WYSI-Help - Index

General functions

Editing functions

Text editing functions

Defining how to run the help application

Help functions

Functions allowing to put Help into its final form

Edit menu

The **Edit menu** includes all the commands needed for managing **object** lists and **link** lists.

Delete

lets you delete selected object(s) or link(s).

Links

lets you create a link by grouping together a certain number of objects, or associate a link to an object (pre-selected objects).

Properties

lets you define the attributes of an object or link according to the type of help that will be triggered if the object or link is designated. For example, defining a link's properties can lead to editing a help message or topic.

The four following choices control the level of information displayed in object and link lists. Up to 5 levels of detailed information can be displayed.

Collapse one level

Using this command adds one more level of details to the information presented in a list.

Expand One Level

This command has the reverse effect of the preceding one in that it removes one level of details from the information being displayed.

Lock to Forced Level

This command saves a picture of the currently displayed level of information. Thus, using the next command:

Force Detail Level

allows you to return to the level that was displayed when the picture was taken.

Preference

Choosing this option lets you customize WYSI-Help's editing functions to suit your particular needs. You can choose a different setting for each application being developed. These settings concern:

- hiding the **control window** (Hide Live Info),
- immediately defining the help associated with each link or object,
- minimizing WYSI-Help's main window when testing a help application,
- automatic saving of the help application before testing,
- hiding the icon bar,
- automatic loading of applications to be loaded (when defined and when loading a WYSI-Help application).

Help application

Help applications are simply programs that you'build with WYSI-Help.

These applications are composed of a certain number of **agents** and can apply to one or several Windows 3 programs.

Right mouse button

Most of the time, you communicate with Windows applications by using the left mouse button.

Some applications, like WYSI-Help, also make use of the right mouse button.

With WYSI-Help, the right mouse button is used to **capture objects**. When an object is the current object of the application being spied on, pressing on the right mouse button is your way of indicating to WYSI-Help that the object will be linked to a **help message** or **help topic**.

Note that you can reverse the roles of the right and left mouse buttons using Windows control panel. If you do so, the left mouse button should be used to capture objects.

See also:

Links

List of error messages

Illegal copy of WYSI-Help.
Error Loading Surveillance File
Surveillance File not Found
No surveillance file specified
String File not found.
Error loading Short help file
Could not load requested program.
No Application selected.
WYSI-Help already Running! Impossible to execute a second copy of WYSI-Help.
Could not load
Error Saving Surveillance file
Error Saving Short Help file
Sorry, this application does not run in Real Mode.
WYSI-Help Help file not Found.
Topic Editor not loaded.
Not enough memory to perform requested operation. Please close some applications and try again.
Not enough memory to perform requested operation, halting program.
Help include file missing or damaged. No test on context help possible.
Link name exists already.
When you rename HMRUN, only one application can be launched.
You cannot load an application with the same name as the renamed HMRUN.EXE.
The Topic help file loaded is not the one for the current application. Do you want to save it ?
Editor in wait state, please answer the dialog box in the editor first.'
File not found
Application is already in the list
You cannot have alternate mouse click both for Topic and Short Help.
You cannot have the same key(s) for Short, Topic and Index Help.
Non supported type of icon.
Error copying icon(s).

Agent

An agent comprehends the following components : **help event on an object** - **link** - **help**.

A help event is the user action on an object detected by WYSI-Help via the mouse or keyboard. WYSI-Help then links the given object to the associated help item and displays it on the screen. In other words, an event is considered as a user's request for help.

In this instance, help can represent both a **short help message** as well as a **help topic**. In fact, you can associate a given **object** with both a **help message** and a **help topic**, but the object will only be associated with one link.

See also:

Help Messages
Help Topics

Author

The person creating a help application is referred to as an author whereas the term "user" designates the person for whom help is intended.

Basic operating principles

WYSI-Help brings together in a single system all of the components needed to develop your own help applications for software products running under Windows 3.

