milliseconds between successive character transfers.
- **Line Delay.** The number of milliseconds between successive line transfers.

Warning

Use of these options will severely impact performance of various SBClient features. They should be used with extreme caution.

Progress Bar

Foreground. Sets the foreground color of the transfer status line that displays when SBClient is transferring a file.

Background. Sets the background color of the transfer status line.

Troubleshooting File Transfer

FTP

If you are having problems with FTP then you should try the following:

1. Does your platform support FTP? If not, you cannot use the FTP option.
2. Ensure FTP is installed and enabled on the server.
3. Ensure that the user may use ftp.
4. Perform a test transfer using the command prompt e.g.:

`ftp hostname`

5. Ensure that the correct login details are entered in the SBClient File Transfer Setup dialog. SBClient needs to be able to log into the FTP server using these details. FTP daemons do not allow null passwords.

Note 

If your FTP service is part of Microsoft Internet Service 2.0. As Administrator on Windows NT, you must create an alias called /TEMP in the configuration of the FTP service in Microsoft Internet Service (MIS). This alias must point to the same directory that the system environment variables TEMP and TMP point to. The MIS FTP service must have write permissions to this directory.

SBZ

If you are having problems with SBZ then you should try the following:

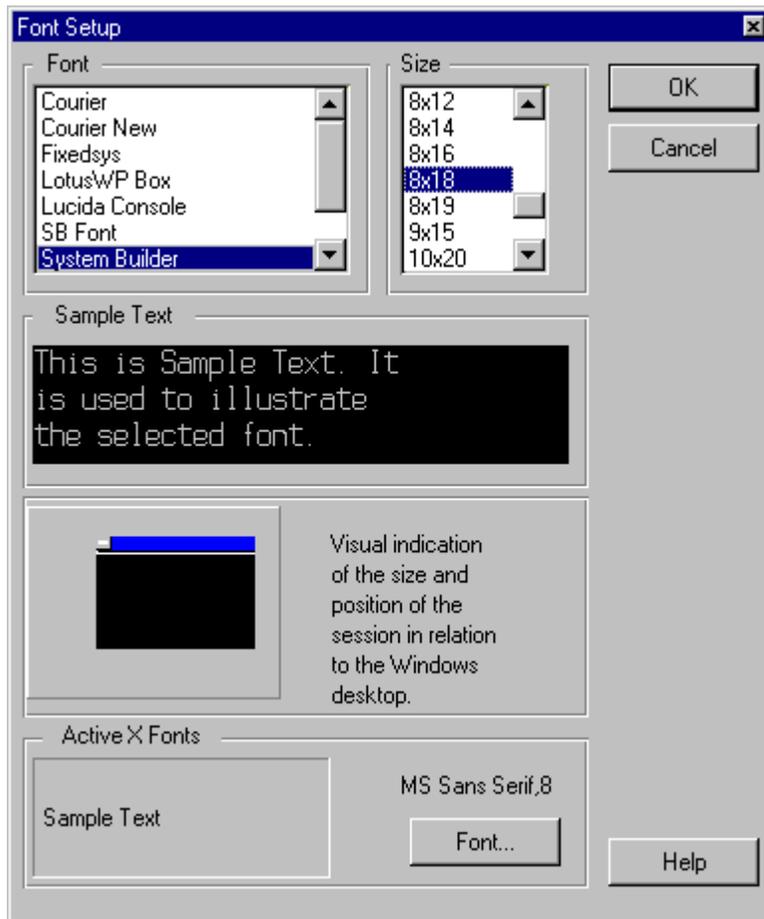
1. Decrease the buffer size.
2. Turn acknowledge packets on.
3. Increase the timeout period (especially for large files).
4. Change the line protocol from 8 bit to 7 bit.

Network Copy

If you are having problems with Network Copy then you should try the following:

1. Check that the Windows NT server can be seen on the network
2. Check that you can access shares on the Windows NT server e.g. \\machinename\c\$\temp
3. Map a network drive to the UNC location of where SBClient/SB+ will be exchanging data.

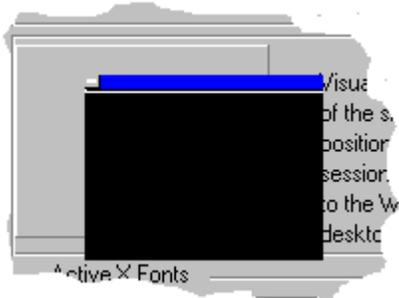
Font Setup



The **Font** option on the **Setup** menu allows you to select the font and size of characters to be displayed on your screen during a session.

An example of the current selection is displayed in the Sample Text box. Below this is a representation of the window that would be required to fully display the font/size combination for the selection. This is shown on a raised area that represents your screen. You should choose a font/size combination that fits within this area.

If the window representation exceeds the screen boundaries:



your actual session window will be larger than can fit on your PC screen.

Parameters are:

Font

A selection list of fixed-pitch fonts.

Size

A selection list of font sizes (in pixel width by height) that are valid for the selected font.

ActiveX Fonts

Allows a user to choose the font to use for ActiveXs in SBClient. This setting controls, for example, the Smart Query pass through to GUI font. This means that international character sets can be selected for the passthrough to GUI.

Troubleshooting Fonts

If you are experiencing problems with displaying SBClient fonts (i.e. SB Font and System Builder font) then try the following:

1. Reboot you computer.
2. Try accessing the fonts from another application e.g. NotePad.
3. Go to the Windows\Fonts directory and check whether SB Font and System Builder font are listed. If they are listed, delete them. Copy all the files from the SBoffice\SBClient\SBopen\Fonts directory to the Windows\Fonts directory.

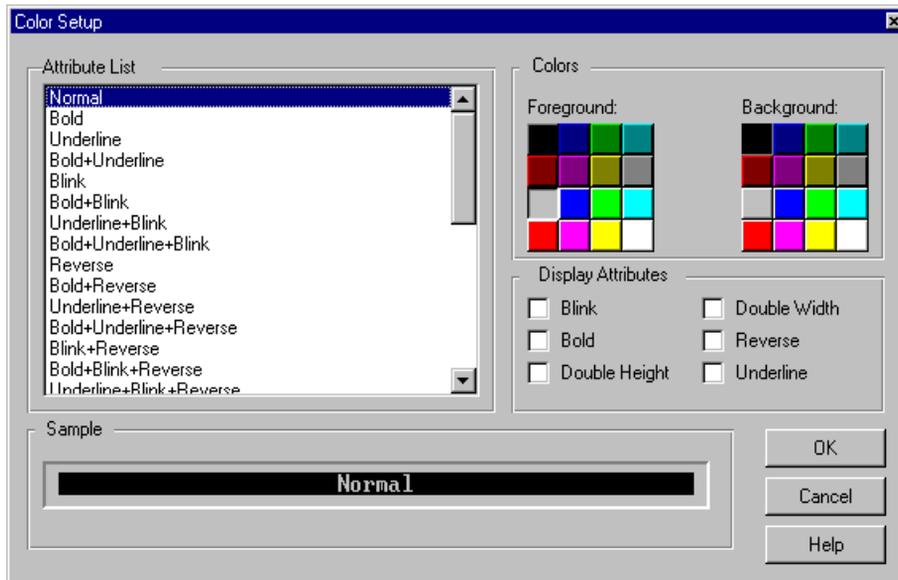
4. Reboot the computer and run SBClient.

Note



The SBClient fonts dialog will only show fixed pitch fonts. If you wish to create your own font and have SBClient use it then your font must be fixed pitch.

Color Setup



The **Color** option on the **Setup** menu allows you to modify the video attributes for your PC emulation.

Parameters are:

Attribute List

A selection list of video attribute combinations.

When you select a combination, SBClient automatically selects the attributes and colors defined for that combination, and displays a sample of text in the Sample box. You may select alternative colors and attributes for a combination, which are saved when you click the OK button.

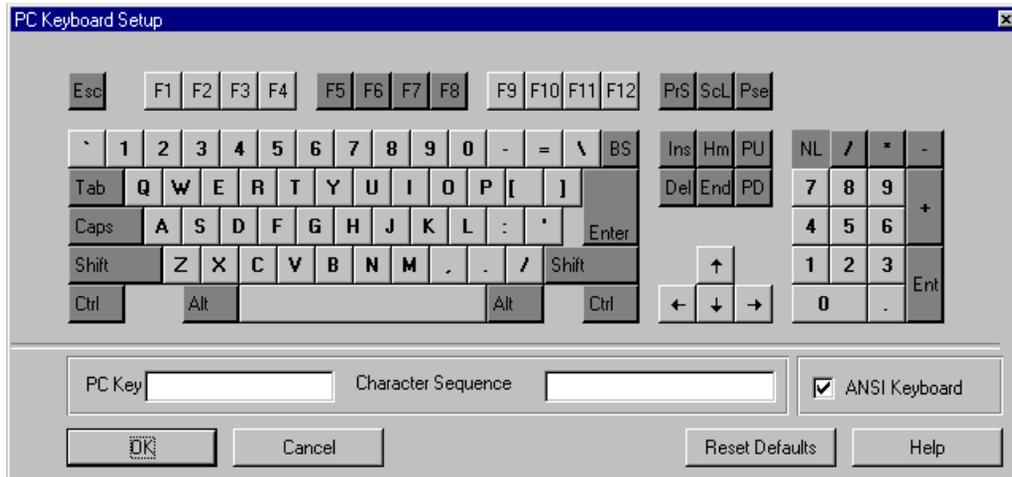
Colors

The foreground and background colors for a combination.

Display Attributes

The video attributes for a combination (Blink, Bold, Double Height, Double Width, Reverse and Underline).

Keyboard Setup



The **Keyboard** option on the **Setup** menu allows you to reprogram any key on the PC keyboard to send a different sequence to the host. This allows you to emulate many different types of keyboard.

To reprogram a key on your keyboard:

1. Click the key you wish to reprogram.

Clicking the Caps button causes uppercase letters to be used when letter keys are clicked. Clicking the Shift button, the Ctrl button or the Alt button corresponds to holding down those keys when clicking other keys.

The selected key is shown in the PC Key box. If the key is already mapped, the character sequence corresponding to the key is shown in the Character Sequence box. The default mappings depend on the terminal being emulated.

Unless you want to remap the PC keyboard to suit a particular application, you should not need to change the default settings. You cannot remap the menu hot keys: Alt + C, E, F, H, S, T, U and v.

2. In the Character Sequence box, enter the character sequence to be produced by the selected key.

Enter printable characters (other than the left-parentheses) as is. Enter non-printable characters, and the left-parentheses, as decimal numbers in parentheses. Spaces are ignored.

For example, the ANSI sequence for Up Arrow is:

(27)[A

The character sequence for the left-parentheses is (40).

3. Repeat steps 1 and 2 for subsequent keys, or the OK button to accept your changes and exit the dialog.

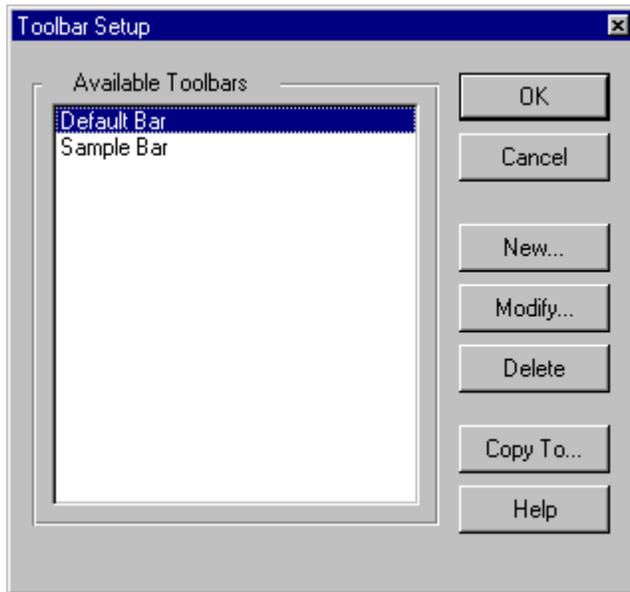
Press the **Reset Defaults** button if you wish to restore all default key mappings for the current terminal emulation.

Ensure that ANSI Keyboard is checked if your keyboard generates ANSI characters (English language keyboards generate ANSI characters).

Warning

Some emulations and UNIX programs will reload the keyboard character mapping. If you have problems with key mappings, it is suggested that you disable the sequences in the emulation or do not run these programs (see UNIX man pages on “tic”).

Toolbar Setup



The **Toolbar** option on the **Setup** menu allows you to create buttons for the toolbar located at the top of the session window. This allows you to set up shortcuts for commonly used operations in that session. You can set up a number of toolbars and select the one that best suits the operations you are performing during a particular session. SBClient comes with a default toolbar.

To select a different toolbar for your session:

1. Highlight the toolbar name in the Available Toolbars list.
2. Click the OK button.

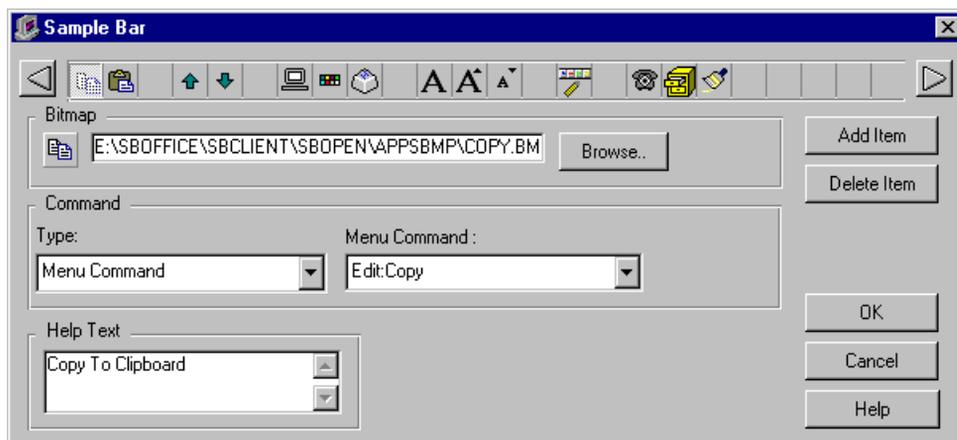
Modifying a Toolbar

You can rearrange, delete, and add to the buttons on your toolbar.

To modify an existing toolbar:

1. Highlight the toolbar name in the Available Toolbars list.

- Click the Modify button. The toolbar maintenance dialog is displayed; its title being the name of the selected toolbar:



The selected toolbar is displayed at the top of the dialog. You can use the arrow buttons to scroll left or right along your toolbar. A blank button is displayed as a gap on the toolbar in your session window.

To rearrange the buttons on your toolbar:

- Drag-and-drop the desired button to its new position on the toolbar.

To delete a button from your toolbar:

- Click on the desired button.
- Click the Delete Item button. The button is removed from the toolbar. All buttons to the right of the deleted button are shifted one place to the left.

To add a blank button to your toolbar:

- Click on the button on the toolbar before which you want to insert the blank button.
- Click the Add Item button. A new (blank) button is added to the toolbar. The button you clicked on, plus all buttons to the right of it are shifted one place to the right.

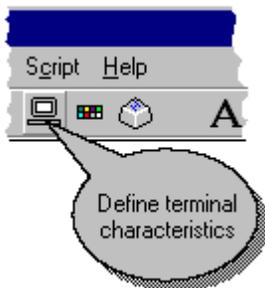
To create a new button on your toolbar:

- Add a blank button to the appropriate position on your toolbar (described above).

2. Select this blank button.
3. In the Bitmap text box, enter the full path of the bitmap file you want to display on the button. You can use the Browse button to find and select the appropriate bitmap. An image of the bitmap will be shown to the left of the path.
4. Select the type of command you want associated with the button from the Type list. Options are:
 - Windows/DOS Command
 - Menu Command
 - Host Command
 - Help File
 - SBDesktop Command
 - Object
 - Macro
5. In the adjoining box (whose label will depend upon the type of command you have chosen), enter the command or filename to be associated with the button.

You should include the full DOS path for a Windows/DOS command or a help filename. You can use the Browse button to find and select the appropriate command or filename.

6. In the Help Text box, enter the text you want displayed in the balloon help for the button. Balloon help is displayed when the user clicks the right mouse on a button:



It is also the text used for the ToolTip.

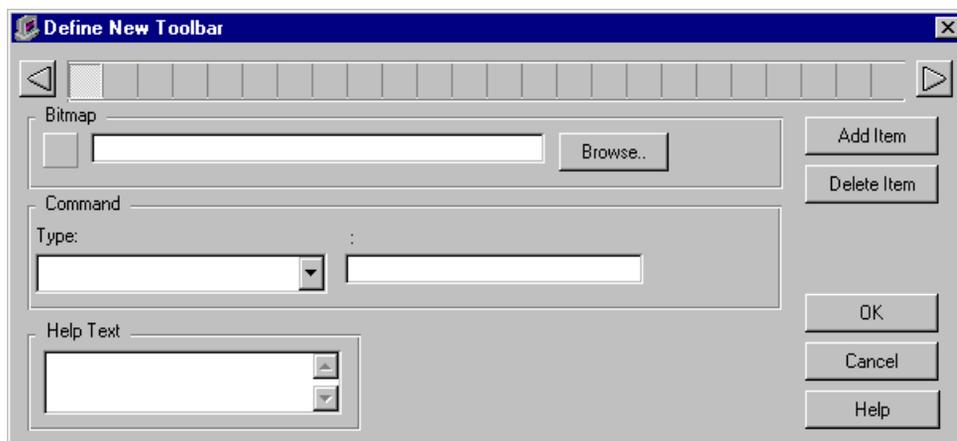
- Click the **OK** button to save the changes and return to the toolbar maintenance dialog.

Creating a New Toolbar

You can create a new toolbar from scratch, or copy an existing toolbar and modify it. You create a new toolbar using the **New** button in the **Toolbar Setup** dialog.

To create a new toolbar from scratch:

- Click the **New** button. SBClient displays the toolbar maintenance dialog with all boxes cleared:

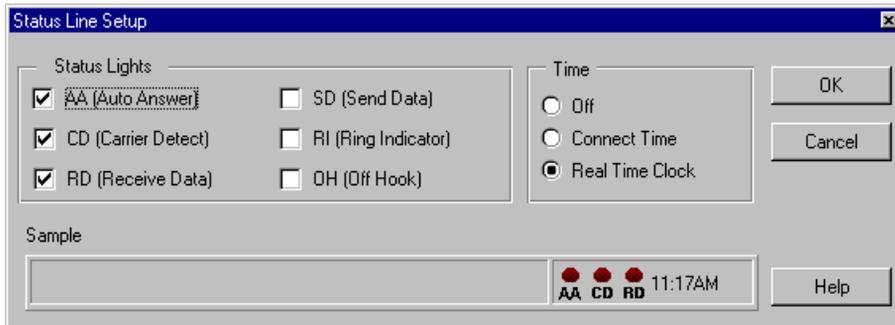


- Create each button.
- When you have finished creating your new toolbar, click the **OK** button. SBClient displays a dialog asking you to enter a name for your new toolbar.
- Enter the new toolbar name and click the **OK** button.

To create a new toolbar from a copy of an existing one:

- Select the toolbar from the Toolbar Setup dialog that matches most closely the set of buttons you want.
- Click the **Copy To** button. SBClient displays a dialog asking you to enter a new toolbar name.
- Enter the new toolbar name and click the **OK** button. SBClient saves a copy of the toolbar with the new name. You can modify the newly named toolbar as required.

Status Line Setup



The **Status Line** option on the **Setup** menu allows you to select the information to be displayed on the status line located at the bottom of your session window for the following:

- the serial or network status lights you want SBClient to emulate on the status line
- the modem status lights you want SBClient to emulate on the status line
- the time display

Parameters are:

Status Lights

Status lights can be on, off or flashing.

For a serial or network connection:

- **CD (Carrier Detect)** indicates a valid connection.
- **RD (Receive Data)** and **SD (Send Data)** indicate incoming and outgoing network traffic for the active session.

For a modem connection:

- **AA (Auto Answer)** indicates that the modem will automatically answer an incoming call.
- **RI (Ring Indicator)** indicates that the modem has detected a ring tone on the phone line.

- **OH (Off Hook)** indicates that the modem is off hook and dialing a number, or otherwise actively connected.

Reducing the number of status lights displayed on the status line may improve terminal emulation performance.

Time

Options are:

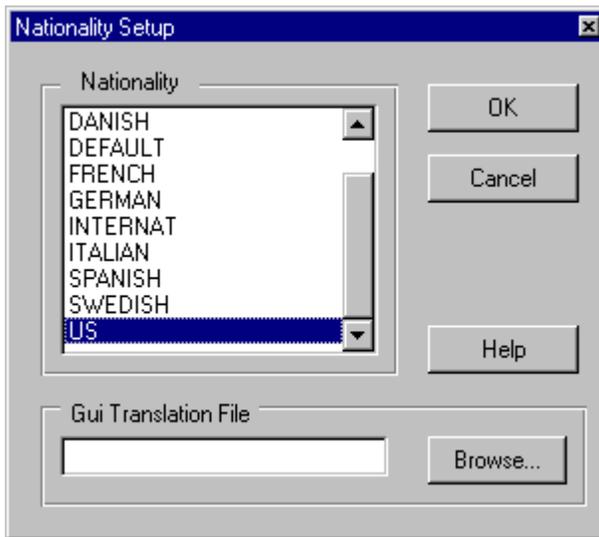
- **Off.** SBClient does not display any time on the status line.
- **Connect Time.** When a successful connection has been established, SBClient displays the elapsed time for the session in hours, minutes and seconds.
- **Real Time Clock.** When a successful connection has been established, SBClient displays the current time.

During a session, you can click on the time displayed on the status line to switch between elapsed and current time.

Sample

The Sample box shows the status line as it will appear on the terminal screen.

Nationality Setup



Note

When using SBClient with non-English character sets, you should follow these steps to ensure correct nationality operation:

In your Windows operating system, ensure that it is setup correctly for your country. For example, in Windows 95 check your Regional Settings in the control panel. See your Windows operating system manuals for more information.

If your keyboard generates ANSI characters (English language keyboards generate ANSI characters) you should check the ANSI Keyboard parameter in **Setup > Keyboard** screen. By default this parameter is checked.

Checking this button causes SBClient to do an ANSI to OEM conversion on keys pressed before they have been sent to the host.

If you are using the VT220 or VT100/VT52 emulation and a non-US, English, keyboard, you should check the Display Control Characters parameter in the Configure screen for the appropriate

terminal definition from **Setup > Terminal**. By default this parameter is unchecked.

VT220 emulations do not allow control parameters to be passed. This causes problems for non-US, English, keyboards, for example, the UK keyboard passes a control character when the pound sign key is pressed.

Under normal circumstances there should be no need to make a selection from the Nationality parameter list.

Examples where you need to make a **Nationality** selection are:

- If you don't have a UK keyboard and have a US keyboard instead, and you want to enter a pound sign by pressing shift 3, you would select BRITISH.
- If you need to do conversions through a conversion table (.cvt file). For example, if the version of Cyrillic on your host uses a different code page to the Windows operating system's code page, you can configure a conversion table (.cvt file) to map the characters between the different code pages. You do this in the input and output sections of the .cvt file.

For more information on conversation tables (.cvt files) see [“Appendix A - Internationalization Issues”](#) and [“Appendix E - Character Mapping and Translation Tables”](#).

Nationality

A selection list of the conversation tables (.cvt files) available with SBClient. If you have the correct keyboard for your country and the correct Windows operating system keyboard driver installed, there is no need to make a selection from the Nationality parameter list.



When programming transfers from SBClient to Microsoft Word or Excel for languages that use the high bit characters of the OEM character set, for example accented characters in Spanish, you must convert these characters from the OEM to the ANSI character set. This is because most hosts use an OEM character set and the Windows operating system uses an ANSI character set. To do this conversion, specify the -I option when using the relevant TU.TO.WORD and TU.TO.EXCEL routines.

GUI Translation File

This is used to assign a translation file to SBClient to dynamically translate characters in GUI mode. The concept is similar but more functional than the CVT files used in character mode. In the **GUI translation File** textbox, enter or select a file.

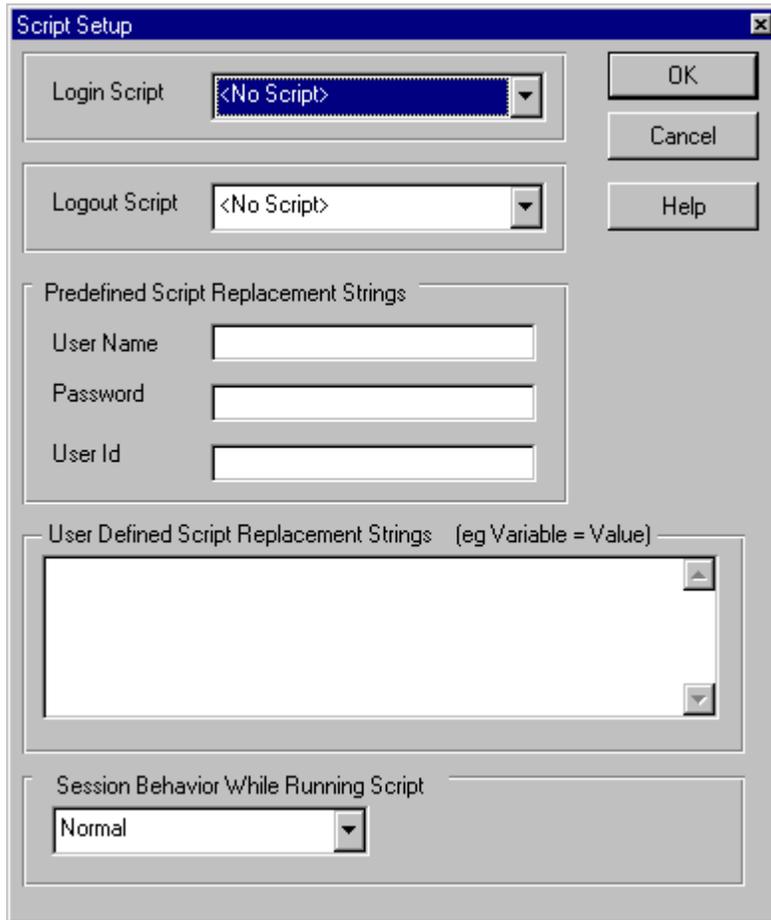
Tip

It is preferable to locate your file in the `..\SBOoffice\SBClient\TU` directory and then not specify the path in the **GUI translation File** textbox. This will ensure that there is no dependency on a hard coded path.

Note

For more information see [“Appendix A - Internationalization Issues”](#) and [“Appendix E - Character Mapping and Translation Tables”](#).

Script Setup



The **Script** option on the **Setup** menu allows you to include login script information required for logging into particular applications.

Parameters are:

Login Script

A selection list of existing login scripts. See [“Chapter 6 - Creating Scripts”](#) for more information.

Logout Script

A selection list of existing logout scripts. See “[Chapter 6 - Creating Scripts](#)” for more information.

User Name

The user name that your host is expecting when you log in.

Password

The password that your host is expecting when you log in. The characters you type are displayed on your screen as asterisks.

User Id

The user id that your host is expecting when you log in.



Tip About the **User Name**, **Password**, and **User Id** parameters: If you are recording a script and you enter your user name and password, these details are embedded in the script. If you share your script with other users, these users will use your login name and password when they run the script. If you enter your user name and password in the Script Setup dialog, SBClient does not embed these values in the script. Instead, SBClient embeds the variables <username> and <password>. These variables are substituted with the values entered at the User Name and Password prompts when the script is run, thus allowing you to share scripts with other users.

User Defined Script Replacement Strings

Statements assigning values to variables used by your login script. For example, if your login script expects values for the variables faxid and sb+servername, you might enter the following replacement strings:

```
faxid=+6123874784
```

```
sb+servername=sb
```

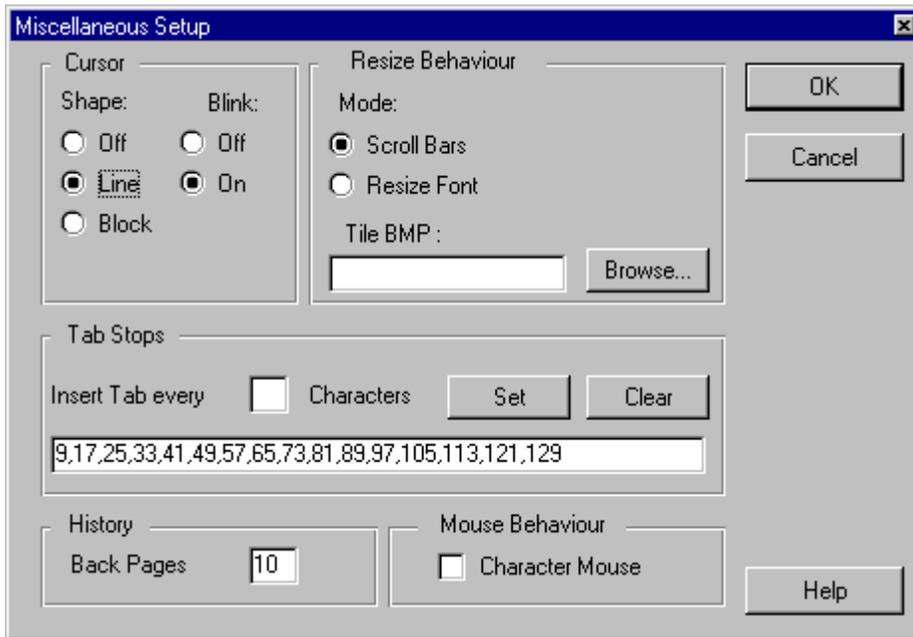
Session Behavior While Running Script

Allows you to specify the session behavior while SBClient runs the scripts. The options are:

- **Normal.** The session window is displayed as normal

- **Minimized Until Done.** The session window is minimized until the script is complete, then the session window is displayed as normal
- **Minimizes Forever.** The session window remains minimized after the script is complete. As the session is displayed in the Windows 95 or Windows NT 4.0 taskbar, you can restore the session.
- **Hidden.** The session window is hidden until the script is complete, then the session window is displayed as normal. This means that the session is not displayed in the Windows 95 or Windows NT 4.0 taskbar while the script is running. A hidden session can only be restored from a System Tray icon menu.
- **No Menu.** If this option is selected and the configuration saved, then, when the session is reinvoked, no menus will be displayed. This is equivalent to the -X option, however this will work in .SBC files as well as normal classical session configurations.

Miscellaneous Setup



The Miscellaneous option on the Setup menu allows you to define additional settings for your session.

Parameters are:

Shape

The shape of the cursor; that is, Off (no cursor), Line or Block.

Blink

Determines whether the cursor will be blinking (On) or not (Off).

Mode

Determines what happens when the session window is resized. Options are:

- **Scroll Bars.** Scroll bars are added to the session window if it is resized smaller than the session window. If you resize the session window so that it is larger than the session window, the bitmap specified in the Tile BMP box is used to fill the extra space around the session window.
- **Resize Font.** The font is rescaled to fit the resized session window. There are a limited number of sizes for each font supplied; therefore, the closest size that fits the resized window is selected. SBClient does not allow the window to be resized greater than the width of the screen.

Tile BMP

The pathname of a bitmap to be used to fill the extra space around the session window. This is relevant only if Scroll Bars mode is selected, and the session window is resized so that it is larger than the session window. Click the Browse button to select a background bitmap.

Insert Tab Every Characters

Indicates the fixed interval to be used for setting tab stops.

When you click the **Set** button, tab stops are set across the screen every *n* columns, where *n* is the number entered for this box. For example, if you enter 8 and click the Set button, tab stops are placed every 8 columns (that is: 1, 9, 17 and so forth).

If you want to set tab stops at intervals that are not fixed, enter the appropriate column numbers (separated by commas) in the text box. For example:

1,12,25,40

Click the **Clear** button to clear the tab settings.

These settings do not apply where the host itself processes the tab characters on output, so that SBClient does not receive them.

Back Pages

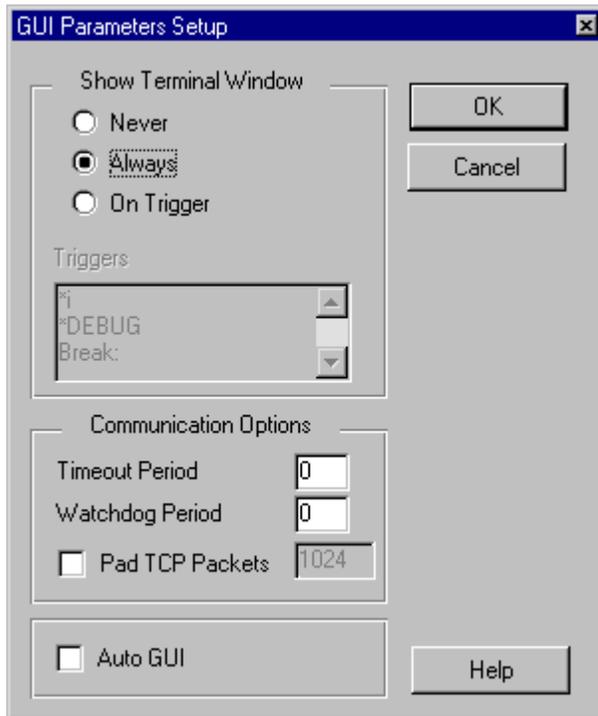
The number of pages to be stored in memory for backpaging. You can scroll backward and forward between session pages using and from the session window toolbar.

Between 0 and 100 pages can be remembered. The more page remembered, the more memory that is used. The default is 10.

Character Mouse

When your session is emulating a character mode terminal, your mouse is normally disabled while you are within the session window. Choose this option if you want SBClient to send mouse escape sequences to your host application program when you perform mouse operations during the session.

GUI Parameters Setup



The GUI Parameters option on the Setup menu allows you to specify parameters required for using GUI.

Parameters are:

Show Terminal Window

In GUI mode, extraneous characters received from the host application not catered for in the GUI environment are trapped by the terminal window. Normally you would not need to know about this activity, but there may be situations when you do (eg to respond to error message from the host). The display of the terminal window is defined here. Options are:

- **Never.** Never show the terminal window.
- **Always.** Always show the terminal window. (The default).

- **Trigger.** Only show the terminal window if it receives one of the triggers specified in the Triggers list.

Triggers

User defined character sequences that when sent from the host application cause the terminal window to be displayed. The defaults, each on a separate line, are: *I, *DEBUG and *BREAK.

Timeout Period

In GUI mode, SBClient can periodically send a sequence to the host to check that it is still 'alive'. If the host fails to respond to the sequence within a certain time, SBClient assumes that the host has died.

Timeout Period is the time to wait after sending a sequence to the host before assuming that it is 'dead'. The default timeout period is 0 seconds.

Setting the Timeout Period and Watchdog Period to 0 (the default) switches off the watchdog timer. If you encounter problems in GUI mode over a slow link, try increasing these values or switching them off if on.

Watchdog Period

The Watchdog Period is the frequency that SBClient send a sequence to the host to check the status of the host. The default value is 0 seconds. See the Timeout Period parameter above.

Pad TCP Packets

It is possible to pad TCP packets. When this option is enabled SBClient is able to work around internal limitations with Microsoft's TCP/IP stack and improve the performance of GUI screens by a factor of 3 to 5 times. You can turn this option on from the **Setup > GUI Parameters** dialog.

Note



On uniVerse Windows NT based installations it is recommended that a value of 1500 be used.

