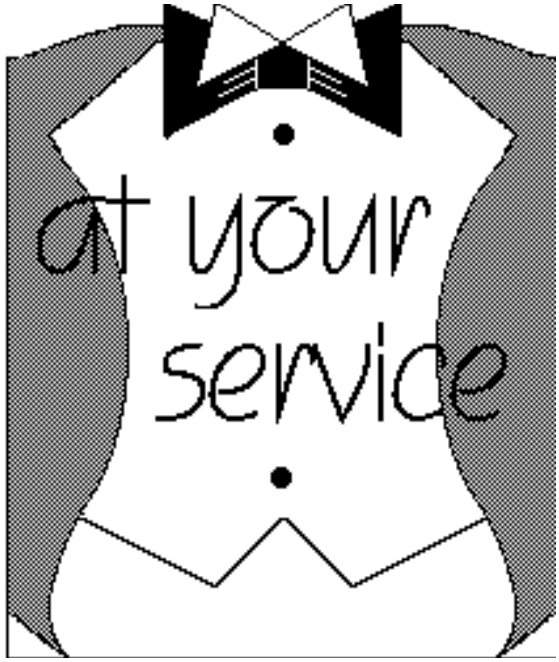


AYS

At-Your-Service
HyperCard
Communications Package

Version 1.3

18th Febuary 1994

Contents

About AYS

Abstract

At Your Service is a HyperCard communications package that provides a series of unique stack based communication facilities.

The package is designed around the concept of a 'service', which is defined as any process running on a local or remote computer, eg. an Online Public Access Catalogue (OPAC) or a Campus Wide Information System (CWIS). A service is uniquely defined via a series of 15 parameters that describe attributes such as the name, service type, the method used to connect to the service, the login and logout scripts, etc.

The AYS Installer stack represents the hub of the package since it is responsible for maintaining service definitions by providing facilities for creation, storage and testing. As the name implies the stack is also responsible for installing/removing definitions into/from other stacks, therefore enabling the creation of custom built communication stacks or the addition of communications facilities to existing stacks. For example, AYS has been used to create an Inter-library loans database and mail system.

The second stack included with the package, the AYS Example stack, is a stand-alone communications 'application' that was generated in just such a fashion. The stack is intended both as an example of what can be achieved as well as a useful, ready-to-use utility.

Installed service definitions allow for multiple, simultaneous sessions to be active at any one time. A typical session might therefore include a serial connection to your local OPAC in one window, whilst another window displays information from a CWIS, and a third displays a remote network connection to a second OPAC.

The package utilises the power of HyperCard's scripting language, HyperTalk, to provide a powerful communication scripting facility. It achieves this by defining a series of communication specific language extensions that enable scripts to communicate directly with services.

AYS is based upon Apple's Communications Toolbox technology, consequently connectivity is only limited by the range of Connection and Terminal Emulation tools that are available on your machine.

Although originally intended as a tool for librarians to connect to remote OPAC's it can as easily be used for other purposes including production control, development and administration.

How to use this Manual

This manual has two major parts to it; a section on the AYS Example stack and a section on the AYS Installer stack. Each section contains an introduction which is followed by a detailed explanation of the menus, fields and buttons associated with that tool.

If you have not already installed AYS then you should read the section 'Installing AYS'.

Users should read the section on 'The AYS Example Stack'.

Installers should also read the section 'The AYS Installer Stack'.

Software developers should also read the section 'Technical Considerations' and the Appendices.

Contact

Please direct all communications to: Postal: Tim Barlow.
PO Box 607
Sandy Bay TASMANIA 7005
AUSTRALIA

E-mail: tim.barlow@lib.utas.edu.au

Registering your Copy

AYS is a shareware product, it is not free.

The distributed versions of the AYS Example stack and the AYS Installer stack will only launch communication sessions with a time duration of 5 minutes, after which they will be automatically closed down. To remove these restrictions you are obligated to pay the registration fee. Upon registration, you will be sent a password that may be used to nullify these restrictions. You will be notified of future versions as they are released.

Disclaimers and Limitations

DISCLAIMER OF WARRANTY ON SOFTWARE

THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, AND TIM BARLOW EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OR CONDITIONS, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. TIM BARLOW DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF THE RESULTS OF THE USE OF THE SOFTWARE OR ANY ACCOMPANYING WRITTEN MATERIALS IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS, OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE AND WRITTEN MATERIALS IS ASSUMED BY YOU. TIM BARLOW DOES NOT WARRANT THAT THE SOFTWARE WILL WORK CORRECTLY IN YOUR MULTI-USER ENVIRONMENT. IF THE SOFTWARE IS DEFECTIVE, YOU AND NOT TIM BARLOW OR HIS DEALERS, DISTRIBUTORS, AGENTS, OR EMPLOYEES, ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

LIMITATION OF LIABILITY

IN NO EVENT WILL TIM BARLOW, OR HIS DEVELOPERS, DIRECTORS, OFFICERS, EMPLOYEES, OR AFFILIATES BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OR INTERRUPTION IN THE AVAILABILITY OF DATA, STOPPAGE OF OTHER WORK OR IMPAIRMENT OF OTHER ASSETS), WHETHER FORESEEABLE OR UNFORESEEABLE, ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE OR ACCOMPANYING WRITTEN MATERIALS, REGARDLESS OF THE BASIS OF THE CLAIM AND EVEN IF TIM BARLOW OR AN AUTHORIZED REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISTDICTIONS DO NOT ALLOW THE EXCLUSION ON LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Installing AYS

What the AYS Package Includes

The AYS Example Stack

The AYS Installer Stack

The TGE TCP Tool

This User Manual

What you need to use AYS

The AYS system has been tested on System 7.0+ Macintoshes. It uses HyperCard version 2.0 or later. In order to use the full range of facilities provided by the AYS installer you will need to be able to set the user level of HyperCard to 5 (scripting). To use the TGE TCP tool you require MacTCP to be installed.

How to install AYS

You should check that your Macintosh meets the requirements stipulated above. If any problems occur please consult the section 'Installation Problems'. You should have a folder which contains the AYS package. See what the AYS Package includes and check that all the parts are present.

Installing the Communications Tool Box Tools

The installation of AYS involves dropping the included tools on the system folder.

Open the folder 'Comms Tools' in which the tools are stored. Select all the tools and drag them and drop over the system folder. The system will store them in the correct place after you respond yes to the warning.

Increasing the HyperCard Memory Partition

You can increase the memory partition of HyperCard by following these instructions. It is recommended that you set HyperCard to a minimum of 1100 kbytes.

Find the HyperCard Stack.

Choose 'Find' from the finder File menu and locate the HyperCard application.

Get its Information

Once the stack is highlighted you choose 'Get Info' from the File menu

Change the Current Size

You locate the current size box and increase it to at least 1100. The default for

HyperCard is 1000 so it is likely that this will need to be changed. Increase it again if you get low memory warnings. Later versions allow for a preferred and minimum memory setting, both of which should be greater than 1100 kbytes.

The AYS Example Stack

Introduction

The AYS Example stack is a demonstration communications stack that has been included in the package to illustrate the kind of stacks that can be created with the AYS Installer stack as well as provide a ready-to-use communications utility. The stack is constructed from an introductory card, a preferences card, an on-line help card and a number of installed service definition cards, each of which offers connection to a particular service.

The stack provides it's own text editor to allow you to capture, edit and save information from/to service windows. The contents of text edit windows can be saved to text files that may then, for example be imported into word processors at a later date.

The stack supports multiple sessions both to a single service and several services when used in a network environment. It has many command keys to tailor the view of the screens such as stacking and tiling and to allow swapping between windows. A session can be logged, its contents may be searched, function keys can be defined and a charging calculations made. Each of these is explained below.

Starting

Double click on the AYS Example icon or choose the 'Open Stack' option from the HyperCard File menu. Since it is a stack all of the HyperCard techniques of manipulation and navigation apply.

Introductory Card

The first card of AYS Example contains the introductory text, with access via buttons to the preference card, online help card, the home stack and the first and last of the service definition cards. The menu bar may optionally contain a services menu. The menus are context sensitive. If you can not see an option it is because all the requirements to use that option have not been completed. There is more than one way of moving between the cards, but as a start you can move one card at a time by clicking on the arrow buttons.

Preferences Card

The preferences card allows you to specify the creation of a Service menu, to change the sort order of service cards, to mark service cards, and to save settings associated with service windows in between sessions. Generally you will only need to set these once. The default preferences are likely to be suitable for most users.

Services Menu

> Clicking this box will allow you to specify the creation or removal of the Services menu. If creation of the menu is specified then its contents and the order in which

they appear will be dependant on the options below.

Sort Order

If you click in one of the service sort boxes then the service cards will be sorted appropriately, and if a Services menu has been specified this will be rebuilt to reflect that ordering.

Marked Services

This option will only be available if there is a Services menu displayed. If you click in this box a further three button items will be displayed, 'Mark Services', 'Unmark Services' and 'Rebuild Menu'. These three buttons allow you to specify, by marking cards, which services names will appear in the Services menu. This feature is useful if the stack contains too many services to comfortably fit in the Services menu

Save Settings

Each service window has an associated series of settings, accessed via the 'Settings' sub-menu of the Windows menu (explained later). Clicking this option will allow you specify whether or not these settings are to be 'remembered' between sessions.

On-line Help Card

Clicking on the Help button gives you access to the online help.
The online help describes, in detail, each of the non-standard HyperCard menu items. The text is arranged in menu order, reflected by the direct access buttons arranged below the text.

The Service Cards

Each service card defines a service that may be launched by the stack.

The Service Card Buttons

At the bottom of each service card are several buttons, 'About', 'Help', ' Text', ' Connect', ' Disconnect', 'Home', 'Next' and 'Previous' Arrows.

About

Clicking on the 'About' button will take you to the first, introductory card of the stack.

Help

Clicking on the Help button gives you access to the online help.
The online help describes, in detail, each of the non-standard HyperCard menu items. The text is arranged in menu order, reflected by the direct access buttons arranged below the text.

Text

Clicking on this button will open up a miniature editor similar to TeachText within HyperCard. Option clicking (ie. clicking on the button with the option key depressed) will allow you to open a text file from disk. This is useful when you wish to copy text to or from your saved electronic mail, etc.

Connect

Clicking on this button will attempt to launch the service defined by the card. Should the connection fail, or time out, a warning dialog will be displayed. See the section below for a more comprehensive explanation.

Disconnect

Clicking on this button will close any active service associated with the card. If there are multiple activations of the same service, the oldest activation will be

closed first. See the section below for a more comprehensive explanation.

Home

The 'Home' button closes all the service and text edit windows and takes you to the home stack.

Quit

The 'Quit' button closes all the service and text edit windows and then quits HyperCard.

Arrows

The 'Next' and 'Previous' buttons allow you to scan the cards one at a time, either forward or backwards through the stack.

Starting a Service

Click on the arrow buttons until you locate a service of interest and then click on the 'Connect' button. The connection will be attempted. If the connection attempt is successful, and there is a

script called 'LoginToService' associated with the service, then this will be executed. This allows automatic connection with no input required by the user. You are prevented from aborting this script (by command-period) in order to provide some degree of service security. After the successful launching of the first service, a new menu, the Windows menu will appear in the menubar. This menu provides options for the manipulation and management of the new window.

Once connected you can move to another card and connect to that service. Note that this is only possible if you are using a network tool such as the TGE tool. You can close the connection by closing from the remote service, by clicking on the 'Disconnect' button on the card associated with the service, selecting the 'Close Window' option from the Windows or by clicking in the 'Go Away' box on the service window. When you do this, if there is a script called 'LogoutFromService' associated with the service, then this will be executed first before the connection is closed. Like the login script you are prevented from aborting the script. If this is the last window to be closed then the Windows menu will be removed from the menubar.

Another useful method of connecting to a service is to choose it from the Services menu. This will immediately attempt to connect you to that service. If you choose a service from the Services menu while holding down the option key you will only be moved to the associated card, no connection will be attempted.

Other Useful Procedures

Find

> The find command from the HyperCard File menu will allow you to find a service by its name. For example, if you type "Aber" into the message box (that will appear after you have chosen the 'Find' option), then HyperCard will locate (and take you to) the service definition card 'Aberdeen University'. NB. This is different to the find option available under the windows menu, see 'Find' under the Search sub-menu.