Creating help consists in building **short help messages** (displayed in the title bar of an application) or **help topics** (a complete help application developed for a particular software product) and in associating these various short help messages or help topics to **objects** (menu choice, dialog box, button...) by means of **links** when help is **context sensitive**.

WYSI-Helps's context editor is thus divided into four main categories of functions:

- functions allowing for the description of **objects**,
- functions allowing for the creation of **agents**,
- functions allowing for editing short help messages and topics,
- functions allowing for the creation of links among topics (**hypertext links**).

With WYSI-Help, creating help becomes simple. All you need to do is use the mouse to designate the objects that will be available with help, and the keyboard to input short help messages and help topics.

WYSI-Help also includes an editor allowing you to create help topics. This editor, which is a true word processor, contains the functions allowing you to create **hypertext links**, as well as all the standard links provided for in Windows Help. Elaborating hypertext links is thus carried out naturally and simply.

To create **objects**, WYSI-Help provides for a capture utility. Using the mouse cursor, you simply point to the objects you want to capture and then you indicate the link that should be associated with them.

With WYSI-Help, you can at any time test and thus verify the help you just built. This is made possible by using the mouse to switch back and forth between define mode and test mode.

You can, of course, use WYSI-Help to build general help (as opposed to **context sensitive help**). In this instance, help will be arranged in topics and no matter how the topics are organized, the user will get access to them by selecting a choice from a help menu.

If this is the type of Help you wish to create, you should only start WYSI-Help's topic editor from WYSI-Help's main window. WYSI-Help's context editor should be used when you want to capture objects and link them so as to create context sensitive help sequences (we will also say "associating" help to an object).

In fact, when using the topic editor, you first build help topics and then you organize them as you wish (hypertext links, browsing sequences, search keywords, **index**, **pop-up help windows**). Access to these topics can be initiated by WYSI-Help or programmed directly in the applications.

Capturing objects

Capturing objects from an application for which help is being built represents one of WYSI-Help's basic operations.

In order to build **context sensitive help**, the actual context in which the user is found when he or she requests help must be identified. In other words, the very last object (current object) that preceded the **user**'s request for help must be located.

The process consists in consecutively capturing all of the objects that were manipulated by the user so as to reproduce the sequential chain of user actions (the actual working situation) that preceded the request for help.

To carry out this procedure, you simply browse through the application and for each object that needs to be captured, you press on the **right mouse button.**

While browsing through an application, the **control window** is **blue** if the object encountered hasn't been captured, and turns **yellow** once it has been.

Remark:

Capturing objects represents the first step in building a help application. Once the objects are captured, you'll need to associate them with **links**. By default, this operation is automatically proposed by WYSI-Help each time you capture an object. However, in certain cases, it may be more pratical to capture a group of objects before defining the corresponding links, help messages and help topics. To do so, you should cancel the **Preferences** option.

For more information about linking objects, refer to the chapter entitled "**Link Objects**" in the "Context Editor Reference Manual".

See also:

Color blue
Color yellow
Control Window
Link
Right mouse button

Choosing a color for short help messages

Overview

Short help messages are written in the title bar of the application for which help is being built. A dialog box will guide you in selecting the color you wish to use.

Procedure

To get access to this feature, choose **Short Help** from the **Define** menu.Then From the "Short Help Text Editor" dialog box, **choose** the Select **Color** button. This opens the "Select a Color" dialog box.

To select the background and foreground (text) colors for your help message:

1. Choose or modify the background color by typing B (for Background) or by clicking the corresponding button. Use the mouse to select the color of your choice among those diplayed.

2. Choose or modify the textcolor by typing T (for Text) or by clicking the corresponding button. Use the mouse to select the color of your choice among those diplayed.

3. To finish, choose the **OK** button.

Of course, you can choose to modify only the background color, or the text color or both the background and text colors.

Since using the other zones in this dialog box is self-explanatory, we will only discuss the Sample button in this section.

Sample:

Using this button will let you "sample" the colors you have chosen so as to make sure they can be seen on the screen. Once you click on the **Sample** button, the title of the "Select a Color" dialog box is replaced by the phrase: "*The Quick Brown Fox Jumps Over The Lazy Dog*". (Note that this phrase has no special meaning, but contains all the letters of the alphabet.)