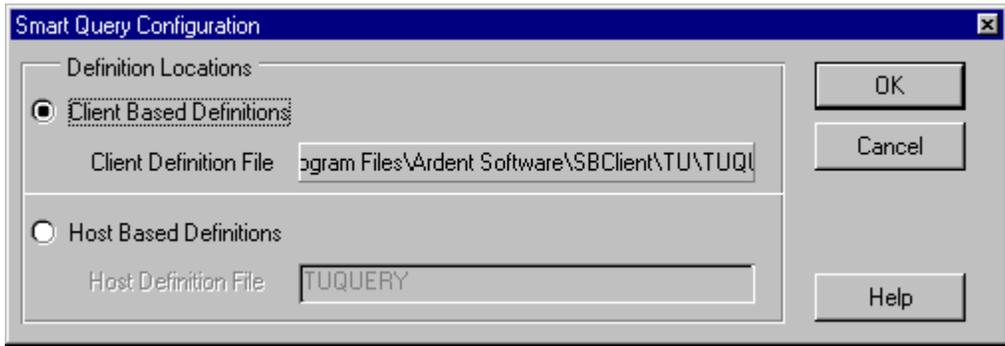
Warning 

This option should only be used where network performance is good. It should not be used for serial or modem communications. The optimum padding size varies between network implementations and it may be necessary to experiment with this value to achieve optimum performance.

Auto GUI

You need to select this checkbox to use AutoGUI. When checked, SBClient will search the character application for trigger strings required for loading forms. If you are not using AutoGUI then deselect this checkbox.

Smart Query Definitions



Smart Query definitions can be stored on either the client or the host. You can configure Smart Query to use client or host definitions through the Smart Query Config screen located on the Setup menu. By default, Smart Query stores definitions on the client.

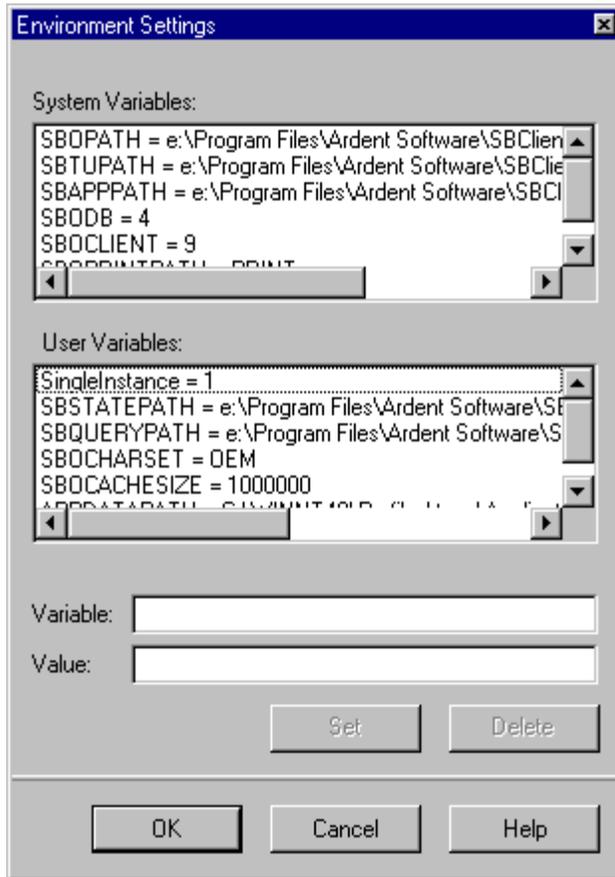
Client Based Definitions

To store the definitions on the client, select the Client Based Definitions option. This will store the definitions in the file specified by "Client definition file". This file is a Btrieve file and should not normally be changed. If you need to change this file or its location, you can do so by changing the SBQUERYPATH environment variable – see Setup Environment..

Host Based Definitions

To store the definitions on the host, select the Host Based definitions option. This will store the definitions on the host in the file specified by Host Definition File. The file must a valid host file. The default file, TUQUERY, is created automatically when installing the host library. It is recommended that this file be used. Unlike the client configuration, the host configuration is session independent; that is, each session may use a different host file to store definitions in.

Environment Variables



The Environment Settings dialog is available from within SBClient by selecting **Setup > Environment Variables**.

The **Environment Settings** dialog allows you to:

- view the SBClient system variables (you cannot edit these values)
- view and edit the SBClient user variables

To edit an SBClient user variable:

1. In the **User Variables** list box, select the variable to edit.

The variable name is displayed in the **Variable** prompt, and the corresponding value is displayed in the **Value** prompt.

2. Edit the value displayed in the **Value** prompt.
3. Click **Set**.

To add a new SBClient user variable:

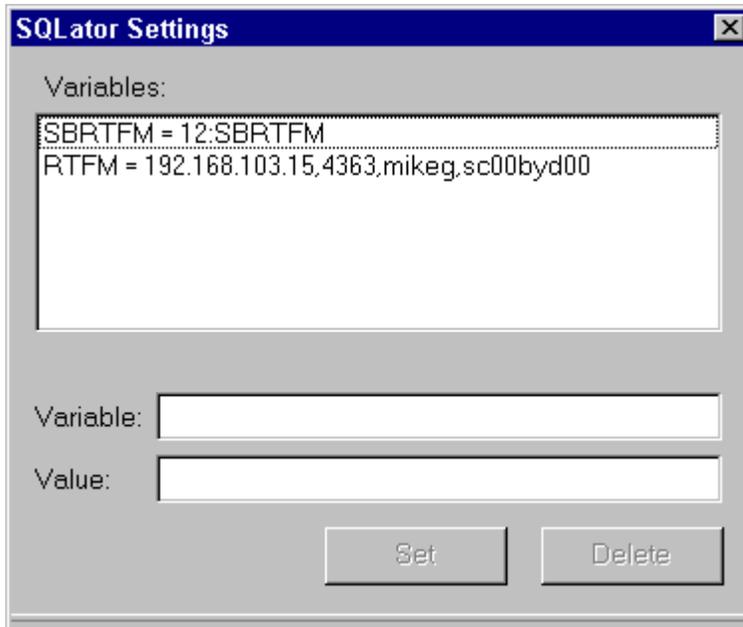
1. Enter the user variable name in the **Variable** prompt.
2. Enter the corresponding value in the **Value** prompt.
3. Click **Set**.

To delete an SBClient user variable:

1. In the **User Variables** list box, select the variable to delete.
2. Click **Delete**.

SQLator Settings

This allows you to view and set SQLator variables. This option is only present if you have Visual Atlas installed on your computer.



It is necessary to create two entries for each oracle server - one to link the Oracle TNS (Transparent Network Substrate) server name to an SQLator Oracle server name (the "12" signifies an Oracle connection) and the other to identify to Objectcall the host (ip address, port, user id and password) on which Oracle resides.

Saving and Sharing Session Configurations

SBClient can be installed in one of two ways:

- Each user does a full install of SBClient on the user's own PC.
- On a network, SBClient can be deployed as an administrator/user combination. In this setup, the administrator account is installed on a designated PC and a user account is created on each user's PC. Each user account contains a minimal amount of local files and relies on the administrator account for the rest (see "Creating a Multiuser SBClient System" in the "Getting Started" guide).

In this case SBClient can store configuration settings in two places:

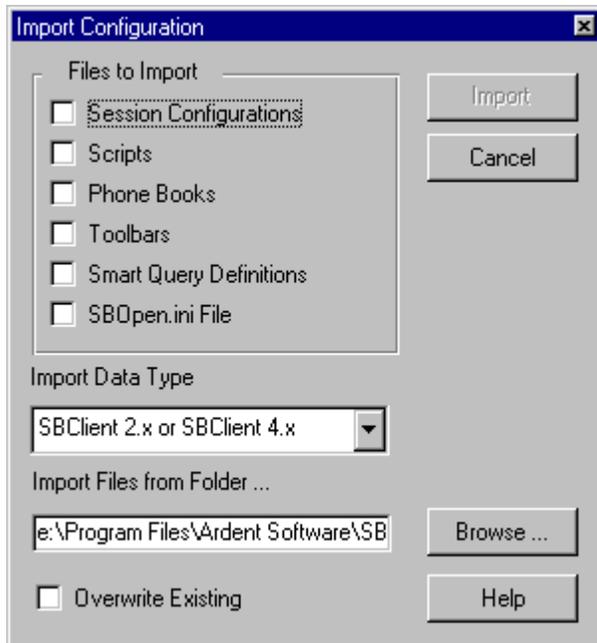
- Administrator account. SBClient default configurations for modem, network and serial sessions are stored in the Administrator account in the SBClient directory. These may be amended only by the system administrator who starts SBClient in the Administrator account. They can then be installed in an individual's account, and used or amended as required.
- User account. When SBClient is installed on the user's PC, a user account is created and the default configuration settings automatically copied to it. User's can modify these settings. They can also update their personal settings with any changes to the default settings. Each new user account will store its particular user settings.

You can import configuration settings from existing SBClient accounts. You can import configuration settings into user and administrator accounts.

Importing Configuration Settings

To import configuration settings:

1. Choose **File > Import Configuration**. The **Import Configuration dialog** is displayed.



2. Select the defaults to import:

- Session Configurations checkbox imports the session configuration from the specified account.
- Scripts checkbox imports the scripts from the specified account.
- Phone Books checkbox imports the phone books (used as the dialing directory) from the specified account.

- Toolbars checkbox. Imports all toolbar definitions from the specified account.
 - SBOpen.ini File checkbox. Imports the entire contents of specified accounts SBOPEN.INI file.
3. Select the Import Data Type.
Select SBClient 2.x for all SBClient and Termulator 2.x accounts. Select Termulator 1.x for Termulator 1.x accounts.
 4. Enter the path to the SBClient or Termulator account that you are importing from, or click Browse to find the account directory.
 5. Select the Overwrite Existing checkbox if you wish to overwrite existing configuration settings with the same name.
 6. Click Import.

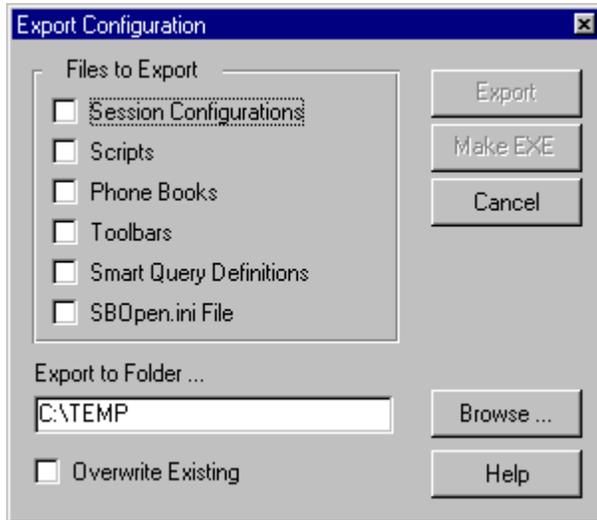
SBClient imports the selected configuration settings to the current account. If the Overwrite Existing checkbox is not checked, a confirmation dialog is displayed each time SBClient detects configuration settings with the same name as the current configuration. You can choose to retain the current setting, or overwrite it with the setting being imported. If the Overwrite Existing checkbox is checked, SBClient will overwrite the current settings, without prompting you.

Exporting Configuration Settings

You can export existing configuration settings to existing or new SBClient accounts.

To export configuration settings:

1. **File > Export Configuration.** The **Export Configuration** dialog is displayed.



2. Select the defaults to export:

- **Session Configurations** checkbox exports the session configuration to the specified account.
- **Scripts checkbox** exports the scripts to the specified account.
- **Phone Books** checkbox exports the phone books (used as the dialing directory) to the specified account.
- **Toolbars** checkbox exports all toolbar definitions to the specified account.

- **SBOpen.ini File** checkbox exports the entire contents of specified account's SBOPEN.INI file.
3. Enter the path of the SBClient account that you are exporting to, or click Browse to find the account directory. You can export configuration settings to a directory that does not contain an existing SBClient account.
 4. Select the **Overwrite Existing** checkbox if you wish to overwrite existing configuration settings with the same name.
 5. Click **Export**.

SBClient exports the selected configuration settings to the specified account. If the Overwrite Existing checkbox is not checked, a confirmation dialog is displayed each time SBClient detects configuration settings with the same name as the destination configuration. You can choose to retain the destinations setting, or overwrite it with the setting being exported. If the Overwrite Existing checkbox is checked, SBClient overwrites the destination configuration, without prompting.

Changing a Session Configuration

You can use the options on the Setup menu to change the configuration settings for a session, from either the SBClient Office or the session window.

Note

Some Setup options are also available as active areas on the SBClient Office.

Any changes you make to a session configuration apply only to the active session, and not to previous sessions opened using the same configuration, or sessions opened by other users on the same network.

Creating a New Session Configuration

You can create a new session configuration by modifying an existing configuration and saving it under a different name.

Saving a Session Configuration

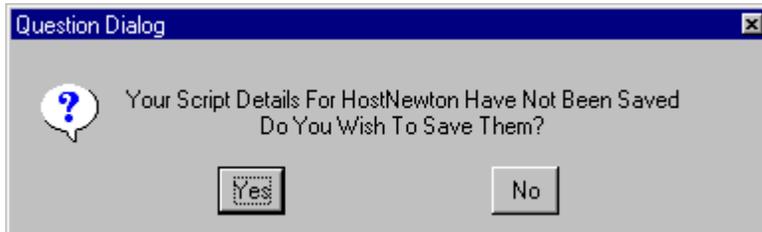
After you have tested the settings for your session configuration you can save them permanently. You can save a session configuration from either the SBClient Office or the session window.

To replace the existing session configuration with your changes:

Choose File > Save Configuration.

The session configuration is updated.

If you have made changes to a session configuration, and then select another session configuration without having saved your changes, the following dialog is displayed, allowing you to save your changes:



To save your settings as a new session configuration:

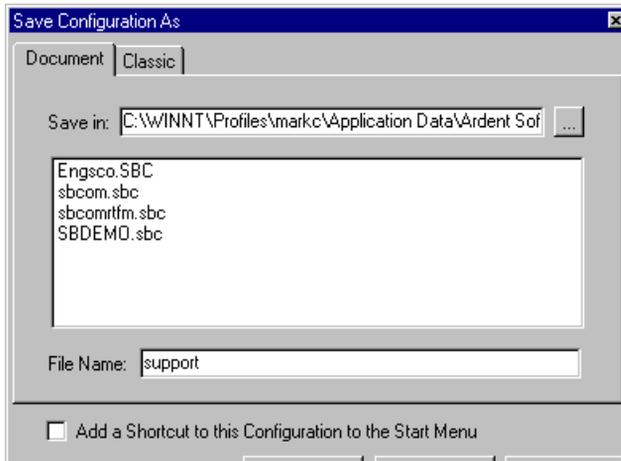
Methods of Saving Configurations

SBClient 4 introduced the concept of .sbc files for storing session configurations. In SBClient 5 the storing of configurations as .sbc files is now the default behaviour of the product. The old behavior of storing configurations within database files which only SBClient can access is now known as "Classic" session maintenance.

The default behavior is known as "Document" session maintenance. SBClient is keeping pace with Microsoft's push for the operating system and applications to become more document centric.

Saving Configuration Using the Document Method

1. Choose **File > Save Configuration As**. This displays the Save Configuration As dialog.



2. A default Save path will be provided. This path is the path to your SBClient data in your user profile. It is possible to amend this path to any location on the machine for which you have both read and write permissions. The default Save path is:

```
C:\windowsdirectory\Profiles\username\Application Data\Ardent  
Software\SBClient\SESSIONS
```

Note



On Windows95 and Windows98 machines if profiling is not enabled then this path will be:

```
C:\Windowsdirectory\Application Data\Ardent Software\SBClient\SESSIONS
```

The path can be amended by either typing in a changed path or by using the Browse button provided to the righthand side of the **Save in** textbox.

3. Either select a **File Name** from the list provided by doubleclicking on it or enter a new **File Name** then click **OK**.

Note

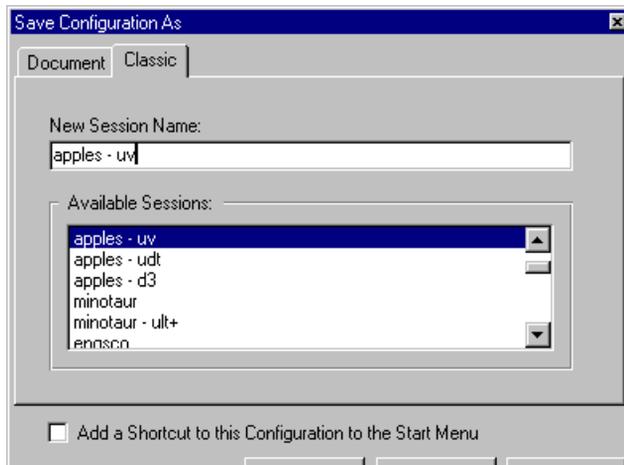


If you are overwriting an existing configuration then SBClient will prompt you to confirm that you want to do this.

4. SBClient will now ask you if you would like the configuration to be added to the Start menu. This will create a SBClient group in your personal settings. This group will contain shortcuts to all SBClient configurations that have been added to the **Start** menu.

Saving Configuration Using the Classic Method

1. Choose **File > Save Configuration As**. This displays the Save Configuration As dialog.



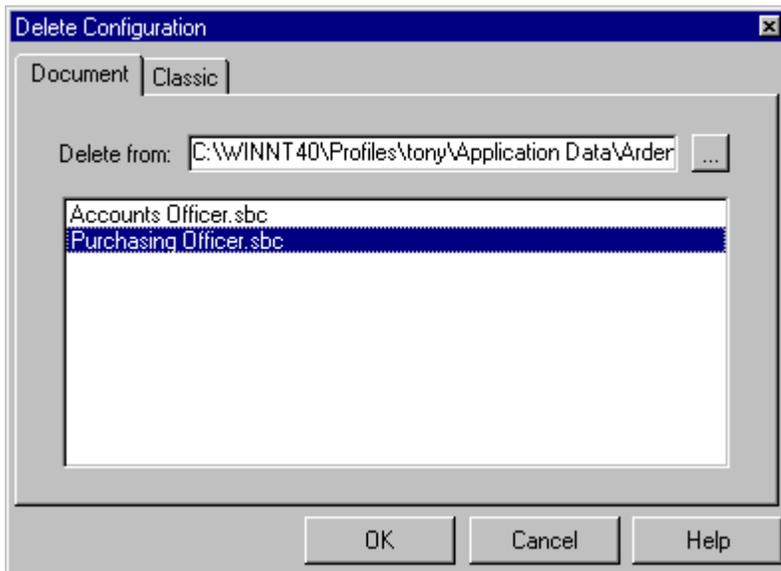
2. Enter a **New Session Name**. If you select an existing session configuration, it will be overwritten (a warning dialog will be displayed).
3. Click the **OK** button.
4. SBClient will now ask you if you would like the configuration to be added to the Start menu. This will create a SBClient group in your personal settings. This group will contain shortcuts to all SBClient configurations that have been added to the **Start** menu.

Deleting a Session Configuration

You can delete a session configuration you no longer require.

To delete a session configuration:

1. Choose **File > Delete Configuration** or press the Delete key when in SBClient office. This displays the **Delete Configuration** dialog:



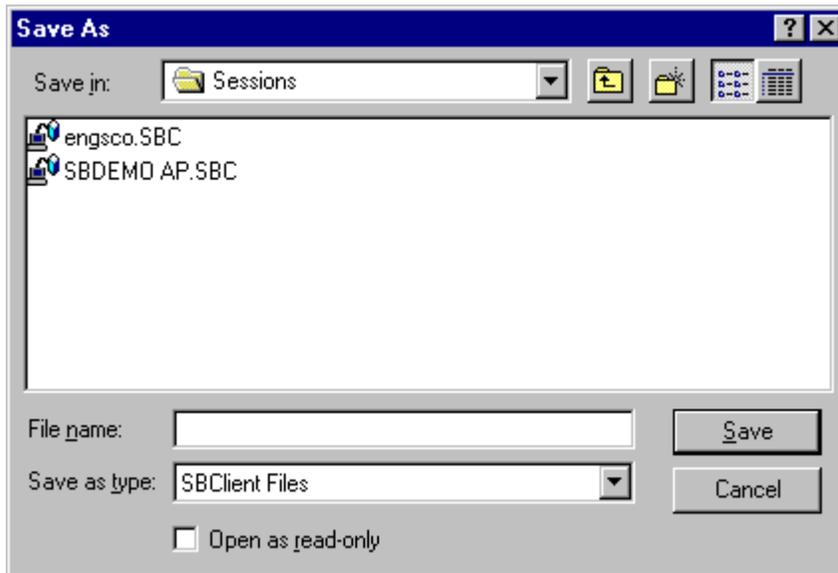
2. Highlight the session configuration you wish to delete in the Non Open Sessions list. Open sessions are not displayed and cannot be deleted.
3. Click the **OK** button. SBClient asks you to confirm your selection.

Portable Session Configurations

SBClient configuration information is normally stored in a Btrieve database file. However it is possible to create a copy and store configuration session information into an SBC file. SBC files can then be double-clicked on to launch that particular configuration.

To create an SBC file:

1. Select **Utilities > Configuration Sharing > Save File**. The following dialog appears:



2. Provide a filename and then click the Save button.

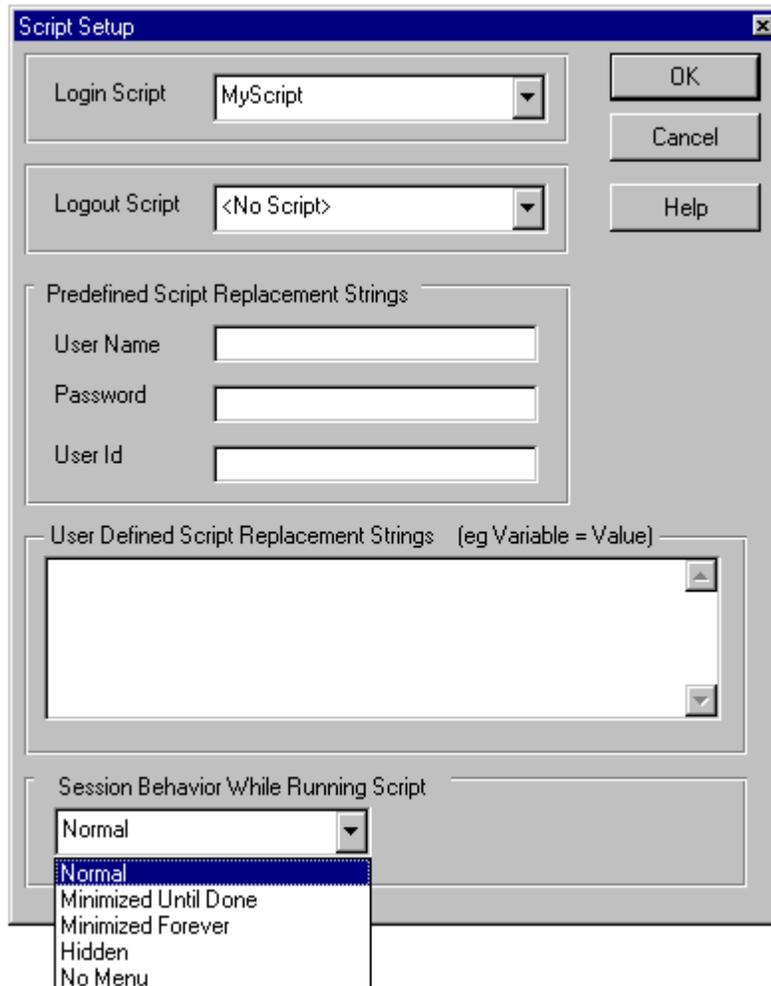
Note ✓

.sbc files are saved in the Sessions directory location by default. The exact location of this directory will depend on where installed SBClient. .sbc files can also be saved to the Windows desktop



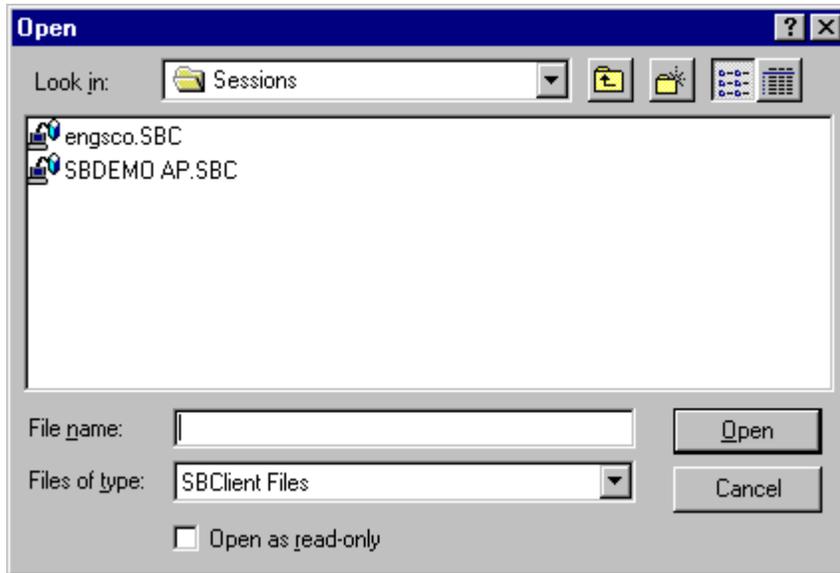
or indeed any other location with read/write access. If you save the .sbc file to the Windows desktop then this may be more convenient as you can double-click on the .sbc file directly from the Windows desktop.

3. If the session configuration being saved has an associated script then this will also be embedded in the .sbc file and will run automatically when the .sbc file is opened later on. It is possible to control the appearance of SBClient via the **Setup > Script** menu.



To launch a session using an SBC file:

1. Either select **Utilities > Configuration Sharing > Open File**, in which case the following dialog will be presented and you should select the .sbc file you want and click on the Save button,



or

double-click on the .sbc file icon directly from either the Windows desktop or via the Windows explorer depending on where the .sbc file is located.

The session configuration and embedded script (if applicable) will run.

Chapter 3 - Host Utilities Installation



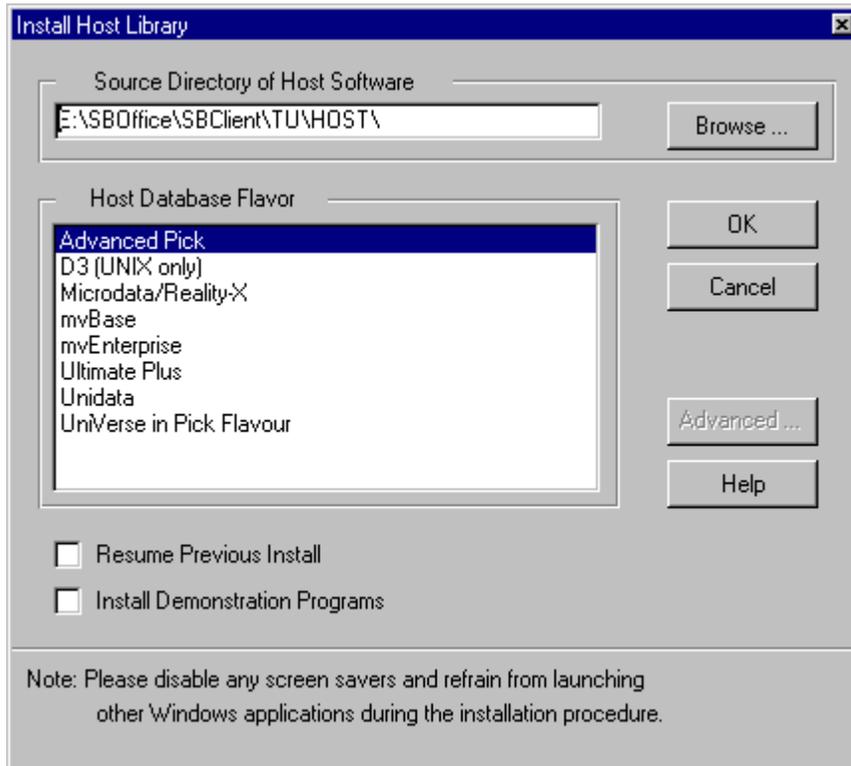
The SBClient host library is a collection of BASIC subroutines that allow programmers to integrate host and PC functionality and to convert character screens and menus to GUI forms. When invoked, the library subroutines send various escape sequences to the PC from the host, where they are interpreted by the SBClient run time engine.

The host library needs to be installed on the host before its subroutines can be used. Host library installation may take between twenty and sixty minutes, depending on the speed of the host computer. Before installing the host library you should close all Windows applications and disable any active screen savers.

Note 

The host library cannot be installed from a client installation.

Host Library Install



To install SBClient host library subroutines:

1. Invoke SBClient.
2. Open a session to the host. See Running an SBClient Session.
3. Create an account on the host where you wish to store the host subroutines (eg SBCLIENT).
4. Log to this account on the host and go to the operating environment prompt (e.g. TCL/ECL).
5. Choose **Transfer > Host Program Installation** from the SBClient menu. This displays the **Install Host Library** dialog (as shown above). If this option is unavailable (dimmed), then SBClient has been invoked in a client installation.

6. From the **Host Database Flavor** list, select the relevant host database.

Note 

When using UniData as the host, SBClient permits users to use local, direct or global catalogs. For advantages and disadvantages of the different catalog options please refer to Ardent's UniData RDBMS product manuals.

7. If you wish to install the SBClient demonstration programs then you should check the **Install Demonstration Programs** checkbox before proceeding.
8. If a previous installation of the host library failed then you can continue with the previous installation by checking the **Resume Previous Install** checkbox.
9. Press the **OK** button. The host library subroutines will now be installed in the account you logged to on the host. This may take from twenty minutes up to a number of hours depending on the speed of the machine and network. A dialog will display the progress of the installation.

Warning 

When installing the bootstrap for the Host Library installation setting the Unix kill character to any character less than 127 can cause the install to abort.

10. To make the SBClient host library available to another account, run TU.UPDATE.ACCOUNT (or TU.UPDATE.ACCOUNT.UVNT in uniVerse for Windows NT, or TU.UPDATE.ACCOUNT.UDT for UniData NT) from the account where the host library was originally installed. This prompts for the name of the target account (or the path to the target account on platforms that provide for paths instead of account names) in which the library is to become available. It then creates file and program pointer entries in the MD (or VOC) of that account.

Warning 

When installing and running the 3GL Host API on Advanced Pick or D3 you must ensure that the extended character mode setting is on. File transfer and Host GUI screens will not function correctly unless this is done. To turn extended character mode on use the following statement at TCL "XCS-ON".

Note 

The first two digits of the version number of SBClient must always match the first two digits of the host library install. The third digit in SBClient can be higher than the third digit of the host library install, but can never be less than this value. If these versions don't match then the SBClient host library will not function correctly.

Once the host library has been installed then to determine version information you need to run the following command at TCL/ECL:

```
SBCVERSION
```

Host Library Demonstrations

There are a number of demonstration programs in the DEMOBP file. These programs demonstrate various features of the SBClient host utilities.

To be able to run these demonstrations you should firstly run

```
BUILD.CARS
```

from the SBClient account. This command builds the CARS and related files of test data required by the demonstration programs.

Below are listed some of the sample demonstration programs which have been supplied with SBClient. You should list the DEMOBP file to obtain a complete listing of all demo programs.

APPLICATION.DEMO

This is a HostGUI screen which demonstrates major features of the SBClient Host Library.

CARS.HOSTGUI.

This is a HostGUI model program.

SCRIPT.DEMO

This is a demonstration program which shows SBClient's interoperability with VBScript. VBScript is Microsoft's language for OLE automation and web development.

OCXDEMO

This demonstrates the use of ActiveX controls in a GUI form.

WORD

This demonstrates SBClient's interoperability with Microsoft Word.

EXCEL

This demonstrates SBClient's interoperability with Microsoft Excel.

Note



Consult the "SBClient Programmer's Guide" for more information on using the host library.

Patching SB+ Server

Post Release 4 of SB+ Server

Version 4 of SB+ has integrated SBClient 4's Host Library. You may perform an upgrade of the host library in SB+ 4 at anytime. To do this you must logto the SB+.DEFN account (SB.DEFN on UniData) and perform a Host Library installation.

Note

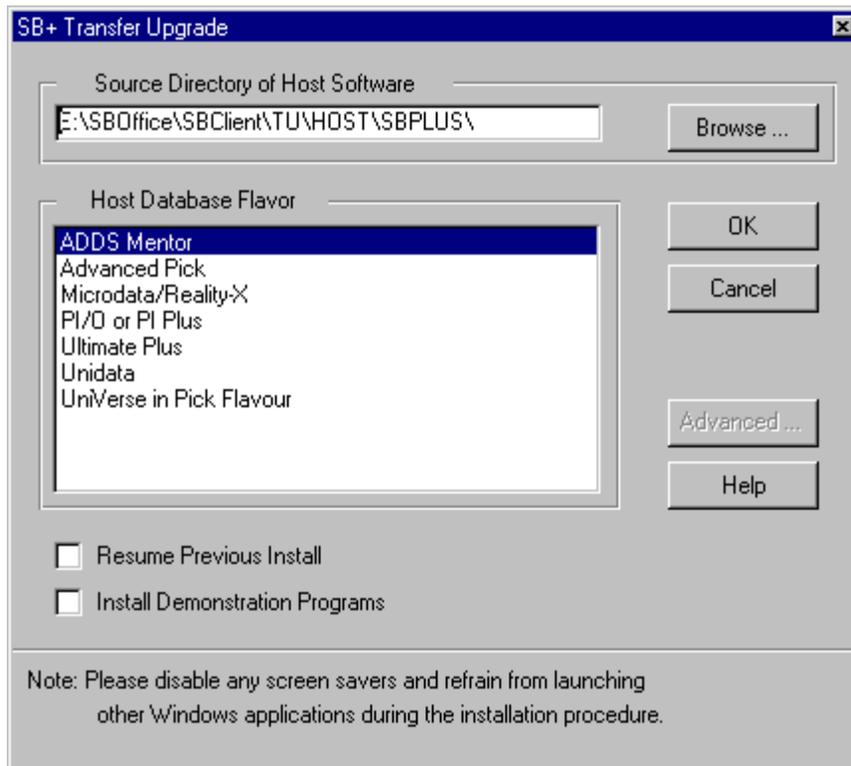


The first two digits of the version number of SBClient must always match the first two digits of the host library install. The third digit in SBClient can be higher than the third digit of the host library install, but can never be less than this value. If these versions don't match then the SBClient host library will not function correctly.

Once the host library has been installed then to determine version information you need to run the following command at TCL/ECL:

```
SBCVERSION
```

Prior to Release 4 of SB+ Server



A new SB+ file transfer protocol (optimized for speed) has been implemented. SB+ releases prior to 3.3.2/2 must be patched as described below to work with this release of SBClient. SBTerm for DOS will not work with this new SB+ file transfer protocol. The only differences the user will notice are the error messages that may be displayed in the Progress dialog.

This patch can only be installed if SBClient is invoked in the administrator account. SB+ patch installation may take between 5 and 15 minutes, depending on the speed of the host computer. Before installing the SB+ patch you should close all Windows applications and disable any active screen savers.

The SB+ patch must be installed from the SB+.DEFN account (SB.DEFN on those systems which do not allow a plus sign in file or account names). To allow for situations where the SB+ and SB+.DEFN accounts have been renamed, the account name test within the patching mechanism

has been extended to look for a \$RENAME record in the master dictionary (MD or VOC) of the renamed SB+.DEFN account. If the patch is being run from an account other than SB+.DEFN and the name of that account matches field 1 of the \$RENAME record, patching will still proceed. In this event, the patch program will require the new name of the SB+ account to be in field 2 of the \$RENAME record.

To install SB+ patch subroutines:

1. Invoke SBClient.
2. Open a session to the host. See *Running an SBClient Session*.
3. Log into the SB+ account (SB+.DEFN). Ignore the ACCESS DENIED message.



On uniVerse accounts it may be necessary to update the VOC before proceeding. To do this type "UPDATE.ACCOUNTS" at the TCL prompt.

4. Choose **Transfer > SB+ Transfer Upgrade** from the SBClient menu. This displays the Install Host Library dialog. If this option is unavailable (dimmed), then SBClient has not been invoked in the administrator account. Exit SBClient and re-invoke from an SBClient Administrator account.
5. From the **Host Database Flavor** list, select the relevant host database.
6. Press the **OK** button. The SB+ patch subroutines will now be installed in the account you logged to on the host. A dialog will display the progress of the installation. Ignore CREATE.FILE error messages regarding pre-existing files.
7. Press the **OK** button when the upload is complete.