Windows Menu Features

When the first service or text edit window is opened a new menu, the Windows menu is added to the menubar. When additional windows are opened this menu is expanded to reflect the new situation. When the last service or text edit window is closed the Windows menu is removed from the menubar.

The menu provides a comprehensive series of options for the manipulation and management of service and text edit windows. Below is a complete description of each feature under its menu option name.

Window Names

The first options in the 'Windows' menu contains the names of the windows that have been opened. The currently active window is denoted by a leading "•" character. Windows that have been hidden are denoted by a leading "◊" character. A visible window may be made the currently active window by a) Clicking anywhere in that window b) Selecting its name from this menu or c) Using one of the special keys for changing windows as described in the 'Use Key Pad' option in the 'Settings' submenu.

Show Window

This option will only be active if one or more windows have been hidden, and will display and activate the most recently hidden window.

Hide Window

This option will hide (render invisible) the currently active window.

Close Window

Closes the currently active window in the appropriate manner. If the option is selected whilst the option key is depressed then all active windows are closed.

Clear Window

If the currently active window is a text edit window then its entire contents are cleared. If the currently active window is a service window then only the terminal emulation screen is cleared, the cached (scrolled) area is not affected. If the option key is depressed when the option is selected then the cached area will also be cleared.

Break Window

This option is only highlighted for a service window, the option will send a long break (1 second) signal to the service. If the option key is depressed when the option is selected then a short break (1/4 second) signal is sent to the service.

Settings Submenu

Copy Submenu

Search Submenu

See below.

Tile Windows

Tiles all active, visible windows, excluding the card window.

Stack Windows

Stacks all active, visible windows, excluding the card window.

Service Info

This option is only highlighted for service windows. Selecting the option will display statistical information relating to the service and allow the specification and calculation of a connection charge per minute.

Settings Submenu

This submenu contains differing options depending on the type of window that is currently active. The following five options are those for service windows.

Data Logging

This option allows the logging of all output from a service to a text file. When the option is selected the operator is prompted for a file name. The menu option is preceded by a 'tick' mark to indicate that the option is 'on'. All data that is subsequently scrolled off the top of the terminal emulation is then written to that file. The facility may be turned off by reselecting the option.

Smooth Scrolling

This option allows the choice between smoother-slower or faster-jerkier terminal emulation text display. If the option is active then the menu item is preceded by a 'tick' mark.

Delete = Backspace

This option allows the character output from the 'Delete' key to be set to either a 'backspace' character (character 8) or to a 'delete' character (character 127) for a service window. NB. This key setting may be overridden by a similar setting in the terminal emulation tool that you are using. If the option is active then the menu item is preceded by a 'tick' mark.

Slow Window Paste

This option allows the slowing down of all pasted data (either Clipboard data or Strings menu data) to a service window to approximately 6 characters per second. The option is provided for those services that cannot accept rapid streams of input data (ie. do not have input buffers). If the option is active then the menu item is preceded by a 'tick' mark.

Use Key Pad

This option allows for the mapping of certain window functions to the keys of the numeric key pad. If the option is active then the menu item is preceded by a 'tick' mark. The mappings are described by the table below :

Function	NumericKeyPad
Activate String no 1 -> 11	0 through 9 and .
Clear window	Clear
Change window Right	+
Change Window Left	-
Zoom window	*
Copy & Paste	=
Show selection (text window) or scroll to cursor (service window)	/

The following three options are those for Edit windows

Save Styling Data

If this option is active (ie. preceeded by a 'tick' mark), then each time that the currently active text window is saved to a file the text styling information will also be written to the same file. The file may subsequently be read back by AYS and all the text styling will be preserved. If the option is deactivated then only the text is written to file, and if the file is subsequently read back into At-Your-Service any styling will be lost. NB. This does not affect the files 'readability' by other applications.

Tab Settings

Selecting this option results in the display of a dialog box that allows the tab character to be replaced by a number of space characters. If the option is active then the menu item is preceded by a 'tick' mark. The facility may be turned off by reselecting the option. If a different number of spaces is required then selecting the active option with the option key depressed will redisplay the dialog box.

Use Key Pad

This option allows for the mapping of certain window functions to the keys of the

numeric key pad. If the option is active then the menu item is preceded by a 'tick' mark. The mappings are described by the table below :

Function	NumericKeyPad
Activate String no 1 -> 11	0 through 9 and .
Clear window	Clear
Change window Right	+
Change Window Left	-
Zoom window	*
Copy & Paste	=
Show selection (text window) or scroll to cursor (service window)	/

Copy Submenu

Print Selection/ Print Text

This item may assume one of two values depending on the type of window that is currently active. If the currently active window is a service window then the item will appear as 'Print Selection'. The option will only be highlighted if there is a selection in the window and choosing it will cause the selection to be printed. If the active window is a text edit window then the option will appear as 'Print Text'. The option will only be highlighted if the window contains text and choosing it will cause the entire contents of the window to be printed.

Save Selection/ Save Text

This item may assume one of two values depending on the type of window that is currently active. If the currently active window is a service window then the item will appear as 'Save Selection'. The option will only be highlighted if there is a selection in the active service window and it will cause the selection to be saved to a text file via the standard file selection dialog. If the active window is a text edit window then the option will appear as 'Save Text'. The option will cause the entire contents of the window to be saved to a text file via the standard file selection dialog. Styling will be saved if the 'Save Styling Data' option is set. If the currently active text window has already been saved to file then a second copy may be made by depressing the option key whilst selecting the menu item. This will result in the redisplay of the standard file selection dialog from which another file name may be specified.

Copy Items

This will copy a selection to another window. If there is a selection in the currently active window and there is at least one other active window (service or text edit) then item 4 onwards of this submenu will contain the names of all the other windows that can receive the selected text. Selecting a service window name from the submenu will result in the selection being pasted into that window. Selecting a text edit window name from the submenu will result in the selection being appended into that window.

Search Submenu

Select All

If the currently active window is a service window then this item will cause the current terminal emulation screen to be selected, the cached (scrolled) area is not affected. If the active window is a text edit window then the option will only be highlighted if the text edit window contains data and it will cause the entire contents of the window to be selected.

Find

If the currently active window is a service window then this item will conduct a find operation on the data that is scrolled off the top of the terminal emulation. The data is searched in reverse order, that is the most recently scrolled data is searched first. If the window is a text edit window then the item provides a standard implementation for text location.

Find Again

Repeats the find operation.

Replace, Replace & Find, Replace All

If the currently active window is a service window then these items will be disabled since their action is inappropriate. If the window is a text edit window then these items provide standard implementations for text replacing.

Other Menus

Strings Pop-up Menu

This menu provides for function key emulation by allowing up to 11 strings to be stored and recalled for each active service window. The menu will appear immediately below the window bar of the active service window if the mouse is clicked in the window bar whilst the 'option' key is depressed. The menu comprises an item for each string that has been 'installed' plus a final item, '*Add String to Menu*' that allows further strings to be installed. Each string item is limited to 255 characters in length and may be modified or deleted by selecting the required item whilst the command key is depressed. The strings may be mapped to one of two sets of keyboard keys, the extended keyboard 'Function keys' or the standard keyboard 'Numeric Key Pad' keys depending on the setting of the 'Use Key Pad' option in the 'Settings' submenu. Selecting a string item from the menu results in the contents of the associated string being sent to the service.

Scripts Pop-up Menu

A pop-up menu of scripts may be associated with each active service window. The menu (if it has any items) will appear immediately below the window bar of the active service window, if the mouse is clicked in the window bar whilst the 'command' key is depressed. If the service window does not have any associated scripts then attempting to display the menu will result in a single 'beep'. The menu comprises an item for each script that may be executed for the service. Selecting an item will result in the execution of that script.

Note. A service script is a special type of HyperCard script that has an extended range of commands that allows communication with a service window.

Tool Menus

The menu bar for a service window may contain up to 2 additional menus. These menus are associated with the Communications Toolbox tools used to connect to a particular service. Whether or not the additional menus will be present will depend on the tools used, as some do not have menus. The display of these menus may be inhibited when defining a service.

Services Menu

This menu is only associated with the AYS Example stack. This menu is optional, depending on the settings of the 'Preferences' card. If the menu is available, then all but the first item will contain the names of the services available for use. Selecting one of these names will result in the activation of the associated service (and display of the card that describes the service in the background). If the option key is depressed whilst the selection is made then only the associated card is displayed,

the service is NOT activated.

The first item, *Text Window* allows you to create of up to 10 text edit windows that may be used to accumulate selections from other service and text windows. Selecting this option from the menu will result in the creation of an 'empty' text window. If the selection is made with the option key depressed then an existing text file may be input via a standard file dialog. Text may be added to the window by typing in data, pasting from the clipboard or by the use of the 'Copy' submenu.

The AYS Installer Stack

Introduction

The purpose of the Installer stack is to -

- Provide facilities for the creation and modification of service definitions from either manually entered data or via a text file scanning facility.
- Provide facilities for the storage and indexing of service definitions.
- Allow the testing (activation) of service definitions.
- Provide facilities for installing service definitions into target stacks, either individually or in batches.
- Provide facilities for removing existing service definitions from target stacks, either individually or in batches.
- Provide facilities for importing service definitions from other stacks either individually or in batches.
- Provide facilities for the creation of one or more text edit windows.

The installer may install/remove/import service definitions in one of two forms, either as HyperCard cards or as Macintosh resources. The AYS stack is an example of a stack constructed from card definitions. The use of resource based definitions is particularly useful for constructing stacks where it is desired to hide as much of the service detail as possible.

Starting

Double click the AYS Installer icon or choose the 'Open Stack' from the HyperCard File menu. Since it is a stack, all of the HyperCard techniques of manipulation and navigation apply.

Registering

Each time the installer stack is launched a dialog box is displayed that requests the input of a password; this password is obtained by registering your copy of the AYS package. Until this is done the AYS Example stack, the AYS Installer stack, and any stack derived from it will only launch services for a duration of 5 minutes, after which they are automatically terminated. Once you have registered your copy and entered the password the AYS Installer will beep three times and a further dialog will be displayed. If this does not happen it is likely that the password has been entered incorrectly. Successful entry of the code will remove the time restriction from the installer stack but not from the AYS Example stack or any other stack derived from the installer. To remove the time restrictions from these stacks you must either install a new service, or reinstall an existing service into each stack WHILST THE OPTION KEY IS DEPRESSED. A dialog box is displayed to indicate successful removal of the time restrictions from each stack.

Navigating in the Installer

The introductory card of the stack contains a scrolling list of the services available with a line of rectangular buttons beneath them.

These buttons are 'Import', 'Install', 'Delete', 'Defaults' and 'Prefs'. Each of these buttons has an associated card. They give you access to all the major operations performed by the Installer apart from creating a new service. The function of these buttons will be described later under individual headings. A service is created when you select 'New Card' from the Edit menu and define the associated parameters such as the Connection and Terminal Settings, and any scripts that might be required. To modify an existing service you click on its name in the scrolling list. This will take you to the definition card for that service.