Remark:

Defining the background and text colors for the short help messages used in a help application can be defined only once for that application. This simply means that all the short help messages that you define will be displayed using the same colors.

Choosing a printer (File menu)

Overview:

Windows 3 supports numerous printers and for each one provides a printer-driver file. WYSI-Help lets you select any of the printer-drivers supported by Windows 3.

Procedure:

1. Choose the **Select Printer** option from the **File menu** (**ALT F, S**).

2. From the list of available printers, choose the one you wish to use.

3. Use the **OK** button to confirm you choice.

Commands Used:

File - Select Printer

Available Printers

WYSI-Help displays the list of available printers. Choosing a printer from this list is carried out as for any Windows application.

Printer

This zone displays the name of the selected printer. The printer selection only takes effect once you've confirmed your choice (**OK** button).

See also:

Print

Collapse one level (Edit menu)

Overview

Two kinds of lists are presented in WYSI-Help's window: the list of **non-linked objects** and the list of existing **links**.

These lists can contain more or less information according to the level of details you wish to have displayed on your screen at a given time.

WYSI-Help provides for four different levels of details. These levels are numbered from 0 to 3. Collapsing a level results in reducing the number of detailed information presented in each list.

The current level is displayed in the **icon bar** and can be identified by a number varying from 0 to 3.

Procedure:

Choose **Collapse One Level** from the **Edit** menu (Alt E, C).

Commands used :

By default, when you choose this option, collapsing one level takes place for both the non-linked objects and the existing links. However, you may just want to obtain more or less details on a specific link or object.

If this is the case, simply click the **detail button** located in front of the corresponding link or object.

The "Collapse One Level" option has the opposite effect of the **Expand One Level** option.

The current level is displayed in black in the icon located to the far left of the icon bar. When this number appears dimmed, it means that the level is locked (see "Lock to Forced Level"). When you go down one level, the number turns back to black (the level is no longer locked).

Example:

Suppose you're currently at level 3 (this level displays the most details):

When you choose **Collapse One Level**, you go back to to level 2:

And so on, until you reach level 0, which displays the least amount of details .

The icon located to the far left of the icon bar gives you direct access to the "Collapse One Level" option. Note that only the arrow pointing to the left in the lower right square of this icon corresponds to the "Collapse One Level" option.

Color: Blue

By the color of the **control window** , you'll be able to know at all times the actual state of each object making up the application being spied on (the application for which help is being developed); its general characteristics and whether it has been **captured** or not.

The control window lets you choose which objects should be captured or not while browsing through the spied-on application. Thus, while focusing on an object with the mouse or when an object is the current one, the control window contains information which characterizes the object.

If the control window is blue, it means that if a request for help is made on the object, it will not trigger **context sensitive help**.

If the control window contains text written in **yellow**, it means that context sensitive help has already been associated with the object.

If you've already developed Windows applications, you'll probably recognize the the information contained in this panel. It's simply information characterizing the object, and indicates:

- a title,
- its class (type of object for Windows),
- an identification number (assigned by Windows),
- an implied link .

See also:

Color: yellow
Object

Color: Green

Any word or expression written in green and contained in a **help topic** signals a **hypertext link**.

Designating one of these words or expressions will bring you to a new help topic.

Color: Yellow

According to the color of the **control window**, you'll be able to know at all times the actual state of each object making up the application being spied on (the application for which help is being developed); general characteristics and whether it has been **captured** or not.

The control window lets you choose which objects should be captured or not while browsing through the spied-on application. Thus, while focusing on an object with the mouse or when an object is the current one, the ontrol window contains information which characterizes the object.

If the control window is blue, it means that if a request for help is made concerning the object, it will not trigger **context sensitive help**.

If the control window contains text written in **yellow**, it means that context sensitive help has already been associated with the object.

Example:

This is an example of the control window containing text written in yellow:

Note that the information written in yellow differs according to if the object is already associated or not with a **link**.