Chapter 4 - Data Transfer



SBCClient's flexible file transfer capabilities give you a variety of methods for transferring and sharing data between your PC and host environments, although there will generally be one method most suited to performing a particular task.

The two main methods for sharing information are:

- **Data File Transfer.** Transfer and convert data files between the host and your PC.
- **Standard File Transfer.** Transfer standard files between the host and your PC using industry-standard transfer protocols without converting the files. Standard file transfer protocols are generally appropriate when you do not want the format of the file to be changed by the transfer.

Four other methods for sharing information are:

- **Windows Clipboard.** Use the Windows clipboard to share session data with other applications.
- **DDE.** Dynamic Data Exchange is a method of interprocess communication. It can be used to control other applications, as well as the exchange of data.
- **OLE.** Object Linking and Embedding is another method of interprocess communication. It is based on DDE and has similar as well as more advanced properties.
- **VBScript.** This is a subset of the popular Microsoft Visual Basic programming language. SBCClient implements VBScript to allow interoperability with other applications.

Note



To run DDE transfers, using OLE or VBScript, you will need to install SBClient's host programs (see "[Chapter 3 - Host Utilities Installation](#)" for more information).

Data File Transfer

SBClient allows you to convert database files from one environment to another in a user-specified format. SBClient's data file transfer facility is unique in its ability to handle files from a nested relational database system such as uniVerse or UniData.

The **Transfer > Host To PC** option allow you to convert nested relational database files to records in specific PC file formats. Conversely, the **Transfer > PC To Host** option allow you to convert files from popular PC software (eg Word and Excel) to records in a nested relational data file format.

Standard File Transfer

SBClient lets you connect your PC to another PC, a Unix system, a bulletin board, an online service or virtually any other remote system. It supports a set of industry-standard data file transfer protocols that allow you to transfer data between your PC and the remote system quickly and easily. These protocols include:

- ASCII transfer
- XModem
- XModem-1K
- XModem-1K-G
- YModem
- YModem-G
- ZModem

The **Transfer > Get File** option allows you to download a file from the host to your PC, while the **Transfer > Send File** option allows you to upload a file from your PC to the host.

Windows Clipboard

You can use the cut, copy and paste capabilities of the Windows clipboard to share session data, such as reports and query listings, with another application or environment. For example, copying data from a report in a session window to a Windows application like Excel, or lines of code from a Windows editor to a host editor.

Note 

On Microsoft Windows95 you must install the Clipboard Viewer. You can do this from **Control Panel > Add/Remove Programs**.

Dynamic Data Exchange

Dynamic Data Exchange (DDE) is a data exchange protocol that allows applications to communicate with one another and share information. The protocol was developed by Microsoft and has been implemented in many Windows software products. Use DDE when you need to transfer data from a host system to a DDE client running locally on your PC or on a network, or from a DDE client to a host system.

While more complex than traditional data transfer techniques, DDE is a very powerful interface for sharing data between applications. For example, using Smart Query (a DDE enabled application) you can build, save and run Sort and List commands for your host computer running a nested relational operating system. The output from your Sort or List command can be sent directly to a Word document, a Lotus 123 spreadsheet or an Excel spreadsheet.

See DDE Functions in the “SBClient Programmer’s Guide” in the online help for more information.

OLE

OLE (Object Linking and Embedding) is Microsoft's framework for a compound document technology. Briefly, a compound document is something like a display desktop that can contain visual and information objects of all kinds: text, calendars, animations, sound, motion video, 3-D, continually updated news, controls, and so forth. Each desktop object is an independent program entity that can interact with a user and also communicate with other objects on the desktop. Part of Microsoft's ActiveX technologies, OLE takes advantage and is part of a larger, more general concept, the Component Object Model (COM) and its distributed version, DCOM). An OLE object is necessarily also a component (or COM object).

See "Generic Object Routines" in the "SBClient Programmers Guide" for more information.

Some main concepts in OLE and COM are:

| Concept | What it is |
|--|--|
| OLE | A set of APIs to create and display a (compound) document. |
| Document (compound document) | A presentation of different items in an "animated desktop". |
| Item (object; also called a component) | An element in a document, such as an animated calendar, a video window, a sound player, a sound file... |
| Container or container application | The program entity that holds a document or a control. |
| Server application | The program entity that holds an item within an OLE container. |
| Embedding | Adding the source data for an item to a document; use the Paste command in a container application. |
| Linking | Adding a link to the source data for an item to a document; use the Paste Link command in a container application. |
| Visual editing | Activating an item that is embedded in a document and "editing" it. |

OLE

| Concept | What it is |
|--|---|
| Automation | Having one container or server application drive another application. |
| Compound files (structured storage) | A standard file format that simplifies the storing of (compound) documents; consists of storages (similar to directories) and streams (similar to files). |
| Uniform Data Transfer (UDT) | A single data transfer interface that accommodates drag-and-drop; clipboard; and dynamic data exchange (DDE). |
| Component Object Model (COM) | Provides the underlying support for OLE items (objects) and ActiveX controls to communicate with other OLE objects or ActiveX controls. |
| ActiveX control | An item (object) that can be distributed and run on top of a COM. |
| Microsoft Foundation Class (MFC) library | A set of ready-made classes or templates that can be used to build container and server applications. |

OLE

VBScript

Microsoft's Visual Basic Host Script language is supported in SBClient. This provides uniVerse and UniData programmers the ability to pass and execute Visual Basic code from a host application. Microsoft provides 3 Visual Basic languages for application development on the desktop. These are the original Visual Basic (VB), Visual Basic for Applications (VBA), and Visual Basic Host Script (VBS).

VB is predominately used to develop stand alone custom applications. VBA was introduced with Office 97 as script language replacement for DDE macro operations. By using VBA in the Office suite Microsoft has defined a common interface for interoperability between applications. A developer may also license VBA for use within their own application. VBS is a logical extension of VBA. Using VBS a developer may from an html document use standard features of VB and VBA. This is immensely powerful. Initially the use of VBS was confined to the execution of html documents in Internet Explorer (IE) 3 and 4. However recently Microsoft released an Active-X control which allows another application to send VBS code to this control and have that code executed.

Note 

For more information on the SBClient VBScript API consult the "SBClient Programmer's Guide" for more information.

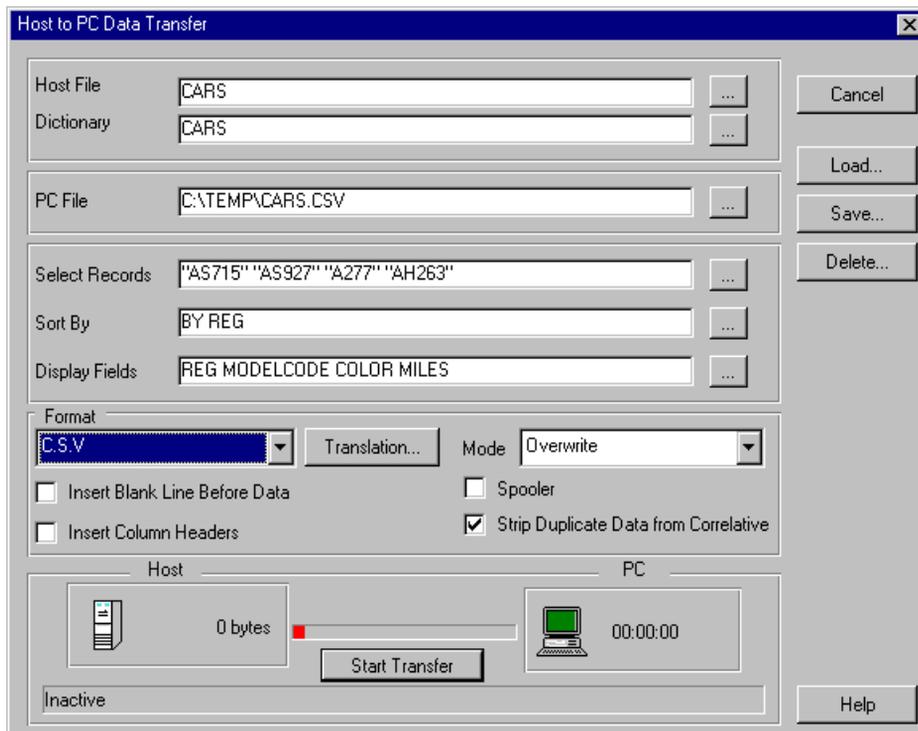
Data Transfer (Host to PC)

This topic shows you how to use the **Transfer > Host To PC** option from the session window to convert data from your PC to a host in a variety of generic formats. You can build, save and run data transfers that extract, sort and format host records, then transfer them to a PC file.

Note



To run a data transfer you need to have installed SBClient's host programs onto the remote computer in the appropriate account. You also need to first open a session, and then log into the appropriate account.



Defining a Data Transfer (Host to PC)

You define a data transfer by specifying the following parameters in the **Host To PC Data Transfer** dialog (you can use an existing data transfer as a basis for the new definition):

- **Select Records.** The selection criteria determining which host record display fields will appear in the PC file. You can define the record selection criteria by field content or by explicit record id.
- **Sort by.** The sort criteria determining the order in which fields are sorted.
- **Display Fields.** The fields on the host record you want to appear in the PC file.
- **Format.** The format in which the host data is transferred.
- **Translation.** The set of character string translations, if any.
- **Mode.** The action to be taken if the destination PC file already exists.

Defining the Selection Criteria

SBClient allows you to define the selection criteria by field content or by explicit record id to accurately pinpoint the records you want.

Click the browse button to the right of the **Select Records** text box to display the **Selection Criteria** dialog.

To select records by explicit record id

1. Display the Selection Criteria dialog. Click the Items tab.
2. Select from one of the five mutually exclusive methods of searching your file:
 - **All.** Search for all records in the file. Ensure the Value text box is clear. Avoid using the All option for files that have large number of records (that is, over 3000) as the list may not be able to display them all.
 - **Contains.** Search for those records whose record ids contain the value you enter in the Value text box.
 - **Range.** Search for those records whose record ids are in the range you enter in the Value text box. Separate the lower and upper limits of the range with a semicolon.

- Ends. Search for those records whose record ids end with the value you enter in the Value text box.
 - Begins. Search for those records whose record ids begin with the value you enter in the Value text box.
3. Click the Search button. SBClient searches your file for all records that match your selection criteria. The record ids are displayed in the Records list.
 4. For each record you want to include in the explicit selection, highlight the record in the Records list and click the Add button. This adds the record to the end of the Selected list. To insert the record in the list, highlight the record in the Selected list that you want to insert before, and click the Insert button. If the Selected list is left empty then all records are selected.

If you are building a List command, the records will be listed in the sequence in which they appear in the selection list. If you are building a Sort command, the sequence in which they appear in the selection list is irrelevant since they will be sorted into the sequence defined by your sort criteria.

To remove a record from the selection list

1. Highlight the record in the selection list.
2. Click the Remove button.

To complete the task

When you have built your explicit selection criteria click the **OK** button.

To select records by field content

1. Display the Selection Criteria dialog. Click the Fields tab if the Fields dialog is not displayed.
2. Construct your selection criteria in the Selection Builder boxes.

For each criterion you define, click the Add button. SBClient constructs the criterion and displays it in the next empty line in the list box. There are four parts to a criterion:

- Operator. Connects multiple criteria. Operators are AND and OR. No operator is required for the first criterion.
- Field. The name of the field on which the selection is based. The field name list originates from the data dictionary you have selected.

- Operand. The relation to apply to the field. The options are: equal (=), not equal (#), greater than (>), less than (<), greater than or equal to (>=), less than or equal to (<=).
- Value. The value to apply to the relation. The value delimiters (double-quotes) are added before and after the value if you do not enter them. If you specify a value of "?" you will be prompted for the value at runtime. This functionality is only available in Smart Query.

To delete a criterion

1. Highlight the criterion in the list box.
2. Click the Delete button.

To replace a criterion

1. Enter the new criterion's operator, field, operand and value.
2. Highlight the criterion in the list box that you want to replace.
3. Click the Replace button.

To insert a criterion

1. Enter the new criterion's operator, field, operand and value.
2. Highlight the criterion in the list box that you want to insert before.
3. Click the Insert button.

To complete the task

1. When you have built your selection criteria, click the OK button.

When you have completed this dialog, the selection criteria portion of the query is displayed in the Select text box.

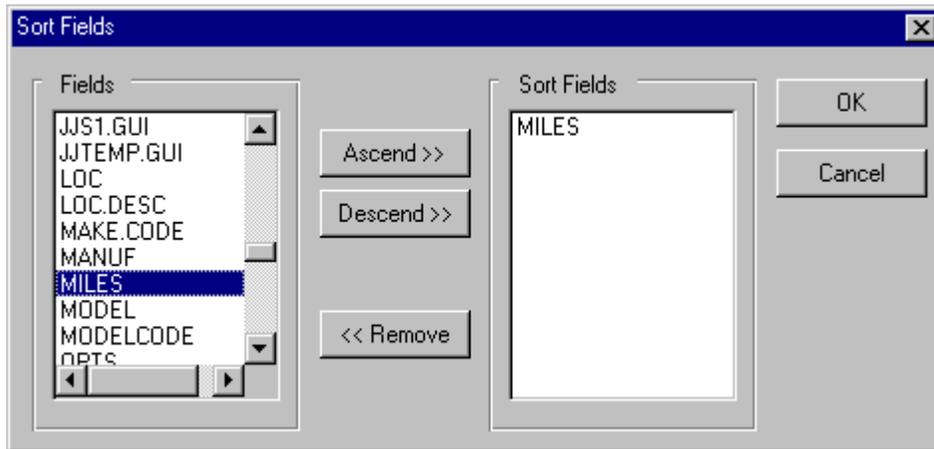
When you have completed this dialog, SBClient builds the selection criteria portion of the data transfer and displays it in the **Select Records** text box.

Note



If you enter a sort criteria directly in the **Select Records** text box, do not enter a BY-EXP sort as SBClient does not support this type of sort.

Defining the Sort Criteria



SBClient allows you to specify the sort criteria for the selected records.

If you do not define any sort criteria, SBClient uses the **List** command to select records for transfer.

To define the sort criteria:

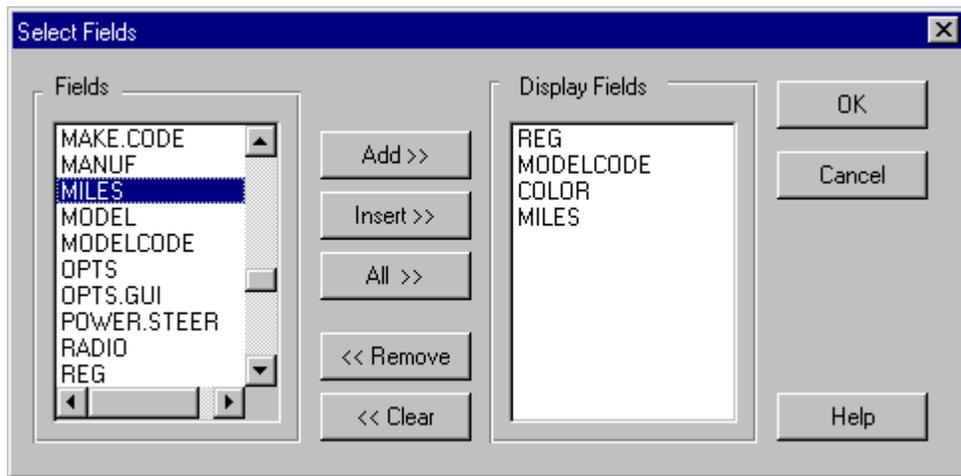
1. Click the browse button to the right of the **Sort By** text box. The **Sort Fields** dialog is displayed.
2. Highlight the primary sort field in the **Fields** list.
3. Click the **Ascend** button if you want to sort the field in ascending order. Click the **Descend** button if you want to sort in descending order.

The selected field is added to the **Sort Fields** list. A (D) appended to the field name indicates a descending sort. You can sort on more than one field.

To remove a field from the **Sort Fields** list, highlight the field and click the **Remove** button. Click the **OK** button. The sort criteria portion of the sort command is built and displayed in the **Sort By** text box.

Note ✓

If you enter a sort criteria directly in the **Sort By** text box, do not enter a BY-EXP sort as SBClient does not support this type of sort.

Defining the Display Fields

The **Display Fields** text box allows you to select the fields in the host record that you want to transfer to the PC file. Leave this field blank if you want to transfer all fields for each record.

Generally you do not need to transfer all record data. For example, when importing a customer file for mail merge purposes, you might not wish to transfer phone, fax and credit information.

Accordingly, you might specify only CO.NAME, CONTACT, ADDRESS, CITY, STATE and ZIP as the required display fields.

Note ✓

Not only are display fields used to determine the data to be imported from the selected records, information from the dictionary is also used to format data in the PC file. Column widths, column headings, field justification, conversions and correlatives are all applied to the PC file during the import process.

To select the fields to appear in the PC file:

1. Click the browse button to the right of the **Display Fields** text box. The **Select Fields** dialog is displayed.
2. For each field you want to appear in the PC file, highlight the field name in the **Fields** list and click the Add button. This adds the field to the end of the Display Fields list. To insert the field in the list, highlight the field in the Display Fields list that you want to insert before, and click the Insert button. To select all fields, click the All button.

The selected fields will be placed in the PC file in the sequence they appear in the selection list.

To remove a field from the Display Fields list:

1. Highlight the field in the Display Fields list.
2. Click the Remove button. To remove all the fields from the Display Fields list click the Clear button.

To complete the task:

1. When you have built your list of display fields, click the OK button. SBClient builds the display fields portion of the sort/list transfer and displays it in the Display Fields text box.

Defining the Transfer Format

SBClient provides a range of formats for changing the host data before it is transferred to your PC.

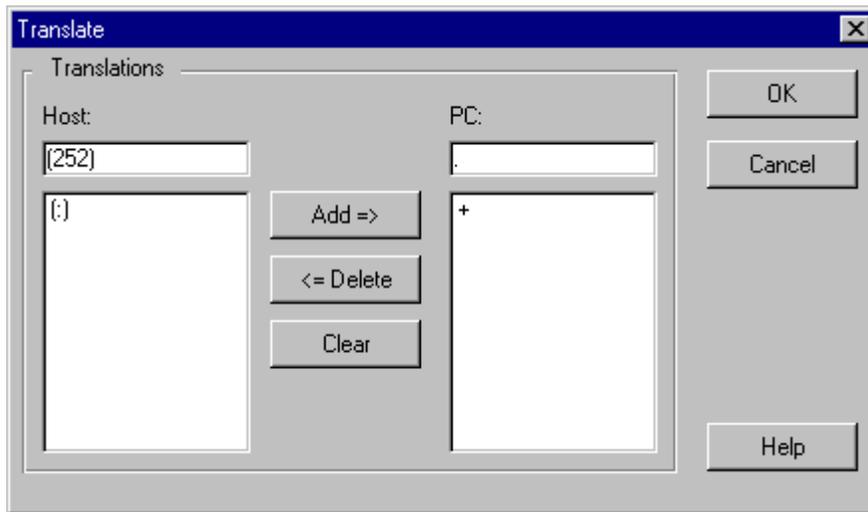
The options available from the **Format** selection list are:

- **Fixed Length.** Each field is formatted according to its dictionary definition. The fields are then combined to create a fixed length record.
- **C.S.V. Comma Separated Variable.** Each field is placed in the record unformatted and separated by commas. This creates variable length records.
- **Tab Delimited.** Each field is placed in the record unformatted and separated by tab characters. This creates variable length records.
- **Space Delimited.** Each field is placed in the record unformatted and separated by spaces. This creates variable length records.

- **dBase.** Each field is placed in the record unformatted, then the record is converted to dBase format. This creates variable length records. This format is recognized by database and spreadsheet applications.
- **SYLK.** Each field is placed in the record unformatted, then the record is converted to SYLK format. This creates variable length records. This format is recognized by spreadsheet applications.
- **ASCII.** Used only for single record transfers (ie no fields).
- **Binary.** Used only for single record transfers (ie no fields).
- **Hexadecimal.** Used only for single record transfers (ie no fields).

These are generic formats understood by a wide number of different applications. For example, Excel understands the spreadsheet compliant dBase and SYLK file formats. However, not all spreadsheets are able to interpret these specific formats, but are able to understand the simpler tab delimited or CSV formats. For more application specific transfers (for example, Excel graphing and Word mail merge) it may be more appropriate to use SBClient's Smart Query tool.

Defining Translations



SBClient allows you to translate specific character strings within the host data into different strings before they are transferred to your PC. For example, the translation of every occurrence of a sub-value mark (ASCII character 252) with a comma.

To translate a character string:

1. Click the Translation button. The **Translate** dialog is displayed.
2. In the **Host** text field, enter the character string you want to translate. To replace a non-printable character, enclose its decimal value within parenthesis.

For example, to replace a sub-value mark (ASCII character 252) with a comma, enter the string (252) in the **Host** text field and a, in the PC text field.
3. In the PC text field, enter the character string that is to replace the host string.
4. Click the **Add** button. The two parts of the translation are added to the Host and PC list boxes.

You can define more than one translation. If you want to delete a translation, highlight it in the Host list box and click the Delete button. Click the Clear button to delete all the translations you have defined.

5. When you have defined all your translations, click the OK button.

Defining the Transfer Mode

SBClient allows you to choose what happens if the destination PC file to which you are transferring data already exists. The options available from the Mode list are:

- **Append.** Adds the data you are transferring from the host to the end of the existing PC file.
- **Overwrite.** Replaces the contents of the existing PC file with the data you are transferring from the host.
- **Preserve.** Does not transfer the data from the host if the PC file exists.

Running a Data Transfer (Host to PC)

In a host to PC data transfer you are transferring a file from the host to a file on your PC.

To run the data transfer you need to open a session to the host, log in to the appropriate account, select the host data file to be transferred to the PC and the PC file to which you are transferring it, choose or define a data transfer and start the transfer.

To run a data transfer:

1. Invoke SBClient.
2. Open a session to the host.
3. Log in to an account on your host system. The account must be one where the host programs are installed or that has been set up to use the data transfer facilities.
4. Choose **Transfer > Host To PC**. The **Host To PC Data Transfer** dialog is displayed.
5. Enter the name of the host file in the **Host File** text box, or use the browse button to select a file from the host's file list.
6. Enter the name of the dictionary for the host file in the **Dictionary** text box, or use the browse button to select a dictionary from the host's dictionary list.
7. Enter the name of the target PC file in the **PC File** text box, or use the browse button to select a file from the PC's file list.

Note



If the PC file already exists, it will be overwritten, appended to or preserved depending on the transfer mode.

8. Once you are satisfied with the data transfer parameters, click the **Start Transfer** button. SBClient converts the data transfer into a nested relational Sort or List transfer, executes the transfer, then transfers the output to the designated PC file. Click the Spooler checkbox if you wish to send the information to the printer as well.

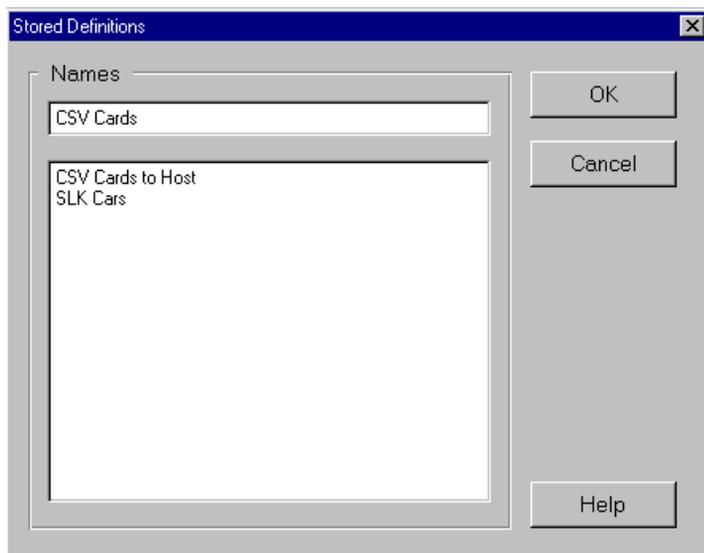
The progress of the data transfer is displayed on the status line.

Saving a Data Transfer Definition

When you have built or modified your data transfer you should save it.

To save a data transfer:

1. From the **Host To PC Data Transfer** dialog, click the **Save** button. The **Stored Definitions dialog** is displayed:

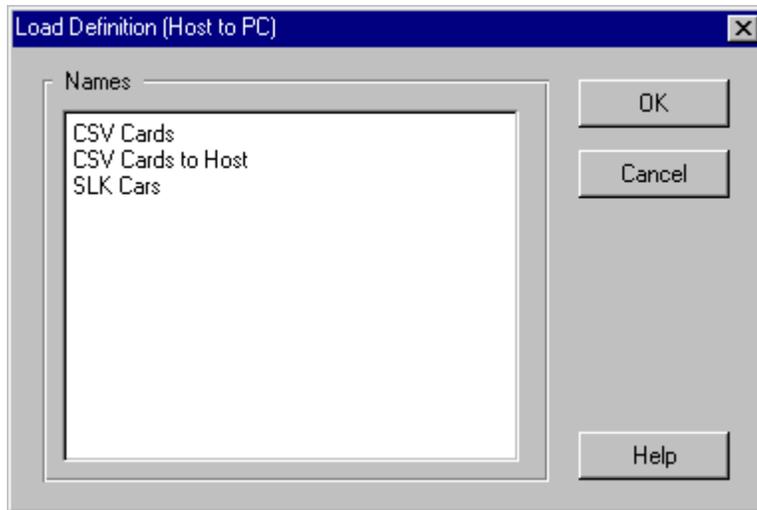


2. Select an existing data transfer or enter a new (case-sensitive) name.

3. Click the **OK** button.

Load Host to PC Definition

1. Click the **Load** button to display the Stored Definitions dialog:



This displays the names of previously saved data transfer configurations.

2. Select the required data transfer and click the **OK** button. SBClient displays the selected data transfer parameters in the **Host To PC Data Transfer** dialog. You may run this data transfer immediately or use it as the basis for building a new data transfer.

Deleting a Data Transfer Definition

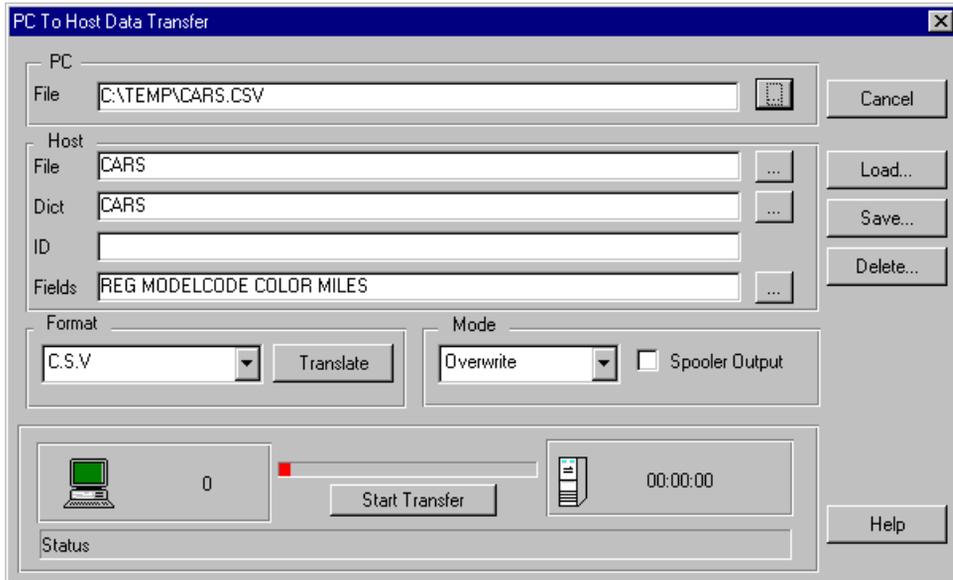
You can delete an existing data transfer definition.

To delete a definition:

1. From the **Host To PC Data Transfer** dialog, click the **Delete** button. The **Stored Definitions** dialog is displayed.
2. Select the data transfer you wish to delete.
3. Click the **OK** button. You will be prompted to confirm the deletion.

4. Click the **OK** button to confirm the deletion or the Cancel button to abort.

Data Transfer (PC to Host)



This topic shows you how to use the **Transfer > PC To Host** option from the session window to convert data contained in a file on your PC, and transfer it to the host as nested relational database records. You can build, save and run data transfers that extract, sort and format PC records, then transfer them to a file on your host.

Note

To run a data transfer, SBClient's host programs must be installed on the remote computer in the appropriate account. You also need to first open a session, and then log into the appropriate account.

Defining a Data Transfer (PC to Host)

You define a data transfer by specifying the following parameters in the PC To Host Data Transfer dialog (you can use an existing data transfer as a basis for the new definition):

- **PC File.** The name of the file that you want to transfer data from.

- **Host File.** The name of the target file on the host.
- **Host Dictionary.** The name of the dictionary where field information is to be extracted from.
- **ID.** Instructions on how to construct the host record ids.
- **Fields.** The fields in the host record to receive the data transferred from the PC file.
- **Format.** The format in which the PC data is transferred.
- **Mode.** The action to be taken if the destination host file already exists.

Note



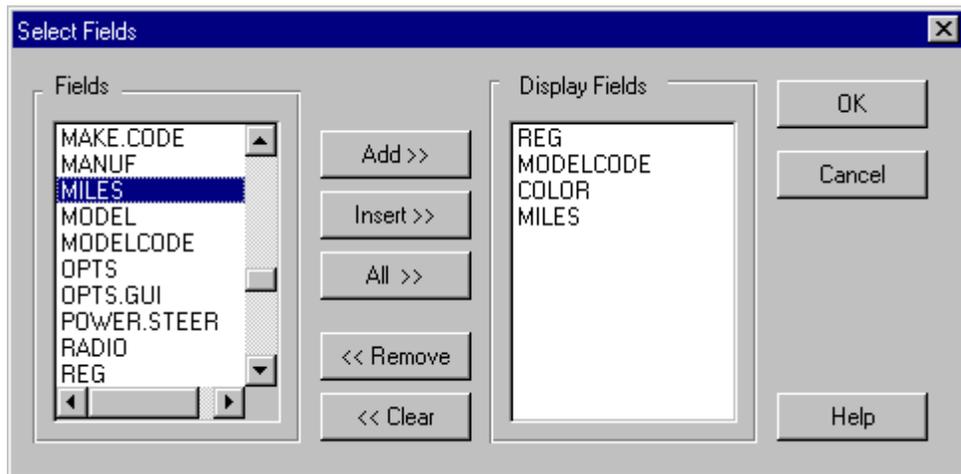
If you do not know the name of the file you are looking for you can click on the browse button (delimited by three periods) located to the right of the textbox in question.

Defining Host Record Ids

When you transfer records from your PC to the host, you must instruct SBClient how it should construct the host record ids. This is achieved in three ways:

- Leave the ID text box blank. This instructs the transfer to use the record id field in the Fields list. Note that you must have a record id field in the Fields list (for example @ID, CODE etc.).
- Specify a literal list of ids. For example: "0003" "0004"
- Use a literal string in conjunction with a mask. For example, using "RECORD.3" or "RECORD.###" will produce a sequential list of record ids: RECORD.001, RECORD.002, RECORD.003 etc. Changing the number after the decimal point (or changing the number of # symbols) will change the size of the right justified counter.

Defining the Fields



The **Fields** text box allows you to select the fields in the host record that will contain data transferred from the PC file. Leave this field blank for single record transfers.

To select the host fields that will receive data:

1. Click the browse button to the right of the **Fields** text box. The **Select Fields** dialog is displayed.
2. For each field to receive data from the PC file, highlight the field name in the **Fields** list and click the **Add** button. This adds the field to the end of the **Display Fields** list. To insert the field in the list, highlight the field in the **Display Fields** list that you want to insert before, and click the **Insert** button. To select all fields, click the **All** button.

If you are exporting records with delimiters, the selected fields are extracted from the PC file in the sequence they appear in the selection list.

If you are exporting fixed length records, the selected fields are extracted from the PC file in the sequence they appear in the selection list combined with the field length defined in the host file dictionary. For example, if you have selected the fields @ID, COLOR and DOORS, whose lengths are 4, 5 and 1 respectively, the first four characters in the exported record are placed in the host's @ID field, the next five characters are placed in the COLOR field and the next character is placed in the DOORS field.

If you are exporting records with delimiters, the number of fields in the selection list should not exceed the number of delimited fields in the exported record. If you are exporting fixed length records, the combined length of the selected fields should not exceed the length of the exported record.

To remove a field from the Display Fields list:

1. Highlight the field in the Display Fields list.
2. Click the Remove button. To remove all the fields from the Display Fields list click the Clear button.

To complete the task:

1. When you have built your list of display fields, click the OK button. SBClient builds the fields selection portion of the transfer and displays it in the Fields text box.

Defining the Transfer Format

SBClient provides a range of formats for changing your PC data before it is transferred to the host.

Formats available from the Format selection list are:

- **Fixed Length (no delimiter).** The length of the data in each field is determined by the dictionary definition. Once data has been placed in the new record in accordance with the specified field definitions, the new record is saved in the host file.
- **C.S.V.** Comma Separated Variable. The data for each field is determined by the commas embedded in the data transferred. Once data has been placed in the new record in accordance with the specified field definitions, the new record is saved in the host file.
- **Tab Delimited.** The data for each field is determined by the tabs embedded in the data transferred. Once data has been placed in the new record in accordance with the specified field definitions, the new record is saved in the host file.
- **Space Delimited.** The data for each field is determined by the spaces in the data transferred. Once data has been placed in the new record in accordance with the specified field definitions, the new record is saved in the host file.
- **dBase.** Each dBase record is converted to an unformatted record. This creates variable length records which are then written to the host.

- **SYLK.** Each SYLK record is converted to an unformatted record. This creates variable length records which are then written to the host.
- **ASCII.** Used only for single record transfers (ie no fields).
- **Binary.** Used only for single record transfers (ie no fields).
- **Hexadecimal.** Used only for single record transfers (ie no fields).

Note



If you are exporting records with delimiters, the number of fields in the selection list (see Defining the Fields previously) must not exceed the number of delimited fields in the exported record. If you are exporting fixed length records, the combined length of the selected fields must not exceed the length of the exported record.

Selecting the Transfer Mode

SBClient allows you to choose what happens if the destination host file to which you are transferring data already exists. The options available on the Mode selection list are:

- **Overwrite.** Completely replaces the corresponding records in the existing host file with the data you are transferring from the PC.
- **Preserve.** Does not overwrite records in the host file if the records already exist.

Running a Data Transfer (PC to Host)

In a PC to host data transfer you are transferring a file from your PC to a file on the host.

To run a data transfer you need to open a session to the host, log in to the appropriate account, select the PC data file to be transferred to the host and the host file to which you are transferring it, choose a data transfer and start the transfer.

To run a data transfer:

1. Invoke SBClient.
2. Connect to the host.

3. Log in to an account on your host system. The account must be one where the host programs are installed or that has been set up to use the data file transfer facilities.
4. Choose **Transfer > PC To Host**. The **PC To Host Data Transfer** dialog is displayed.
5. Enter the name of the PC file in the **PC File** text box, or use the browse button to select a file from the PC's file list.
6. Enter the name of the target host file in the Host File text box, or use the browse button to select a file from the Host's file list.
7. Enter the name of the dictionary for the host file in the Dictionary text box, or use the browse button to select a dictionary from the host's dictionary list.

Note 

If the host file already exists, then records in the file will be overwritten, appended to or preserved depending on the transfer mode.

8. Once you are satisfied with the data transfer parameters click the **Start Transfer** button. SBClient executes the transfer, then transfers the output to the designated host file. Click the **Spooler Output** checkbox if you wish to send the information to the printer as well. See [“Defining a Data Transfer \(PC to Host\)”](#) on page 167.