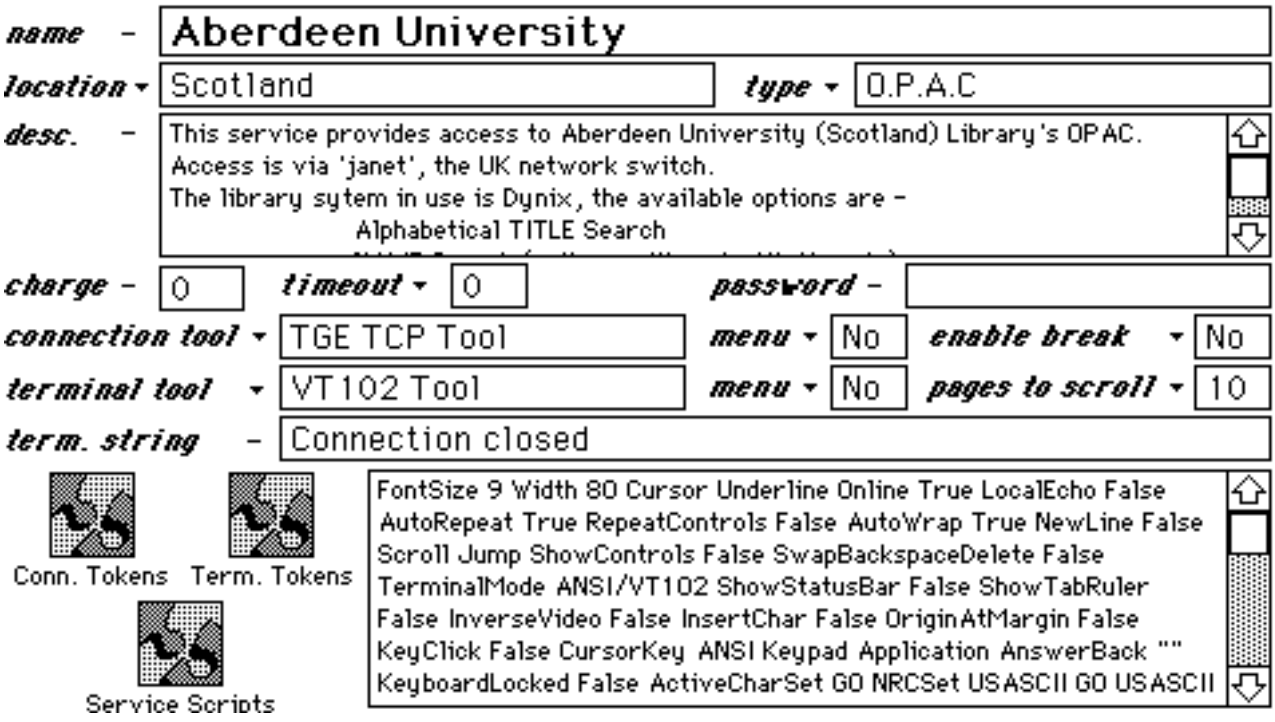
The stack has a special menu, the 'Function' menu that is only highlighted when a service definition card is displayed. The menu provides a series of service definition manipulation facilities such as installing and removing definitions into/from other stacks. The last option on the menu provides for the creation of a text edit window that may be used in an identical way to that described for the AYS Example stack.

Creating a New Service

The following step by step guide will enable you to create a service definition. Start by defining a default service, and then create a new definition card and complete the details. Test the new definition and when you are happy with its performance then install it into a target stack.

Service Definition Parameters

Each service definition is constructed from a series of parameters that are held on a single card, for example -



The parameters are as follows -

1. The unique name of the service eg. “NewsReader”. This field may be used to sort the cards via the 'Preferences' card.
2. The geographic location of the service, this field may be used to sort the cards via the 'Preferences' card.
3. The type of service provided, this field may be used to sort the cards via the 'Preferences' card.
4. Notes and general information about the service.
5. The charge in cents per minute associated with the service. Setting this value to 0 will turn this feature off.
6. The number of minutes before the service will 'timeout' and terminate the connection. Setting this value to 0 will turn this feature off.
7. The number of screen pages to be scrolled by the Terminal Emulation tool being used.
8. The local password to protect the service.
9. The communications toolbox Connection tool to be used to connect to the target computer upon which the service is available eg. “TGE TCP Tool”.
   The tools available are the Connection tools that have been installed on your

system. Each time a selection is made the operator is presented with a tool specific interface (dialog box) by which the various communications parameters may be specified. For the TGE TCP tool the network address, terminal type in use and many other parameters may be changed. Each tool is different.

10. Whether or not any menu associated with Connection tool is to be added to the service menubar.
11. Whether or not the break key is to be activated for the service
12. The Communications toolbox Terminal Emulation tool that will be used to display the output generated from the service eg. “VT102 Tool”.

The tools available for selection are all those Terminal Emulation tools that have been installed on your system. Each time a selection is made the operator is presented with a tool specific interface (dialog box) by which the various emulation parameters may be set. The VT102 tool allows the character set, cursor type, local echo and many other parameters to be changed. Each tool is different.

13. Whether or not any menu associated with Terminal Emulation tool is to be added to the service menubar.
14. A character string that will, upon detection signal the closure of the service. That is, AYS will check all input from the service against this parameter string. If a match is found then the stack will terminate the service by executing the logout script and closing the connection, etc.
15. The HyperTalk script handlers that are to be associated with the service.

Defining the Default Parameters

Since many of the features of a service will remain constant for your environment you should firstly set your service defaults to minimise the amount of work required to create new services. On the introductory card click on the 'Defaults' button to take you to the 'Service definition defaults' card.

Setting the default communications tool will depend on the type of network access you have, many users will have TCP/IP access using MacTCP. In this case you should set the TGE TCP tool as the default tool.

The Terminal tool most people will use is the VT102 Tool. If this is the case you should set this as your default terminal tool.

Complete all of the other fields as necessary, the on-line help should provide sufficient information to complete this task.

Create the New Service

Choose 'New Card' from the Edit menu. A new service card will be created with the name 'New Service' and it will contain the default values previously defined. Fill in the remaining fields such as the name and modify any of the defaults that do not apply to this particular service. The on-line help should provide sufficient information to complete this task.

Select the script window by clicking on the 'Service Scripts' button and create any HyperTalk scripts required by the service.

Scripting

A service definition may have any number of scripts associated with it. The names of the associated script handlers (not functions) will appear in the 'Scripts' pop-up menu that appears below the service window bar when the mouse is clicked in the bar and

the command key is depressed.

The scripts are written in HyperTalk, HyperCard's scripting language. AYS extends HyperTalk with several commands, functions and properties to enable it to talk directly to service and text edit windows. These extensions are described in detail in appendix D. An example of the use of some of these extensions appears in appendix F.

AYS maintains a series of HyperTalk globals that together reflect the status of a stacks active service and text edit windows. You can query these globals within your scripts simply by including a 'Global' statement at the head of the script. A full description of all the globals maintained by AYS appears in Appendix C.

To create scripts, click on the 'Service Scripts' button on the service definition card. This will place you in the HyperCard script editor with which you may start entering scripts. The scripts are stored in the 'Card Script Area' of the card. This is an important fact that will be referred to later.

A script handler indicates successful execution to AYS by setting ‘the Result’ to true, (via the command line ‘return true’). Conversely it indicates failure by setting ‘the Result’ to false, (return false)

Two special scripts are recognised. The first is called 'LoginToService'. If a script handler of this name is associated with the service then it will be executed immediately following the successful launching of a service. If the script fails then the service will be closed down. The second special script is called 'LogoutFromService'. If a script handler of this name is associated with the service then it will be executed immediately prior to closing the service.

Warning

When a service script is executed via the 'Scripts' menu the actions differ depending on whether it's service definition is held on a card, (as in the case of the stacks AYS Example and AYS Installer) or if it is held as a resource.
If the definition is held on a card, the scripts associated with the service are held in the card script area and it is from here that they are executed. If the definition is held as a resource then the scripts are also held in the resource, and when they are executed via the 'Scripts' menu or by the use of the function 'ExecuteServiceScript' (see appendix D) they are firstly copied to the script area of the current card. **This results in the OVERWRITING of any scripts that were there originally.**

Testing

The whole process may now be tested by choosing 'Test Service' from the 'Function Menu'. You may have to change your scripts to get the service to function in the manner you require. The events and conditions that follow the successful launching of a service are virtually identical to that for the AYS Example stack. That is, if the connection attempt is successful, and there is a script called 'LoginToService' associated with the service, then this will be executed. However unlike the AYS Example stack this script may be aborted by typing command-period. This allows for the debugging of login scripts. After the successful launching of the first service, a new menu, the Windows menu will appear in the menubar. This menu provides options for the manipulation and management of the new window.

Once connected you can move to another card and connect to that service. Note that this is only possible if you are using a network tool such as the TGE tool. You can close the connection by closing from the remote service, by selecting the 'Close Window' option from the Windows menu or by clicking in the 'Go Away' box on the service window. When you do this, if there is a script called 'LogoutFromService' associated with the service, then this will be executed first before the connection is closed. Like the login script, this script may be aborted by typing command-period. This allows for the debugging of logout scripts. If this is the last window to be closed then the Windows menu will be removed from the menubar.

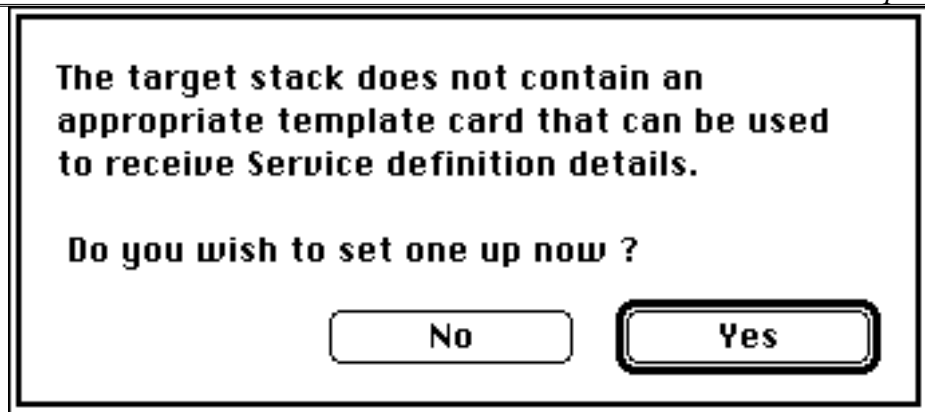
Installing

Once the service is tested the next step is to install it into a target stack. You must

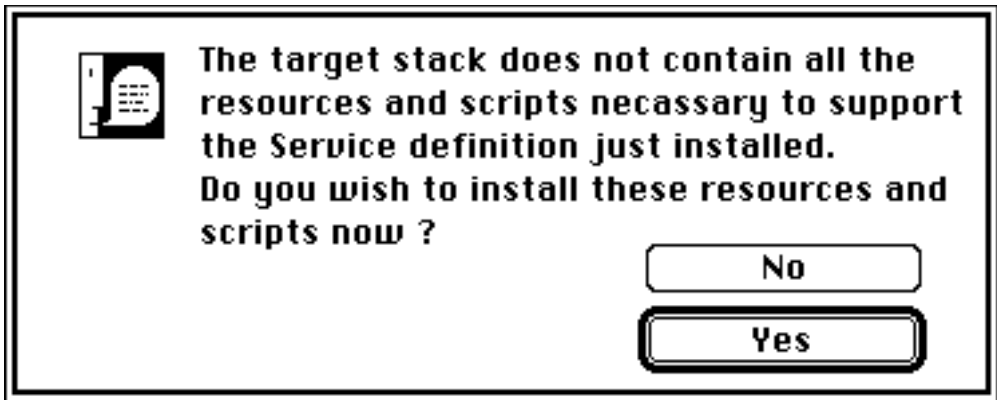
firstly define the stack into which you wish to place this new service. Chose 'Define Target' from the 'Function Menu' and select the stack into which the service is to be installed. You will be asked if you wish to install service definitions as HyperCard cards or as Macintosh resources. Select whichever is appropriate to the type of stack that you are constructing.

Once the target has been defined you install it into the stack by choosing the 'Install into Target' option from the 'Function Menu'. The first time that you install a service you will be prompted by a further one or two dialog boxes.

If you are installing service card definitions and the stack into which you are installing does not contain a card of the type required by AYS then the following dialog is shown. If you click on the 'No' button the process will be aborted.



Regardless of the type of service definition being installed the following dialog will be displayed. The dialog box will be displayed each time a service is installed into the stack until the 'Yes' button is clicked. For details of the resources and scripts that are copied to the target stack see the section 'Technical Considerations' below.