If the object hasn't been attached to a **link**, the information presented in the control window is the same as when the object hasn't been captured (designated by a blue control window). Refer to the example above.

On the other hand, when the object has been attached to a link, the name of the link and the defined **help message** appear in yellow as illustrated below:

See also:

Color: blue
Link
Object

Context sensitive help

Context sensitive help refers to help which is directly associated   with the user's working situation when he or she requests help. The requested help sequence is automatically displayed on the screen without the user having to search for help himself.

When context sensitive help is requested, WYSI-Help analyses the user's working situation (for example, to know which dialog box has been opened or which word has been selected in a field, ...) and establishes a connection between the user's actual working situation and the associated **help topic or topics**.

Control window (panel)

As its name implies, the **control window** lets you control the **capturing of objects** while browsing through the spied-on application (the one for which help is being built).

Thus, while focusing on an object with the mouse or when an object is the current one, the control window contains information which characterizes the object.

By the color of the control window, you can determine if the current object has been captured (**yellow** control window) , or not (**blue** control window).

As you can see, the control window is divided into two parts.

The left side contains WYSI-Help's title, written in red.

The right side contains the characteristics of the objects being focused on.

When editing help applications, both parts are visible.

In test mode, only the left side remains visible. In fact, clicking the left side of the control window (the WYSI-Help title), lets you enable and disable **test mode.**

Let's take a closer look at the right side of the control window.

As long as the current object hasn't been associated with a **link**, the control window displays the following information:

- a title,
- its class (type of object for Windows),
- an identification number (assigned by Windows),
- a proposed, **implicit link**.

If you've never programmed Windows 3 applications before, don't try to decipher this information now.

When an object has been attached to a link, the control window displays:

- the link name,
- the defined help message.

Even though WYSI-Help tries to position the control window where it least interferes with what you're doing, in certain cases, its presence may hinder your work. If this is the case, you can hide it by selecting the **Preferences** choice from the **Edit menu.**

Converting files

Note that this feature is not supported in version 1.0 of WYSI-Help.

In future versions of WYSI-Help , you'll be able to convert files containing help applications into .**RTF** files (**R**ich **T**ext **F**ormat) and vice versa.

The RTF format is the basic format allowing for the construction of Windows help sequences outside of the WYSI-Help context.

This will allow you to use and modifiy help files that haven't been developed with WYSI-Help.

Creating help

The WYSI-Help System is within the reach of all Windows application users.

With a little practice, you'll be able to produce effective help applications responding to your particular needs.

As described in the section entitled **Basic operating principles**, the process of creating help sequences is carried out in several steps:

- **defining help events** (keys triggering access to help),
- **defining applications to spy** on (or load),
- **capturing objects**,
- defining **links** (between objects and the corresponding help messages or help topics),
- defining **help messages** and their links,
- defining **help topics** and their links,
- defining WYSI-Help's **Default Behavior** ,
- testing the help application.

The above steps are presented in the order in which you'll normally develop a help sequence (logical order). However, nothing prevents you from carrying these steps out in a different order.
For example, you could define all your help topics, and only afterwards capture and link the various objects needed to trigger the help sequence.

The Learner's Manual provided with the WYSI-Help System will let you discover simply and progressively the various steps you'll need to use to create your own help application.

Default behavior (Define menu)

Overview:

The Default Behavior command lets you define the way in which WYSI-Help will handle a help request when the current object (the object that triggered help) hasn't been captured.

In this situation, WYSI-Help will try to do one of the following:

- associate an automatic **link** name,
- look for a parent **object**,
- branch to the help **index** if a **help topic** or **help message** has not been found.

So as to avoid the situation where nothing happens when the user requests help, a brief discussion of the **Default Behavior** command is presented immediately after the following section.

Procedure:

1. Choose **Default Behavior** from the **Define** menu (**Alt D, D**).
2. Choose the boxes corresponding to the default behavior option(s) you want to use.
3. Use the **OK** button to confirm your choice(s).

Note: clicking a check box containing an X clears the selection.