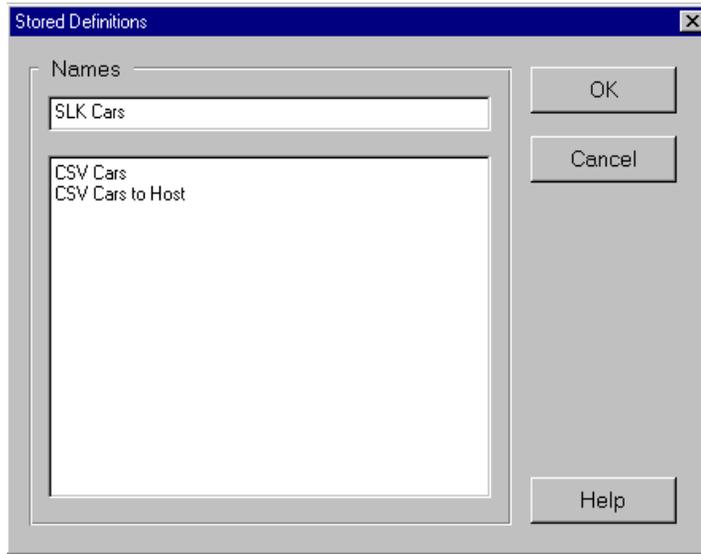
The progress of the data transfer is displayed on the status line.

Saving a Data Transfer (PC to Host)

When you have built or modified your data transfer you may wish to save it.

To save a data transfer:

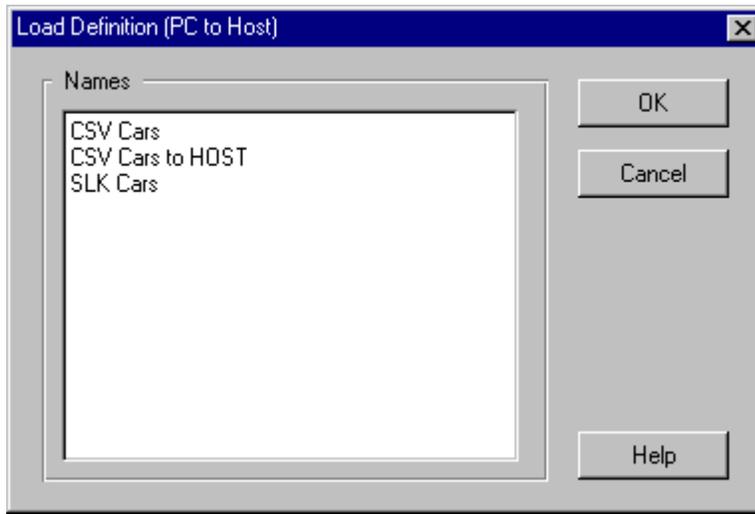
1. From the PC To Host Data Transfer dialog, click the Save button. The Stored Definitions dialog is displayed:



2. Select an existing data transfer from the list to overwrite or enter a new (case-sensitive) name.
3. Click the **OK** button.

Load PC to Host Definition

1. Click the **Load** button to display the **Load Definition (PC to Host)** dialog:



This displays the names of previously saved data transfer configurations.

2. Highlight the data transfer you wish to recall and click the **OK** button. SBClient displays the selected data transfer parameters in the **PC To Host Data Transfer** dialog. You may run this data transfer immediately or use it as the basis for building a new data transfer.

Deleting a Transfer (PC to Host)

You can delete an existing data transfer.

To delete a data transfer:

1. From the **PC To Host Data Transfer** dialog, click the **Delete** button. The **Stored Definitions** dialog is displayed.
2. Highlight the data transfer you wish to delete.
3. Click the **OK** button. You will be prompted to confirm the deletion.
4. Click the **OK** button to confirm the deletion or the **Cancel** button to abort.

Downloading and Uploading Files

SBClient allows you to transfer data files between your PC and a remote system without reformatting. The remote system may be another PC, a Unix system, a bulletin board, or an online service.

You download files to your PC and upload files from your PC.

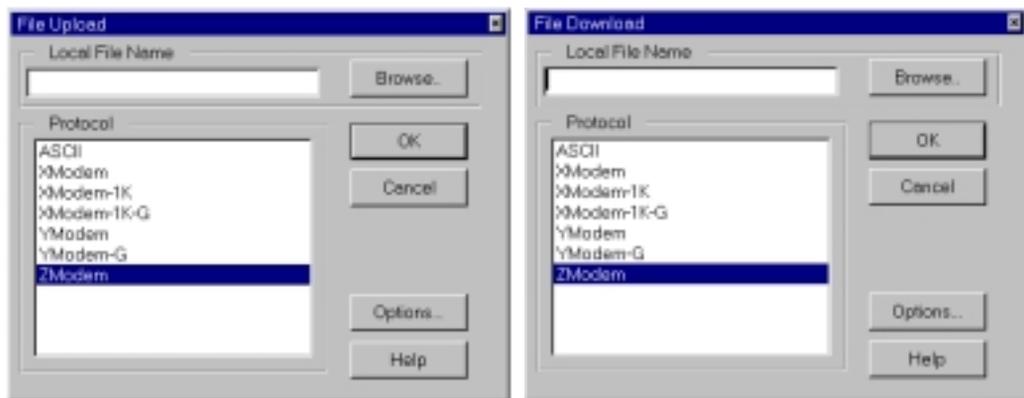
To download or upload a file:

1. Connect to the remote system.
2. Where appropriate, log in to the relevant account.
3. If you want to use the remote system to initiate the upload or download, enter the appropriate commands on the remote system.

For example, you can run the programs `rc` and `sc` on many Unix platforms. If you want your PC to initiate the transfer, you must enter a command in the Host Command text box on the File Transfer Options dialog when you get to step 7. This is applicable to ASCII transfers only.

4. To download a file, choose **Transfer > Get File**. To upload a file, choose **Transfer > Send File**.

The **File Download** or **File Upload** dialog is displayed:



5. Enter the PC filename (including the full DOS path) in the Local File Name text box, or click the Browse button to select a file from a file selection dialog.

The PC filename is irrelevant if you are downloading using ZModem and select the Auto Download checkbox in the File Transfer Options dialog (see step 7).

6. Select the appropriate protocol from the Protocol list. The protocol you select must match the remote system's protocol.
7. Click the **Options** button. The **File Transfer Options** dialog is displayed:
8. Select the **No User Response Required** checkbox if you do not need to respond to SBClient's message on the successful completion of a file transfer. This is not enabled for ASCII transfer as the user must decide when an ASCII transfer is complete.
9. If you are downloading a file, select the appropriate **Download Options**. These options are inactive (grayed) if you are uploading. Options are:
 - **Overwrite If File Exists** checkbox. If the file exists on your PC, replace the file's contents with the data you download.
 - **Auto Download** checkbox. This option applies if the ZModem protocol is selected. The file is downloaded when the download is initiated by the remote system. The filename is carried across with the file from the remote system and overrides any file selection you have defined in the **Local File Name** text box.
10. Enter the host command in the **Host Command** text box.

This command applies if you have selected the ASCII protocol and you want your PC to initiate the file transfer. Enter the command you would normally type on the host system to initiate the upload or download. When you start the transfer, SBClient sends this command to the remote host.

11. Click the **OK** button in the **File Transfer Options** dialog.
12. To start the file transfer, click the **OK** button in the **File Upload** or **File Download** dialog. SBClient uploads or downloads the relevant file.

Using the Clipboard

You can copy information displayed in your session window to the Windows clipboard. Using the clipboard you can then pass the information to other applications on your PC (eg Microsoft Word).

You can:

- copy the entire contents of the session window to the clipboard as text.
- graphically copy a selected part of the session window to the clipboard as a Windows bitmap.
- copy selected text from the session window to the clipboard.
- record all session output to the clipboard.

Clipboard Setup

The **Edit > Clipboard Setup** option allows you to change the following settings (these are not mutually exclusive):

- **Join Lines.** Carriage Return/Line Feed (CRLF) control characters are removed from any lines copied from the session window.
- **Append a <CR>.** A Carriage Return (CR) control character is appended to the end of the characters copied from the session window.

Copying Entire Session Window

You can copy the contents of the current session window to the Windows clipboard, using either the **Edit > Copy Screen** option or **Edit > Snapshot** option.

Tip

Using the Copy Screen option to copy a screen to the clipboard is the same as choosing **Edit > Snapshot** and selecting the clipboard as your destination using the **Edit > Capture To** option. The Copy Screen option, however, allows you to copy a screen to the clipboard regardless of the destination you have set using **Edit > Capture To**.

To copy all printable characters using the **Copy Screen** option:

1. Choose the **Edit > Copy Screen** option. All printable characters displayed in your session window are copied to the Windows clipboard.

To copy all printable characters using the **Edit > Capture To** option:

1. Choose **Edit > Capture To**.
2. Select the clipboard as your destination (you may also select a file or a printer).
3. Choose **Edit > Snapshot**. All printable characters displayed in your session window are copied to the Windows clipboard.

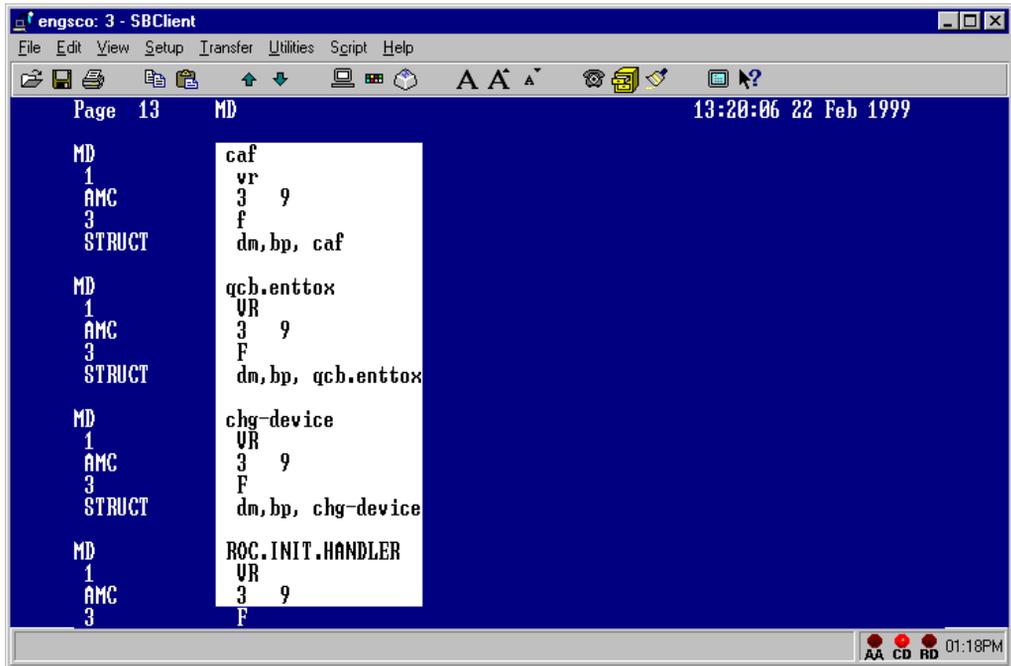
To copy a bitmap image:

1. Press Alt +Print Screen. A bitmap image of the session window is copied to the Windows clipboard.

Copying Selected Text from the Session Window

You can copy selected text from the current session window to the Windows clipboard. You need to first select the text to be copied. This may be a block of text or consecutive lines (ie complete lines of text). You may select the text using the mouse or using the keyboard.

To select a block of text using the mouse:



1. If the **Setup > Miscellaneous > Character Mouse** checkbox is not selected, press the left mouse button and drag the mouse across the block of text you wish to select (only the text you highlight is selected).

If the **Setup > Miscellaneous > Character Mouse** checkbox is selected, select the block of text using any of the following key and mouse combinations:

- Alt + left mouse button
- Ctrl + left mouse button
- Alt + Ctrl + left mouse button

Note

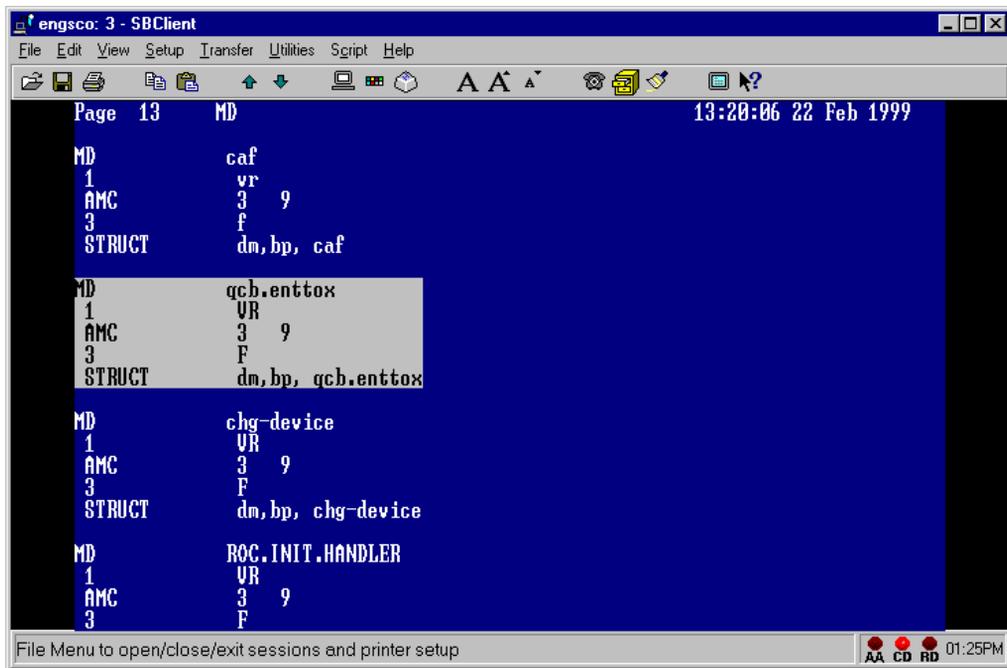
The behavior of the mouse depends on the application running on the host.

To select a block of text using the keyboard:

1. Position the cursor at the start of the selection using Alt + Ctrl + Up Arrow, Down Arrow, Left Arrow and Right Arrow.
2. Begin the selection using Alt + Ctrl + Home.
3. Grow the selection region using Alt + Ctrl + Up Arrow, Down Arrow, Left Arrow and Right Arrow.

You may cancel the selection using Alt + Ctrl + End.

To select consecutive lines of text using the mouse:



1. If the **Setup > Miscellaneous > Character Mouse** checkbox is not selected, press the right mouse button and drag the mouse through the lines of text you wish to select (complete lines of text are selected).

If the **Setup > Miscellaneous > Character Mouse** checkbox is selected, select the block of text using any of the following key and mouse combinations:

- Shift + left mouse button
- Alt + Shift + left mouse button
- Alt + right mouse
- Alt + Shift + right mouse

The behavior of the mouse depends on the application running on the host.

To select consecutive lines of text using the keyboard:

1. Position the cursor at the start of the selection using Alt + Shift + Up Arrow, Down Arrow, Left Arrow and Right Arrow.
2. Begin the selection using Alt + Shift + Home.
3. Grow the selection region using Alt + Shift + Up Arrow, Down Arrow, Left Arrow and Right Arrow.

You may cancel the selection using Alt + Shift + End.

To copy the selected text:

1. Choose **Edit > Copy** or press Alt + Ctrl + Left Arrow. The highlighted text is copied to the Windows clipboard.
2. Choose **Edit > Graphic Copy** to copy the selected text as a Windows bitmap to the Windows clipboard.

Recording Screen Activity

You can use the **Edit > Start Capture** option to record all activity in your session window (both host messages and your responses) and save it to the Windows clipboard (or file or printer). The **Start Capture** option records just the printable characters that the host sends. The destination of you output is shown on the status line.

To record screen activity:

1. Choose the **Edit > Capture To** option.
2. Select the destination: clipboard (file or printer).

3. Select the **Edit > Start Capture** option. SBClient records all the printable characters displayed in the session window from that point forward until the recording is paused or stopped.

The **Edit > Pause** option temporarily pauses the recording. To resume recording choose the **Edit > Start Capture** option again.

4. To completely stop the recording, choose the **Edit > Stop** option. SBClient copies all screen activity it has recorded to the selected destination.

Back Page History

Data displayed in the session window is saved as screen pages. The number of pages can be set to between 0 and 100, using the Back Pages box in the **Miscellaneous Setup** dialog. The default is 10. The higher the number of pages set, the more memory is used.

Use Alt + Page Up to display the previous saved page and Alt + Page Down to display the next saved page. Alt + Home displays the first saved page and Alt + End displays the current page (which will have control of the cursor).

All saved pages are displayed in black on grey to distinguish them from the current page. You can copy data from a saved page in the same way as for the current page. You can scroll back through previous pages to view or copy information from more than one page.

Chapter 5 - Smart Query



Smart Query is a Windows-based tool that allows you to build, save and run report queries for your host computer running a nested relational operating system. The output from your queries can be directed to your screen or printer, or sent to a PC file in Word, Lotus 123 or Excel format.

A screenshot of the 'Smart Query' dialog box. The window has a title bar with 'Smart Query' and a close button. It is divided into several sections: 'File' with 'Dict File' and 'Data File' both set to 'CARS'; 'Select' with the query 'WITH COLOR = "BROWN" OR WITH COLOR = "SILVER"'; 'Sort' with 'BY MODELCODE BY COLOR'; 'Disp Fields' with 'MODELCODE COLOR MILES AVAILABLE'; 'Heading' with 'Car availability by color and miles 'D'; and 'Footing' which is empty. There are 'Options' and 'Output' sections at the bottom. 'Options' includes checkboxes for 'Id Suppress', 'Detail Suppress', 'Header Suppress', and 'Column Header Suppress'. 'Output' includes a dropdown menu set to 'Terminal' and checkboxes for 'DEM To ANSI Conversion', 'Insert Blank Line Before Data', and 'Strip Duplicate Data from Correlative' (which is checked). On the right side, there are buttons for 'Cancel', 'Load...', 'Save...', 'Delete...', 'Help', 'Clear', and 'Run Query'.

You can use Smart Query to:

- recall, modify and run a stored query
- build a new query

To start Smart Query:

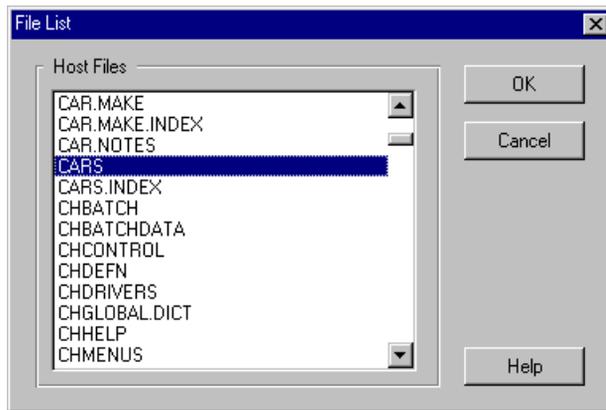
1. Open a session.
2. Log in to an account on your host system.
3. Choose **Utilities > Smart Query**. The Smart Query dialog is displayed.

Building a New Query

You can build a new query either from scratch or based on an existing query.

To select the dictionary and data file you want to access:

1. Click the browse button to the right of the **Dict File** text box (or enter the dictionary file name directly). The dictionary **File List** dialog is displayed:



2. Select the desired dictionary from the **Host Files** list. The selected dictionary name is displayed in the **Dict File** text box, and the name of its associated data file is displayed in the **Data File** text box. You may change the name of the data file if necessary.

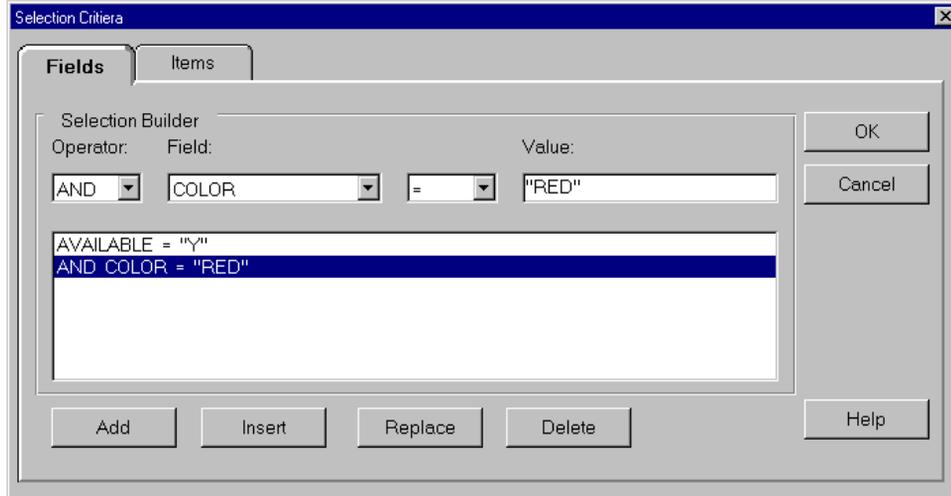
Defining the Selection Criteria

SBClient allows you to define the selection criteria by field content or by explicit record id to accurately pinpoint the records you want.

Click the browse button to the right of the **Select** text box, to display the **Selection Criteria** dialog.

To select records by field content

1. Display the Selection Criteria dialog. Click the Fields tab if the Fields dialog is not displayed.



2. Construct your selection criteria in the Selection Builder boxes.

For each criterion you define, click the Add button. SBClient constructs the criterion and displays it in the next empty line in the list box. There are four parts to a criterion:

- **Operator.** Connects multiple criteria. Operators are AND and OR. No operator is required for the first criterion.
- **Field.** The name of the field on which the selection is based. The field name list originates from the data dictionary you have selected.
- **Operand.** The relation to apply to the field. The options are: equal (=), not equal (#), greater than (>), less than (<), greater than or equal to (>=), less than or equal to (<=).
- **Value.** The value to apply to the relation. The value delimiters (double-quotes) are added before and after the value if you do not enter them. If you specify a value of "?" you will be prompted for the value at runtime. This functionality is only available in Smart Query.

To delete a criterion:

1. Highlight the criterion in the list box.

2. Click the Delete button.

To replace a criterion:

1. Enter the new criterion's operator, field, operand and value.
2. Highlight the criterion in the list box that you want to replace.
3. Click the Replace button.

To insert a criterion:

1. Enter the new criterion's operator, field, operand and value.
2. Highlight the criterion in the list box that you want to insert before.
3. Click the Insert button.

To complete the task:

1. When you have built your selection criteria, click the OK button.

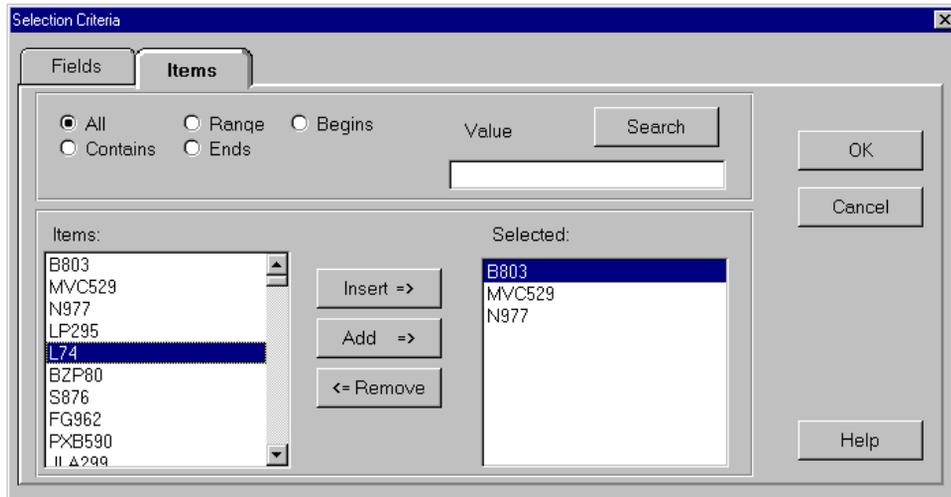
When you have completed this dialog, the selection criteria portion of the query is displayed in the **Select** text box.

Note 

If you enter a selection criteria directly in the **Select** text box, do not enter a BY-EXP selection as SBClient does not support this type of selection.

To select records by explicit record id

1. Display the Selection Criteria dialog. Click the **Items** tab.



2. Select from one of the five mutually exclusive methods of searching your file:
 - All. Search for all records in the file. Ensure the Value text box is clear. Avoid using the All option for files that have large number of records (that is, over 3000) as the list may not be able to display them all.
 - Contains. Search for those records whose record ids contain the value you enter in the Value text box.
 - Range. Search for those records whose record ids are in the range you enter in the Value text box. Separate the lower and upper limits of the range with a semicolon.
 - Ends. Search for those records whose record ids end with the value you enter in the Value text box.
 - Begins. Search for those records whose record ids begin with the value you enter in the Value text box.
3. Click the Search button. SBClient searches your file for all records that match your selection criteria. The record ids are displayed in the Records list.

- For each record you want to include in the explicit selection, highlight the record in the Records list and click the Add button. This adds the record to the end of the Selected list. To insert the record in the list, highlight the record in the Selected list that you want to insert before, and click the Insert button. If the Selected list is left empty then all records are selected.

If you are building a List command, the records will be listed in the sequence in which they appear in the selection list. If you are building a Sort command, the sequence in which they appear in the selection list is irrelevant since they will be sorted into the sequence defined by your sort criteria.

To remove a record from the selection list:

- Highlight the record in the selection list.
- Click the Remove button.

To complete the task:

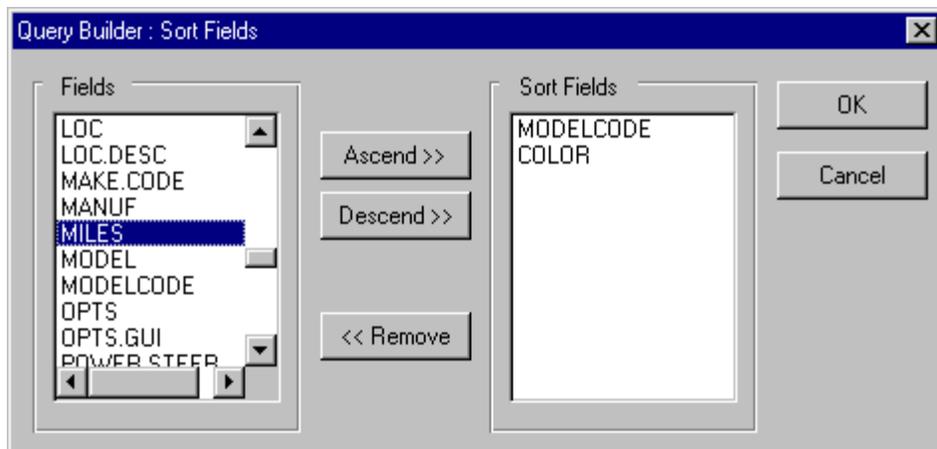
When you have built your explicit selection criteria click the **OK** button.

Defining the Sort Criteria

If you do not define sort criteria, Smart Query uses the **List** command.

To define the sort criteria:

- Click the browse button to the right of the **Sort By** text box. The **Sort Fields** dialog is displayed:



2. Highlight the first sort field in the **Fields** list.
3. Click the **Ascend** button if you want to sort the field in ascending order. Click the **Descend** button if you want to sort in descending order.

The selected field is added to the **Sort Fields** list. A (D) appended to the field name indicates a descending sort. You can sort on more than one field.

To remove a field from the **Sort Fields** list, highlight the field and click the **Remove** button.

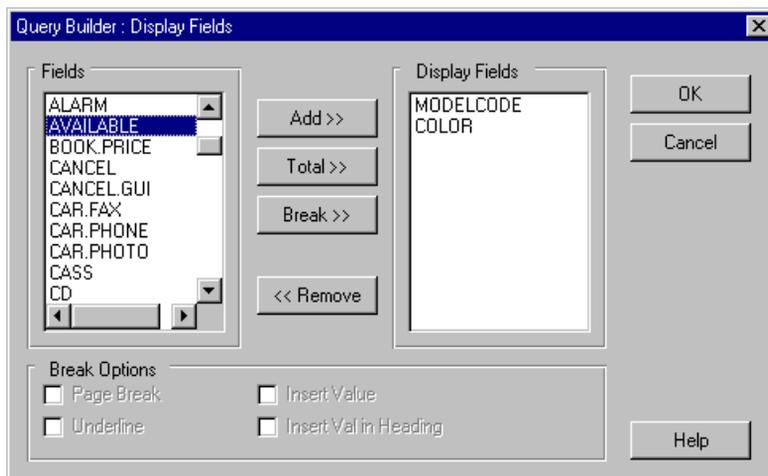
4. Click the **OK** button. The sort criteria portion of the query is built and displayed in the Sort By text box in the **Smart Query** dialog.

If you enter a sort criteria directly in the **Sort By** text box, do not enter a BY-EXP sort as SBClient does not support this type of sort.

Defining the Display Fields

To select fields you want to display:

1. Click the browse button to the right of the **Disp Fields** text box. The **Query Builder > Display Fields** dialog is displayed:



2. For each field you want to display, highlight the field in the Fields list and:
click Add to display the selected field,

-or-

click Total to generate a total for the selected field,

-or-

click the Break button to make this field a break field.

The field name is added to the Display Fields list. A (T) appended to the field name indicates it is a Total field. A (B) appended to the field name indicates it is a Break field.

To assign options to a break field:

1. Highlight the break field in the Display Fields list. The Break Options become active.
2. Select the desired Break Options:
 - **Page Break.** Generates a page break when the contents of the selected field change.
 - **Underline.** Underlines the value displayed for the selected field (before outputting the total).
 - **Insert Value.** Inserts the value in this field in the same line as the total.
 - **Insert Val in Header.** Inserts the value in this field in the header/footer. The Break Value option must be selected in the Header/Footer dialog.

To remove a field from the **Display Fields** list:

1. Highlight the field in the **Display Fields** list.
2. Click the **Remove** button.

To complete the task

1. When you have built your list of display fields, click the **OK** button. The display fields portion of the query is displayed in the **Disp Fields** text box in the **Smart Query** dialog.

Creating Headers and Footers

To create a header or a footer:

1. Click the browse button to the right of the Header or Footer text box, depending on whether you wish to create a header or a footer. The Header/Footer dialog is displayed:

2. Enter your heading or footing in the text box.
3. Select the options you wish to add to the header or footer:
 - **Center.** Aligns the header/footer to the center of the page.
 - **Date.** Inserts the system date in the header/footer.
 - **Break Value.** Inserts the value from the break field in the header/footer.
 - **Line Feed.** Outputs the header/footer on a new line.
 - **Time.** Inserts the system time in the header/footer.
 - **File Name.** Inserts the filename in the header/footer.
4. Click the **OK** button. **Smart Query** builds the header or footer portion of the query and displays it in the **Header** or **Footer** text boxes.

Selecting Additional Options

There are four additional options you can incorporate into your Sort or List command:

- **Id Supp.** Suppresses the output of record ids.
- **Detail Supp.** Suppresses detail output, when used with total or break fields.
- **Header Supp.** Suppresses the output of the header.
- **Col Header Supp.** Suppresses the output of the column headings.

Directing Your Output

Use the Output selection list on the Smart Query dialog to specify the destination for your output. Destinations are:

- **Terminal.** Your PC monitor.
- **Auxiliary Printer.** Your Windows default printer.
- **Host Printer.** The printer connected to your host.
- **Word.** A PC file in Microsoft Word format.

- **GUI.** A window on your PC.
- **Excel.** A PC file in Microsoft Excel format. Additional Excel formats are also provided.
- **123.** A PC file in Lotus 1-2-3 format. Additional 1-2-3 formats are also provided.

Note

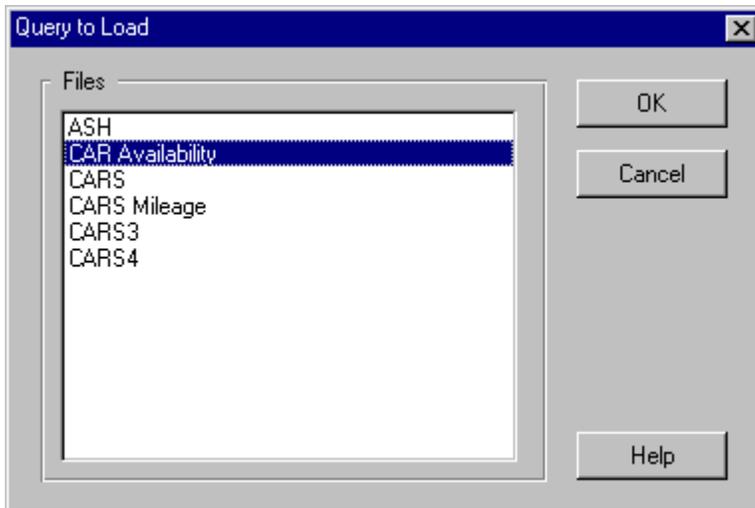
When you direct your output to a PC file, the selected PC application opens a new (untitled) document. Your output from the query is transferred from the host to this newly opened document.

Recalling a Stored Query

You can recall a previously saved query to either run immediately, or use as the basis for building a new query.

To recall a previously saved query:

1. Click the **Load** button. The **Query to Load** dialog is displayed:



2. Highlight the query name you wish to recall.
3. Click the **OK** button. **Smart Query** displays the selected query in the **Smart Query** dialog text boxes. You can:

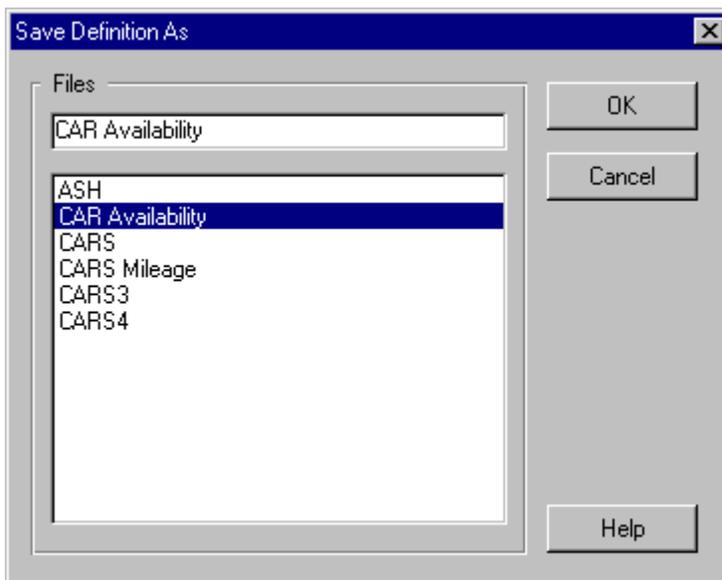
- run the query unchanged
- modify the query and save it as a new query
- modify the query, save it as a new query and run it

Saving a Query

When you have built or modified your query you can save it for future use.

To save your query:

1. Ensure all the parameters are set up correctly and tested.
2. Click the **Save** button. The **Save Definition As** dialog is displayed:



3. Select an existing query name from the **Files** list or enter a new (case-sensitive) query name.
4. Click the **OK** button.

Running a Query

To run a query:

1. Build a new query or use the **Load** button to recall a previously saved query.

Tip 

If you are building a new query, or modifying an existing one, we recommend you use the Save button to save the query before you run it.

2. Click the **Run Query** button on the **Smart Query** dialog. Smart Query converts the query displayed in the Smart Query dialog into a nested relational Sort or List command, passes the generated query statement to the host, which runs the command. The query is directed to the destination specified in the Output box.

Deleting a Query

You can delete an existing query.

To delete a query:

1. Click the Delete button. The Stored Query List dialog is displayed.
2. Highlight the query name you wish to delete.

Click the OK button. A dialog is displayed, asking you to confirm the deletion.

Chapter 6 - Creating Scripts



An SBClient script is like a keystroke macro recorder and player. It can record a series of keystrokes and mouse operations that you perform when communicating with a host, then play them back on request. For example, you could create a script that logs in to a special account on the host system, provides the password, then runs a particular host application program. You could instruct SBClient to run this script when you connect to that host so that the entire process of logging in and running the application is automated.