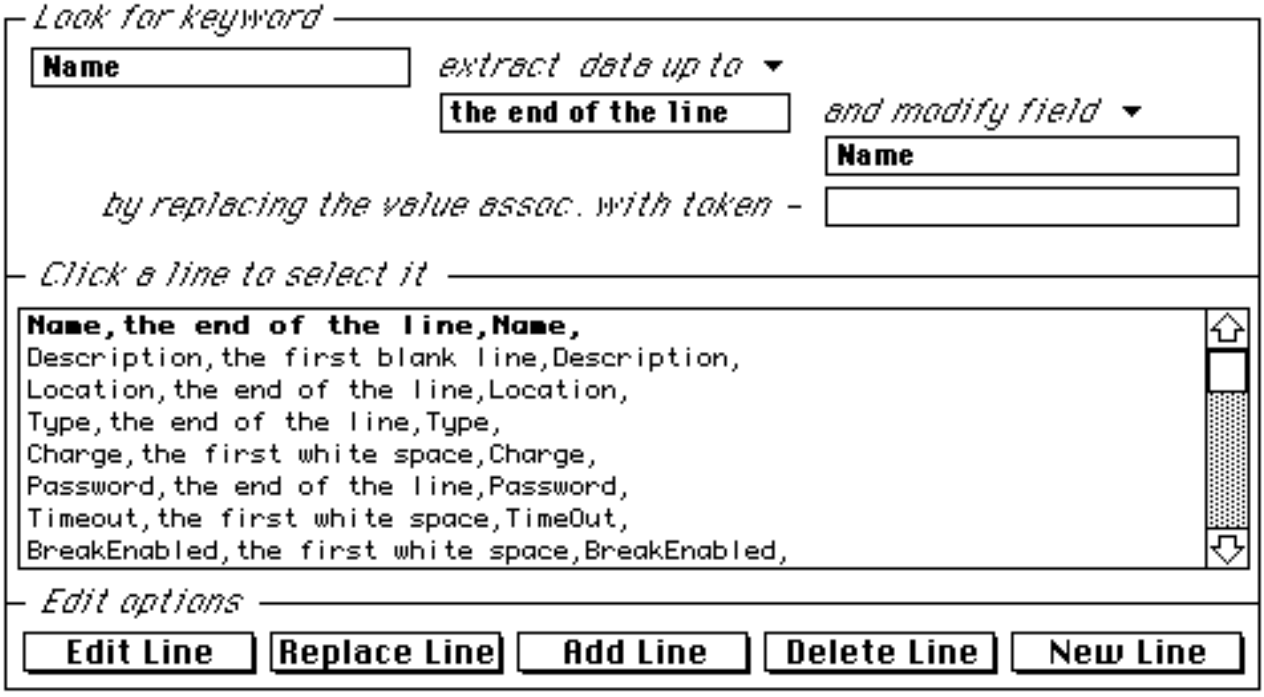


Aids to Defining a Service

To assist in the creating and archiving of service definitions the Installer stack provides the following two features.

Creating a Service Definition from a Text File

To assist in the creation of service definitions you can use the 'Load from Text File' option of the Function menu to scan any text file for a definable list of keywords that may describe AYS service parameters.
When you select this menu option the first action is to replace the Connection and Terminal names and configuration 'tokens' of the 'New Service' card with those defined on the 'Keyword Default' card. This is necessary to ensure that a known set of 'tokens' are present in the 'New Service' definition prior to scanning the text file. The text file is then scanned by interpreting the keyword table contained on the 'Keyword definition' card.



The keyword table may contain any number of lines, each line containing information on a single keyword.

1. The first item of a keyword line contains the keyword to be scanned for, only single words, containing alpha-numeric characters are allowed. The rules for scanning are as follows -

   Only alphanumeric characters are used to construct keywords.
   eg. Service-Name yields ServiceName **and** $Amount yields Amount.

   Comments are prefixed by either a hash or an exclamation character (**#** or **!**) and terminated by a carriage-return.
   eg. ServiceName=Uni Opac # Provides access to the university's OPAC Password # No password required for this service.

   The characters used to determine the end of a keyword are a **space**, an **equal** sign, **colon** or a start comment character.
   eg. Password Fred **or** Password=Fred **or** Password:Fred **or** Password = Fred **or** Password : Fred **or** Password# **or** Password!

2. If a keyword is detected in the scanned file then the second item of the table line defines how much of the data that follows it is to be extracted. The options are either to extract data up until the first space (or comment character), the first carriage-return (or comment character) or up until the first blank line

(comments are treated as part of the data).

3. The extracted data is then used to modify the HyperCard 'New Service' card field named by the third item of the table line. The rules that apply here are dependent on whether there is a fourth table item or not.

4. If there is no fourth item then the contents of the named field are overwritten with the extracted data. Otherwise the fourth item contains a single word configuration 'token' that is to be searched for in the named field. If the 'token' is found in the field then the single word parameter that follows it will be replaced by the extracted data. This feature is only of use if Communications Toolbox tools being used have recognisable 'tokens'.

For example -

a) ServiceName,the end of the line,Name

is interpreted as -

Scan the text file for the keyword ‘ServiceName’ and if found extract the data that follows the keyword up until the end of the line and then replace the contents of the background field ‘Name’ with the extracted data.

b) Address,the first white space,ConnectionTokens,HostName

is interpreted as -

Scan the text file for the keyword ‘Address’ and if found extract the data that follows the keyword up until the occurrence of the first space. Locate the configuration token ‘HostName’ in the background field ‘ConnectionTokens’ and replace it with the extracted data.

Archiving a Service Definition to a Text File

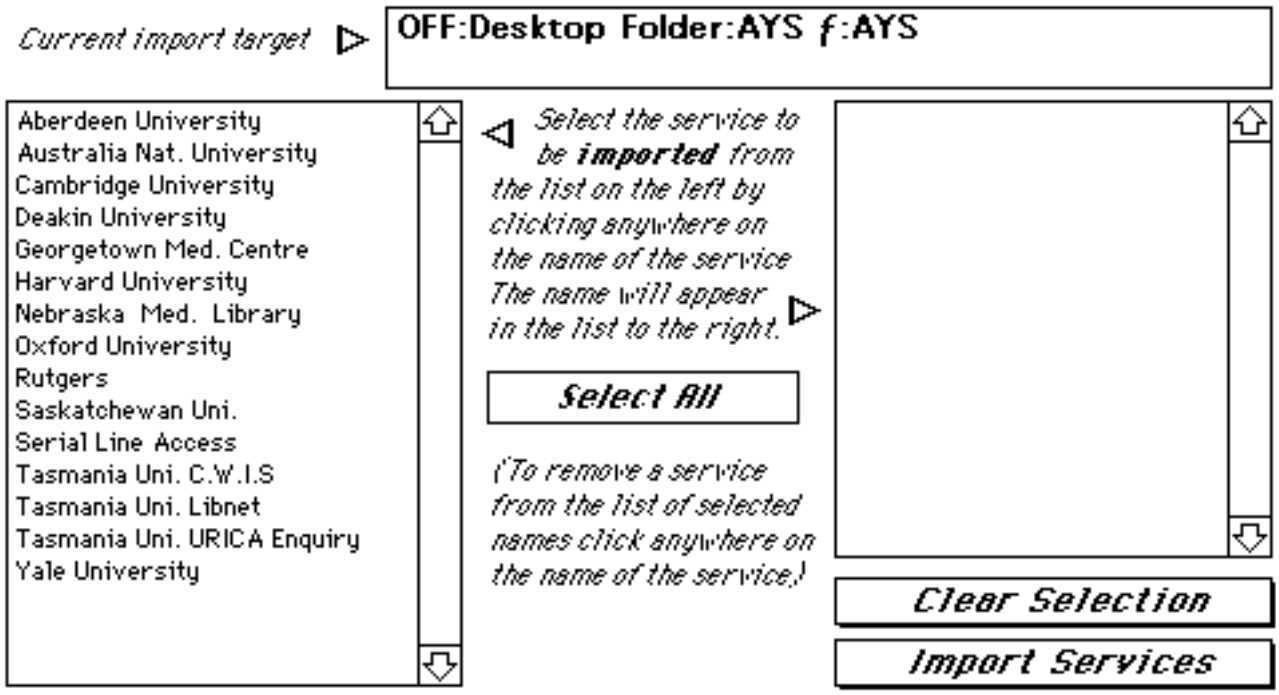
To allow you to save a service definition to file you can use the 'Create Parameter File' option in the 'Function' menu. This option will encode the service definition parameters and write them to file. The file is written in a format that is understood by the previous option and consequently the file may be used to reinstall the service into the installer stack.

Import

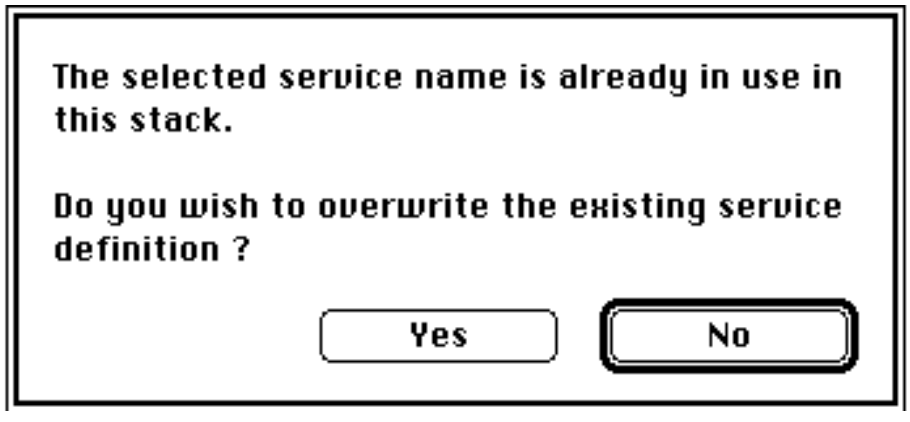
You may import service definitions from any AYS compatible stack. If you click on the 'Import' button a standard open file dialog box will enable you to choose a stack from which to import service definitions.

Once a stack has been chosen you will be asked if you wish to import service definition cards or resources. If the selected stack does not contain any of the chosen type of definition then an error message will be displayed.

The following card will then be shown with the service definitions that are available for importation appearing into the left hand field.



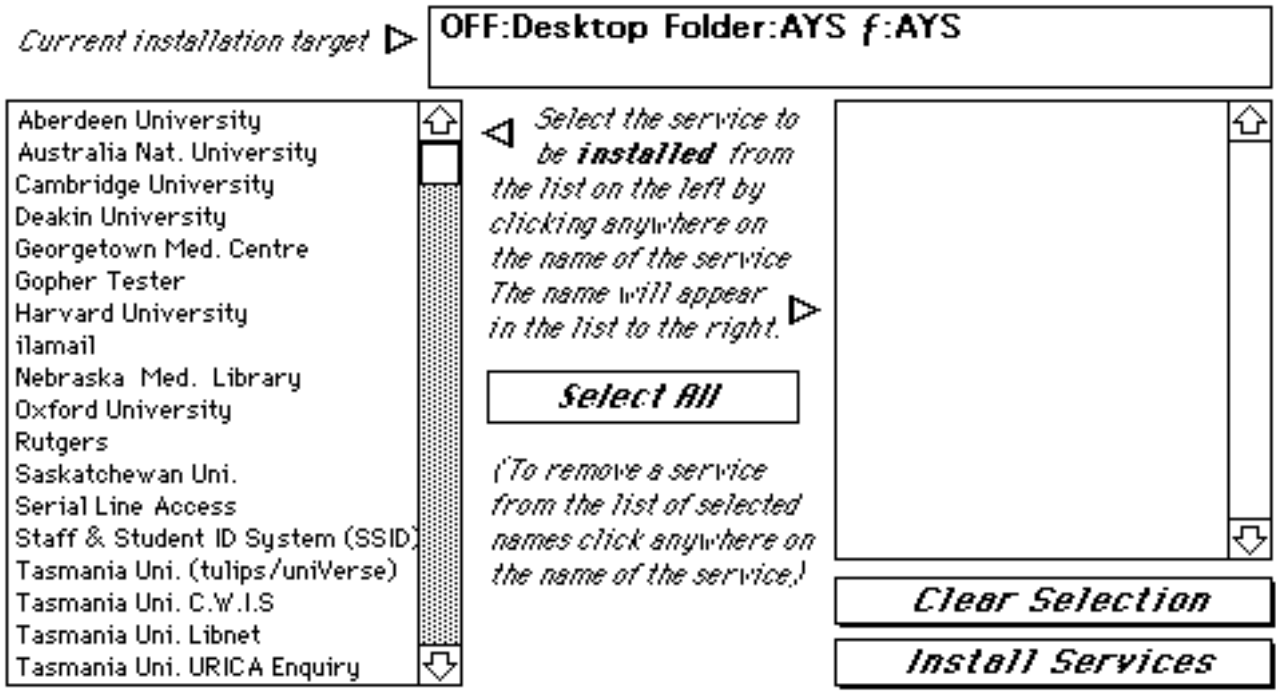
These may be chosen singularly by clicking on each service name that you wish to add to the Installer, or all at once by clicking the 'Select All' button. Click on the 'Import Services' button at the bottom of the card, and the selected definitions will be added to the installer. If the installer stack already contains a service with the same name as one of the services being imported then the following warning message is output.