Define, Default Behavior

The rest of this section will explain how WYSI-Help's Default Behavior command operates. This discussion is divided into two parts: the first explains the individual commands that can be used with this option. The second will explain what happens when you combine the three commands.

Automatic Link Name

If you haven't associated a **link** with an **object** and this object is the current object when the user requests help, WYSI-Help will automatically search for the associated link by using either the object's title or the editing field's content (appears in the **control window** while editing help for that object).

Look For Parent

For those situations where a request for help is made on a non-linked object, it's recommended that you trigger help that is linked to a parent object.

As an example, let's examine a situation that you'll probably encounter frequently.

Let's suppose that a dialog box contains three objects: A, B, and C. Object A corresponds to a new feature, so you link it to help. Objects B and C are options frequently used in your application, and you feel that it isn't necessary to provide help for them. You would, however, like to provide help for the overall dialog box.

Being that the dialog box is the parent object for the other three objects (A, B, and C),
you can immediately see that the default behavior option "Look for Parent" is the one to use in this situation.

Index

Still assuming that the current object hasn't been associated with help, you may wish that the user's request for help takes him to the the help **index** (the menu containing the list of **help topics**).

Of course, this option should only be used for events triggering **help topics**.

Possible Combinations

If none of the default behavior options has been set and the user's request for help is triggered by a non-linked object, nothing happens (no help is provided).

Having studied how each one of the default behavior options operates when used individually, we'll now examine what happens when all three of these options are selected and the object triggering help is a non-linked one.

In this case, WYSI-Help will do one of the following :

A. Searches to know if the implied link corresponds to an existing link and according to the type of event found, will display either a help message or help topic.

B. Searches for a parent object. If a parent object exists, it becomes the current object. The same procedure carried out in step A is then applied (returns to A).

C. Proposes the help index if one exists.

We will not discuss the situation where only two out of three default behavior options are set, since this simply represents a sub-set of the situation where three options are set.

Remember that the default behavior options only become active once they have been set. By default, when you start a new application, these options are not set .

From one work session to the next, WYSI-Help records the default behavior settings. This, of course, applies only to the current application for which help is being developed. The default behavior options must be explicitly defined for each help application.

See also:

Control window
Color: blue
Index
Link
Object
Short help
Help Topic

Deleting an object (Edit menu)

Overview:

It is possible, at all times, to delete all of the non-linked objects that are highlighted (inverse video) in the non-linked objects list.

Procedure:

1. **Select** one or several objects from the list.

2. Choose **Delete** from the **Edit menu** (Alt E, D).

The deletion is carried out without any request for confirmation.

See also:

Selecting objects

Deleting a link (Edit menu)

Overview:

It is possible, at all times, to delete all of the existing links that are highlighted (inverse video) in the existing links list.

Procedure:

1. **Select** the links from the "Existing links" list.

2. Choose **Delete** from the **Edit menu** (Alt E, D).

A confirmation for deletion is requested:

See also:
Selecting a link

Defining applications to load (Define menu)

Overview:

WYSI-Help lets you embed help in one or more Windows applications at the same time.

As you develop your help application, you will need to specify the applications to be spied on.

You can do this in either of two ways.

The first consists in choosing **Applications to Spy** on from the **Define** menu. In this case, it's your responsability as well as the user's to load the applications manually (unless they're already loaded).

The second method consists in asking WYSI-Help to take care of loading the files to be spied on (**Applications to Load** from the **Define menu**). The applications to be loaded will then be started and stopped by WYSI-Help itself.

Procedure:

1. Choose **Applications to Load** from the **Define menu** (Alt D, L).

2. Select an application by entering its name or by looking through the list of files that are located on hard disk by using the Browse button.

3. Select the **Add button**. The application is then added to the list of applications to load.

4. To confirm, choose **OK**. If the **Automatic Launch** check box (**Edit** menu) has already been set (contains an X), all new applications will be automatically loaded by WYSI-Help and all deleted applications will be stopped.