An SBClient script is stored as a series of trigger sequences and responses. A trigger sequence is normally a message sent from the host to your PC, and a response is what you would enter in response to that host message. When you run a script, SBClient checks each message it receives to see whether it is one of the trigger sequences stored in the script. If it is, SBClient sends its stored response automatically. Otherwise, SBClient returns control to your keyboard so that you can enter your own response manually. This behavior allows you to add new triggers and responses to a script while it is running.

An SBClient script can respond to trigger sequences in any order. Rather than run a script from first entry to last entry (like a normal macro player), it responds to each trigger sequence as it is detected. This allows you to respond to unusual or unforeseen events.

Multiple scripts can be run simultaneously, monitoring the same or other sessions. For example, a login script may detect that you have mail and launch a mail retrieving script, suspending the current login script until the mail script has completed.

There are two ways to create a script:

- write the script from scratch using the script editor
- use SBClient's auto script learner feature, then edit it using the script editor

A script consists of an unlimited number of trigger sequences and responses (local disk space or memory being the only limiting factor).

Trigger Sequences

A trigger sequence may be any of the following:

- a single word
- a single character
- a sentence
- a phrase
- a mouse selection (for example, clicking the right mouse button)
- a mouse drag
- a special GUI event (for example, the opening of a Windows error message dialog)

SBClient scans each host message for a match against the triggers stored in the script you are running. You can apply the following rules to find a match:

- case insensitive match or a case sensitive match
- an exact match or an index match where a match is recognized if the trigger is seen anywhere within the sequence of characters sent from the host
- a match if seen anywhere, anytime or matched only when SBClient believes that the host has paused for user response

Not all trigger sequences need to be used or seen for a successful scripting session.

You can add a response to a trigger sequence that instructs SBClient that the current script has logically run its course. This allows you to automatically terminate the script under specific conditions.

Trigger Responses

The response to a trigger can be anything from characters typed on the keyboard through to DDE execute macro functions (or special script control functions).

When SBClient learns a response, it records each trigger, then converts your response into a line of text. For example, assume the host sends the following message:

Enter User Name:

During auto learn mode, SBClient stores that exact string of characters as the trigger. Now assume you enter the following response and press the Enter key:

USER123

As the response to the trigger, SBClient stores the following string of characters:

USER123{enter}

Printable characters (in this example, USER123) are stored exactly as they are entered. Special keys are converted into a string of characters surrounded by braces. In this example, SBClient converts the Enter key into the character string {enter}.

If the trigger is repeated during auto learn mode, SBClient stores the trigger once only. Your responses are stored with that trigger in the same sequence that you enter them. When you run the script, SBClient uses your first response for the first occurrence of the trigger, your second response for the second occurrence of the trigger, and so forth.

If a trigger sequence is detected more times than there are response sequences, the last available response sequence is used. For example, for the fifth and subsequent occurrence of a trigger sequence that has only four responses stored in the script, the fourth response will be re-used.

Auto learn mode adds a [script_end()] function to the final trigger it learns (that is, when you stop the script). You can edit your script to add this function to the response for other triggers in the script as well. It instructs SBClient to issue the response and then stop the script. This allows you to set up your script so that it can decide when to stop (for example, when a particular error message is received).

You can also edit your script to incorporate more sophisticated script features.

Maximum Hits

A maximum hits value is associated with each trigger sequence. This value identifies the number of times the script responds to the same trigger sequence. After the limit has been reached, the script no longer responds to that trigger sequence. Should a response be required again, it must be done manually via the keyboard or mouse, and auto-learned for future reply. This may be intentional, to allow you to enter a response or choose a menu option, or it may be a diagnostic feature when testing scripts to see that all contingencies have been catered for.

If you set Maximum Hits to 0 (zero) in the Script Editor dialog, your script will respond to the trigger sequence indefinitely until another trigger generates a [script_end()] or [script_exit()] response (or until the communications session is closed). For example, if the trigger sequence is Press Enter to Continue then a single response sequence of {enter} is all that is required, regardless of how many times the trigger sequence occurs.

When you create a script in auto learn mode, SBClient stores a trigger sequence once only. If the trigger is repeated, your responses are stored with that trigger and SBClient automatically updates the Maximum Hits box to reflect the number of responses detected.

For example, the password prompt will usually be sent just once from the host, so you would want to set the Maximum Hits value to 1 for that trigger sequence. After it is detected, the response sequence is sent, and the trigger sequence is then deactivated. However, with trigger sequences that you have set to use multiple response sequences, make sure that the Maximum Hits value is set at least to the number of response sequences you have defined or to 0, otherwise all the response sequences will not be used.

Where you do not know how many times a prompt might appear (for example, pause or press a key...), you should use 0 as the Maximum Hits value.

The more trigger sequences you have active, the greater the potential of slowing the session's response due to the quantity of tests being performed by SBClient, every time characters are received from the host. Therefore, you should deactivate trigger sequences when you know they will not be needed any more.

Auto Script Learner

The easiest way of creating a script is to instruct SBClient to learn the series of triggers and responses. When you select auto learn mode, SBClient creates a script that records each trigger along with your response.

You can use the Script Replacement Strings box in the Script Setup dialog to create parameters for your script responses. You should do this before you use auto learn mode. You should still respond to the host triggers manually but when the auto learn script is built and saved, it will then substitute your literal responses with the parameters you have entered in the Script Setup dialog.

To use auto learn mode to create a script:

1. Choose **Script > Auto Learn Script**. The Auto Script Learner dialog is displayed:



2. Select the **Diagnostics** checkbox if you want SBClient to display diagnostic messages while the session is in auto learn mode.

3. Select the Initial CR checkbox if you want SBClient to send a CR (Carriage Return) to the host when a connection is established. After a connection has been established, some host computers will not respond with a login request until the user presses Enter to generate a CR character. This option generates the CR automatically on connection.
4. Enter the script name in the **Enter Script Name** text box.
5. Click the **OK** button. If you have specified an existing script name, SBClient displays a confirmation dialog. You can instruct SBClient to:
 - replace (overwrite) the existing script with the new responses, or
 - append the new responses to the existing script.

SBClient returns you to the session window. The following message is displayed on the status line while the session is in auto learn mode:

```
Learning Script scriptname
```

6. Enter your responses to the host's messages.

SBClient records both the host's messages and your responses. Each message is stored as a trigger and your response is stored as the trigger's response.

To temporarily pause auto learn mode while you perform an operation that you do not want SBClient to record, choose **Script > Pause Script**. After pausing, you can restart auto learn mode by choosing **Script > Resume Script**.

7. When you have completed all the responses you wish SBClient to record, choose **Script > Stop Script**.

You can now run your script automatically from the session window.

Creating Login and Logout Scripts

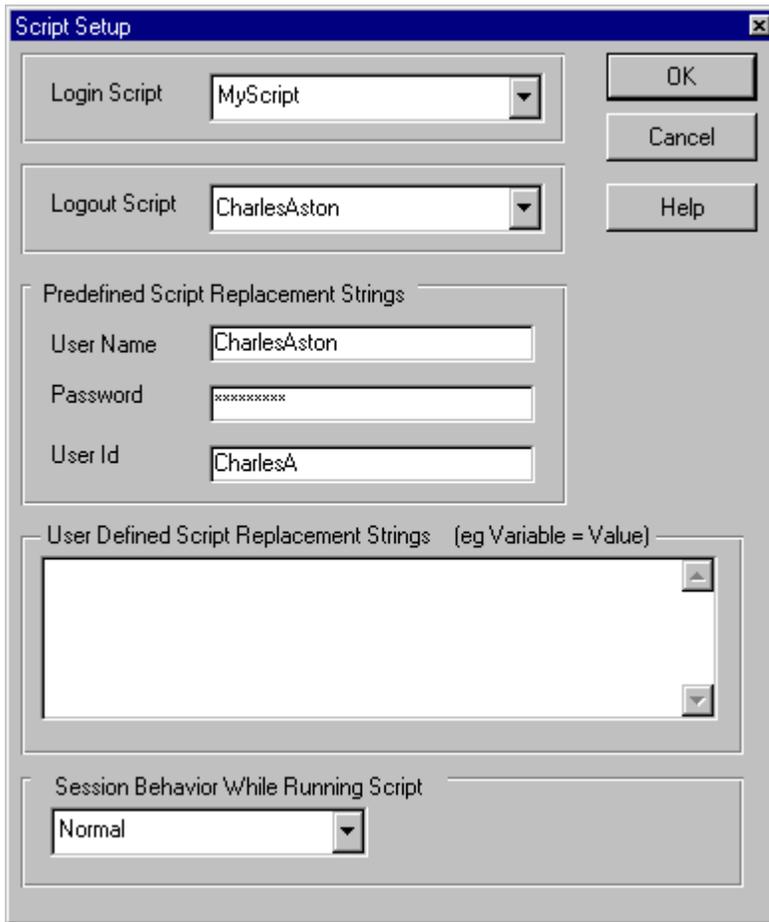
A login script is one that SBClient runs when it establishes a connection to a host. It usually includes the user's name and password responses that they would enter when logging in to an account on the host system. It may also contain a user id and replacement strings required for a particular application program on the host.

A logout script is one that SBClient runs when the user exits the session. Its aim is to take you completely out of the system before closing the session. This will ensure that the database integrity is maintained by not terminating incorrectly.

Users can create a login script for their personal use. An administrator, however, can create generic login scripts that can be used by all users who are performing a common login procedure. Generic login scripts can be created in the administrator account only. They can be installed in a user account using the **Utilities > Configuration Sharing > Import Configuration** option.

To create a login script:

1. Choose **Setup > Script**. The **Script Setup** dialog is displayed.



2. Select **<No Script>** from the **Login Script** list. This is necessary, as otherwise any existing login script would be run when you logged in and you would not have a chance to create a new one.
3. Enter your user name in the **User Name** text box and your password in the **Password** text box. This can also be done after you have created your login script, but you will have to edit the

script, replacing literal responses into functions that refer to script parameters instead. Enter any user defined script replacement strings.

4. Select **Normal** from the session behavior while running the script.
5. Close the **Script Setup** dialog.
6. Open a session (but do not respond to the host's first message).
7. Choose **Script > Auto Learn Script** to record your responses.
8. Follow your normal procedure for logging in to the host system.
9. SBClient records your responses. Your responses to the login id and password requests will be replaced in the script by the terms [user_name()] and [user_password()] respectively (if you have matching entries in the **User Name** and **Password** text boxes in the **Script Setup** dialog). This ensures that, in response to the host's user name and password requests, SBClient will respond with the contents of the **Script Setup** dialog.
10. When you have completed all the responses you wish SBClient to record, choose **Script > Stop Script**. You now have a script that can be used as a login script.
11. Choose **Setup > Script** again.
12. Select your new login script name from the **Login Script** list.
13. Enter your user name and password, (and replacement strings if required) if you have not done so already.
14. Close the dialog.
15. If necessary, edit the script so that any literal responses now refer to your script parameters.
16. Save your session configuration by selecting either **File > Save** or **File > Save As**.

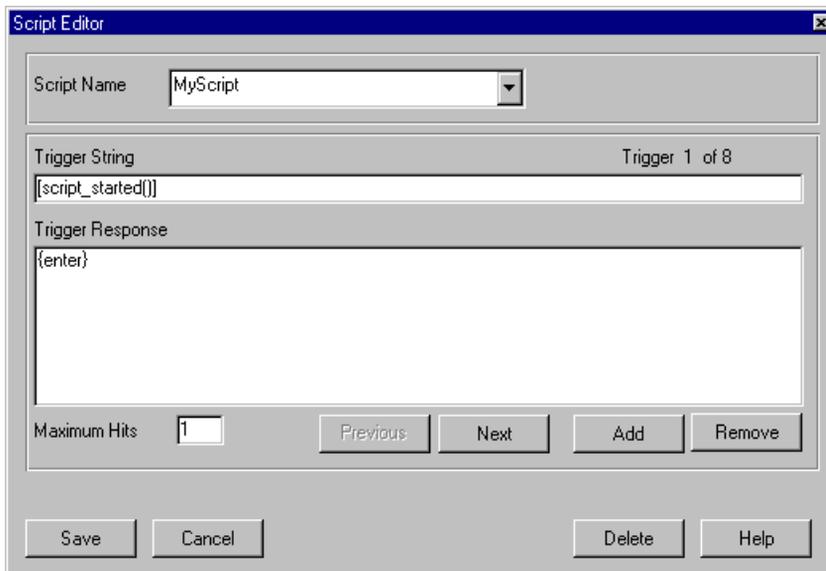
Next time you open a session, your login script is run when a connection is established to the host. If another user specifies this script in their Script Setup dialog, SBClient responds to the host's user name and password requests with the contents of that user's User Name and Password text boxes (from the Script dialog).

Editing Scripts

Since each trigger and response is stored as a character string, SBClient allows you to edit the script to modify the response, or add more sophistication to the script itself. See for details of the script syntax.

To edit a script:

1. Open a session window.
2. Choose **Script > Edit Script**. The **Script Editor** dialog is displayed:



3. Select the script you want to edit from the **Script Name** list. SBClient displays the first trigger and its associated response.
4. To view your trigger responses use the **Previous** button and the **Next** button.
5. To add a new trigger to the script, click the **Add** button. The order of triggers is unimportant (only the order of responses for a given trigger is important).

SBClient clears the **Trigger String** and **Trigger Response** text boxes. You must enter the new trigger and its response as text that conforms to the script syntax.

6. To change an existing trigger or its response, display the trigger and make the appropriate change to the text.
7. To delete a trigger response pair, select the response and click the **Remove** button.
8. When you have completed your edits, click the **Save** button.

Adding Extra Responses to the Same Trigger

If the trigger is repeated during auto learn mode, SBClient stores the trigger once only. Your responses are stored with that trigger in the same sequence that you enter them. When you run the script, SBClient uses your first response for the first occurrence of the trigger, your second response for the second occurrence of the trigger, and so forth.

You can add extra responses to those already stored for any trigger.

To add extra responses to a trigger sequence:

1. Choose **Script > Edit Script**.
2. Select the appropriate scripts if not already selected.
3. Display the appropriate trigger.
4. Enter the extra responses on separate text lines in the **Trigger Response** text box. The sequence in which they appear in the text box is the sequence SBClient will use the responses when the trigger is detected.
5. Enter the new number of responses in the **Maximum Hits** text box.
6. When you have completed your edits, click the **Save** button.

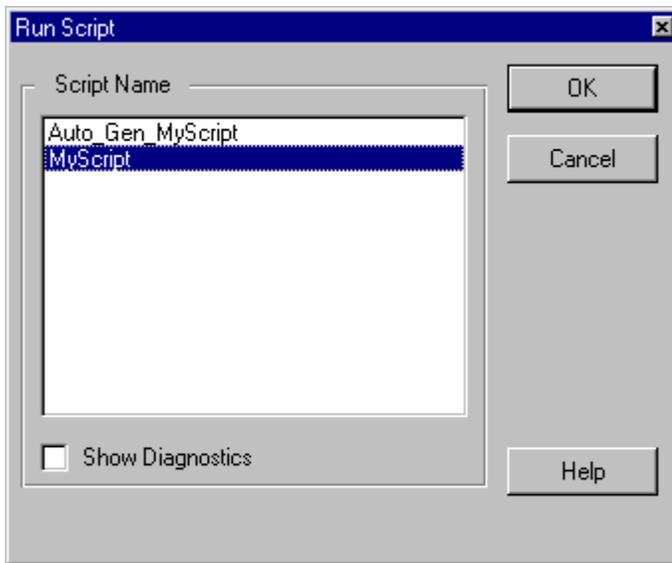
Running Scripts

There are two ways a script can be run after it has been created:

- Automatically, as a login script. A login script is run automatically when SBClient establishes a connection to a host system (or has received carrier from a modem connection).
- Manually, after you have opened a session and connected to a host system.

To manually run a script during a session:

1. Open a session. You can run a script any time after SBClient has established a connection and opened the session window. Whether or not you need to log in to an account before you run a particular script depends upon the script itself.
2. Choose **Script > Run Script**. The **Run Script** dialog is displayed:



3. Highlight the name of the script you wish to run.
4. Select the **Show Diagnostics** checkbox if you want SBClient to display diagnostic messages while the script is running.
5. Click the **OK** button. SBClient runs the selected script. The following message is displayed on the status line of the session window:

Running Script *scriptname*

The script continues running until the response it makes to a trigger includes a `script_end` instruction, or all triggers have been seen or you choose **Script > Stop Script**.

If you want to temporarily pause the script, choose **Script > Pause Script**. After pausing, you can restart the script by choosing **Script > Resume Script**.

Learning Responses While a Script is Running

While a script is running, SBClient may detect conditions that are not catered for in the script. When this occurs, SBClient returns control to the keyboard, allowing you to enter your own responses before responding. You can instruct SBClient to add the triggers and your responses to the running script. You can add a single trigger and response to a running script, or multiple triggers and responses.

To add a trigger and response to a script that is running:

1. When SBClient returns control to the keyboard (with the script still running), choose **Script > Learn Next Response**. The following message is displayed on the status line of the session window:

```
Learning Script scriptname
```

2. Enter your response. When you press Enter or use the mouse in response to the host, SBClient resumes running the script.

If the response you enter is for a trigger already in the script, SBClient adds your response to the end of the stored responses for that trigger. If the response is for a trigger not in the script, SBClient adds both the trigger and your response.

To add multiple triggers and responses to a script that is running:

1. When SBClient returns control to the keyboard (with the script still running), choose **Script > Auto Learn Script**.
2. Recall the name of the script that is already running. Ensure the **Initial CR** checkbox is not selected and press **OK** button.
3. Choose the **Append** button on the **Script Already Exists** dialog.
4. Enter you responses to any triggers that are not catered for in the running script.
5. Choose **Script > Stop Script** to finish you auto learn append. This can be done before or after the running script has completed.
6. If necessary, choose **Script > Edit Script** and remove erroneous [script_end()] responses after your last appended auto learn response, or remove it from its original position.

Logout Scripts

It is possible to assign a script to be run when the user tries to close SBClient. If the 'Logout Script' is assigned a value in the **Setup Script** dialog, then whenever the user clicks on the **Close** button on the SBClient screen or selects **File > Close Session** or **File > Exit All** from the SBClient menu or System Tray, the script will be run.

This will only be run if the session is connected to a host. It is important that the script itself then ultimately shuts down SBClient otherwise the script will be run but SBClient will not be shut down. A typical way to shut down the session from the script would be to assign [exit_session(265)] to a particular trigger. The exit_session parameter now has a possible value of 256 that tells SBClient that it really must shut down and not simply run the shut down script. The number 264 in the above e.g. is made up of $256 + 1 + 8 = 265$ (the 1 and the 8 control the shut down confirmation dialogs see the online help on exit_session for details.).

Note the only way to shut down SBClient, if a shutdown script is assigned yet the required trigger for the exit_session response never occurs, is to go to the "Script Setup" dialog and select <No Script> for the "Shutdown Script" parameter and then close SBClient.

This logout script can be used as a way to suppress the Exit confirmation dialogs by setting up a logout script that just does an [exit_session(265)] as a response to the script_started prompt.

Creating Advanced Scripts

This section explains how to use the advanced features available in SBClient scripts. It explains the use and syntax of commands that can be incorporated as trigger responses in a script.

To build a complex script, you should first create a script that records the basic dialog between your PC and the host. You can then edit that script to add the extra commands described in this topic.

The syntax of a response string conforms to the DDE execute macro syntax used by most Windows DDE server applications.

This syntax represents normal keyboard characters as plain text with special keys represented by tokens surrounded by {}. For example, user name{enter}.

Functions to perform special tasks are surrounded by [] and their internal arguments are surrounded by (). For example, [char_mouse(1,10,5)].

The script functions in this topic may be used as either response strings to script trigger events or generated internally for use as trigger sequences. These sequences can only be used in scripts. You may also use general purpose macro functions as response sequences.

Script Functions

Script functions are:

script_case_sensitive affects how trigger sequences are matched.

script_clear de-activates a script on the script stack.

script_end terminates the current script.

script_exit terminates all scripts on the script stack.

script_indexed affects how trigger sequences are matched.

script_max_retries can end the script due to inactivity.

script_monitor controls display of the diagnostic script window.

script_retrigger_timeout adjusts inactivity retry delays.

script_run stacks current script and runs another.

script_started is an internal script trigger to which you can respond.

script_trigger_delay detects when the host has paused for a user response.

[script_case_sensitive(*value*)]

Specifies the case sensitivity of comparisons made between script trigger strings and the incoming characters and events from the session. You should only set this value to TRUE if you wish to distinguish similar prompts by case differences.

value

Options are:

- TRUE if the trigger string comparisons to incoming characters or events are to be case sensitive.
- FALSE if trigger string comparisons are to be case insensitive (the default, for the first script run).

Example

```
[script_case_sensitive(TRUE)]
```

See also

script_indexed, script_trigger_delay

[script_clear(*scriptname*)]

Clears (deactivates) a previously suspended script from the nested script stack.

scriptname

The name of the script to be deactivated. If the name is not found in the script stack, this function will be ignored. *scriptname* is skipped when popped from the stack.

Example

A login script being run may detect NEW MAIL and launch the getmail script to process the mail. The login script launches the getmail script via a [script_run("getmail")] response. This suspends the initial login script you are running and pushes the current state of the login script onto the

script stack. While running the getmail script you then launch an uploadfile script to send your latest mail responses, which similarly pushes the getmail script onto the script stack. While running the uploadfile script, SBClient detects the error message "Error Out Of Disk Space" from the host. To effect a clean and tidy abort due to this error, you can disable the remaining mail script processing with `[script_clear("getmail")]` and stop the uploadfile script with `[script_end()]`. This will pop you back into the initial login script, which will then process the normal disconnect script procedures.

Trigger string:

Error Out Of Disk Space

Trigger response:

```
[script_clear("getmail")][script_end()]
```

See also

`script_end`, `script_exit`, `script_run`

[script_end()]

Ends the current script, then pops the script stack to enable any previously suspended script. Just before ending the current script, it internally issues a trigger string of "`[script_end()]`" to allow this script to perform any final clean-up responses before the script is terminated.

See also

`script_clear`, `script_exit`, `script_run`

[script_exit()]

This function is similar to `script_end()` except that it ends all previously running script levels. That is, it pops all scripts from the script stack and terminates the script session completely. Just before ending the script, `script_exit()` issues an internal trigger string of "`[script_exit()]`" to allow the script to perform any final clean-up responses before the script session is terminated. Note that a different clean up process is possible per script level.

See also

`script_clear`, `script_end`, `script_run`

[**script_indexed**(*value*)]

Specifies how comparisons are made between script trigger strings and the incoming characters and events from the session. The default for the first script run is TRUE, for indexed comparisons to be made. Indexed means that the script trigger string may be contained anywhere within the last line of characters received from the host. This allows the script trigger string to be smaller and less precise, as well as providing flexibility. In this way, certain lead-in spaces or words can be filtered out by not including them in the script trigger sequence. You may also elect not to record trailing punctuation characters in the script trigger string. This may allow a given script to work for multiple user names, if recorded user names are removed from the trigger strings, leaving the script to trigger on the rest of a prompt.

value

Options are:

- TRUE if the trigger string comparison to incoming characters or events is to be indexed.
- FALSE if trigger string comparisons are not to be indexed.

Example

A login script is being run that detects a string (eg Winnie The Poo's Password:) from the host. If the login script was using indexed comparisons, a trigger string of Password would satisfy detection and the appropriate response dispatched: [send_password()]. By not including the logon user id in the script trigger, this script may still work for another user id logging onto the same host computer. Furthermore, if you are using case insensitive comparisons, a trigger string of password or PASSWORD would also suffice. For example, the script will still work even if the case of the login prompt is changed due to an upgrade to the operating system.

Trigger string:

```
[script_started()]
```

Trigger response:

```
[script_indexed(TRUE)]
```

Trigger string:

```
password
```

Trigger response:

[send_password()]

See also

script_case_sensitive, script_trigger_delay, user_name, user_password

[script_max_retries(*value*)]

Enables automatic script ending upon a successive number of trigger string timeouts. Note that timeout does not occur if the script has been paused, either by the user via the user interface or when doing file transfers etc.

If no new characters are received from the host for a value number of [script_retrigger_timeout()]seconds, then the [script_end()] sequence will be initiated, ending this script and popping any previous script from the script stack.

value

The number of consecutive trigger re-tries attempted before this script is ended. If value is set to 0, automatic script ending on timeout will be disabled.

See also

script_end, script_retrigger_timeout

[script_monitor(*value*)]

Creates a script monitoring window of value lines deep. It contains a scrolling stack of trigger strings and associated responses sent or learnt for diagnostic purposes. You can turn this window on and off as required within a script.

value

The depth (in lines) of the script monitoring window. If value is set to 0 (the default) the monitoring window is not displayed.

[script_retrigger_timeout(*value*)]

Sets the inactivity re-trigger delay.

For example, the host may stop at a menu item and by the nature of the response sequences sent (for example, initiating a file transfer) the host may not redisplay its prompt string. By retriggering, SBClient re-scans the last host sequence for trigger sequences if there has been no activity from the host. It also allows newly invoked scripts to look again at the last host sequence in case it was started just after their first trigger string had arrived. In other words, it allows a trigger sequence in one script to run another; the new script can be triggered from the same prompt that caused it to be called and take appropriate action.

value

The number of seconds of inactivity from the host after which the last seen host sequence is retested for trigger sequences. If value is set to 0, no re-testing will occur. The default value is 3 seconds.

See also

`script_max_retries`, `script_run`

[`script_run(scriptname)`]

Suspends the current script and saves its state by pushing it onto the script stack. It then runs the `scriptname` script.

Note that the first trigger sequence seen by the newly run script will be [`script_started()`]. If no more new host sequences are received by SBClient and the function [`script_retrigger_timeout()`] has a value, the last host sequence received will be tested again by the new script if this timeout value elapses.

Upon completion of the `scriptname` script, the previous script will be popped from the script stack and continue running.

If the script name is suffixed with `-Dn` or `-dn` or `(Dn`, a diagnostic script monitoring window will be displayed during the running of the script. All received host sequences will be displayed in this window along with a scrolling stack of user responses sent. `n` represents the window depth, which may be specified to increase it from its default size of 5 lines.

scriptname

The name of the script to run. You can suffix the name with `-Dn` or `-dn` or `(Dn` to display a diagnostic window.

See also

`script_end`, `script_retrigger_timeout`, `script_started`, `script_monitor`

From the user interface, a user can run other scripts, each having its own script stack. Up to 100 suspended scripts can be stored on the script stack.

[`script_started()`]

This is not a response function but is sent internally as a trigger string to test against when a script is started. This is provided as a trigger event so you can respond by setting string comparison method options, timeout values, max-trigger values or just sending a wake-up sequence such as {enter} to the host.

See also

`script_end`, `script_run`

[`script_trigger_delay(value)`]

Detects when the host has paused for user response. Incoming host data is not looked at nor tested until value seconds has elapsed without any new sequences being received from the host, then it will test the last line received from the host. This is a good way of preventing the script from responding prematurely to complex sequences such as large bodies of text. The value should be kept small for response performance but may be increased for reliability. If the value is set to 0, scripts can trigger on any passing text or event if required. In this mode, a script can read a screen as it is being drawn rather than just looking at the last prompt displayed.

value

The number of seconds to wait without receiving sequences from the host. This allows you to detect when the host is waiting for a user response. The default value is 0.5 seconds.

Example

After logging in and responding to the user id and password, you could add to the password response sequence: [`script_trigger_delay(0)`]. This would allow SBClient to detect, for example, a YOU HAVE NEW MAIL message from the host before presenting the command prompt. Upon detecting mail or the command prompt, you should use [`script_trigger_delay(0.5)`] to look at prompts only again.

See also

`script_case_timeout`, `script_case_sensitive`, `script_indexed`

Macro Syntax

This section describes the syntax of SBClient internal macros. These can be used to send key-strokes within a session, and can include edit keys, control sequences, methods and functions.

Edit keys within a macro are represented by one or more characters. For example, a represents the character "a"; abc represents the characters "abc".

To specify the following special characters, include them within curly braces: plus (+), caret (^), percent sign (%), left square brace ([) and right square brace (]). For example, {+} represents the plus sign. To specify an actual left curly brace ({) or right curly brace (}), use {{} and {}} respectively.

To specify keys that represent actions (such as Enter and Tab) rather than characters, use the following codes:

| Key | Code | Key | Code |
|------------|-------------------------------------|------------|--------------|
| Backspace | {backspace} or {bs} or {bksp} | Break | {break} |
| Caps Lock | {capslock} | Clear | {clear} |
| Del | {delete} or {del} | Down Arrow | {down} |
| End | {end} | Enter | {enter} or ~ |
| Esc | {escape} or {esc} | Help | {help} |
| Home | {home} | Insert | {insert} |
| Left Arrow | {left} | Num Lock | {numlock} |
| Page Down | {pgdn} | Page Up | {pgup} |

Edit Keys

| Key | Code | Key | Code |
|--------------|---------|-------------|---------|
| Print Screen | {prtsc} | Right Arrow | {right} |
| Up Arrow | {up} | F1 | {f1} |
| F2 | {f2} | F3 | {f3} |
| F4 | {f4} | F5 | {f5} |
| F6 | {f6} | F7 | {f7} |
| F8 | {f8} | F9 | {f9} |
| F10 | {f10} | F11 | {f11} |
| F12 | {f12} | F13 | {f13} |
| F14 | {f14} | F15 | {f15} |

Edit Keys

To specify a key combination that includes one of the following keys: Shift, Ctrl and Alt, precede the regular key code with +, ^ and % respectively. To specify that Shift, Ctrl and/or Alt should be held down while several keys are pressed, enclose the keys in parentheses.

For example, to hold down the Shift key while pressing E, use +E. To hold down the Shift key while pressing E then C, use +(EC). To hold down Shift while pressing E, followed by C without the Shift key, use +EC.

To specify repeating keys, use the form {key number} where there is always a space between key and number. For example, {left 42} means press the left arrow key 42 times. {z 10} means press the character z 10 times.

Control sequences

Control sequences within a macro are identified by their surrounding quotation marks.

The following example (on Windows NT4) takes the currently selected text, copies it to the clipboard then runs Notepad and pastes the selected record there:

```
"%EC^{esc}RNOTEPAD{Enter}%EP"
```

where:

- %EC calls the Edit menu and chooses Copy
- ^{esc}R calls the Start menu and selects the Run option
- NOTEPAD {Enter} runs the Notepad application
- %EP chooses the Edit and Paste menu option to paste the text

Methods

Methods within a macro are identified by their surrounding angled brackets. A method may be invoked using the syntax:

```
[objectname<method>]
```

The only two methods currently supported by SBClient are <EXIT> and <FORM_CLEAR>, which relate to the form object. They would be specified as:

```
[form<EXIT>]
```

-and-

```
[form<FORM_CLEAR>]
```

The <EXIT> method is used to order the destruction of the form. The <FORM_CLEAR> method will clear all the dynamic objects on the form (that is, all those objects with the dynamic property set to TRUE). This is useful when a character window may have been cleared by printing spaces (instead of @(-1)).

Functions

Functions within a macro are identified by their surrounding square braces:

```
[functionname(parameters)]
```

where functionname is a an internal SBClient or Data/C++ function.

For example, the following macro string calls Microsoft Word:

```
[os_call('&word')]
```

Internal macro functions may be called from toolbar buttons, scripting response sequences, DDE server execute sequences, host library macro sequences or from any GUI object that can execute macro sequences. These macro functions are evaluated, and depending on the function may: substitute themselves with the resultant text to be sent as keystrokes, or send their result directly to a session host or just perform an external action.

Internal Macro Functions

Internal macro functions are:

- `char_mouse` sends character-mode mouse button actions to the host.
- `com` generates an external COM event to the program that created SBClient using the SBCOM interface.
- `create_session` creates another session.
- `destroy_session` closes this or another session.
- `exit_session` closes the current session and responds to close dialogs automatically.
- `file_transfer` initiates a file transfer in either direction.
- `gui_respond` sends a GUI mouse or keyboard event to the host.
- `keep_session_minimized` toggles the minimized state of a session window.
- `os_call` calls a local DOS or Windows application or command.
- `play_sound` plays a Windows .WAV sound file.
- `run_command` runs any Data/C++ VOC command.
- `run_menu_opt` runs an SBClient menu option.
- `send_eventstr` sends a GUI event sequence internally to SBClient.
- `send_string` sends characters to an identified non-GUI session without the need for the screen to take focus.
- `sendkey` sends keystrokes to the window that currently has input focus.
- `set_attribute` modifies GUI object parameters.

- `suppress_session_mode` toggles the drawable state of a session window.
- `user_data` sends saved data associated with a given keyword.
- `user_id` sends the user id saved in session configuration.
- `user_name` sends the user name saved in session configuration.
- `user_password` sends the user password saved in session configuration.
- `user_script` runs an external Data/C++ subroutine or interpreted script.

char_mouse

[`char_mouse`(button, col, row)]

Sends a character mode mouse sequence to the host. This is a non-GUI mouse action where the host supports terminals that send mouse click actions.

button

A number representing the mouse buttons pressed, if any. Valid values are:

- | | |
|---|-------------------------------------|
| 0 | all buttons up |
| 1 | left button down |
| 2 | right button down |
| 3 | left and right buttons down |
| 4 | middle button down |
| 5 | left and middle buttons down |
| 6 | middle and right buttons down |
| 7 | left, middle and right buttons down |

col

The character column coordinate number on a terminal screen where the mouse click occurred. The first valid column number is the left-most column number 0.

row

The character row coordinate number on a terminal screen where the mouse click occurred. The first valid row is the top row number 0.

Note 

Mouse events usually exist in pairs. A typical mouse sequence consists of a button down sequence followed by a button release sequence.

Example

```
[char_mouse(1, 10, 5)][char_mouse(0, 10, 5)]
```

See also

`gui_respond`, `send_eventstr`, `sendkey`, `set_attribute`, `user_id`, `user_name`, `user_password`

create_session

```
[create_session(sessionname)]
```

Creates a session using a saved session configuration. Note: this can be executed via the DDE "SBClient" , "System" topic.

sessionname

The name of a saved session configuration.

com

```
[com(data)]
```

Only applicable when using SBClient's COM interface. This sends an external Windows event to the program that created SBClient through the SBCOM interface.

data

Any user data that you wish to send. For further information see the SBCOM interface documentation in the online help.

destroy_session

[destroy_session(exitmethod,[sessionname])]

Closes and exits the current session, all sessions or a specific session by name. Note: this can be executed via DDE "SBClient" , "System" topic.

exitmethod

Valid values are:

- 0 full confirmation required
- 1 say NO to save configuration changes
- 2 say YES to save configuration changes
- 4 say NO to exit session dialog or reconnect
- 8 say YES to exit session dialog
- 16 say NEW SESSION to exit session dialog
- 32 say NO to exit SBClient dialog
- 64 say YES to exit SBClient dialog
- 128 will close ALL sessions
- 256 will close the session even if a logout script is used.



The exitmethod can be a combination of methods by adding the value of these together e.g. 129 (a combination of 128 + 1) is valid.

sessionname

The optional name of the session to close.

exit_session

[exit_session(exitmethod)]

Exits the current session or all sessions, controlling the appearance of confirmation dialogs.

exitmethod

Valid values are:

- 0 full confirmation required
- 1 say NO to save configuration changes
- 2 say YES to save configuration changes
- 4 say NO to exit session dialog or reconnect
- 8 say YES to exit session dialog
- 16 say NEW SESSION to exit session dialog
- 32 say NO to exit SBClient dialog
- 64 say YES to exit SBClient dialog
- 128 will close ALL sessions
- 256 will close the session even if a logout script is used.

Tip

The `exitmethod` can be a combination of methods by adding the value of these together e.g. 129 (a combination of 128 + 1) is valid.

file_transfer

[`file_transfer(filename, direction, method)`]

Initiates a file transfer.

filename

The pathname of the file to transfer.

direction

This can be UP, SEND, UPLOAD or PUT for uploading to a host, or anything else to download from the host.

method

The file transfer protocol to use for the transfer.

gui_respond

[gui_respond(string)]

Sends a GUI sequence to the host via SBClient's Remote Object Call transport layer. The response is usually one that has been captured by the Auto Script Learner. Although you may enter your own string sequence here it is better to recall one that has been intercepted via the Auto Script Learner.