An alternative to this 'batch' importation feature is to import a single service at a time, This may be achieved by going to the card associated with the service to be imported, choosing 'Define Target' to set the stack from which you wish to import/recover and then choosing the 'Function' menu option 'Import from Target'.

Install

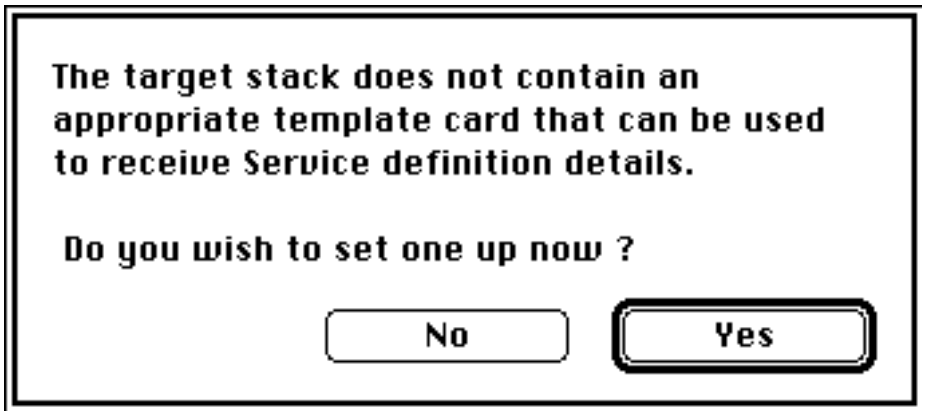
You may install service definitions into any stack. If you click on the 'Install' button a standard open file dialog box will enable you to choose a stack into which to install service definitions.
Once a stack has been chosen you will be asked if you wish to install service definition cards or resources.
The following card will then be shown with the service definitions that are available for installation appearing in the left hand field.



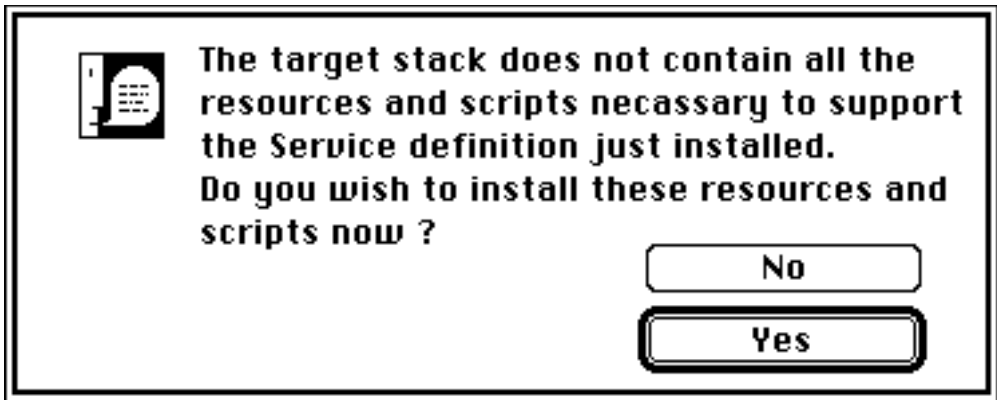
These may be chosen singularly by clicking on each service name that you wish to install, or all at once by clicking the 'Select All' button. Click on the 'Install Services' button at the bottom of the card, and the selected definitions will be installed into the target stack. If the stack being installed into already contains a service with the same name then the following warning message is output.

The first time that you install a service you will be prompted by a further one or two dialog boxes.

If you are installing service card definitions and the stack into which you are installing does not contain a card of the type required by AYS then the following dialog is shown. If you click on the 'No' button the process will be aborted.



Regardless of the type of service definition being installed the following dialog will be displayed. The dialog box will be displayed each time a service is installed into the stack until the 'Yes' button is clicked. For details of the resources and scripts that are copied to the target stack see the section 'Technical Considerations' below.



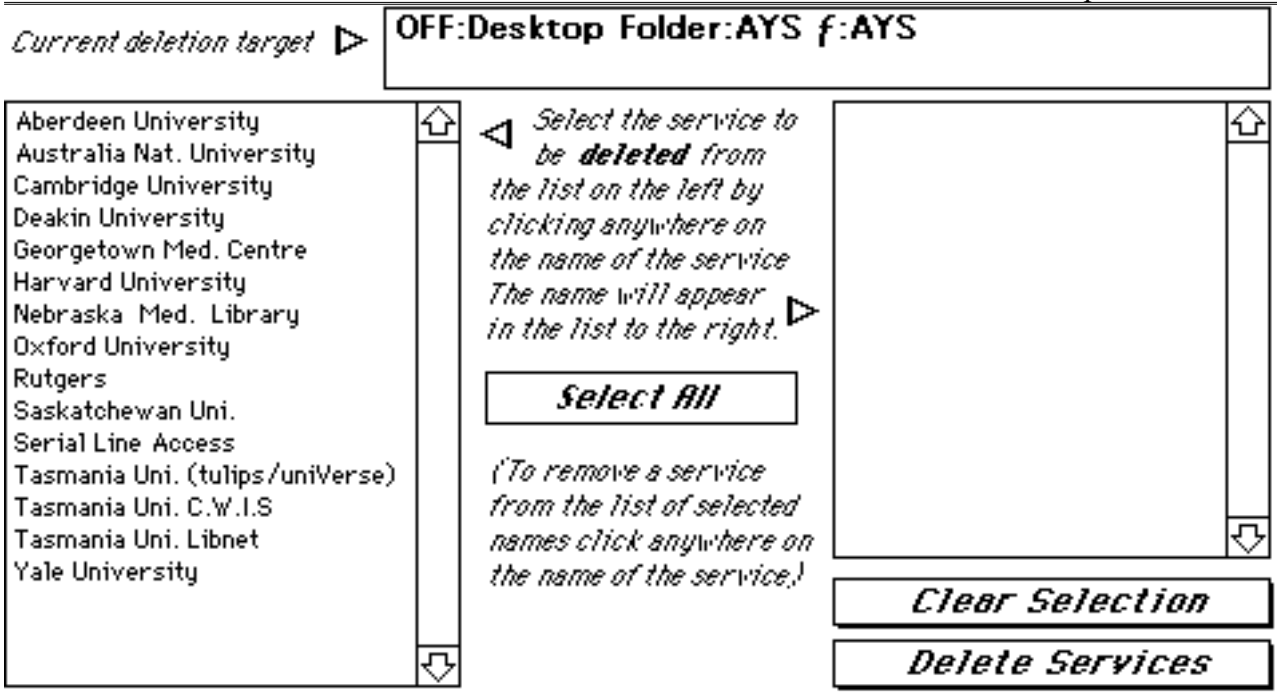
An alternative to this 'batch' importation feature is to install a single service at a time, This may be achieved by going to the card associated with that service, choosing 'Define Target' to set the stack into which you wish to install the service and then choosing the 'Function' menu option 'Install into Target'.

Delete

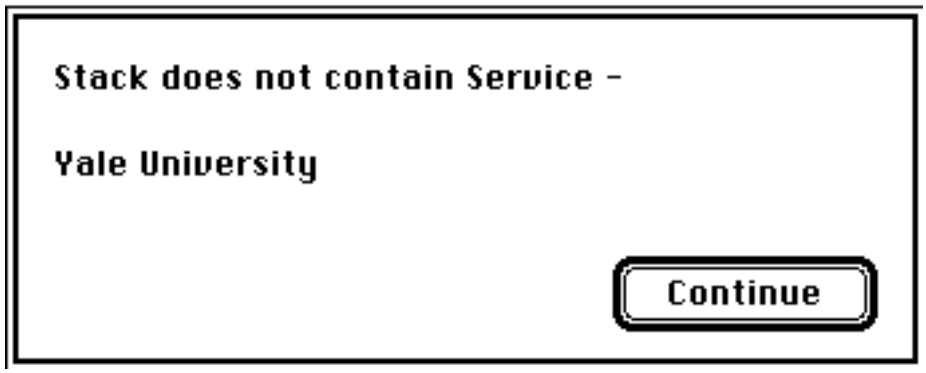
You may delete service definitions from any stack. If you click on the 'Delete' button a standard open file dialog box will enable you to choose a stack from which to delete service definitions.
Once a stack has been chosen you will be asked if you wish to delete service definition

cards or resources.

The following card will then be shown with the service definitions that are available for deletion appearing into the left hand field.



These may be chosen singularly by clicking on each service name that you wish to delete from the target stack, or all at once by clicking the 'Select All' button. Click on the 'Delete Services' button at the bottom of the card, and the selected definitions will be deleted from the selected stack. If the stack being deleted from does not contain the named service then the following warning message is output.



An alternative to this 'batch' deletion feature is to delete a single service at a time, This may be achieved by going to the card associated with that service, choosing 'Define Target' to set the stack from which you wish to delete the service and then choosing the 'Function' menu option 'Delete from Target'.

Defaults

Clicking on the 'Defaults' button will take you to the first of the two default cards.
The first default card, the 'Service definition default card' enables you to define default

values for creating new services definitions (see the section 'Defining the Default Parameters').

The second default card, the 'Keyword default card' allows you to define the default connection and terminal emulation tools that will be loaded into the 'New Service' card immediately prior to scanning the selected text file for keywords, (see the section 'Creating a Service Definition from a Text File').

Preferences

Clicking on the 'Prefs' button will take you to the 'Preferences' card; that allows the setting of two options.

The first option allows you to change the sort order of the service definition cards in the stack. If you click in one of the service sort boxes then the service definition cards will be sorted appropriately.

Each service window has an associated series of settings, accessed via the 'Settings' sub-menu of the Windows menu. Clicking the second option will allow you specify whether or not these settings are to be 'remembered' between sessions.

Technical Considerations

Scripting Areas

When creating new communications stacks care must be taken not to place scripts in the card script area of any card that can launch an AYS service. This area is reserved by AYS for scripts associated with the service and any script placed in the area may be overwritten by the service scripts.

Stack StartUp

The HyperTalk scripts that are installed into target stacks to support AYS communication services, contain the Communications Toolbox initialisation code in the 'OpenStack' handler. Consequently, any attempt to launch a service before the 'Openstack' handler is executed will result in disaster.

Installation Process

Whenever a service definition is installed into a target stack the installation process checks to see if the stack contains the full set of supporting resources, and scripts that are necessary to interpret the service definitions being installed. If there isn't then you are informed and asked if the stack is to be updated. If the reply is ‘Yes’ then the stack is updated with the appropriate HyperCard scripts and resources.

(NB. If you wish to replace all of the supporting resources in the target stack with a fresh set from the installer then install a new service or reinstall an existing service into the target stack with the command key depressed.)

The supporting resources that are copied to a target stack at installation time include the code segments (XFCN and XCMDs) necessary to interpret the service definitions. A slightly different set are copied to stacks whose definitions are held as cards as to those whose definitions are held as resources. The definition of each set of code resources are held in appendices A and B.

Although it is possible to install both ‘resource’ and ‘card’ service definitions into the same stack, the set of supporting HyperTalk scripts will only support ‘resource’ **or** ‘card’ definitions. If a custom stack is required that contains mixed service definition types then the supporting HyperTalk script handlers must be modified appropriately.

There are two sets of HyperTalk script handlers that are copied to the target stacks script area by the installation process. The first set contain 'essential' handlers and functions that are necessary to carry out such operations as launching and terminating the service definitions. The deletion of one or more of these scripts will render the

target stack inoperable.

The second set of scripts that are copied are 'optional' since they are not essential to the operation of the stack. These scripts are responsible for creating a set of card buttons whose names reflect the installed service definitions and when pressed will launch the named service. The scripts are activated by typing the command 'AYS' into the message box. The script will attempt to set the host stack to level 4 (authoring level). **If the host stack cannot be set to this level of operation these scripts will fail.** These scripts may be deleted from the target stack with no resulting ill effect.

Resources

The installer stack uses Pop-Up menus to define a service definition's 'Location' and 'Type' parameters. These menus may be modified or extended, by the addition of further sub-menus, by using ResEdit to edit the stack's MENU resources.

Several of the error messages generated by AYS contain the word ‘AYS’ or refer to AYS specifically. Should you be using AYS as a toolkit to build your own stacks then you may wish to change the wording of some of these error messages. This may be done by using ResEdit to edit the STR# resources held in the installer stack.
Likewise you may wish to change some of the DLOG, ALRT or DITL resources in order to customise your dialog boxes.