Example

The following trigger response selects the LOGTO process menu option after the menu container "SBWIN_#_0" has been given GUI input focus.

The # represents an object instance handle number, and the actual number is extracted and kept by the trigger matching process. Its value is reapplied by substituting any # sequence in the response string with received trigger string value.

Trigger string:

GUIinput SBWIN_#_0*containerclass

Trigger response:

```
[gui_respond("\251SBWIN_#_0\251\251":AM:"menu;P*LOGTO*")]
```

See also

char_mouse, send_eventstr, sendkey, set_attribute, user_id, user_name, user_password

keep_session_minimized

[keep_session_minimized(state)]

Toggles the minimized state of a session window.

state

If state is 1, the session window is kept minimized. If the session is also in GUI mode, the drawable property of the session becomes FALSE (that is, an icon is not displayed for the minimized session window). See the -M command line option in Command Line Options.

os_call

[os_call(command)]

Calls a DOS or Windows application or command.

command

The command to be called. If command is not specified, a command shell is invoked; that is, you are placed at the underlying OS prompt, from which you can issue the appropriate commands. To exit the shell, issue the appropriate exit command (for example, EXIT for DOS).

See also

run_command**play_sound**

[play_sound(soundname, mode)]

Plays a Windows .WAV format sound file. This feature assumes that your Windows setup supports the playing of .WAV files.

soundname

Either the file name of a .WAV file with a path and .WAV extension, or the name of a [SOUND] event listed in your WIN.INI file.

mode

A number representing how the sound is played. Valid values are 0 to 31, where:

0 or SND_SYNC. Play synchronously (the default).

1 or SND_ASYNC. Play asynchronously.

2 or SND_NODEFAULT. Do not use default sound if soundname is not found.

4 or SND_MEMORY. Point to a memory file (not a valid option).

8 or SND_LOOP. Loop the sound until this function is called again.

16 or SND_NOSTOP. Do not stop any currently playing sound.

Example

```
[play_sound(tada.wav, 1)]
```

```
[play_sound(tada.wav, SND_ASYNC + SND_LOOP)]
```

run_command

```
[run_command(command, capt)]
```

Calls a Data/C++ command from the VOC in a new environment.

command

The command to be called.

capt

Suppresses output.

See also

`os_call`

run_menu_opt

```
[run_menu_opt(menucommand, args)]
```

Runs an SBClient menu option, optionally passing it arguments.

menucommand

This can be the literal menu text of a menu record as seen in the SBClient menu tree. If menu text is used, a case sensitive search is done to find it in the menu structure. If this fails, '...' are filtered out and a case insensitive search is done to identify the menu record to run.

args

The arguments to be passed to the option.

menucommand can also be passed as the name of the internal function to call.

Example

```
[run_menu_opt("Save Desktop", "")]
```

```
[run_menu_opt("file manager", "")]
```

send_eventstr

```
[send_eventstr(string)]
```

Sends a GUI event sequence internally to SBClient to manipulate objects in SBClient's user interface or in another application with a GUI interface. The event sequence is usually one that has been captured by the Auto Script Learner.

Example

The following trigger response enters the requested user id directly into the window's editclass object, then sends a keystroke event directly to this object as if the user had given this object focus and pressed the Enter key on the keyboard. This event will trigger SBClient to fetch what is in the editclass object and send it to the host.

The # represents an object instance handle number, and the actual number is extracted and kept by the trigger matching process. Its value is reapplied by substituting any # sequence in the response string with received trigger string value.

Trigger string:

```
GUIget SBWIN_#_0*sbloginclass<userid>
```

Trigger response:

```
[set_attribute("SBWIN_#_0","string","sb")][send_eventstr("SBWIN_#_0","special;SBWIN_#_0;13;0;0;0")]
```

See also

char_mouse, gui_respond, sendkey, set_attribute, user_id, user_name, user_password

send_string

```
[send_string(sessionname, string)]
```

Sends characters to an identified non-GUI session without the need for the screen to take focus.

sessionname

The SBClient session name.

It is recommended that you do not have multiple sessions with the same name. If you do have multiple sessions with the same name you can target a specific session by specifying a number in the second attribute of sessionname, for example, if sessionname is Unidata on Minotaur":AM:3, this targets the third session of Unidata on Minotaur that has not been destroyed.

Note that the number may not necessarily be the same as the number shown in the title of the session. Consider the following example:

If you create 3 instances of the same session, the title of each session will have a sequential number next to it. For example, if the session name is Test, the session titles for the 3 instances are Test:1, Test:2, and Test:3. These numbers are generated in sequence as you create new sessions and do not reset when you destroy a session. For example, if you destroyed Test:1 and created another Test session, the new session title will be Test:4. If you were to use the send_string macro with sessionname set to Test:AM:3, SBClient sends the string to Test:4, as it is the third session remaining. This way of numbering sessions is based on the way Microsoft numbers documents, for example, in Word.

string

The string to send.

Example

The examples below show 2 methods of calling this macro.

- Method 1 - using the SBClient host library:

```
CALL TU.MACRO(APP.MACRO, '[send_string("UDT on Doom", "hello\013")]', ")
```
- Method 2 - using an escape sequence:

```
CRT CHAR(27):"_m[send_string('UDT on Doom','hello\013')]":CHAR(27):"\
```

sendkey

[sendkey(string)]

Allows you to direct keystrokes to other windows or other Windows applications. Keystrokes are sent to the window that currently has input focus. The function uses the same DDE keystroke syntax as this scripting language. The difference is, by default, all scripting keystroke macros are sent directly to the associated communication session regardless of whether it is visible or has input focus or not.

See also

char_mouse, gui_respond, send_eventstr, set_attribute, user_id, user_name, user_password

set_attribute

[set_attribute(object, attribute, value)]

Sets a specified attribute to the given value in the specified GUI object. The set_attribute sequence is usually one that has been captured by the Auto Script Learner. Although you may enter your own string sequence here it is better to recall one that has been intercepted via the Auto Script Learner.

If object is the word this, the current instance of the SBClient session is referred to, otherwise it is the handle or name of an object.

Example

The following trigger response enters the requested user id directly into the window's editclass object, then sends a keystroke event directly to the object as if the user had given this object focus and pressed the Enter key on the keyboard. This event will trigger SBClient to fetch what is in the editclass object and send it to the host.

The # represents an object instance handle number, and the actual number is extracted and kept by the trigger matching process. Its value is reapplied by substituting any # sequence in the response string with received trigger string value.

Trigger string:

```
GUIget SBWIN_#_0*sbloginclass<userid>
```

Trigger response:

```
[set_attribute("SBWIN_#_0","string","sb")][send_eventstr("SBWIN_#_0","special;SBWIN_#_0;13;0;0;0")]
```

The following sequence is used to create another session instance by setting a method in the current session:

```
[set_attribute("this","create_session","AP_on_SCO,override_script")]
```

See also

char_mouse, gui_respond, sendkey, set_attribute, user_data, user_id, user_name, user_password

suppress_session_mode

```
[suppress_session_mode(state)]
```

Toggles the drawable state of a session window.

state

If TRUE, the session window will not be drawable (that is, it will be hidden) until a login script has completed or this function is called again with a value of FALSE. If there is not a login script, the session window will become visible upon successful connection.

user_data

```
[user_data(type)]
```

Retrieves the user data referenced by type, stored in SBClient's host session configuration (script replacement strings), then inserts it in the script as text. If the host requires a termination sequence, you can add this as required.

type

The Auto Script Learner will substitute your user data in a learnt script if it sees you type the data and if the data has been stored in SBClient's host session configuration under type.

Note, if type contains the word password then its corresponding data is stored encrypted but will be decrypted automatically for execution.

Example

The following example shows a simple keystroke response to the host. The user id is fetched from the session configuration, then an Enter key is sent to terminate the host's input subroutine.

Trigger string:

login:

Trigger response:

```
[user_data(loginid)]{enter}
```

The following trigger response enters the requested user id directly into the window's editclass object after fetching it from the session configuration. The response sequence then sends a key-stroke event directly at this object as if the user had given this object focus and pressed the Enter key on the keyboard. The special event will trigger SBClient to fetch what is in the editclass object and send it to the host.

Trigger string:

```
GUIget SBWIN_#_0*sbloginclass<userid>
```

Trigger response:

```
[set_attribute("SBWIN_#_0","string",[user_data(loginid)])][send_eventstr("SBWIN_#_0","special;SBWIN_#_0;13;0;0;0")]
```

See also

char_mouse, gui_respond, send_eventstr, sendkey, set_attribute, user_id, user_name, user_password

user_id

```
[user_id()]
```

Retrieves the user id stored in SBClient's host session configuration, then inserts it in the script as text. If the host requires a termination sequence, you can add this as required.

The Auto Script Learner will substitute your user id in a learnt script if it sees you type the id and if the id has been stored in SBClient's host session configuration.

Example

The following example shows a simple keystroke response to the host. The user id is fetched from the session configuration, then an Enter key is sent to terminate the host's input subroutine.

Trigger string:

login:

Trigger response:

```
[user_id()]{enter}
```

The following trigger response enters the requested user id directly into the window's editclass object after fetching it from the session configuration. The response sequence then sends a key-stroke event directly at this object as if the user had given this object focus and pressed the Enter key on the keyboard. The special event will trigger SBClient to fetch what is in the editclass object and send it to the host.

Trigger string:

```
GUIget SBWIN_#_0*sbloginclass<userid>
```

Trigger response:

```
[set_attribute("SBWIN_#_0","string",[user_id()])[send_eventstr("SBWIN_#_0","special;SBWIN_#_0;13;0;0;0")]
```

See also

char_mouse, gui_respond, send_eventstr, sendkey, set_attribute, user_id, user_name, user_password

user_name

```
[user_name()]
```

Retrieves the user name stored in SBClient's host session configuration then inserts it in the script as text. If the host requires a termination sequence, you can add this as required.

The Auto Script Learner will substitute your user name in a learnt script if it sees you type it and it has been stored in SBClient's host session configuration.

Example

Trigger string:

login:

Trigger response:

```
[user_name()]{enter}
```

See also

char_mouse, gui_respond, send_eventstr, sendkey, set_attribute, user_data, user_id, user_password

user_password

```
[user_password()]
```

Retrieves the password stored in SBClient's host session configuration, decrypts it, then inserts it in the script as text. If the host requires a termination sequence, you can add this as required.

The Auto Script Learner will substitute your real password in a learnt script if it sees you type it and it has been stored in SBClient's host session configuration.

Example

Trigger string:

Password:

Trigger response:

```
[user_password()]{enter}
```

See also

char_mouse, gui_respond, send_eventstr, sendkey, set_attribute, user_data, user_id, user_name

user_script

```
[user_script(scriptid)]
```

This function executes the given script as an interpreted script or as a user-defined Data/C++ script function with special script arguments passed.

Launching an interpreted script

The syntax for running an interpreted scriptid is as follows:

`script_name`

is the name of a saved script in the TUSCRIPT file.

`file_name,script_name`

will run the `script_name` located in the file, `file_name`.

Note, that `script_name` may contain additional parameters or arguments as follows:

`script_name arg1 arg2 arg3...`

- or -

`script_name=arg1`

This is used to pass one argument on command lines where the space character is used for delimiting other types of information.

An optional `-Dn` or `(Dn` will cause a diagnostic window of `n` lines deep to be displayed when the script is started. See *Creating Advanced Scripts* for information on controlling this display from within a script.

Launching a Data/C++ script function

The syntax for running a Data/C++ script function is as follows:

`script_name`

is the name of a cataloged script function.

`lib_name@script_name`

is the name of a script function inside a cataloged `lib_name` function.

`file_name,script_name`

refers to Data/C++ source or compiled object code of a script function inside a given `file_name`.

`file_name,lib_name@script_name`

refers to a script function inside Data/C++ source or compiled object code of a lib_name function inside a given file_name.

The function arguments of a user defined script function are:

function userscriptname(hSBClientappinst, hSBClient sessinst, args)

Chapter 7 - Troubleshooting



Help Menu

The Help menu includes the following options to help you use SBClient and to troubleshoot communication problems:

Display Help Prompts

This option displays a question mark cursor. Clicking this cursor on an option or certain areas of the screen displays on-line help for that item.

Contents

Displays the contents page for SBClient's on-line help.

View Readme

Provides information for this release of SBClient that may not be in the manual.

How to Use Help

This presents a Windows Help file which explains how to use Windows Help files.

About SBClient

Displays information about this release of SBClient.

Troubleshooting

Allows you to:

- **Send Break.** Sends a break signal to the host system. Depending on what is happening on the host system at the time, you can often send a break signal to interrupt the host and return control to your session window. This option is also found on the Edit menu. The Edit menu also includes a Pause Display option, allowing you to pause incoming host data.
- **Reset Terminal.** Is the equivalent of switching your terminal off and back on again. This option will also abort any data transfer currently in progress.
- **Reset Port.** Disconnects and reconnects your active session.
- **Unlock Keyboard.** Enables the keyboard if it has been disabled by an escape sequence sent from the host.
- **Log Options.** The **Screen Log Options** form allows you to set specific debug traces and configure SBClient's log directory. See "[Log Options](#)" on page 245 for more information.
- **Support Information.** This is used to report SBClient related problems.

View Readme

This presents the Readme text file which contains late breaking information that may not have been documented in the manuals or the online Help supplied with SBClient.

Register SBClient

This is used if you want to license SBClient for the first time or if you want to increase the number of licenses. Depending on the circumstances, the dialog presented when you select this option will ask you to provide the following information:

- Serial Number
- User Name
- Organization
- Key

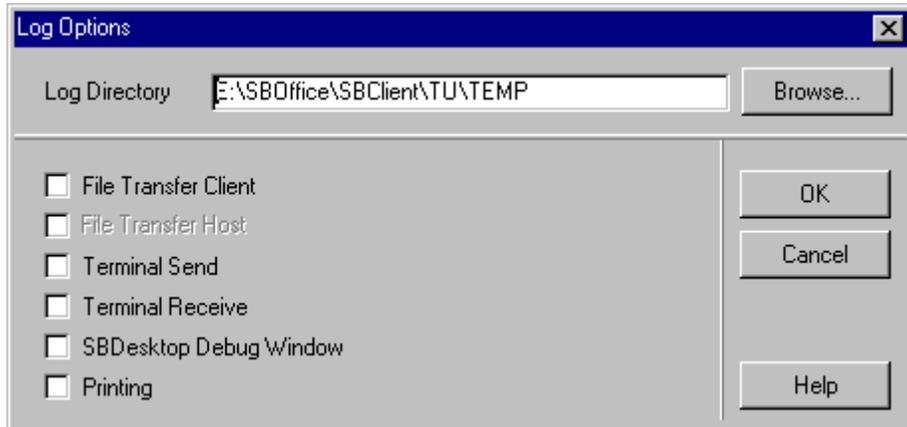
Registration Details

This presents a dialog which displays summary information about the current license arrangements. The information provided includes:

- Serial Number
- Organization
- User Name
- User Limit
- User Count
- Expiry Date

It is also possible to view a list of users who are using the license at the moment on the network by clicking on the Details button.

Log Options



Parameters are:

Log Directory

The full path to a temporary directory to store all log information.

File Transfer Client

This option turns on the following client debug logs:

- The SBClient debug log captures all packets sent between that client and the transfer DLLs. The log file is named SBCFT.LOG and is stored in the log directory.
- The DLL debug log captures all data packets sent and received by the DLL. The log file is named FT_EVENT.LOG and is stored in the C: root directory.

Terminal Send

This option captures all data sent from the client to the host. The log file is named TUSEND.LOG and is stored in the log directory.

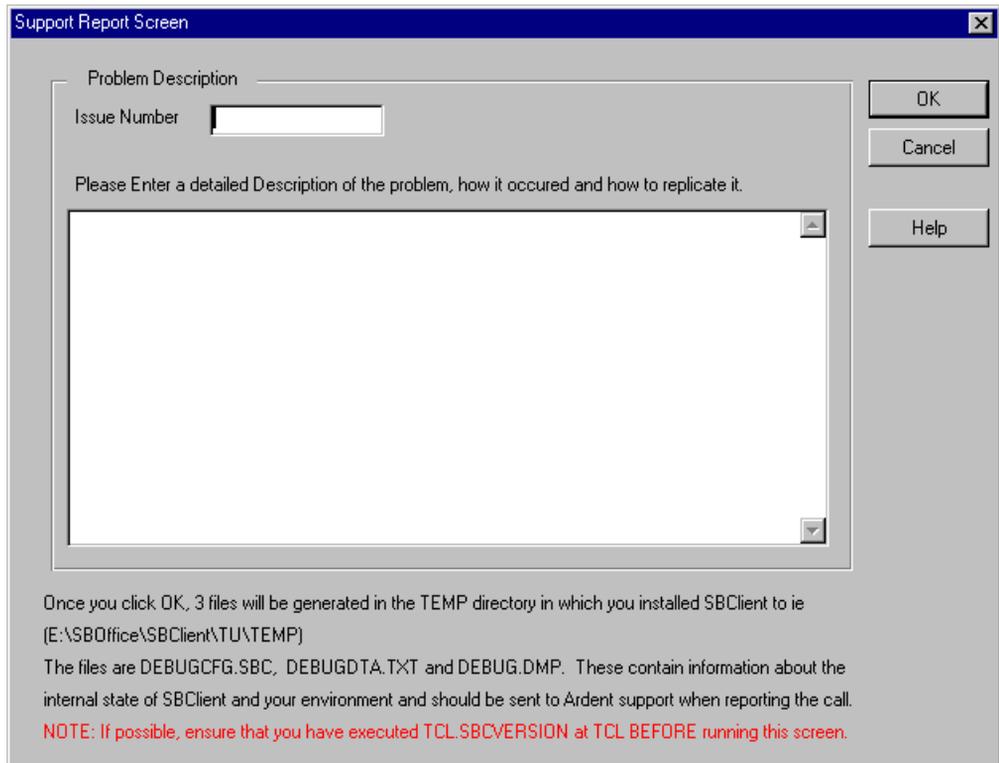
Terminal Receive

This option captures all data received by the client from the host. The log file is named TURCVE.LOG and is stored in the log directory.

SBDesktop Debug Window

This option launches the debug window for SBDesktop. See the SBDesktop Reference Manual for further information.

Support Report Screen



The screenshot shows a dialog box titled "Support Report Screen". It contains a "Problem Description" label, an "Issue Number" text box, and a large text area for a detailed description. The text area is currently empty. To the right of the text area are three buttons: "OK", "Cancel", and "Help". Below the text area, there is a paragraph of text explaining that clicking OK will generate three files in the TEMP directory (E:\SBOoffice\SBClient\TU\TEMP): DEBUGCFG.SBC, DEBUGDTA.TXT, and DEBUG.DMP. A red note states: "NOTE: If possible, ensure that you have executed TCL.SBCVERSION at TCL BEFORE running this screen."

This is available from **Help > Troubleshooting > Support Information**.

This should be completed when reporting problems with SBClient. After completing the details and pressing the OK button three files will be generated. These files are:

DEBUGCFG.SBC

DEBUGDTA.TXT

DEBUG.DMP

SBClient will then attempt to email this information to the Support section at Ardent Software Inc.

Before using this option it is highly recommended that you got to TCL and run TCL.SBCVERSION BEFORE selecting Help > Troubleshooting > Support Information. This will add information regarding the internal variables of SBClient to the DEBUGDTA.TXT file.

These files , which will contain important information about the internal state of SBClient, will need to be sent to your support representative.

These files can be found at the following location:

...SBoffice\SBClient\Temp

The Support Resport Screen contains the following prompts:

Issue Number

If given to you by your support office then this should be entered here, otherwise leave it blank.

Description

This is where you provide a full account of the problem that you have been experiencing.

Possible Problems/Solutions

To clear a communication problem between your PC and a host, try sending a break signal first. If that does not clear the problem, try resetting the terminal. If that does not work, try resetting the port.

Specific problems you may experience are:

- cannot connect to the host computer
- a GUI form hangs
- losing characters in the screen display
- extraneous characters in a session window

Cannot connect to the host computer

See appropriate troubleshooting section in this manual for the communications protocol you are using. If problems persist then contact your Systems Administrator.

A GUI Form Hangs

Generally when a GUI form hangs in SB+ or Host GUI this can be contributed to a conflict in terminal line settings between SBClient and the SB+ Server. The communications protocols used by SBClient make extensive use of the ASCII characters 1 and 2, Ctrl A and Ctrl B respectively.

You can use the system menu option Switch To Terminal to view host character output or to access SBClient's menu options (eg Help > Troubleshooting). In character mode this option will be Switch To GUI. These options minimize the current display and restore the GUI or character display, as appropriate.

If you are experiencing GUI form hangs then you should check for the following:

- On UNIX run the command "stty -a" and ensure that none of the line setting are using ^A, ^B, or ^X.
- Make sure that your line settings are using 8 bit not 7 bit. Be careful with using commands such as telnet or rlogin as depending on the UNIX flavor these may be 7 bit protocols.

- If you are using a terminal server make sure that Ctrl A, Ctrl B, Ctrl X, and character 27 concatenated with '_' are being passed through correctly.
- On UNIX systems disable the LNEXT token. This token is usually set to Ctl V and is a literal next character on SVR4 and 4.4+BSD. The LNEXT character is recognized on input in extended mode only. It will cause any special meaning of the next character to be ignored. The LNEXT character is discarded when processed, but the next character entered is passed to the process. Extended mode can be turned off by doing a 'stty -iexten'.

There are no known problems with terminal line setting when using SBClient and SB+ Server on a Windows NT system. This is due to the fact that Windows NT doesn't have the concept of a terminal, hence no line settings.

Losing Characters in Screen Display

Check that:

- the parity and stop bit settings for your session are correct
- the host and SBClient flow control protocols are consistent. For example, if the host is set up for software flow control, SBClient should also be configured to use software flow control
- the baud rate is not too high

Losing characters may also result from a high system load, caused by mouse, keyboard, network or other activity (eg the production of sound).

Extraneous Characters in a Session Window

This may be due to incompatibility in port settings between the PC and the host system.

Check the settings of the host port (from a working terminal) using the appropriate host command (eg stty -a in UNIX). Check these with the serial configuration settings. The most likely causes for mismatch are the baud rate or parity settings for asynchronous communications.

Appendix A - Internationalization Issues



This appendix discusses various internationalization issues associated with character sets. An overview of the historical perspective is given.

Explanation of Code pages, character sets, OEM and ANSI.

Historically, the original ASCII character set utilized 7 bits. This meant that a maximum of 128 characters could be defined by this set (ie $2^7 = 128$). This was suitable for the English character set but was too limiting for other languages.

This led to the development of the IBM PC Extended Character Set (ECS) which is used by DOS. The lower 128 values are the same as the ASCII character set, but values 128-255 are used for European and other special characters.

The ECS codes 0-31 and 128-159, which are assigned to non-printing control functions such as Return and Line feed, are also assigned special graphics for use in contexts where control functions are not needed. That is how you can obtain symbols such as happy faces and arrows in ECS displays and line-drawing characters as well as European characters.

It became clear that one ECS, even with its 256 characters, was not enough to encompass all European languages let alone the Asian languages. This led to different sets being developed for different groups of countries - each new set known as a code page - often referred to as an OEM code page. Below is a list of some of the common code pages. Note that most West European countries including England use code page 850 while the US uses code page 437.

| | |
|-----|------------------------|
| 437 | United States |
| 850 | Multilingual (Latin I) |
| 852 | Slavic (Latin II) |
| 855 | Cyrillic (Russian) |
| 857 | Turkish |
| 860 | Portuguese |
| 861 | Icelandic |
| 862 | Canadian-French |
| 865 | Nordic |
| 866 | Russian |

869 Modern Greek

With the arrival of Windows the issue became more complicated, since Windows uses another standard for its code pages that is different to the DOS OEM code pages. Windows 3.x and Windows 95 support the ANSI character set, which is also an 8 bit set (ie 28 = 256 characters, numbered 0 to 255) but the positions of the characters within the set are different from the equivalent OEM set. Values 0 to 127 are the same as in the ASCII character set. Values 128 to 255 are similar to the ISO Latin-1 character set (OEM code page 850), but there are also extensions and incompatibilities. Unfortunately, this means that the ANSI character set is incompatible with the ECS character set used by DOS. For example, the British pound symbol is code 163 in ANSI, but code 156 in ECS. Worse still, ANSI does not include many of the ECS symbols, particularly the line drawing characters. If you try to display a DOS document in Windows, Windows attempts to convert the ECS characters to ANSI. Any character that does not convert is replaced by an arbitrary graphic. Therefore a round-trip conversion from DOS to Windows and back will not re-create the original document.

Unicode

Windows NT uses the 16-bit Unicode character set. Windows 95 still uses the 8-bit ANSI character set, but is a transition toward Unicode (32-bit OLE uses Unicode and Windows 98 includes some Unicode as well).

Unicode is a 16-bit character set designed to cover all the world's major living languages, including the far east, in addition to scientific symbols and dead languages that are the subject of scholarly interest. It eliminates the complexity of multibyte character sets that are currently used on UNIX and Windows to support Asian languages. Unicode was created by a consortium of companies including Apple, Microsoft, HP, Digital and IBM and merged its efforts with the ISO-10646 standard to produce a single standard in 1993. Unicode is already the basis for at least one operating system, namely Windows NT.

Unicode is a 16-bit character set where all characters occupy the same space. The first 256 values are the same as the ISO-Latin character set, which is also the basis for the ANSI Character set used in Windows 3.1 and Windows 95. But Unicode goes on to define 34,168 distinct coded characters. In most character sets a single value is often assigned to several characters. For example, in ASCII a "-" is used to represent a hyphen, a minus sign, a dash and a non-breaking hyphen. In Unicode each meaning is given its own code. The Unicode standard contains only one instance of each character and assigns it a unique name and code value. It also supports "combining" accent characters, which follow the base character that they are to modify.

SBCClient Handling of Charsets, Fonts, Code Pages and Printing

There are various areas where the international issues occur. These are now discussed in the following sections.

Mapping of Data from OEM to ANSI and ANSI to OEM for Display

The differences between ANSI and OEM character sets have been discussed above. Each international version of Windows has its own internal representation of what an 'ANSI' code page is and what an 'OEM' code page is depending on what country it is in. This means that eg the US version of Windows has an internal OEM code page of 437 and an internal ANSI code page of 1252 whereas the UK, German and French versions of windows have an internal OEM code page of 850 although they also have an internal ANSI code page of 1252. In another example, the Russian version of windows has an internal OEM code page of 866 and an internal ANSI code page of 1251. In a system where multiple keyboards are installed, the internal ANSI code page changes when the keyboard is changed eg from the system tray. The internal OEM code page is NOT however changed when this setting is changed).

It is also possible however to create Windows fonts that use the OEM character set eg System Builder Font (Note System Builder Font is an OEM font based on the US OEM code page 437 which includes line drawing characters but not all the European characters that are present in code page 850).

In order to display the characters correctly SBCClient needs to know what character set the host data is using and compare that to the character set of our chosen font and do any necessary conversions. SBCClient has an ini variable called SBOCHARSET in the environment section of sbopen.ini to tell SBCClient what the host is using. This value should be set to what the host is using and since most hosts are OEM this has a default value of OEM. Remember though that internally to windows, OEM means something different on systems where the internal OEM codepages differ eg US versus UK.

When data arrives in SBCClient from the host, SBCClient checks what font it is using and determines whether it is ANSI or OEM. If it is different to that value set in SBOCHARSET, SBCClient does either an ANSI to OEM or OEM to ANSI conversion whichever is applicable. To do the conversion, SBCClient calls standard windows routines to do the conversion. These standard routines have internal mappings between the internal Windows OEM and internal Windows ANSI character sets.

When entering keys SBClient performs an ANSI to OEM conversion (if the host is OEM) since the keyboard generates an ANSI key. The character that is sent to the host is then an OEM character. For example a UK keyboard will generate char 163 when the pound key (£) is entered. SBClient will then do an ANSI to OEM conversion on the character resulting in the OEM (£) character 156 being sent to the host. In rare cases this conversion should not be performed. In such cases the toggle on the keyboard mapping titled ANSI Keyboard should be disabled.

When a Character in One Code Page is Unavailable on Another

When a character in one code page is unavailable on another and the conversion occurs, the character gets converted to its closest equivalent character in the other code page. For example, ASCII 224 (*œ*) in code page 437 is converted to ASCII 97 (a) in code page 1252 because this is the most similar character in the 1252 code page. When you convert this ANSI character back to code page 437, the result is ASCII 97 (a), because ASCII 97 (a) exists in both code pages. The original 437 character (*œ*) is lost. This means that incorrect data will be saved in your database if the character exists in one code page but not the other, and an OEM host is used with an ANSI font. This often manifests in the case of line drawing characters.

Line Drawing Characters

There is an additional issue when the host code wishes to utilize line drawing characters. The host code can send down these OEM specific characters and if the Windows font is an ANSI font eg Courier new, there are no equivalent characters in the font so the character graphics will not be as good as if an OEM font was used on the client. Note that if the host code sends specific emulation dependent esc sequences that draw lines, then SBClient will know that this is a special case and draw these characters independently of any OEM to ANSI translations.

Determining the internal code page of the OEM font

To determine what internal OEM codepage the system you are on is using, you can simply type `chcp [nnn]` in a dos box. Note that this function will display the number of the active console code page, or change the active console code page that Windows NT is to use for the console. Note: Only the OEM code page installed with Windows NT will display correctly in a command prompt window using Raster fonts. Other code pages will display correctly in full-screen mode or command prompt windows that are using TrueType fonts.

chcp [nnn]

Parameter

none

Used without parameters, chcp displays the number of the active console code page.

nnn

Specifies the code page. The following list shows each code page that Windows NT supports and its country or language:

| | |
|-----|------------------------|
| 437 | United States |
| 850 | Multilingual (Latin I) |
| 852 | Slavic (Latin II) |
| 855 | Cyrillic (Russian) |
| 856 | Turkish |
| 860 | Portuguese |
| 861 | Icelandic |
| 862 | Canadian-French |
| 865 | Nordic |
| 866 | Russian |
| 869 | Modern Greek |

Usual (National) Case

In the usual case (ie where client and host are in the same country). Windows will interpret ANSI to OEM conversions correctly. This is because the client will have an internal representation of the OEM code page the same as that of a host in that country. This means that if we are in the UK on a UK Windows system talking to a UK host that uses code page 850 (and our font is ANSI eg Courier New) Windows will do the conversion correctly. This is because the Windows UK version also has an internal representation of 850 as its OEM character set. Of course if we use a font in SBClient that is an OEM font that matches the OEM codepage on the host, SBClient will do no conversion as the SBOCHARSET is OEM and the font is also OEM. If however the font is OEM but the wrong OEM codepage, no conversion will be done but the display will still be incorrect. Since system builder font is an OEM font that uses code page 437 it CANNOT be used for this purpose except when talking to a US host ie one that also uses code page 437.

Multinational Case

This is the case whereby there is a Single European OEM host with clients in different geographical locations (that have differing code pages).

If we wish to log into a European host, (eg UK that is using code page 850), from a US Windows PC we will hit a problem since the US Windows has an internal OEM code page of 437. What would then occur if we chose say Courier New as our font and set SBOCHARSET to OEM (since the host is OEM not ANSI), SBClient would call the windows OEM to ANSI conversion routine. Internally Windows will think that it is converting characters from OEM code page 437 to ANSI 1252 when in fact the characters arriving from the host are encoded using the 850 charset. This will cause Windows to map the characters incorrectly and the display and entering of data would be incorrect. The question is how can we solve this issue?

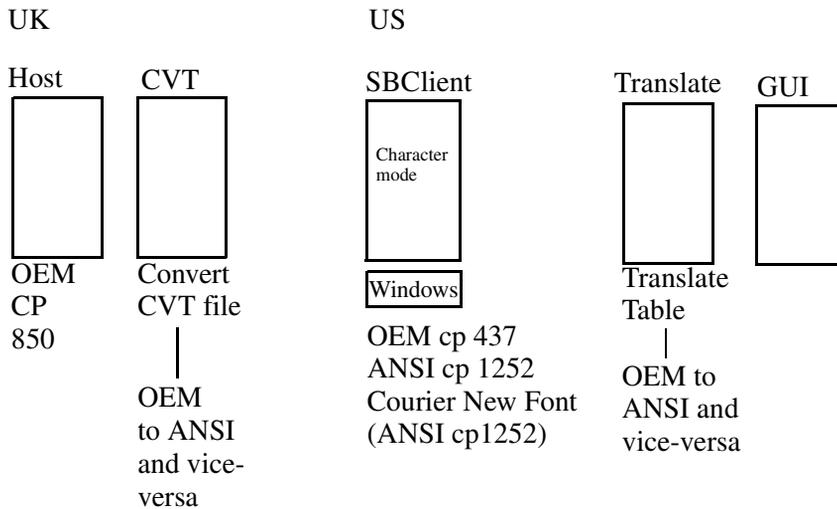
One possibility would be to store data on our hosts in ANSI and then set the SBOCHARSET to ANSI for both European and US systems. This is not feasible for the following reasons:

1. The conversion of existing data is time consuming and error prone.
2. Real terminals eg Wyse or VT would stop working as they only know about OEM character sets.
3. The ANSI table would use char 252, char 253 and char 254, i.e. ü, ý, and þ for data and these values clash with the Unidata/Universe/Pick standard delimiters as well as SBClient's standard delimiters.

Another possible solution would be to use SBClient's conversion table (CVT) files to map the incoming characters (which are in code page 850) to the equivalent character in code page 437 and vice versa on the outgoing characters on the US system only. This would mean that when Windows gets to do the OEM to ANSI conversion, the characters are already in the code page that the US version internally does the conversion from/to ie code page 437. A problem with this solution would manifest in the case where there is no equivalent character in the 437 codepage eg Å ie char 181 in code page 850 has no equivalent in code page 437. We could pick a similar character eg char 142 Ä but this is not ideal (This would be a simple solution and recommended for languages where all the required characters exist in code page 437).

SBClient could be 'fooled' into thinking the host is ANSI on US Windows by setting SBOCHARSET to ANSI. This would mean that SBClient would do no OEM to ANSI conversion when using an ANSI font such as Courier New. Of course this on it's own is not enough, as OEM characters cannot simply be displayed in an ANSI charset with no conversion. In this case we would set up the CVT file to do our own OEM to ANSI and ANSI to OEM conversions (cvt files are discussed

in Appendix E). We know the character set on the host is OEM charset 850 and we can set up our own map to translate to ANSI. We are thus bypassing what US windows thinks is the internal OEM set and doing it correctly ourselves. We would also need to uncheck the ANSI keyboard toggle (see discussion on input) on the key mapping screen. If we do not disable this option then every time we hit a key, an ANSI to OEM conversion will be done. This will corrupt our data as we are now taking care of all conversions ourselves in the CVT file. This should work fine in character mode since there is no string processing relying on delimiters in character mode. We will potentially have problems in GUI mode though because the data that will be processed in SBClient e.g. read or write the data out of a GUI field could have char 252 char 253 and char 254 in it. Note that in the usual (National) case discussed in the previous section (ie Host and client in the same country) we do not have this char 252 char 253 and char 254 problem in GUI. This is because the data is all processed as OEM (which doesn't use those chars) until the last minute when it is passed to the GUI text fields and the text field then does the conversion whereas in this case the conversion is done as soon as data arrives or just before it leaves SBClient. This solution would therefore require significant changes to SBClient. The following diagram and explanation illustrates another option i.e. the option of using translate tables.



From the diagram you can see the different places where conversions can occur.

Basically the aim is to convert data on the host which is using CP 850 to ANSI CP 1252 for use with Courier New Font in the CEO and, for example MS Sans Serif in GUI fields. The issue is that it is not possible to use Windows internal OEM to ANSI routines since internally Windows thinks that the OEM code page is 437 (since it is a US Windows not a UK Windows).