With the exception of the ‘vers’ resource **DO NOT CHANGE THE NAME OR ID’s** of any of the resources. Doing so will almost certainly render the AYS stacks and any stacks derived from them inoperable.

Installation Problems

Incorrect version of HyperCard

AYS will caution and terminate if this is the case.

The communications tool box is not installed.

AYS will caution and terminate if this is the case. System 7 comes with the Communications Tool Box installed. It is likely that you are running system 6. If this is the case you can install the Communications Tool Box or install System 7.

The terminal emulation tools are not present in the extensions folder

The included tools should be dropped on the system folder, this can be done all at once.

Open the folder, 'Comms Tools' in which the tools are stored, select all the tools and drag them and drop over the system folder. You will be asked if they should be stored in the Extensions Folder. Answer 'yes' and the system will store them in the correct place.

The Communications tools are not present in the extensions folder

The included tools should be dropped on the system folder, this can be done all at once.

Open the folder, 'Comms Tools' in which the tools are stored, select all the tools and drag them and drop over the system folder. You will be asked if they should be stored in the Extensions Folder. Answer 'yes' and the system will store them in the correct place.

HyperCard is not configured with sufficient memory

The symptoms may be 'Low Memory' warning messages or unexpected applications failures. You should increase the amount of memory available to HyperCard by doing the following.

Find the HyperCard stack.

Choose 'Find' from the finder File menu and locate the HyperCard application.

Get its information

Once the stack is highlighted you choose 'Get Info' from the File menu

Change the current size

You locate the current size box and increase it. The default for HyperCard is 1000 so it is likely that this will need to be increased. Increase it again if you get low memory warnings.

Warnings

a) We seem to have found a bug with the VT102 Terminal emulation tool. If the tool's 'Keyclick Sound' option (in the 'keyboard' panel of the tool's dialog) is selected and the tool's menu, (Keys) is shown, then selecting from the menu will crash AYS. The problem goes away if the 'Keyclick' option is turned off.

b) If AYS has one or more open service or text edit windows in one or more active stacks when HyperCard is terminated via the 'Quit HyperCard' option, (or command Q) then -

   1) The service windows may not be closed correctly. That is, if a service has an associated logout script then it will not be executed prior to the window being closed.
   2) If the text edit windows associated with any ‘background’ stacks contain unsaved text then this will be lost on closure.

   This problem is unavoidable due to the manner in which HyperCard handles external windows at termination time. It is therefore recommended that all windows are closed ‘manually’ prior to quitting HyperCard.

c) The TGE tool that is included with AYS contains a memory leak that results in a 2-3k loss of memory each time a connection is created. Consequently, if sufficient number of TGE configured services are opened/closed, AYS will eventually run out of memory. To help prevent this, there is a built in memory check, executed each time a service is launched, that warns of low memory.

   The author of the TGE tool, Tim Endress has been informed of this problem.

Distribution Rights and Legal Information

No charge may be levied for this program, nor may it be included as part of any package sold for profit without express authorisation from the author. It may be distributed freely via BBSs and other electronic data services that do not explicitly charge for file downloads, that is, do not charge a per file rate as opposed to a per hour rate. Further, no service or distributor may attach additional restrictions to this program or to its distribution. This program must be distributed complete with all documentation and additional programs.

AYS is provided “as-is” without warranty of any kind, including the implied warranties of merchantability and fitness for a particular purpose. The authors do not warrant, guarantee, or make any representations regarding the use, or the results of the use, of the software or accompanying instructions in terms of correctness, accuracy, reliability, or otherwise. The entire risk as to the results and performance of the software is assumed by you. If the software or instructions are defective, you, and not the authors, assume the entire cost of all necessary servicing, repair or correction. The authors shall not be liable for any damages whatsoever arising out of your use or inability to use this product, even if the author has been advised of the possibility of such damages.

Appendices

A. XFCN and XCMDs used in 'Card' Stacks.

a) InitialiseCWindows

Code Type XFCN.

Function Initialises the Communications toolbox and AYS globals.

Arguments No arguments required.

Returns a) An 'OK' code = 0.

or

b) An error code = -1

b) ServiceWindow

Code Type XFCN.

Function Creates and controls a 'HyperCard' external service window.

Arguments None.

Returns a) A numeric value that represents the window created.

or

b) An error code = -1

c) DoCService

Code Type XFCN.

Function Performs one of two operations on a service.

a) OPEN Launches the named service.

b) CLOSE Terminates the named service.

All information necessary to open or close the service is obtained from the HyperCard 'Card' of the same name as the service being operated on.

Arguments a) The operation required, either 'OPEN' or 'CLOSE'

b) if closing a service then

the name of the service to be closed.

if opening a service then

i) the name of the service to be opened.

ii) a 'true' or 'false' value that specifies whether the 'Windows' menu 'Settings' sub-menu flags should be reinstated from

the last session.

iii) an optional 'true' or 'false' value that specifies whether the service window should be shown or left hidden following successful opening of the service. if the argument is not included the value will be assumed to be 'true' and thus the service window will be shown.

Returns a) A name of the window that was successfully opened or closed.

or

b) An error code = -1

d) **EditWindow**

Code Type XFCN.

Function Creates and controls a 'HyperCard' external text edit window.

Arguments None.

Returns a) A numeric value that represents the window created.

or

b) An error code = -1

e) **DoEdit**

Code Type XFCN.

Function Performs one of two operations on an text edit window.

a) OPEN Launches the named editor.

b) CLOSE Terminates the named editor.

Arguments a) The operation required, either 'OPEN' or 'CLOSE'

b) if closing a text edit window then

the name of the window to be closed.

if opening a text edit window then

i) the name of the text editor to be opened.

ii) a 'true' or 'false' value that specifies whether the text edit window should be shown or left hidden following successful opening of the text edit window.

iii) an optional value that specifies the default font to be used. If the value is not supplied 'Monaco' is assumed.

iv) an optional value that specifies the default font size to be used. If the value is not supplied 9 is assumed.

Returns a) A name of the window that was successfully opened or closed.

or

b) An error code = -1

f) **SuspendWindows**

Code Type XCMD.

Function Suspends all active windows belonging to the stack by deactivating and hiding them. Only called upon receipt of a 'SuspendStack' message.

Arguments No arguments required.

Returns	No codes returned.

g) **ResumeWindows**

Code Type	XCMD.
Function	Resumes all active windows belonging to the stack by reactivating and showing them. Only called upon receipt of a 'ResumeStack' message.
Arguments	No arguments required.
Returns	No codes returned.

h) **ExecuteServiceScript**

See appendix D

i) **SendToService**

See appendix D

j) **ExpectFromService**

See appendix D

B. XFCN and XCMDs used in 'Resource' Stacks.

a) **InitialiseRWindows**

Code Type XFCN.

Function Initialises the Communications toolbox and AYS globals and builds a comma separated list of available service names (ie. 'srvc' resources).

Arguments No arguments required.

Returns a) A comma separated list of available service names

or

b) An error code = -1

b) **ServiceWindow**

Code Type XFCN.

Function Creates and controls a 'HyperCard' external service window.

Arguments None.

Returns a) A numeric value that represents the window created.

or

b) An error code = -1

c) **DoRService**

Code Type XFCN.

Function Performs one of two operations on a service.

a) OPEN Launches the named service.

b) CLOSE Terminates the named service.

All information necessary to open or close the service is obtained from the 'srvc' resource of the same name as the service being operated on.

Arguments a) The operation required, either 'OPEN' or 'CLOSE'

b) if closing a service then

the name of the service to be closed.

if opening a service then

i) the name of the service to be opened.

ii) a 'true' or 'false' value that specifies whether the 'Windows' menu 'Settings' sub-menu flags should be reinstated from the last session.

iii) an optional 'true' or 'false' value that specifies whether the

service window should be shown or left hidden following successful opening of the service. if the argument is not included the value will be assumed to be 'true' and thus the service window will be shown.

Returns a) A name of the window that was successfully opened or closed.

or

b) An error code = -1

d) **EditWindow**

Code Type XFCN.

Function Creates and controls a 'HyperCard' external text edit window.

Arguments None.

Returns a) A numeric value that represents the window created.

or

b) An error code = -1

e) **DoEdit**

Code Type XFCN.

Function Performs one of two operations on an text edit window.

a) OPEN Launches the named editor.

b) CLOSE Terminates the named editor.

Arguments a) The operation required, either 'OPEN' or 'CLOSE'

b) if closing a text edit window then

the name of the window to be closed.

if opening a text edit window then

i) the name of the text editor to be opened.

ii) a 'true' or 'false' value that specifies whether the text edit window should be shown or left hidden following successful opening of the text edit window.

iii) an optional value that specifies the default font to be used. If the value is not supplied 'Monaco' is assumed.

iv) an optional value that specifies the default font size to be used. If the value is not supplied 9 is assumed..

Returns a) A name of the window that was successfully opened or closed.

or

b) An error code =-1

f) **SuspendWindows**

Code Type XCMD.

Function Suspends all active windows belonging to the stack by deactivating and hiding them. Only called upon receipt of a 'SuspendStack' message.

Arguments No arguments required.

Returns No codes returned.

g) **ResumeWindows**

Code Type	XCMD.
Function	Resumes all active windows belonging to the stack by reactivating and showing them. Only called upon receipt of a 'ResumeStack' message.
Arguments	No arguments required.
Returns	No codes returned.

h) **ExecuteServiceScript**

See appendix D

i) **SendToService**

See appendix D

j) **ExpectFromService**

See appendix D

C. HyperCard Globals Maintained by AYS.

AvailableResources	Contains a comma separated list of the names of all the available service resources within a custom built ‘resource’ stack.
AvailableCards	Contains a comma separated list of the names of all the available service cards within within a custom built ‘card’ stack.
CapturedData	Contains the text captured by the text capturing feature, see appendices D and F.
WindowsToClose	Contains the names of the windows for which there are close requests outstanding.
Services	Contains the number of active service windows for this stack.
Editors	Contains the number of active text edit windows for this stack.
Windows	Contains the number of active windows for this stack (the sum of globals 'Services' and 'Editors').
Externals	Contains the total number of HyperCard external windows that have been opened during the current session (the sum of the 'Windows' globals for each active AYS stack).
ActiveService	Contains the name of either the currently active service window, or if the frontmost window is a text edit window then the most recently active service.
ActiveEditor	Contains the name of either the currently active text edit window, or if the frontmost window is an service window then the most recently active text edit window.
ActiveWindow	Contains the name of the currently active window
ServiceNames	Contains a comma separated list of active service window names.
EditorNames	Contains a comma separated list of active text edit window names.

D. HyperTalk Extensions

Several extensions have been provided to enable HyperTalk scripts to communicate with AYS services, these extensions are defined below

1) **Functions**

a) **ExecuteServiceScript**(Arg1, Arg2, Arg3)

Arg1 = Name of service for which the handler is to be executed.

Arg2 = Name of handler to be executed.

Arg3 = Where the handler is to be sent

1 = The handler is sent to the card which has the same name as Arg1 (ie. the name of the service).

2 = The handler is sent to the background of the current card.

3 = The handler is executed in the normal fashion and is subject to the normal message sending order.

Returns a) true if, following execution of the handler 'the Result' contains true.

b) false if, following execution of the handler 'the Result' contains 'false' or the handler failed to execute properly or the handler was aborted.

NB. Using 'ExecuteServiceScript' to execute service scripts is, in most cases preferable and in some cases absolutely necessary. For instance in a stack that contains 'resource' services it is the only way to run scripts associated with the Service other than using the pop-up 'Scripts' menu.

b) **ExpectFromService**(Arg1, Arg2, Arg3, Arg4 Arg5)

Arg1 = Name of service to be scanned.

Arg2 = Number of seconds before a timeout occurs.

Arg3 = 'true' or 'false' determines whether a 'moretime' dialog is to be shown when a timeout occurs.

Arg4 = Mandatory search string, max. 255 characters.

Arg5 |

Arg6 | Optional additional search strings.

Arg7 |

Arg8 |

NB. The text strings to be used for searching may contain character pairs that together define a control character. That is, the character "^" (shift 6) is interpreted as defining the following character to be a control character. The character is converted to its numeric control value by subtracting 64 from its character value, eg. ^A = control A and ^[= escape.

Returns -1 = Error occurred whilst executing function.
0 = None of the search string were found
1-5 = The number of the search string found.

NB. This function will not recognise the termination string associated with the service should it occur in the text received from the service. However, as soon as the script that contains this function terminates, string recognition is triggered, resulting in the subsequent closing of the service.

2) **Commands**

a) **SendToService** Arg1, Arg2, Arg3

Arg1 = Name of service to which the text string is to be sent.

Arg2 = Text string to be sent to the service

NB. The text string may contain character pairs that together define a control character. That is, the character "^" (shift 6) is interpreted as defining the following character to be a control character. The character is converted to its numeric control value by subtracting 64 from its character value, eg. ^A = control A and ^[= escape.

Arg3 = Text modifier

i)'slowly' text sent at 6 chars per second.

ii)'break' a long break is executed before the text is sent.

Returns -1 = Error occurred whilst sending data.

0= Function terminated ok.

3) **Properties**

All windows...

b) get the visible of window 'windowname'
set the visible of window 'windowname' to true/false

b) get the location of window 'windowname'
set the location of window 'windowname' to 'point'

c) get the height of window 'windowname'

d) get the width of window 'windowname'

Text Edit windows...

a) set the AppendText of window "editname" to "text"
...appends the text contained in "text" to the text edit window.

b) get the Text of window "editname"

...returns the “text” of the text edit window.

Service windows...

a) get the scripts of window ‘servicename’

...returns a "/" separated list of service scripts that are associated with the service window.

b) get the terminalSelection of window ‘servicename’

...returns the text framed by the current service window selection.

c) set the AttachLogFile of window 'servicename' to 'path'

...attaches the file defined by the path name contained in 'path' to the service window, all data subsequently scrolled 'off-the-top' of the service window will be written to the log file.

NB. If there is no text file of the same name as that specified then a text file will firstly be created. If a text file of the specified name already exists then the logged data will overwrite the existing file data.

4) **Messages**

All windows...

a) send 'ClearWindow' to window 'windowname'

...clears the contents of the window. If the window is a service window then only the contents of the terminal emulation will be cleared, the scrolled area is untouched. If the window is a text edit window then the entire contents are cleared.

b) send 'Activate' to window 'windowname'

...activates the named window ie. the window will be shown and made the frontmost active window.

Service windows...

a) send ‘Copy’ to window ‘servicename’

b) send ‘Paste’ to window ‘servicename’

c) send ‘SelectAll’ to window ‘servicename’

...selects (highlights) the contents of the window. If the window is a service window then only the contents of the terminal emulation will be selected, the scrolled area is untouched. If the window is a text edit window then the entire contents are selected.

d) send ‘DetachLogFile’ to window ‘servicename’

...detaches the attached file from the service window.

e) send ‘StartCapture’ to window ‘servicename’

...starts the capture of all text received by the service into the global variable 'capturedData'. There is only one global variable, consequently, if this feature is used simultaneously by more than one service the variable will be overwritten.

f) send ‘StopCapture’ to window ‘servicename’

...stops the capture of all text received by the service.

g) send ‘SetupService’ to window ‘servicename’

...initialises several essential facets of a service.

5) **Useful HyperTalk Scripts**

Functions...

a) **OpenService(servicename, retainSettings, showWindow)**

Launches the service named by 'servicename'. If the second argument is equal to 'true' then the settings of the 'Settings' submenu will be reinstalled from the previous activation of the service. If the argument is equal to 'false' then the 'settings' submenu will be set to default settings. If the third argument is equal to 'true' then the service window will be shown (and made the frontmost active window) after the successful launching of the service. If the argument is equal to

'false' then the window will remain hidden following the successful launching of the service. Returns either true or false to indicate whether the service was launched successfully or not.

a) **OpenEditor(editorname, showWindow)**

Launches a text editor named by editorname. If the first argument is equal to 'true' then the text edit window will be shown (and made the frontmost active window) after the successful launching of the editor. If the argument is equal to 'false' then the window will remain hidden following the successful launching of the editor. Returns either true or false to indicate whether the editor was launched successfully or not.

Commands...

a) **CloseService** servicename

Closes the named service.

b) **CloseEditor** editorname

Closes the named text edit window.

c) **CloseAllWindows**

Closes all service and text edit windows.

E. AYS Error Codes.

1. This XFCN/XCMD has been called with invalid arguments or with an incorrect number of arguments.

   Check appendices A , B and D to determine the number and type of arguments that the specified XFCN/XCMD requires.

2. The requested action on the specified window is invalid since there are no currently open AYS windows.

   According to the internal data structures maintained by AYS there are no currently open (service or text edit) windows. AYS can only recognise windows that it has created.

3. AYS has exhausted its memory allocation resulting in an internal HyperCard error.

   An action requested by AYS of HyperCard has failed and is almost certainly due to the lack of available memory. HyperCard can be reconfigured to use a greater amount of memory. To check and/or change this setting, refer to the TeachText document ‘Installation Instructions’.

4. There is insufficient memory remaining for the following operation.

   The specified operation has failed due to the lack of available memory. HyperCard can be reconfigured to use a greater amount of memory. To check and/or change this setting, refer to the TeachText document ‘Installation Instructions’.

5. The Connection tool will not accept the 'termination' string as a search string.

   The Communications toolbox Connection tool that you are using has refused to accept the termination string defined for the service. This error is likely to be Connection tool specific and we therefore suggest that you refer to the accompanying tool documentation.

7. AYS is attempting to use an invalid resource of the following type.

   AYS is unable to locate a specific resource. Either the stack has been corrupted or resources have been removed from the resource fork of the stack. To ensure that the stack contains a full set of resources reinstall one of the existing services.

8. This document has already been opened for writing and will therefore only be opened with 'Read Only', (RO) access

   This message is output if the text file specified at the launching of a text edit window is already opened for writing (by AYS, another stack or application) . AYS is thus prevented from writing to the text file.

9. AYS is unable to initialise the specified Connection tool.

   The Connection tool associated with the service being launched cannot be initialised. Firstly check that your system has all the necessary network software installed eg. MacTCP and then check the connections in the back of your machine.

If neither of these are at fault then try reinstalling the affected service Connection tool.

10. This document or its disk is locked. You MAY not be able to save any modifications.

    This message is output if the text file specified at the launching of a text edit window is locked by another stack/application or that the disc that it resides on is locked. You may edit the file, but you may lose any changes made at the time that you attempt to save the file.

11. AYS is unable to initialise the specified Terminal Emulation tool

    The Terminal Emulation tool associated with the service being launched cannot be initialised. Firstly check that your system has all the necessary network software installed eg. MacTCP and then check the connections in the back of your machine. If neither of these are at fault then try reinstalling the affected service Terminal Emulation tool.

12. The maximum number of strings, (11) have already been created

    The maximum number of strings that may be added to the ‘Strings’ menu has been reached. To add a new string, one of the existing strings must firstly be removed.

13. AYS is unable to obtain environment data for specified tool.

AYS is unable to complete an internal Comms. toolbox related task. It is suggested that the system be rebooted.

14. AYS is unable to open the specified service Connection.

    The Connection tool is unable to open the required connection. Firstly check that your system has all the necessary network software installed eg. MacTCP and then check the connections in the back of your machine. If neither of these are at fault then investigate the network address that the service is configured for. Failing this, check your network connection.

15. The service script has failed to execute to completion.

    The service script just activated has failed. Correct the script and retry.

16. The Comms. Toolbox, system software has not been installed on this system. This must be done before AYS can proceed .

    It is recommended that AYS is run under System 7.0 however if you running under System 6.0.x and your system does not have the Communications toolbox installed then this message will result. You must install the Communications toolbox before proceeding.

17. AYS is unable to initialise one or more of the following system modules - the Connection, Terminal Emulation or Comms. Resource Managers.

    It is recommended that AYS is run under System 7.0 however if you running under System 6.0.x and the Communications toolbox has not been installed correctly then this message may result. You must reinstall the Communications toolbox before proceeding.

18. The maximum number of services, (10) have already been opened.

    The number of active services is limited to 10.

19. The maximum number of text edit windows, (10) have already been opened.

    The number of active text edit windows is limited to 10.

20. There are no recognisable Connection tools available for use with the Comms. toolbox on this system.

    The Communications toolbox on your system does not contain any Connection tools. You must install at least one before proceeding. To install the Connection tool that is bundled with AYS read the document ‘Installation Instructions’.

21. There are no recognisable Terminal Emulation tools available for use with the Comms. toolbox on this system.

    The Communications toolbox on your system does not contain any Terminal Emulation tools. You must install at least one before proceeding. To install the Terminal Emulation tool that is bundled with AYS read the document ‘Installation Instructions’.

22. The Connection being read from or written to is no longer open, (possibly timed

out!).

This may occur if length delays are occurring on the affected host connection. Try relaunching the service to obtain a better connection.

23. AYS is unable to obtain the Connection status data

    Indicates that the Connection being read from/written to is no longer open. Try relaunching the service.

24. Error has been detected whilst opening or reading from the specified file.

    Indicates that there is an fault with the attached file. Try closing and then reopening the file.

25. AYS is unable to locate the specified Connection tool in the system folder.

    The Connection tool that the service being launched is configured for is not installed in your System. Reconfigured the service or install the required Connection tool.

26. AYS is unable to locate the specified Terminal Emulation tool in the system folder.

    The Terminal Emulation tool that the service being launched is configured for is not installed in your System. Reconfigured the service or install the required Terminal Emulation tool.

27 The specified Connection tool failed to accept the configuration tokens.

The Connection tool associated with the service being launched is unable to accept the configuration tokens defined for the service. This indicates that the too or the string of configuration tokens have been corrupted. It is suggested that you reinstall the affected service.

28 The specified Terminal Emulation tool failed to accept the configuration tokens.

The Terminal Emulation tool associated with the service being launched is unable to accept the configuration tokens defined for the service. This indicates that the tool or the string of configuration tokens have been corrupted. It is suggested that you reinstall the affected service.

29 An error was detected whilst writing to the active log file.

Indicates that there is an fault with the attached file. Try detaching and then reattaching the log file.

29 An error was detected whilst writing to the active log file.

Indicates that there is an fault with the attached file. Try detaching and then reattaching the log file.

30 The text accumulation window has reached its maximum size of 32000 characters.

AYS is limited to 32k text edit windows.

31 The specified file exceeds the max. allowable size (32000 chars.) and will be truncated upon input.

AYS is limited to 32k text edit windows.

33 The document cannot be saved since there is insufficient disk space remaining.

Try freeing up some disk space by deleting a file and then try to save the file again.

36 This file cannot be written too since it has already been opened for writing.

This message is output if the text file specified for logging purposes or for the saving of a service window selection has already been opened for writing (by AYS, another stack or application).

37 This file cannot be written too since it or its disk is locked.

This message is output if the text file specified for logging purposes or for the saving of a service window selection is locked by another application or that the disc that it resides on is locked.

38 The specified text file does not contain any recognisable keywords.

Output following an unsuccessful scan of a text file. service parameters will have to ‘manually’ specified.

39 The specified service name does not point to a valid HyperCard 'card' within the current stack.

An action requiring the location of a HyperCard card associated with an AYS window has failed since the card cannot be found within the active stack. Either the

stack has been corrupted or window has been incorrectly named.

40 The specified window name cannot be found in the list of active windows.

According to the internal data structures maintained by AYS there is no currently open window of the specified name (service or text edit). Either the stack has been corrupted or the window has been misnamed. AYS can only recognise windows that it has created.

41 The installer is unable to locate the target stack. Please redefine the target via the 'Define Target' item in the 'Function' menu.

Attempting to install or delete a service from a nonexistent target.

42 The resource fork of the target file cannot be opened.

Almost certainly due to the fact that the target resource fork has already been opened by another application. Terminate the offending application before proceeding.

43 The target stack does not contain the specified service definition.

Attempting to remove a service from a target stack that does not contain the service. Check the contents of the target stack.

44 The amount of memory that HyperCard has been loaded with is set too low. The minimum size necessary to run AYS is 1100k.

In order to run AYS successfully it is recommended that HyperCard is configured with a minimum of 1100kb memory partition. To check and/or change this setting, refer to the document 'Installation Instructions'

45 The resource to be copied to the target stack cannot be found in this stack.

The installer stack is unable to locate one or more of the resources that are to be copied to the target stack. This indicates that the stack has been corrupted.

46 Invalid password.

Output if the password supplied to the 'protected' service does not match that defined for the service.