If the conversion is performed in the CVT file, then as can be seen from the diagram, from then on SBClient is dealing with ANSI characters which make use of char 252, 253, 254. This is fine in character mode as SBClient will not need to process this data further.

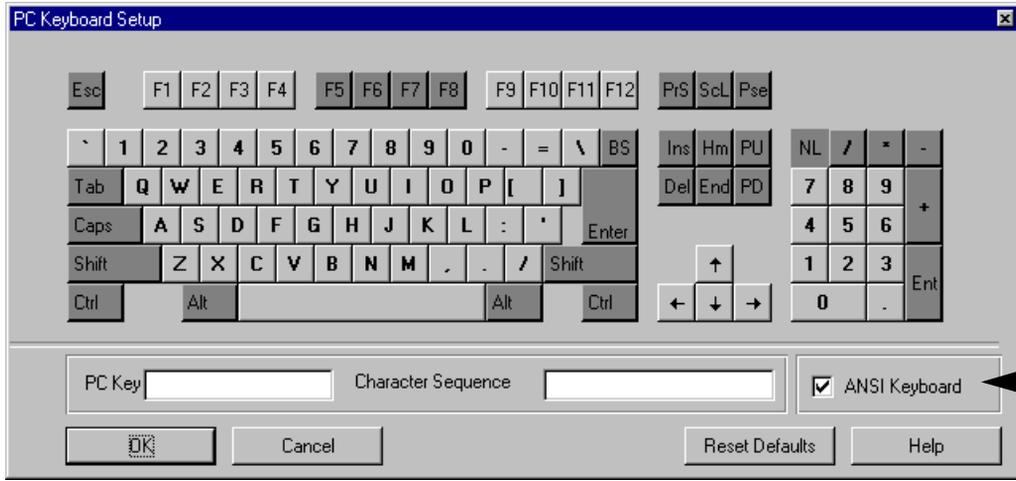
If GUI is required then the translate tables should be used to do the OEM to ANSI conversion instead of the CVT tables. This will then only do the conversion at the last minute so that the integrity of data in all processing in SBClient is preserved.

If a translation table is specified then at the point at which SBClient normally does it's calls to the windows OEM to ANSI and ANSI to OEM routines it does it's own conversion based on the user defined translation table.

See also the discussion on CVT files and Translate Files in Appendix E.

Entering data that requires conversion from one character set to another.

The Keyboard Mapping ANSI Keyboard setting



In countries that use the extended characters eg France or Germany, the most common scenario is to have an OEM host (eg France would use code page 850) talking to an ANSI client (France would use code page 1252). To display the extended characters, a font that contains these characters needs to be chosen. The most common choice is Courier New (for character mode) and MS San Serif (GUI mode). These are both ANSI fonts that in France would use code page 1252. The mapping of OEM to ANSI has been discussed above with respect to displaying the OEM character from the host in the ANSI font on the client. What still needs to be discussed is the manner in which the data entered on a Window's client gets transmitted to the host.

When a key is pressed in a Windows environment, the character that is generated is an ANSI character. This means that SBClient receives an ANSI character but this character needs to be sent to the host as the equivalent OEM character (since our host is OEM). What happens is that SBClient does this conversion and sends the OEM character to the host. The host then receives the correct character and echoes back the OEM character. SBClient receives the echoed OEM character and since we are using an ANSI font (Courier New) we display it correctly by doing the OEM to ANSI conversion. This illustrates that every keystroke needs to be converted when talking to an OEM host. Having the ANSI keyboard toggle set enables this conversion.

If, however, the host is ANSI, then `sbocharset` should be set to ANSI and the ANSI keyboard toggle should not be checked.

There is a known problem with Eastern European and Russian systems that causes some confusion on this issue in all 16 bit releases and SBClient 4.3.x. The problem occurs in the situation where eg you have a Russian OEM host (code page 866) and a Russian version of windows that is using code page 1251 and Courier New font (Cyrillic script). The `sbocharset` should be set to OEM and the ANSI keyboard toggle should be checked. When a key is entered, SBClient correctly converts the character to OEM and sends it to the host. The host echoes the OEM character back and SBClient displays the character. The problem is that the echoed character that is displayed in the CEO is not converted back to ANSI and so the incorrect character is displayed. The workaround for this is to set up a CVT file and in the display section only, set up the oem to ansi map eg map OEM character 129 to ANSI 193 and all other relevant characters. Everything should then work as expected. Note this has been resolved in SBClient 4.4.3 and later (however, if for some reason the old behavior is required in SBClient 4.4.3 and later, then add the line "`convertnonansi = 0`" to the [ceo] section of the `sboopen.ini` file).

Entering Extended Characters with Alt-Num

You can enter any ANSI character into a Windows application. If you see it on the keyboard, just press it. Even if it isn't on your keyboard, you can still enter it using a special Alt-Num key sequence.

DOS and Windows allow you to create Extended Characters such as y-umlaut with the help of a numeric code. To enter any character from the IBM PC Extended Character (ECS), which is standard on DOS, hold down the Alt key while you enter the 3-digit decimal code. For example, Alt-152 is y-umlaut in ECS. To be compatible with DOS, you can still use Alt plus a 3-digit ECS code, but only if the character exists in the ANSI Character Set used by Windows. Windows then converts the character from the ECS code to the equivalent ANSI code. For example, y-umlaut is entered as Alt-152 but is translated to decimal 255. Or you can enter the ANSI code, but must convert it to 4-digits by preceding it with a 0 (zero). Thus Alt-0255 is also y-umlaut.

Warning

Be sure to use the Numeric Keypad when keying in an Alt-Number and remember to engage the Num Lock key.

To find the 4-digit Windows code for a character, use the Character Map (from start/accessories) that is supplied with windows. Select some font, such as Arial, click on the character you want, then note the 4-digit code displayed at the bottom of the window.

Note 

When using Alt-Num sequences to enter keys, Windows will use its internal ANSI and OEM codepages so if you have changed keyboards this will not work as expected. In other words, only use this for the original installed language on the windows machine.

Note 

Generating OEM data to store on the host.

Windows generally treats all data as ANSI unless told otherwise. If, for example, the data was created in Word and then saved as text, windows would save it as ANSI text. If the host is OEM, this is not what you want to upload to it. What you would want to do is to save the data from word as "Dos text" since this will save it as OEM data. If the host is OEM, this data can then be uploaded without any problems.

Conversion of one OEM font to another OEM font

Russian and Some East European Character Sets

All the above discussions regarding OEM and ANSI are relevant to Russian Windows systems talking to Russian hosts. The internal ANSI code page on a Russian Windows is 1251 and the internal OEM code page is 866. Some Russian hosts use a different OEM code page to 866 and so this different code page needs to be mapped in the CVT file to 866. One such case is provided with the Cyrillic.CVT file that ships with SBClient.

Printing

In the 16 bit SBClient before release 2.4.3.0 it was not possible to choose any font for printing. The font used when Windows drivers were selected was either an internal SBClient font (SB Font (OEM code page 437) when the **Use SBClient Printer Font** toggle was checked, or Courier New (ANSI code page 1252) when the toggle was unchecked). It was possible to make SBClient do an OEM to ANSI conversion when printing, by selecting that option on the print options dialog, so

OEM host data could be printed. If however another code page eg Cyrillic (1251) was required for printing it was not possible. The only workaround was to switch off the Windows printer driver option and the OEM to ANSI option, and use printers with firmware or flash memory and loadable OEM Cyrillic fonts.

In SBClient 2.4.3.0, it is now possible to specify a printer font in the sbopen ini file for use with printing with Windows printer drivers. To do so add the following to the [Environment] section of the Sbopen.ini file.

```
SBOPRINTFONT=Font Name, Normal Point Size, Compress Point Size
```

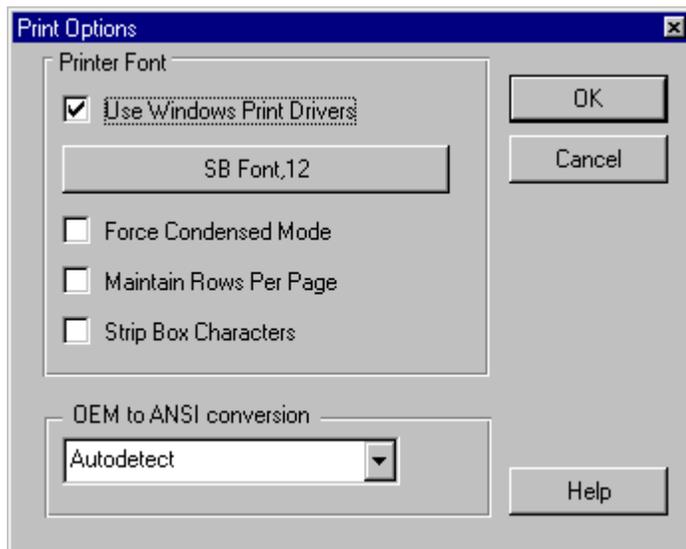
```
Eg SBOPRINTFONT=Arial, 8, 12
```

In SBClient 4.3 and above, it is possible to select a printer font from the print options dialog box when windows print drivers is selected. Below is the new Print options dialog in SBClient 4.3.

Note



The OEM to ANSI option also has an autodetect which when selected will attempt to automatically determine whether a conversion is required or not. Note that there was a bug in the implementation of OEM to ANSI SBClient 4.3 and Sbclient 4.3.1 but this has been fixed in SBClient 4.4.3. and later.



OEM to ANSI during File Transfers and Smart Query and DDE

The case may arise that an OEM file on the host needs to be converted to an ANSI file on the client for use in eg notepad with a Courier Font. To do this the file can be transferred as usual and then the routine TU.OEM.TO.ANSI can be called to translate the file. (TU.ANSI.TO.OEM is available for ANSI to OEM conversions). Alternatively when using the HOST to PC or PC to HOST utilities, the translation can be done on the fly by setting up the translation tables and specifying eg the notation (162) for the host and (243) for the client to translate OEM character 162 to the ANSI equivalent 243. The resulting file will have the ANSI data in it.

When performing a smart query to Word or Excel from a host containing OEM data, it may be necessary to convert the data from OEM to ANSI depending on the font that is to be used in word. If, as is the usual case, an ANSI font is to be used, the OEM to ANSI option should be checked on the Smart Query dialog before starting the query.

When programming transfers from SBClient to Microsoft Word or Excel for languages that use the high bit characters of the OEM character set, for example accented characters in Spanish, you must convert these characters from the OEM to the ANSI character set if the font to be used in Word or Excel is an ANSI font. To do this conversion, specify the -I option when using the relevant TU.TO.WORD and TU.TO.EXCEL routines.

SBClient defaults to doing ANSI to OEM conversions for TU.DDE.READ and vice-versa for TU.DDE.WRITE since windows apps are ANSI, and hosts are usually OEM. A problem may arise when the data in the spreadsheet/document is OEM already. To stop this conversion happening, the NOANSIOEMXFER variable in the environment section of the SBOPEN.INI file should be set to to 1. For example, if you are exchange data to EXCEL with DDE and you do not want SBClient to convert ANSI to OEM or OEM to ANSI, set noansioenxfer to 1.

Network and 7 bit link Problems

There are many networks still in existence today, especially WANS, that use 7 bits for their transmission of data. This means that if you have SBClient on a PC talking to a host using high bit (8bit) characters eg a German host across a 7 bit WAN, the data will be corrupted when it is transferred across the network. The high bit will be stripped of the data. This will manifest as in the following eg. The host has OEM code page 850 and is sending the character ü ie char 154. The character that SBClient will be receive is character 154 less the high bit ie less 128 = 26 ie the right arrow symbol. The only solution to this problem is to upgrade the link to 8 bit.

Note that for file transfers SBClient uses the full 8 bits to encode data for transmission. This can be set to 7 bit mode to allow SBZ to work correctly over 7 bit links. In this case SBClient encodes the data in a format similar to UUENCODE. This would enable SBClient to be used across 7 bit links for languages that don't require the high bit characters (eg English). This does not help the case in the above paragraph.

Note



When talking to Advanced Pick (AP), the TCL setting (Extended character set) XCS must be on to enable 8 bit data to be transferred. It can be checked by typing XCS return. To set it on type XCS-ON.

Known Issues

1. SBClient does not support multibyte. This means that Asian languages are not supported. Further release specific information may be found at the Ardent website, or with your Ardent Support representative.
2. Char (250) to char(254) can be a problem where the language used uses those characters as valid characters ie an ANSI host that uses those characters.
3. The SBOCHARSET variable in the ini file applies to all sessions so there may be problems if different sessions from within the same instance of SBClient talk to different hosts where the charset on the host is eg oem on one host and ANSI on another.
4. In editions of SBClient prior to release 4.5.3 some SMART QUERY pass through routines would not work with non English versions of OFFICE where the internal DDE topics were not represented in English. For example, System topic in English is systÈme.
5. Error and setup dialogs are currently all in English and it is currently not possible for the user to change them to other languages.
6. The first release that had most issues resolved was 2424. Do not attempt to resolve any international issues on releases prior to that release. It is best to be using the latest release of SBClient for 32 bit Windows based systems and in 16 bit Windows based systems to use release 2.4.3.0 or later.
7. SBFont and system builder font are both OEM US English (code page 437) only fonts. If you wish to display other characters choose an appropriate font that has all the correct characters.

The character map that is supplied with the windows operating system is very useful in this respect. (Start/programs/accessories/character map).

8. Prior to SBClient 4.5.3/SB+ Server 4.5.3, first level help in SB+ was not customizable and therefore could not display characters other than those on code page 1252. The same problem with tooltips and second and 3rd level help that existed in SBClient 4.3 and SBClient 4.3.1 was fixed in SBClient 4.4.3
9. The default cvt files may need to be altered by the user as they have been generated with certain assumptions that may not be valid in all cases.
10. Releases of SBClient prior to 4.4.3 had a problem with Cyrillic character sets. This was discussed in section 2 of this document and a workaround was proposed.
11. Prior to release 4.4.3, OEM to ANSI during printing did not work correctly and gui text fields that automatically expanded beyond their initial character limit also had a problem with OEM to ANSI conversions. The /MPX menu explorer did not convert from OEM to ANSI correctly.
12. When the user types alt-enter SBClient goes to full screen mode. SBClient uses the same font family that is currently being used but if the family has no size that fits the full screen, SBClient will default to use System Builder font. If the font requires European characters, they will not be displayed correctly when this occurs. If this happens, the user must ensure they are using a font that has all the required characters and a valid full screen size before going to full screen mode. Courier New should suffice for most cases.
13. On the international version of Unidata, the default tokens are not char 252, char 253 and char 254. SBClient does not support this remapping but will work to some degree (in the DDE area) if the following line is added to the SBOPEN.ini file in the environment section.

```
TOKENS=\250\251\252\253\254
```

Where the numbers eg 250 represent the numbers you wish to use. The above line is the default and if the line is not there, those are the values that will be used for delimiters. If for eg you want to map 250 to 170 and 251 to 171 you would use

```
TOKENS=\170\171\252\253\254
```

Troubleshooting and things to check when problems occur

1. Read this entire appendix and ensure you understand the implications of all the parameters. Read the SBClient readme and help file.

2. Ensure that the user is using the latest version of SBClient.
3. Ensure that you have all the relevant code tables in front of you. Determine what code tables Windows is using (OEM and ANSI). You can use chcp to get the OEM page and you can use table in the accompanying pdf file to see what ANSI pages are used for what languages.
4. Always try to ensure character mode works before debugging GUI.
5. Check whether the data on the host is OEM or ANSI and set SBOCHARSET appropriately.
6. If the host code page is ANSI (unlikely) and char 252 to 254 is being used, then set up a cvt file that maps the characters to oem on the input and back to ansi on the output and set sbocharset to OEM. Ensure the ANSI keyboard setting is checked.
7. If the host code page is OEM then ensure that the code page the host is using matches the internal codepage of windows. If it does match ensure that the ANSI keyboard setting is checked.
8. Ensure that the host NT or Unix Machines, are set up for the correct language eg it is essential that a German user is running a German version of the operating systems. This ensures that the host will be using the correct code page.
9. Use a font that has all the characters required - Courier New is a safe choice for European systems. If not Western European, ensure that the correct version of Courier New is used eg Courier New (Cyr) for Russian.
10. If printing is a problem, ensure that the font chosen has all the characters and ensure that the correct script setting is chosen from the print common dialog. Try forcing OEM to ANSI from the Print options dialog.
11. Replicate the problem - have test data on the host and write simple BASIC programs to CRT the SEQ of the character received.
12. Eliminate the 7 bit network as a possible source of the problem by connecting to a host via a modem.
13. As a last resort, get all the settings mentioned in this document ie OEM and ANSI code pages on the client as well as the version (and language) of windows as well the version of SBClient. Also get some sample data and a step by step description of how to replicate the problem. Get all the host settings ie code pages, OS details and database details. Also get screenshots of what you are seeing and descriptions of what you expect to see. Obtain a copy of any cvt files

that are used. Get the SBClient settings, the sbopen.ini file, whether the problem is character or GUI and the TUSTATE file. Then send this to Ardent Technical Support.

Charts of Character Sets

Published charts of character sets are available from a number of professional organizations and should be consulted when developing applications with SBClient.

The following websites also contain charts of character sets:

<http://www.czyborra.com/charsets/>

<http://www.hermessoft.com/wincp.html>

<http://utopia.knoware.nl/users/eprebel/Communication/CharacterSets/index.html>

<http://www.cyrillic.com/ref/cyrillic/cp1251.html>

<http://www.robelle.com/library/smugbook/char.html>

Note



The organizations and websites quoted above are not affiliated with Ardent Software Inc in any way. Consequently the content of the above websites may be subject to change without notice.

Appendix B - SBClient Defaults



This appendix provides the following SBClient configuration default values:

- **Communication Defaults.** Provides defaults for serial, network and modem configurations.
- **Terminal Emulation Defaults.** Provides configuration defaults for the terminal emulations supported by SBClient.
- **Other Defaults.** Provides defaults for other configuration settings.

Communication Defaults

SBClient caters for three types of communication method, namely:

- Serial.
- Network. The network default values are shown for each network type.
- Modem. The modem default values include the settings for a default modem type.

Serial

Port COM2

Baud Rate 19200

Data Bits 8

Stop Bits 1

Parity NONE

Advanced Serial

Flow Control XON/XOFF

XON Character 17 (CTRL-Q)

XOFF Character 19 (CTRL-S)

RTS/CTS No

DTR/DSR No

Break Duration 100 ms

CTS Timeout 0 ms

DSR Timeout 0 ms

Send CR on Connect No

Network

Network Type Windows Sockets (TCP/IP)

Host No Default

Term Type No Default

Port 23

Enable Echo Yes

Mode ASCII

Conversion CRLF

Echo Remote

Flush on Break Yes

Buffered Output No

Break Method Abort Output (AO)

Break Signal `tn_interrupt`

Debug No

Autostart No

Modem

Port COM2

Baud Rate 38400

Data Bits 8

Stop Bits 1

Parity None

Prefix No default

Telephone Number No default

Modem Default Settings

Model GENERIC HIGH SPEED

Speaker ModeUntil Connected

Speaker VolumeLow

Initialization StringATZ^M~AT&C1&D2V1X4S0

Hangup String~~~+++~~~ATH0

Connect StringCONNECT

Reset StringATZ

No Connect StringNO CARRIER,BUSY,VOICE,NO DIAL

Pulse DialNo

Detect Busy ToneYes

Detect Dial ToneYes

Dial Retries3

Retry Every60 secs

Wait for Carrier60 secs

Auto AnswerNo

Answer AfterNo default

Drop DTR to HangupYes

Advanced Modem

XON/XOFFNo

RTS/CTSYes

DTR/OSRNo

XON Char17

XOFF Char19

Break Duration100

CTS 0

DSR 0

Send CR On ConnectNo

Terminal Emulation Defaults

The following terminal emulations configurations are catered for:

- ADDS Viewpoint, ADDS Viewpoint Enhanced
- ANSI
- IN4407
- Prism 8
- SM9400
- VT52 and VT100
- VT220
- Wyse 50, 60 and 120

ADDS Viewpoint Defaults, ADDS Viewpoint Enhanced

Smooth Scroll No

Cursor Home at Bottom Yes

Auto Scroll Yes

Auto Wrap Yes

Advanced ADDS Viewpoint

Character Mapping No Default

ANSI Defaults

Smooth Scroll No

Auto Wrap Yes

Advanced ANSI

Character MappingNo Default

IN4407 Defaults

Auto Wrap Yes

Advanced IN4407

Character MappingNo Default

PRISM 8 Defaults

Auto Wrap Yes

Smooth Scroll No

Advanced PRISM 8

Character MappingNo Default

SM9400 Defaults

Display Screen Size80x24

AttributeCharacter

Auto Wrap Yes

Advanced SM9400

Smooth ScrollNo

Auto Scroll VerticalYes

Auto Scroll Horizontal No

EmbeddedYes

Character MappingNo Default

VT52 and VT100 Defaults

Smooth Scroll No

Screen Size 80x24

Keypad Numeric

ANSI Color No

New LineNo

Auto WrapYes

Display Control CharactersNo

VT220 Defaults

Smooth Scroll No

Screen Size 80x24

Keypad Numeric

PersonalityVT200

Auto Wrap Yes

New Line No

ANSI ColorNo

Display Control CharactersNo

Advanced VT52, VT100 and VT220

Local Echo No

Insert No

Reverse No

Origin No

Autoprint No

Print FF No
Auto Repeat No
8-bit Codes No
Character Mapping No Default
Print Extent Full
Cursor Key ANSI

Wyse 50, 60 and 120 Defaults

Enter CR
Return CR
Received CR CR
Screen Size 80 x 24
Attribute Character
Insert No
ApplicationNo

Advanced Wyse 50, 60 and 120

Smooth Scroll No
Auto Scroll Yes
Auto Font LoadYes
Auto WrapYes
Status Line OFF
Monitor No
Wyse Word No
Enhance No

Embedded No (does not apply to Wyse 50)

Auto Repeat No

Answer Back No

Label Line No (does not apply to Wyse 50)

Send ACKNo

Character MappingNo Default

Other Defaults

This section details defaults for other configuration settings.

Terminal Emulation

SB+ Mode On

Emulation Type VT220

File Transfer

Input Buffer Size Auto

Transfer Type SBZ

Acknowledge All Packets No

Timeout Auto Yes

Timeout Seconds 5, if Timeout Auto is not set

Line Protocol 8 Bits

ASCII Options

Character delay No default

Line delay No default

Font

Font Type System Builder

Size 7 x 14

ActiveX Fonts MS Sans Serif, 8

Toolbar

Name Default Bar

Status Line

Status Lights All Off

Time Off

Color Setup

Foreground Gray

Background Black

Note



Each attribute defaults to its description. For example, Blink and Underline and Reverse means that the blink, underline and reverse checkboxes will be on. All non erase attributes are green.

Nationality

Nothing set.

GUI Translation FileNo default

Miscellaneous

Cursor Shape Line

Blink On

Resize Behavior Resize font

Title BMP No default

Tab Stops No default

History Back Pages 10

Character Mouse No

SCRIPT

Login Script <No script>

Logout Script <No Script>

User Num No default

Password No default

User Id No default

User defined replacement strings No default

Session Behavior Normal

Print Options

Use Windows Printer Drivers No

Font SB Font, 12

Force Condensed Mode No

Maintain Rows Per Page No

String Box Characters No

OEM to ANSI Conversion Auto detect

GUI Parameters

Show Terminal Window Always

Triggers*I

*DEBUG

BREAK:

Timeout Period 0

Watchdog Period 0

Pad TCP Packets N0, 1024

AutoGUI No

Appendix C - Default Key Mappings



This appendix provides default key mappings for:

- VT220
- Wyse60/120
- Extra SB+ Server keys for SB+ Server terminal types

Note

"Appl+" means the emulation is in application mode.

| VT220 Key | PC Key | Character Sequence |
|----------------|-----------------|--------------------|
| DataTalk | Shift+F8 | |
| DelChar | Backspace | (8) |
| DelWord | Shift+Backspace | (127) |
| Do | Ctrl+F1 | (27)[29~ |
| DownArrow | DownArrow | (27)[B |
| Appl+DownArrow | DownArrow | (27)OB |
| F6 | F6 | (27)[17~ |
| F7 | F7 | (27)[18~ |
| F8 | F8 | (27)[19~ |

Appendix C - Default Key Mappings

| | | |
|----------------|-----------|----------|
| F9 | F9 | (27)[20~ |
| F10 | F10 | (27)[21~ |
| F11 | F11 | (27)[23~ |
| F12 | F12 | (27)[24~ |
| F13 | Shift+F10 | (27)[25~ |
| F14 | F5 | (27)[26~ |
| F17 | Shift+F1 | (27)[31~ |
| F18 | Shift+F2 | (27)[32~ |
| F19 | Shift+F3 | (27)[33~ |
| F20 | Shift+F4 | (27)[34~ |
| Find | Home | (27)[1~ |
| Help | Shift+F12 | (27)[28~ |
| HoldScreen | Shift+F5 | |
| InsertHere | Insert | (27)[2~ |
| LeftArrow | LeftArrow | (27)[D |
| Appl+LeftArrow | LeftArrow | (27)OD |
| NextScreen | PageDown | (27)[6~ |
| NumPad, | NumPad+ | , |
| Appl+NumPad, | NumPad+ | (27)OI |
| NumPad- | NumPad- | - |
| Appl+NumPad- | NumPad- | (27)Om |
| NumPad. | NumPad. | . |
| Appl+NumPad. | NumPad. | (27)On |
| NumPad0 | NumPad0 | 0 |

| | | |
|------------------|---------|--------|
| Appl+NumPad0 | NumPad0 | (27)Op |
| NumPad1 | NumPad1 | 1 |
| Appl+NumPad1 | NumPad1 | (27)Oq |
| NumPad2 | NumPad2 | 2 |
| Appl+NumPad2 | NumPad2 | (27)Or |
| NumPad3 | NumPad3 | 3 |
| Appl+NumPad3 | NumPad3 | (27)Os |
| NumPad4 | NumPad4 | 4 |
| Appl+NumPad4 | NumPad4 | (27)Ot |
| NumPad5 | NumPad5 | 5 |
| Appl+NumPad5 | NumPad5 | (27)Ou |
| NumPad6 | NumPad6 | 6 |
| Appl+NumPad6 | NumPad6 | (27)Ov |
| NumPad7 | NumPad7 | 7 |
| Appl+NumPad7 | NumPad7 | (27)Ow |
| NumPad8 | NumPad8 | 8 |
| Appl+NumPad8 | NumPad8 | (27)Ox |
| NumPad9 | NumPad9 | 9 |
| Appl+NumPad9 | NumPad9 | (27)Oy |
| NumPadEnter | Enter | (13) |
| Appl+NumPadEnter | Enter | (27)OM |
| PF1 | F1 | (27)OP |
| PF2 | F2 | (27)OQ |
| PF3 | F3 | (27)OR |

| | | |
|-----------------|-------------|---------|
| PF4 | F4 | (27)OS |
| PrevScreen | PageUp | (27)[5~ |
| PrintScreen | Shift+F6 | |
| Remove | Delete | (27)[3~ |
| Return | Enter | (13) |
| RightArrow | RightArrow | (27)[C |
| Appl+RightArrow | RightArrow | (27)OC |
| Select | End | (27)[4~ |
| SendBreak | Shift+Pause | NULL |
| SetUp | Shift+F7 | |
| UpArrow | UpArrow | (27)[A |
| Appl+UpArrow | UpArrow | (27)OA |

| Wyse 50, 60 and 120 Key | PC Key | Character Sequence |
|--------------------------------|----------------------|---------------------------|
| Backspace | Backspace | (8) |
| Shift+Backspace | Shift+Backspace | (8) |
| ClrLine | End | (27)T |
| Appl+ClrLine | End | (222) |
| ClrScrn | Shift+End | (27)Y |
| Appl+ClrScrn | Shift+End | (244) |
| Del | Ctrl+Backspace | (127) |
| Shift+Del | Shift+Ctrl+Backspace | (127) |
| DelChar | Alt+Delete | (27)W |
| Appl+DelChar | Alt+Delete | (221) |

| | | |
|-----------------|-----------------|----------|
| DelLine | Shift+Delete | (27)R |
| Appl+DelLine | Shift+Delete | (243) |
| Down | DownArrow | (10) |
| Appl+Down | DownArrow | (210) |
| Shift+Down | Shift+DownArrow | (10) |
| Appl+Shift+Down | Shift+DownArrow | (215) |
| ESC | Esc | (27) |
| Shift+ESC | Shift+Esc | (27) |
| F1 | F1 | (1)@(13) |
| Shift+F1 | Shift+F1 | (1)'(13) |
| F2 | F2 | (1)A(13) |
| Shift+F2 | Shift+F2 | (1)a(13) |
| F3 | F3 | (1)B(13) |
| Shift+F3 | Shift+F3 | (1)b(13) |
| F4 | F4 | (1)C(13) |
| Shift+F4 | Shift+F4 | (1)c(13) |
| F5 | F5 | (1)D(13) |
| Shift+F5 | Shift+F5 | (1)d(13) |
| F6 | F6 | (1)E(13) |
| Shift+F6 | Shift+F6 | (1)e(13) |
| F7 | F7 | (1)F(13) |
| Shift+F7 | Shift+F7 | (1)f(13) |
| F8 | F8 | (1)G(13) |
| Shift+F8 | Shift+F8 | (1)g(13) |

| | | |
|-----------------|---------------|----------|
| F9 | F9 | (1)H(13) |
| Shift+F9 | Shift+F9 | (1)h(13) |
| F10 | F10 | (1)I(13) |
| Shift+F10 | Shift+F10 | (1)i(13) |
| F11 | F11 | (1)J(13) |
| Shift+F11 | Shift+F11 | (1)j(13) |
| F12 | F12 | (1)K(13) |
| Shift+F12 | Shift+F12 | (1)k(13) |
| F13 | Ctrl+F5 | (1)L(13) |
| Shift+F13 | Shift+Ctrl+F5 | (1)l(13) |
| F14 | Ctrl+F6 | (1)M(13) |
| Shift+F14 | Shift+Ctrl+F6 | (1)m(13) |
| F15 | Ctrl+F7 | (1)N(13) |
| Shift+F15 | Shift+Ctrl+F7 | (1)n(13) |
| F16 | Ctrl+F8 | (1)O(13) |
| Shift+F16 | Shift+Ctrl+F8 | (1)o(13) |
| Home | Home | (30) |
| Shift+Home | Shift+Home | (27){ |
| Appl+Home | Home | (212) |
| Appl+Shift+Home | Shift+Home | (217) |
| Ins | Insert | (27)q |
| Appl+Ins | Insert | (245) |
| InsChar | Alt+Insert | (27)Q |
| Appl+InsChar | Alt+Insert | (220) |

| | | |
|--------------------|-------------------|-------|
| InsLine | Shift+Insert | (27)E |
| Appl+InsLine | Shift+Insert | (242) |
| Left | LeftArrow | (8) |
| Shift+Left | Shift+LeftArrow | (21) |
| Appl+Left | LeftArrow | (208) |
| Appl+Shift+Left | Shift+LeftArrow | (213) |
| NextPage | PageDown | (27)K |
| NumPad, | NumPadEnter | , |
| Shift+NumPad, | Shift+NumPadEnter | , |
| Appl+NumPad, | NumPadEnter | (204) |
| Appl+Shift+NumPad, | Shift+NumPadEnter | (204) |
| NumPad- | NumPad- | - |
| Shift+NumPad- | Shift+NumPad- | - |
| Appl+NumPad- | NumPad- | (205) |
| Appl+Shift+NumPad- | Shift+NumPad- | (205) |
| NumPad. | NumPad. | . |
| Shift+NumPad. | Shift+NumPad. | . |
| Appl+NumPad. | NumPad. | (206) |
| Appl+Shift+NumPad. | Shift+NumPad. | (206) |
| NumPad0 | NumPad0 | 0 |
| Shift+NumPad0 | Shift+NumPad0 | 0 |
| Appl+NumPad0 | NumPad0 | (176) |
| Appl+Shift+NumPad0 | Shift+NumPad0 | (176) |
| NumPad1 | NumPad1 | 1 |

| | | |
|--------------------|---------------|-------|
| Shift+NumPad1 | Shift+NumPad1 | 1 |
| Appl+NumPad1 | NumPad1 | (177) |
| Appl+Shift+NumPad1 | Shift+NumPad1 | (177) |
| NumPad2 | NumPad2 | 2 |
| Shift+NumPad2 | Shift+NumPad2 | 2 |
| Appl+NumPad2 | NumPad2 | (178) |
| Appl+Shift+NumPad2 | Shift+NumPad2 | (178) |
| NumPad3 | NumPad3 | 3 |
| Shift+NumPad3 | Shift+NumPad3 | 3 |
| Appl+NumPad3 | NumPad3 | (179) |
| Appl+Shift+NumPad3 | Shift+NumPad3 | (179) |
| NumPad4 | NumPad4 | 4 |
| Shift+NumPad4 | Shift+NumPad4 | 4 |
| Appl+NumPad4 | NumPad4 | (180) |
| Appl+Shift+NumPad4 | Shift+NumPad4 | (180) |
| NumPad5 | NumPad5 | 5 |
| Shift+NumPad5 | Shift+NumPad5 | 5 |
| Appl+NumPad5 | NumPad5 | (181) |
| Appl+Shift+NumPad5 | Shift+NumPad5 | (181) |
| NumPad6 | NumPad6 | 6 |
| Shift+NumPad6 | Shift+NumPad6 | 6 |
| Appl+NumPad6 | NumPad6 | (182) |
| Appl+Shift+NumPad6 | Shift+NumPad6 | (182) |
| NumPad7 | NumPad7 | 7 |

| | | |
|------------------------|------------------|-------|
| Shift+NumPad7 | Shift+NumPad7 | 7 |
| Appl+NumPad7 | NumPad7 | (183) |
| Appl+Shift+NumPad7 | Shift+NumPad7 | (183) |
| NumPad8 | NumPad8 | 8 |
| Shift+NumPad8 | Shift+NumPad8 | 8 |
| Appl+NumPad8 | NumPad8 | (184) |
| Appl+Shift+NumPad8 | Shift+NumPad8 | (184) |
| NumPad9 | NumPad9 | 9 |
| Shift+NumPad9 | Shift+NumPad9 | 9 |
| Appl+NumPad9 | NumPad9 | (185) |
| Appl+Shift+NumPad9 | Shift+NumPad9 | (185) |
| NumPadEnter | Ctrl+Enter | (13) |
| Shift+NumPadEnter | Shift+Ctrl+Enter | (13) |
| Appl+NumPadEnter | Ctrl+Enter | (186) |
| Appl+Shift+NumPadEnter | Shift+Ctrl+Enter | (186) |
| PrevPage | PageUp | (27)J |
| Print | Ctrl+F11 | (27)P |
| Repl | Delete | (27)r |
| Appl+Repl | Delete | (223) |
| Return | Enter | (13) |
| Shift+Return | Shift+Enter | (13) |
| Right | RightArrow | (12) |
| Appl+Right | RightArrow | (209) |
| Appl+Shift+Right | Shift+RightArrow | (214) |

| | | |
|-----------|-------------|-------|
| Send | Ctrl+F12 | (27)7 |
| SendBreak | Shift+Pause | |
| Tab | Tab | (9) |
| Shift+Tab | Ctrl+Tab | (27)I |
| Up | UpArrow | (11) |
| Appl+Up | UpArrow | (211) |

| SB+ Server Key | PC Key | Character Sequence |
|-----------------------|---------------|---------------------------|
| Appl+Shift+Up | Shift+UpArrow | (216) |
| SBPlus_InsertLine | Ctrl+Insert | (2). |
| SBPlus_BottomOfText | Ctrl+PageDown | (2)v |
| SBPlus_TopOfText | Ctrl+PageUp | (2)(32) |
| SBPlus_DeleteLine | Ctrl+Delete | (2)/ |
| SBPlus_BackTab | Shift+Tab | (2)(15) |

Appendix D - ASCII Code Table



This appendix lists the complete set of ASCII character codes. ASCII (American Standard Code for Information Interchange) characters are each represented by one byte according to a standard code. The "low-byte" characters (0 to 127) represent the alphabet and common typewriter symbols, as well as some special screen and printer control codes (such as a page feed). The extended or "high-byte" characters (128 to 255) have several different character mappings, allowing printers to output italics, graphics, international characters, and so on. Thus, only half of the ASCII character set is actually a universal standard. This is referred to as 7-bit ASCII, to distinguish it from the full byte.

ASCII is not designed to represent non-Western alphabets, such as Kanji. There are simply not enough codes to do these symbol sets justice.

It should be noted that Microsoft Windows uses the ANSI (American National Standards Institute) character set, which is not fully compatible with ASCII. See [“Appendix A - Internationalization Issues”](#) for more information.

ASCII Code Table

| ASCII | Dec | Hex | Ctrl | ASCII | Dec | Hex | ASCII | Dec | Hex |
|-------|-----|-----|------|-------|-----|-----|-------|-----|-----|
| NUL | 00 | 00 | @ | + | 43 | 2B | V | 86 | 56 |
| SOH | 01 | 01 | A | , | 44 | 2C | W | 87 | 57 |
| STX | 02 | 02 | B | - | 45 | 2D | X | 88 | 58 |
| ETX | 03 | 03 | C | . | 46 | 2E | Y | 89 | 59 |
| EOT | 04 | 04 | D | / | 47 | 2F | Z | 90 | 5A |
| ENQ | 05 | 05 | E | 0 | 48 | 30 | [| 91 | 5B |
| ACK | 06 | 06 | F | 1 | 49 | 31 | \ | 92 | 5C |
| BEL | 07 | 07 | G | 2 | 50 | 32 |] | 93 | 5D |
| BS | 08 | 08 | H | 3 | 51 | 33 | ^ | 94 | 5E |
| HT | 09 | 09 | I | 4 | 52 | 34 | _ | 95 | 5F |
| LF | 10 | 0A | J | 5 | 53 | 35 | ' | 96 | 60 |
| VT | 11 | 0B | K | 6 | 54 | 36 | a | 97 | 61 |
| FF | 12 | 0C | L | 7 | 55 | 37 | b | 98 | 62 |
| CR | 13 | 0D | M | 8 | 56 | 38 | c | 99 | 63 |
| SO | 14 | 0E | N | 9 | 57 | 39 | d | 100 | 64 |
| SI | 15 | 0F | O | : | 58 | 3A | e | 101 | 65 |
| DLE | 16 | 10 | P | ; | 59 | 3B | f | 102 | 66 |
| DC1 | 17 | 11 | Q | < | 60 | 3C | g | 103 | 67 |
| DC2 | 18 | 12 | R | = | 61 | 3D | h | 104 | 68 |
| DC3 | 19 | 13 | S | > | 62 | 3E | i | 105 | 69 |
| DC4 | 20 | 14 | T | ? | 63 | 3F | j | 106 | 6A |
| NAK | 21 | 15 | U | @ | 64 | 40 | k | 107 | 6B |

| ASCII | Dec | Hex | Ctrl | ASCII | Dec | Hex | ASCII | Dec | Hex |
|-------|-----|-----|------|-------|-----|-----|-------|-----|-----|
| SYN | 22 | 16 | V | A | 65 | 41 | l | 108 | 6C |
| ETB | 23 | 17 | W | B | 66 | 42 | m | 109 | 6D |
| CAN | 24 | 18 | X | C | 67 | 43 | n | 110 | 6E |
| EM | 25 | 19 | Y | D | 68 | 44 | o | 111 | 6F |
| SUB | 26 | 1A | Z | E | 69 | 45 | p | 112 | 70 |
| ESC | 27 | 1B | [| F | 70 | 46 | q | 113 | 71 |
| FS | 28 | 1C | \ | G | 71 | 47 | r | 114 | 72 |
| GS | 29 | 1D |] | H | 72 | 48 | s | 115 | 73 |
| RS | 30 | 1E | ^ | I | 73 | 49 | t | 116 | 74 |
| US | 31 | 1F | _ | J | 74 | 4A | u | 117 | 75 |
| SP | 32 | 20 | | K | 75 | 4B | v | 118 | 76 |
| ! | 33 | 21 | | L | 76 | 4C | w | 119 | 77 |
| " | 34 | 22 | | M | 77 | 4D | x | 120 | 78 |
| # | 35 | 23 | | N | 78 | 4E | y | 121 | 79 |
| \$ | 36 | 24 | | O | 79 | 4F | z | 122 | 7A |
| % | 37 | 25 | | P | 80 | 50 | { | 123 | 7B |
| & | 38 | 26 | | Q | 81 | 51 | | 124 | 7C |
| ' | 39 | 27 | | R | 82 | 52 | } | 125 | 7D |
| (| 40 | 28 | | S | 83 | 53 | ~ | 126 | 7E |
|) | 41 | 29 | | T | 84 | 54 | DEL | 127 | 7F |
| * | 42 | 2A | | U | 85 | 55 | | | |

Appendix E - Character Mapping and Translation Tables



You can use the character mapping feature of SBClient when character translations or mappings are required for proper application behavior. They can be specified in the advanced configuration dialog for all emulations or using **Setup > Nationality**.

There is a set of standard tables that ship with SBClient or the user can define their own files. If you have the correct keyboard for your country and the correct Windows operating system keyboard driver installed, there should be no need to make a selection from the Nationality parameter list. (ie choose a cvt file).

Three situations that require character mappings are when the host uses:

- an ASCII 7-bit national character set. For example, for the code 0x23, the UK ASCII 7-bit character set has the graphic £ instead of the # in standard ASCII (if you don't have a UK keyboard and have a US keyboard instead, and you want to enter a pound sign by pressing shift 3, you would select BRITISH).
- a non-Windows standard 8-bit character set. For example, there are two Cyrillic OEM character sets in common use in Russia. Windows supports one of them, so when the other is used a character mapping file, CYROEM.CVT, must be specified.
- To implement the workaround for the incorrect display of cyrillic characters on SBClient prior to release 4.4.3 as discussed in the previous section (Entering data that requires conversion from one character set to another).

Three types of translations can be specified for any terminal emulation: input, output and display (see Data Flow). Correspondingly, there are three sections allowed in a character mapping file with the headings:

[input] translates incoming data

[display] translates displayed data

[output] translates outgoing data

Any of these sections can be omitted or be empty.

Within each section, character translations can be described using the syntax:

lvalue=rvalue

This specifies that when *lvalue* is encountered, *rvalue* is substituted. No space characters are allowed before *lvalue* and both *lvalue* and *rvalue* must be specified using the standard C syntax for octal, decimal, or hexadecimal constants with values between 0 and 255 (decimal). The missing mappings are assumed to be identity mappings (that is, 0x00=0x00, 0xff=0xff). Comments may be introduced in a line with a semicolon. (125 (decimal), 0175 (octal), 0x70 (hex))

Below is an example of the British CVT file that would allow the US user to type the # key (shift 3 on a US keyboard) and generate the correct £ symbol that will be stored on an OEM host as char (156). A UK user viewing that data would then see the correct symbol as well.

```
;File :          BRITISH.CVT
;Description:    British ASCII OEM Character Set Conversion Table
```

```
[display]
```

```
.overlay
0x23=0x9c
```

```
[input]
```

```
0x9c=0x23
```

```
[output]
```

```
0x23=0x9c
```

Data Flow

The specified input and output translations can be inserted into the following diagram in any of the three possible positions: before, after or instead of the corresponding default translations:

Input: Host> Default input translations> Emulator

Display:Emulator>Default display translations> User screen

Output: Keyboard> Default output translations> Host

The following keywords can be used to specify the position of the specified translations:

`.first` insert specified translations before the default (internal) translations

`.last` insert specified translations after the default translations

`.exclusive` the specified translations replace the default translations

In the display translation, the following keywords can be used:

`.pure` the specified translations replace the default translations

`.overlay` each individual specified character translation is added or replaces an individual default character translation

A keyword can appear anywhere in a section; the last one to occur takes effect if there is more than one.

Translation Support

It is also possible to assign a translation file to SBClient to dynamically translate characters in GUI mode. The concept is similar but more functional than the CVT files used in character mode. To assign a translate file to a session, select the Setup Nationality screen and under the "GUI translation File" section, enter or select a file.

It is preferable to locate your file in the `..\SBOffice\SBClient\TU` directory and then not specify the path in the "GUI translation File" field. This will ensure that there is no dependency on a hard coded path.

The file is similar in format to the CVT files. The file comprises of an [INPUT], [OUTPUT] and [DISPLAY] sections.

The [input] section is applied at the point just before data is displayed in any GUI field.

The [output] section is applied just after data is retrieved from any GUI field.

The [display] section is applied every time a ctrl or alt key is pressed in combination with a regular key when on a GUI field. This means eg you could map umlaut to be ctrl u.

The characters can be mapped in the same way as the CVT files using hexadecimal character numbers. The characters can also be mapped using simple explicit assignments e.g. a=b

Each mapping must be on a new line. The [input] and [output] mappings may be one to many e.g. a = test or many to one e.g. test=a. the [Display] section can only be used to map ctrl or alt characters to other characters. Ctrl is designated by ^ and alt is designated by ~.

Example

```
[INPUT]
E^o=0xF1
A=MRK
```

```
[Output]
MRK=A
0xF1=E^o
```

```
[Display]
^G=0xF1
~E=242
```

This functionality can be used to replace the OEM to ANSI conversions that generally need to be performed when the host is OEM and the windows font is ANSI. A table can be set up to do the mapping which then bypasses Windows' internal representation of what it thinks the OEM code page is. Bypassing this internal representation solves the problem of different national versions of windows having different internal representations of code pages. This led to problems when having a host in one country being accessed by clients in multiple countries.

Appendix F - SBOPEN.I NI Initialization File



The SBOPEN.INI file contains parameters used to configure SBClient. It has a common format:

```
[sectiontitle]
name=value
```

The sectiontitle is dependent upon the application being used, but is outlined for each product described below.

The resolution of section names is case insensitive. Whether or not the data within a given section is case sensitive will depend upon the application and the operating system being used.

You may add a comment anywhere within the INI file by placing a semicolon in front of the first non-blank character on the line. For example:

```
; This is a comment
```

Section Information

Warning

Variables which are not listed here should not be modified. Doing so may cause SBClient to cease functioning. It is recommended that if you need to update any of these settings that you do this via **Setup > Environment Variables**.

The following describe the contents of each section within SBOPEN.INI:

```
[ENVIRONMENT]
```

| | |
|----------------|---|
| NOANSIOEMXFER | Set this parameter to 1 when you want to transfer high bit characters to and from SBClient and the host. For example, if you are exchange data to EXCEL with DDE and you do not want SBClient to convert ANSI to OEM or OEM to ANSI, set noansioenxfer to 1. |
| OLDDDETOKEN | Set this parameter to 1 when you are using SBClient with SB+ Server Release 3.3.2.4 or earlier. |
| SBAPPPATH | The location of the application specific forms, classes and Windows resources (bitmaps, cursors and help files). |
| SBOCHARSET | The default host character set (ANSI, OEM). |
| SBODATE | The type of date formatting to be used in the default conversion of dates. If specified, it can have two values: AMERICAN or INTERNATIONAL. If not specified, it will default to INTERNATIONAL. |
| SBQUERYPATH | The location of the TUQUERY file. Set this variable so all users on a network can use the same stored queries. |
| SBSTATEPATH | The location of the files that cache the host GUI forms when SB+ GUI or SBClient HostGUI are in operation. This can be changed to point to any valid SBClient account, including any user account on the PC's hard drive. The default value is set to the same value as SBTUPATH. Set this parameter when your PC is in a networked environment and: <ul style="list-style-type: none">• connected to a read-only server, or• connected through slow devices (for example WANs, or modem lines) |
| SBTUPATH | The location of the SBClient installation directory. This should never be changed. |
| SINGLEINSTANCE | Determines whether SBClient can only run one instance (this is 1 by default to allow only one instance). |
| SINGLESESSION | When this and SINGLEINSTANCE are both set then only one session can be launched. Whereas SINGLEINSTANCE refers to the entire instance of SBClient, SINGLESESSION prevents multiple sessions from being launched within a single instance of SBClient. |

NEWFONTCALC

This can be used to toggle between the new faster calculation and the old type of calculation. This variable should only be set to false if users are having font sizing problems using the new font calculation algorithm.

Appendix G - Command Line Options



This appendix lists the options that can be appended to the SBClient command line.

The syntax is:

```
[sbobinpath\]SBClient[.exe] [ SBCLIENT sessionname [ @loginscript]] [ -options]
```

where:

sbobinpath

The path to the location of the SBOpen program directory (the default is: \SBOFFICE\SBOPEN\BIN).

SBCLIENT *sessionname*

The name of a saved session configuration. Use underscore characters instead of space characters within the session name. For example: NT_to_minotaur. You can specify multiple SBCLIENT *sessionname* @*loginscript* pairs.

loginscript

This overrides any login script defined for the session. It must be preceded by an @ to identify it as a script. An @@ will run this script with the S option. Use underscore characters instead of space characters within the script name.

options

Options are:

- B Invokes the AutoGUI IDE.

- D Diagnostic mode. Allow the user to break into the Data/C++ debug prompt.
- H Hides the session display until the host application restores it.
- M Keeps the session minimized.
- N Ignores the saved SBClient Office settings.
- S Minimizes the session display until the script is finished. If there is no script, minimizes the session display until connected.
- X Starts SBClient with no menus. Use this in conjunction with scripts to prevent users gaining access to restricted menu options, such as file manager.

If you are using SB+ GUI or SBClient Host GUI, the session window is hidden. You can restore the session by choosing **Switch to Terminal** from the system menu of any SB+/Host GUI dialogs or forms.

Appendix H - Error Codes



This appendix provides explanations of the error codes built into SBClient.

General

- 1 Wrong configuration data. Configuration string was not recognized or was formed incorrectly.
- 2 Session already started. A TERM session has already been started (warning).
- 3 Can't load generic TE module. Error loading generic TERMINAL EMULATION module (E_GEN.DLL). Check your installation.
- 4 Can't load generic FT module. Error loading generic FILE TRANSFER module (X_GEN.DLL). Check your installation.
- 5 TE initialization failed. Cannot initialize generic terminal emulation module.
- 6 Cannot load generic comm library.
- 7 Cannot load addresses of comm's function.
- 8 Comm not ready. TERM session hasn't been successfully set up before starting this operation. As a result comm module is not ready to operate. Try to configure session first.
- 9 Can't initialize specified comm module. A new communication channel cannot be opened. It's possible that the limit of available channels for the communication method that you specified has been reached. Please also check your transport system installation.
- 10 Swapping current session failed.

- 11 Can't configure static parameter. After successful start of a session, COMM module cannot be changed. It's considered to be session static parameter.
- 12 Filter name is wrong. Terminal emulation or file transfer name specified was not accepted. Check if the name was correct.
- 13 TE session creation failed. Terminal emulation session creation failed. One of the reasons could be that TE cannot allocate memory for a new session.
- 14 FT session creation failed. File transfer session creation failed. One of the reasons could be that FT cannot allocate memory for a new session.
- 15 Filter's open session failed. Cannot open file transfer session. External conditions to open a file transfer session are not satisfied.
- 16 Can't add filter to already opened session.
- 17 Session has not been defined yet. The module that you are trying to configure was not specified as a current session element. For instance, you are trying to configure TE when FT is a current session element.
- 18 Wrong TERM's interface number requested. Wrong interface number was used to initialize TERM. Please upgrade TERM or ULATOR as the version you are using may not be fully functional.
- 19 TERM was not initialized. TERM wasn't initialized properly. The Generic comm module was not found or not loaded.
- 20 Wrong session number.
- 21 Can't allocate memory for a new session. TERM was not able to allocate enough memory to create a new session. Try to free some memory by exiting some applications or closing another session.
- 22 Invalid Windows handle. A handle that was used to create a session was invalid.
- 23 Break operation failed or not implemented. The SendBreak operation failed in communication module. There may be a hardware problem.
- 24 Comm or filter returns FALSE on read. Terminal emulation or File Transfer reports error on read operation.

-
- 25 The operation would block if in blocking mode TERM was busy. An application should wait until a message TERM_WRITE is posted or TERM was busy waiting for data to arrive. An application should wait until a message TERM_READ is posted. Please note that this error may occur when you call suReceiveData right after suSendBreak.
 - 26 Wrong parameters on function call. Parameters in function call were faulty. Please check values you pass to TERM before calling again.
 - 27 A session must be started before. Session was established but a StartSession command is required to start normal operation. Operation was not executed.
 - 28 Comm or filter returns FALSE on write. Data cannot be sent. Please check the number of bytes specified as it might happen that nothing was actually sent.
 - 29 Wrong context operation rejected An attempt to change an emulation to a different one was not successful at this moment or the actual session configuration doesn't allow the operation. Please repeat this operation sometime later or check your session configuration.
 - 30 Paused. Terminal emulation was waiting for synchronization signal when SendBreak, ReceiveData or SetConfiguration was called. Call SendInfo before calling SendBreak.
 - 31 Configure session failed on comm.
 - 32 Connection refused. Opening a comm connection failed. Wrong parameters were used or underlying transport is not operating.
 - 33 Configuration buffer was too small. An application must supply a buffer where TERM will copy an appropriate configuration string. Unfortunately the buffer supplied in function call was not sufficient to keep the whole configuration string.
 - 34 Wrong comm handle. Communication channel was not successfully opened and can't be configured.
 - 35 Wrong TE handle. Terminal emulation session was not successfully opened and can't be configured.
 - 36 Wrong FT handle. File transfer session was not successfully opened and can't be configured.
 - 37 Get configuration failed in module. Function failed on get configuration in addressed module.
 - 38 Capture failed. Start capture operation failed in COMM module.

Communications

- 257 Buffer supplied is too small.
- 258 Comm module must be in "suspended" mode.
- 259 The session has been connected.
- 260 The session in not connected.
- 261 Bad communication channel handle.
- 262 Operation would block on non-blocking socket.
- 263 The operation has failed.
- 264 Operation not supported for that interface.
- 265 Bad configuration string received.
- 266 The operation not possible in current state.
- 267 The operation has already been in progress.
- 268 Wrong parameter passed in function call.
- 269 Invalid session descriptor.

Terminal Emulation

- 513 Failed to initialize interface.
- 514 Cannot create session.
- 515 Cannot open session.
- 516 Cannot close session.
- 517 Error receiving data.
- 518 Internal error.
- 519 Error sending data.
- 520 Internal parser error.

-
- 521 Cannot reset session.
 - 522 Cannot reset emulation.
 - 523 Attempt to set configuration failed.
 - 524 Attempt to get configuration failed.
 - 525 Attempt to get capabilities failed.
 - 613 Invalid SB string pointer.
 - 614 Invalid length of SB string.
 - 615 Memory allocation failure.
 - 616 Invalid item <<item>.
 - 617 Wrong item name or invalid value <<item>.
 - 618 Static item <<item>.
 - 619 Wrong item name <<item>.
 - 620 Cannot set <<item>.

File Transfer

- 1025 Invalid Session.
- 1026 Invalid Configuration.
- 1027 Unable to Create Session.
- 1028 Module Not Found.
- 1029 Cannot Load Communication Module Interface.
- 1030 Communication Module is Not Connected.
- 1031 Read From COM Failed.
- 1032 Send To COM Failed.
- 1033 Memory Lock Failed.
- 1034 Memory Allocation Failed.

- 1035 Buffer too small or corrupted.
- 1036 Module Has Not Been Loaded.
- 1037 Null pointer in parameter.
- 1038 Temporarily unable to send.
- 1039 Connection has been lost.
- 1040 Communication module failed.
- 1041 Receive Buffer Too Small.

Host Library

- 1 Cancelled at user request.
- 2 Unable to open PC file.
- 3 Unable to read from PC file.
- 4 General communications failure.
- 5 Retry limit exceeded.
- 6 Unable to write to PC file.
- 7 Function/Option not supported.
- 8 Exists on file.
- 9 Upload field is correlative.
- 10 Unknown error.
- 11 Item already exists in File.
- 20 Too many items.
- 24 Invalid dictionary item.
- 101 Unable to open specified database file.
- 102 Unable to delete database file.
- 201 Unable to open host file TUXFER.DATA.<port no>.

-
- 202 Unable to read database item from database file.
 - 203 Unable to write database item to database file.
 - 204 Invalid selection criteria specified.
 - 205 Unable to delete database item from database file.
 - 206 Invalid fields list.
 - 207 Invalid application specified.
 - 401 No items found in database file.
 - 905 File transfer aborted by user.
 - 950 Conversion error; corrupt source file.
 - 951 Invalid conversion class specified.
 - 952 Conversion error; unable to open target file on PC.
 - 953 Conversion error; unable to open source file on PC.
 - 954 Conversion error; invalid header block in target file.
 - 955 Invalid transfer; database record written do not match number of records in source file.
 - 956 Conversion error; unable to write to target file.
 - 958 Communications error.
 - 959 Target file already exists.
 - 960 Communications breakdown.
 - 960 Invalid Remote Object Call Server request.
 - 961 File open error.
 - 962 File write error.
 - 964 Remote Object Call Server error.
 - 965 Invalid option.
 - 966 Conversion object error.

- 967 OEM to ANSI conversion error.
- 999 Unable to open database file TUSTATE.
- 5000 Invalid database dictionary item.
- 5001 Invalid database file item id.
- 5002 Invalid DOS file used.
- 5003 Invalid directory used.
- 5004 Invalid database dictionary item id.
- 6001 FTP error; unable to load winsock interface.
- 6002 FTP error; login failure.
- 6003 FTP error; connection failure.
- 6004 FTP error; operation failure.
- 6005 FTP error; operation aborted.
- 6006 FTP error; winsock stack error.
- 6007 FTP error; unable to access file.
- 6008 SBZ error; End of file encountered.
- 6009 SBZ error; protocol error detected.
- 6010 SBZ error; configuration error.
- 6011 SBZ error; unable to start file transfer.
- 6012 SBZ error; internal error.
- 6014 SBZ error; invalid file pointer position.
- 7001 DDE error; unable to open transportation file.
- 7002 DDE error; unable to write to transportation file.
- 7003 DDE error; unable to close transportation file.

SB OLE Server - SBClient errors

SBClient can return various error codes to the calling code of the client application. The errors which are returned can be the result of an error that took place within SBClient itself, or the host or a COM error in the Windows interface.

- 0 This is not an error.
- 1 SBClient Server cannot be found.
- 3 The configuration could not be read.
- 5 Option has not initialized a value or invalid option.
- 6 A window handle has not been specified for debug.
- 7 It was not possible to create the debug window or an invalid window handle was passed.
- 2 Server state=2 - server was not ready to accept the request.
- 3 Server state=3 - server was not ready to accept the request.
- 4 Server state=4 - server was not ready to accept the request.
- 5 Server state=5 - server was not ready to accept the request.
- 6 Server state=6 - server was not ready to accept the request.
- 7 Server state=7 - server was not ready to accept the request.

SB OLE Server - Host Errors

- 1 Invalid state or invalid event.

SB OLE Server - Windows COM Errors

Errors are returned to the method return variable (see fx column in table) or to one of the parameters (see Param column in table) specified in the method call by the client application code. The following table lists both types of error return values.

| Methods | OK | | Timeout | | Memory Problem | | Other Reasons | |
|---------------------|------|-------|---------|-------|----------------|-------|---------------|-------|
| | fx | Param | fx | Param | fx | Param | fx | Param |
| StartServer() | Data | Data | -100 | Null | -101 | Null | -102 | Null |
| SendEvent() | Data | Data | -100 | Null | -101 | Null | -102 | Null |
| CheckServer-State() | Data | N/A | -100 | N/A | -101 | N/A | -102 | N/A |
| SetOptions() | Data | N/A | -100 | N/A | -101 | N/A | -102 | N/A |
| GetOptions() | Data | N/A | -100 | N/A | -101 | N/A | -102 | N/A |
| ShutdownServer() | N/A | N/A | N/A | N/A | N/A§ | N/A | N/A§ | N/A§ |

Index



A

About SBClient, [19](#)
ADDS ViewPoint, [66](#), [67](#)
ANSI, [66](#), [69](#), [252](#), [254](#)
 auto wrap, [69](#)
 character mapping, [69](#)
 smooth scroll, [69](#)
Answer After Rings, [58](#)
Auto Answer, [58](#)
Auto Scroll, [68](#)
Auto Wrap, [68](#)

B

Baud rate, [42](#), [54](#)

C

Character
 mapping, [68](#)
Character sets, [252](#)
chcp, [255](#)
Clipboard, [145](#), [149](#)

 recording screen activity to, [181](#)
 setup, [177](#)
 using, [177](#)
 viewer, [33](#)
Close, [25](#)
Code pages, [252](#)
Color, [19](#), [40](#)
Color Setup
 attribute list, [98](#)
 colors, [98](#)
 display attributes, [99](#)
Command line, [30](#), [32](#)
Communication, [18](#), [39](#)
 setup, [41](#)
Communication setup, [41](#)
 modem, [53](#)
 network, [46](#)
 serial, [41](#)
Configuration
 modem, [55](#)
Connect string, [56](#)
Contents, help, [243](#)
conventions, [12](#)
Copy, [28](#)
Criteria

- selection, 185
- sort, 189

Cursor

- home at bottom, 68

CVT files, 257, 258, 262

D

Data bits, 42, 54

Data file transfer, 145

Data formats

- ASCII, 161, 171
- binary, 161, 171
- CSV, 160, 170
- dBase, 161, 170
- fixed length, 160, 170
- hexadecimal, 161, 171
- space delimited, 160, 170
- SYLK, 161, 171
- tab delimited, 160, 170

Data transfer

- host to PC, 154
 - defining a, 155
 - defining display fields, 159
 - defining selection criteria, 155, 155
 - defining sort criteria, 155, 158
 - defining transfer format, 160
 - deleting a, 165
 - load a, 165
 - running a, 163
 - saving a, 164
- PC to host, 167
 - defining a, 167
 - defining fields, 169
 - defining host record IDs, 168
 - defining transfer format, 170
 - deleting a, 174

- running a, 171
- saving a, 172
- selecting transfer mode, 171

DDE, 145, 150

Defining data transfer

- translation, 162

Desktop

- saving, 32

Detect Busy Tone, 57

Detect Dial Tone, 57

Dial

- pulse, 57
- retries, 57

Dialing Directory, 19, 29, 40, 60

- host login details, 64

Dialing Directory dialog, 60

Dialing Directory Maintenance dialog, 61

Display area, 33

Display Help Prompts, 243

Drop DTR to Hangup, 58

Dynamic Data Exchange, 150

E

Environment Variables, 40

Exit, 19

Expiry Date, 245

Export configuration, 129

Export Configuration dialog, 128

F

File transfer, 40

- data, 145
- standard, 145
 - ASCII, 148

- XModem, [148](#)
- XModem-1K, [148](#)
- XModem-1K-G, [148](#)
- YModem, [148](#)
- YModem-G, [148](#)
- ZModem, [148](#)
- File Transfer Setup, [89](#)
 - ASCII options, [93](#)
 - ftp options, [90](#)
 - input buffer size, [90](#)
 - network copy options, [91](#)
 - progress bar, [93](#)
 - SBZ options, [92](#)
 - transfer type, [90](#)
 - troubleshooting, [93](#)
- Files
 - downloading, [175](#)
 - uploading, [175](#)
- Font, [40](#)
 - netxt size up, [29](#)
 - next size down, [29](#)
 - resize, [34](#)
- Font Setup
 - ActiveX fonts, [96](#)
 - font, [96](#)
 - size, [96](#)
- Footers
 - creating, [191](#)
- Force condensed mode
 - 132 columns, [87](#)
- Full screen, [33](#)

G

- GUI parameters, [40](#)
- GUI Parameters Setup, [118](#)
 - auto GUI, [120](#)

- show terminal window, [118](#)
- timeout period, [119](#)
- triggers, [119](#)
 - watchdog period, [119](#)
- GUI screen, [86](#)
- GUI Translation File, [111](#)
- GUI windows, [86](#)

H

- Hangup string, [56](#)
- Headers
 - creating, [191](#)
- Help
 - pointer, [29](#)
- Help menu
 - About SBClient, [243](#)
 - Contents, [243](#)
 - Display Help Prompts, [243](#)
 - How to Use Help, [243](#)
 - Troubleshooting, [244](#)
 - View Readme, [243](#)
- Host Login Details, [64](#)
 - password, [65](#)
 - script name, [64](#)
 - script replacement strings, [65](#)
 - user ID, [64](#)
 - user name, [64](#)
- How to Use Help, [243](#)

I

- Import Configuration dialog, [126](#)
- Import data type, [127](#)
- IN4407, [66](#), [70](#)
 - Auto wrap, [70](#)

character mapping, [71](#)
Initialization string, [56](#)
ions, [244](#)

K

Key, [244](#)
Keyboard, [19](#), [40](#)
Keyboard Setup, [100](#)
Keyboard, unlock, [244](#)

L

Log Options, [244](#)
Login scripts, [206](#)
Logout Script, [113](#)
Logout scripts, [206](#)

M

Maintain rows per page, [87](#)
Menu bar, [20](#), [27](#)
Miscellaneous, [40](#)
Miscellaneous Setup, [115](#)
 back pages, [116](#)
 blink, [115](#)
 character mouse, [117](#)
 insert tab every characters, [116](#)
 mode, [115](#)
 shape, [115](#)
 tile BMP, [116](#)
Model, [56](#)
Modem Communications Setup dialog, [55](#)
Modem configuration, [55](#)

N

Nationality, [40](#)
Nationality Setup, [109](#)
 nationality, [110](#)
New Session, [25](#)
No connect strings, [57](#)

O

OEM, [252](#)
OEM to ANSI conversion, [88](#)
Organization, [244](#), [245](#)

P

Pad TCP Packets, [119](#)
Parity, [42](#), [54](#)
Paste, [28](#)
Phone books, [128](#)
Port, [41](#), [54](#)
Port, reset, [244](#)
Print options, [86](#)
Print screen, [86](#)
Print setup, [86](#)
Printer drivers, [86](#)
Printing, [86](#)
 GUI screen, [86](#)
 GUI windows, [86](#)
 print options, [86](#)
 print screen, [86](#)
 print setup, [86](#)
 text, [86](#)
PRISM 8, [66](#), [71](#)
 Auto wrap, [71](#)
 character mapping, [72](#)

Smooth scroll, [71](#)
Pulse dial, [57](#)

R

Readme, view, [243](#)
Re-Connect, [25](#)
Register
 registration details, [36](#)
 SBClient, [35](#)
Registration Details, [245](#)
Reset Port, [244](#)
Reset string, [56](#)
Reset Terminal, [244](#)
Retry Every, [57](#)

S

SB+ mode, [67](#)
SBOpen.ini file, [129](#)
Screen
 size, [33](#)
Screen Log Options, [244](#)
Screen mode, [33](#)
Script, [40](#)
 auto script learner, [204](#)
 creating, [199](#)
 advanced scripts, [214](#)
 editing, [209](#)
 functions, [214](#)
 learning responses, [212](#)
 login, [206](#)
 logout, [206](#)
 maximum hits, [202](#)
 running, [210](#)
 trigger responses, [201](#)

 trigger sequences, [201](#)
Script Setup, [112](#)
 login script, [112](#)
 password, [113](#)
 session behavior while running script, [113](#)
 user defined script replacement strings, [113](#)
 user ID, [113](#)
 user name, [113](#)
script_case_sensitive, [215](#)
script_clear, [215](#)
script_end, [216](#)
script_exit, [216](#)
script_max_retries, [218](#)
script_monitor, [218](#)
script_retrigger_timeout, [218](#)
script_run, [219](#)
script_started, [220](#)
script_trigger_delay, [220](#)
Scripts, [128](#)
Scroll
 auto, [68](#)
 back a page, [28](#)
 forward a page, [28](#)
 smooth, [67](#)
Send Break, [244](#)
Serial Number, [244](#), [245](#)
Session
 area, [27](#)
 closing, [24](#)
 configuration deleting, [133](#)
 configuration, changing, [129](#)
 configuration, creating new, [129](#)
 configuration, importing, [126](#)
 configurations, [128](#)
 configurations, maintaining, [125](#)
 name, [27](#)
 opening, [22](#)

- Opening additional sessions, 23
- views, 32
- window, 27
 - back page history, 182
 - copying, 177
 - copying selected text only, 178
- window resizing, 30, 31
- Session Name List, 18, 18
- Setup
 - color, 28
 - dialing directory, 29, 60
 - font, 29
 - GUI parameters, 118
 - keyboard, 28, 100
 - miscellaneous, 115
 - nationality, 109
 - script, 112
 - status line, 107
 - terminal, 28, 66
 - toolbar, 102
- SM9400, 66, 72
 - attribute
 - character, 73
 - line, 73
 - auto scroll horizontal, 74
 - auto scroll vertical, 74
 - auto wrap, 73
 - character mapping, 74
 - screen size, 73
 - smooth scroll, 73
 - video attributes, 74
- Smart query, 183
 - building a new query, 185
 - defining a, 197
 - directing output, 192
 - recalling stored query, 193
 - running a, 196

- saving a, 195
- Smart Query Definitions, 40
- Smooth scroll, 67
- Speaker
 - mode, 56
 - volume, 56
- Standard file transfer, 145
- Status line, 30, 32, 40
- Status Line Setup, 107
 - sample, 108
 - status lights, 107
 - time, 108
- Stop bits, 42, 54
- Strip box characters, 87
- Support Information, 244
- Switch, 19

T

- Terminal, 19, 39
 - setup, 66
- Terminal Setup
 - emulation, 66
 - SB+ mode, 66
- Terminal Setup dialog, 66
- Terminal, reset, 244
- text, 86
- Toolbar, 28, 32, 40, 128
 - creating a, 105
 - modifying a, 102
- Toolbar Setup, 102
- Troubleshooting, 244
 - A GUI form hangs, 249
 - cannot connect to host, 249
 - extraneous characters in session window, 250
 - Losing characters in screen display, 250

U

- Unicode, 253
- Unix, 250
- Unlock Keyboard, 244
- User Count, 245
- User Limit, 245
- User Name, 244, 245
- Utilities
 - file manager, 29
 - form painter, 29

V

- View
 - full screen, 29
- VT220, 76
 - ANSI color, 77
 - auto wrap, 77
 - display control characters, 77
 - keypad, 77
 - new line, 77
 - personality, 77
 - screen size, 76, 76
 - smooth scroll, 76
- VT220 Configuration dialog, 76
- VT220, VT52, VT100
 - 8 bit codes, 79
 - auto print, 79
 - auto repeat, 79
 - character mapping, 79
 - cursor key, 79
 - insert, 78
 - local echo, 78
 - origin, 78
 - print extent, 79
 - print FF, 79

- reverse, 78
- VT52 and VT100, 74
 - ANSI color, 75
 - auto wrap, 75
 - display control characters, 75
 - keypad, 75
 - new line, 75
 - screen size, 75
 - smooth scroll, 75
- VT52, VT100 and VT220, 66

W

- Wait For Carrier, 57
- Windows
 - printer drivers, 86
- Wyse 50, 60 and 120
 - answer back, 85
 - attribute, 82
 - auto font load, 83
 - auto repeat, 85
 - auto scroll, 83
 - auto wrap, 84
 - character mapping, 85
 - embedded, 84
 - enhance, 84
 - enter, 81
 - insert, 82
 - label line, 84
 - monitor, 84
 - received CR, 81
 - return, 81
 - screen size, 81
 - send ack, 85
 - smooth scroll, 83
 - status line, 84
 - wyse word, 84

Wyse 50, 60, and 120, [66](#)

X

XModem, [148](#)

XModem-1K, [148](#)

XModem-1K-G, [148](#)

Y

YModem, [148](#)

YModem-G, [148](#)

Z

ZModem, [148](#)



THE ART OF DATA MANAGEMENT

Using SBClient

DSC-5002