47 The time restrictions associated with services launched by this target stack have now been removed.

Output to indicate successful unlocking of the target stack following registration.

48 AYS is unable to locate the specified logging file.

The path named supplied to the HyperTalk 'property extension cannot be located. Check that the file exists and respecify.

49 There is a logging file already attached to this service.

Attempting to attach a logging file to a service that already has a logging file attached.

F. Sample Scripts

```
on DownLoadMail LogFileName
 Global ActiveService, ServiceNames
 --
 -- If the service is not already active then launch it
 --
 if "Sample Service" is not in ServiceNames then
   if not(OpenService("Sample Service", true, true)) then exit DownLoadMail
 else
   send activate to window "Sample Service"
 end if
 --
 -- If there is enough room, position the service window below
 -- the card window
 --
 PositionWindows
 --
 -- Attach the receiving file, download the mail, clear the screen to
 -- force all of the last screen to the log file and then detach
 -- the file
 --
 set AttachLogFile of window ActiveService to LogFileName
 put ExecuteServiceScript(activeService, "ReadMail", 3) into status
 send ClearWindow to window ActiveService
 send "DetachLogFile" to window ActiveService
 --
 -- If reading of mail went OK ask if it's OK to log out otherwise
 -- post an error and logout
 --
 beep
 if not(status) then
   answer "Error detected whilst reading mail" with "Continue"
   CloseService ActiveService
 else
   answer "Logout from Service now ?" with "No" or "Yes"
   if it = "Yes" then CloseService ActiveService
 end if
end DownLoadMail
o=============================================================================o
on ReadMail
 Global ActiveService
 --
 -- Start process off by sending a carriage-return to bring up the
 -- menu, and then select the appropriate option to download the mail
 --
 SendToService ActiveService, return, ""
 if the Result = false then return false
 if ExpectFromService(ActiveService, 20, true, "Select option") = 1 then
   SendToService ActiveService, "2" & return, ""
   if the Result = false then return false
   --
   -- Go into endless an loop grabbing mail items until
   -- 'End of Mail' detected
   --
   repeat
     put ExpectFromService(ActiveService, 30, true,
                    "--More--", "End of Mail.") into xx
     if xx = 0 or xx = -1 then return false
     SendToService ActiveService, return, ""
     if the Result = false then return false
     if xx = 2 then exit repeat
```

```
   end repeat
   return true
 end if
 return false
end ReadMail
o=============================================================
====o
on PositionWindows
 Global ActiveWindow
 --
 -- Determine if there is enough space on the screen to position the
 -- Service window below the card window.
 --
 put the height of the card window into xHeight
 put the width  of the card window into xWidth
 put (item 4 of the screenrect) - (item 2 of the screenrect) into yHeight
 put (item 3 of the screenrect) - (item 1 of the screenrect) into yWidth
 put the height of window ActiveWindow into zHeight
 put the width  of window ActiveWindow into zWidth
 if (yHeight - xHeight) > (zHeight + 50) then
   put the trunc of ((yWidth - xWidth)/2) into xLeft
   set the left of the card window to xLeft
   set the top  of the card window to 45
   put the trunc of ((yWidth - zWidth)/2) into zLeft
   put (zLeft - xLeft) into item 1 of sLoc
   put (xHeight + 25)  into item 2 of sLoc
   set the loc of window ActiveWindow to  sLoc
 end if
end PositionWindows
o=============================================================
====o
on EditServiceText
 Global editorNames, activeService
 --
 -- Select the terminal emulation screen and then grab the selection
 --
 send SelectAll to window ActiveService
 get the TerminalSelection of window ActiveService
 put it into Text
 --
 -- If the specified editor window has not already been created, create
 -- it leaving the window hidden. Place the selected text into the window
 -- and the show the window
 --
 if "Selected Text" is not in EditorNames then
    get OpenEditor("Selected Text", false)
 end if
 send ClearWindow to window "Selected Text"
 set AppendText of window "Selected Text" to text
 set visible of window "Selected Text" to true
end EditServiceText
o=============================================================
====o
on CaptureReport
 Global ActiveService, ServiceNames, CapturedData
 --
 -- Firstly start the service keeping the window hidden
 --
 if OpenService("Sample Service", true, false) then
   --
   -- Open succeeded so the login script should have been run and
   -- ActiveService global should now be set
   --
   -- Now run the data capture script
   --
```

```
if ExecuteServiceScript(activeService, "ReportCapture", 3) then
```

```
    --
    -- Capture script worked OK so we can now assume that there is
    -- something in 'capturedData'
    --
    -- <DO SOMETHING WITH CapturedData>
    --
  end if
  --
  -- Regardless of what happened close the service
  -- NB. I don't know why, but you may have to wait 10 ticks before
  --     before closing the service.I have, on occasion had a strange
  --     'Out of memory' error when it has heaps left (pun!). What
  --     seems to happen is that the global isn't recognised since
  --     CloseService activeService      ---- will fail
  --  BUT
  --     CloseService "the actual name"  ---- will succeed
  --
  wait 10
  CloseService ActiveService
end CaptureReport
o=============================================================================o
on ReportCapture
 Global ActiveService, CapturedData
 --
 -- Step through the various prompts looking for the required one
 -- and then set capture on until the terminating prompt is found
 --
 if ExpectFromService(ActiveService, 10, true, "Some prompt") = 1 then
    SendToService ActiveService, "Some command" & return, ""
    if the Result = false then return false
    --
    -- Found it so start capturing now
    --
    send StartCapture to window ActiveService
    SendToService ActiveService, "5" & return, ""
    put ExpectFromService(ActiveService, 10, true, "Continue") into status
    --
    -- Stop capturing regardless of whether the string was found or not
    --
    send StopCapture to window ActiveService
    if status = 1 then
       SendToService ActiveService, "Quit" & return, ""
       if the Result = false then return false
       return true
    end if
 end if
 return false
end ReportCapture
o=============================================================================o

function openService ServiceName, RetainSettings, ShowWindow
 set cursor to busy
 put false into Status
 set the cantAbort of this stack to true
 put ServiceWindow() into Ptr
 if Ptr > 0 then
   put DoCService("Open",ServiceName,RetainSettings,ShowWindow) into reply
   if reply <> -1 then
     put true into Status
     send SetupService to window ServiceName
     get the Scripts of window ServiceName
     if it contains "/LoginToService/" then
       if not(ExecuteServiceScript(ServiceName, "LoginToService", 1)) then
```

```
    get DoCService("Close", ServiceName)
```

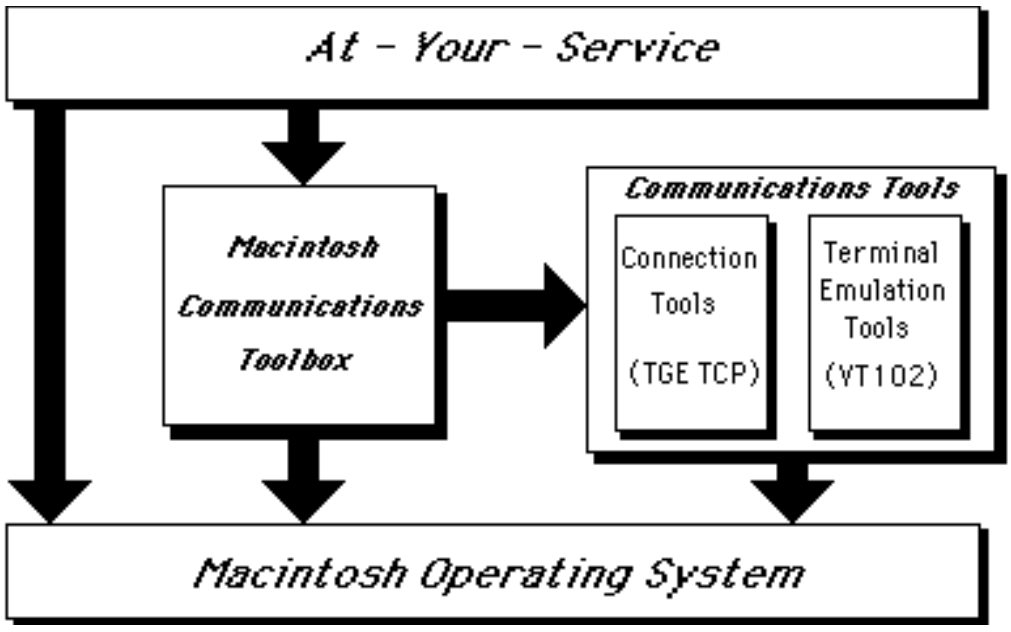
```
        put false into Status
       end if
     end if
   end if
 end if
 if not(Status) then close window "New-Window"
 set the cantAbort of this stack to false
 return Status
end openService
o=============================================================================o
on closeService ServiceName
 set the cantAbort of this stack to true
 get the Scripts of window Servicename
 if it contains "/LogoutFromService/" then
   get ExecuteServiceScript(ServiceName, "LogoutFromService", 1)
 end if
 get DoCService("Close", ServiceName)
 close window ServiceName
 set the cantAbort of this stack to false
end closeService
o=============================================================================o
function openEditor EditorName, ShowWindow
 put EditWindow() into Ptr
 if Ptr > 0 then
   if DoEdit("Open", EditorName, ShowWindow) <> -1 then
     return true
   end if
 end if
 close window "New-Window"
 return false
end openEditor
o=============================================================================o
on closeEditor EditorName
 get DoEdit("Close", EditorName)
 close window EditorName
end closeEditor
o=============================================================================o
on closeAllWindows
 Global  Services, Editors, ServiceNames, EditorNames
 repeat with x = Services down to 1
   closeService item x of ServiceNames
 end repeat
 repeat with x = Editors down to 1
   closeEditor item x of EditorNames
 end repeat
end closeAllWindows
```

G. The Communications Toolbox

The Macintosh Communications Toolbox is system software that provides Apple Macintosh applications with standard access to communication services, including data connection and terminal emulation.

The Macintosh Communications Toolbox is Apple computer's strategic communications development platform. It is designed to support multivendor connectivity for Macintosh computers in environments such as Digital, IBM, OSI, TCP/IP, and the Appletalk network system.

An extension of the Macintosh Toolbox system software, the Macintosh Communications Toolbox consists of several managers such as the Connection manager and the Terminal Emulation manager. These managers have been designed to work in concert with tools, such as a modem connection tool or VT102 terminal emulation tool, created by Apple and third party developers to provide applications with standard communications functions.



Communications Tool Box tools are stored in the extensions folder inside the System 7's 'Systems' folder. You need to determine which tools to use, which is dependant on the hardware you use to connect to other machines.

Serial Cable This type of connection requires the serial tool, only one connection is possible at a time with most systems.

Modem This type of connection requires a modem tool of which several are

available. Only one connection is possible at a time.

Appletalk If you have access to a network that is TCP/IP based you can use MacTCP and a TCP/IP tool such as TGE TCP tool supplied with AYS (written by Tim Endres).

Ethernet If you have access to a network that is TCP/IP based you can use MacTCP and a TCP/IP tool such as TGE TCP tool supplied with AYS (written by Tim Endres).

You will need to determine which terminal emulation is supported by both your Macintosh and the remote computer. The most common emulation is vt100 and fot this you can use the vt102 tool.

Other emulations are available and you should check with vendors of Communications Toolbox tools.

Configuring the Tools.

Each tool has its own set of configuration fields The values of each field depend on the environment in which the tool is used. Expert help should be sought to determine these field values. With the TGE TCP tool it is probable that all that is required is the address of the host computer.

Sources for CTB Tools

The following information (the accuracy of which cannot be guaranteed) may be of assistance in acquiring Communications Toolbox tools.

a. The Hayes modem tool may be retrieved by anonymous ftp from sumex-aim.stanford.edu directory = info-mac/comm.
b. Intercon produce a TCP/IP/telnet tool as part of their TCP Toolz And Toyz package. (InterCon Systems Corporation. 950 Herndon Parkway, Herndon, VA 22070 +1 703 709 9890).
c. Advanced Software Concepts (email 'ADV.SOFT@applelink.apple.com) produce amongst others a TCP/IP/telnet tool called TCPack.
d. Synergy Software produce the VersaTerm telnet tool .
e. Pacer Software produce the PacerTCP TCP/IP tool .
f. Software Ventures produce the MPTelnet telnet tool .
h Global Village Tech Support, globalvill@aol.com are planing a modem tool.