Note: this option is also used to delete an application from the list. Simply select the file to delete (steps 2 and 3), then choose **Delete** .

Application name

This field is used to type in the name of the application. If necessary, you should designate the full pathname. You are not, however, obliged to furnish its extension (.EXE). Thus, the names WRITE and WRITE.EXE would be considered as equivalent. If the application is started with parameters, they must also be entered.

The application's presence on hard disk is controlled as soon as you choose the **Add button**.

You can also use the **Browse** button to add an application to the list.

Browse

The browse button lets you search through the various directories found on your hard disk for the application you wish load. This functionality is a common one in the Windows 3 environment.

Applications

WYSI-Help always lists the applications that you have defined as the ones to load. You can choose an application from this list so as to make it appear in the **Application Name** list box (for example, to delete it from the list).

Add

If there are no applications   listed in the Applications list box, the **Add** button appears dimmed. Choosing this button adds the selected application to the list of applications to load. WYSI-Help then checks its presence on hard disk. If it cannot be located, the following error message is displayed: "file not found".

Delete

If there are no applications   listed in the **Applications** list box or if an application from the list has not been selected, the **Delete** button appears dimmed. The Delete button erases the selected application from the list of applications to be loaded.

OK

It's only after having chosen the OK button that WYSI-Help effectively records the lists of applications to be loaded. As usual, as long as you haven't chosen the **OK** button, you can always change your mind and **Cancel** the current operation. Remember that the automatic loading of files only takes place if the **Automatic Launch** choice has been set (**Preferences** from the **Edit** menu). If you haven't set this option, you can always choose to load a file by selecting the **Launch Application(s)** choice from the **Surveillance menu**.

Defining applications to spy on

Overview:

WYSI-Help lets you embed help in one or more Windows applications at the same time.

As you develop your help application, you will need to specify the applications to be spied on.

You can do this in either of two ways.

The first consists in choosing **Applications to Spy** on from the Define menu. In this case, it's your responsability as well as the user's to load the applications manually (unless they're already loaded). This is the command that we'll be discussing here.

The second way consists in asking WYSI-Help to take care of loading the files to be spied on. In this case, you should choose **Applications to Load** from the **Define** menu.

Procedure:

1. Choose **Applications to Spy on** from the **Define menu** (Alt D, S).

2. Select an application by entering its name or by looking through the list of files that are located on hard disk by using the Browse button.

3. Select the **Add button**. The application is then added to the list of applications to load.

4. To confirm, choose **OK**.

Note: this option is also used to delete an application from the list. Simply select the file to delete (steps 2 and 3), then choose **Delete**.

Application name

This field is used to enter the name of the application. If necessary, you should designate the full pathname. You are not, however, obliged to furnish its extension (.EXE). Thus, the names WRITE and WRITE.EXE would be considered as equivalent. If the application is started with parameters, they must also be indicated.

Since its always possible to install an application to spy on after having added it to the list of applications to spy on, the application's presence on hard disk is not controlled.

You can also use the Browse button to fill in the information needed in this editing field.

Browse

The Browse button displays the list of applications that are currently running.

Applications

WYSI-Help always lists the applications that you have defined as the ones to spy on. You can choose an application from this list so as to make it appear in the **Application Name** editing filed.

Add

If there are no applications listed in the **Applications** list box, the **Add** button appears dimmed. Choosing this button adds the selected application to the list of applications to spy on. No verification is made at this point.

Delete

If there are no applications listed in the **Applications** list box or if an application from the list has not been selected, the Delete button appears dimmed. The **Delete button** erases the selected application from the list of applications to be spied on.

OK

It's only after having chosen the **OK** button that WYSI-Help effectively records the lists of applications to bebe spied on. As usual, as long as you haven't chosen the OK button, you can always change your mind and Cancel the current operation. No verification is made at this point.

Applications selected

This field is used for designating the name of one of the active applications. You can either directly type in the name of the application, or choose it from the list of "Active Applications".

Active applications

WYSI-Help presents the list of applications currently running and which can be spied on. You can only choose one of the applications from this list.

OK

This choice is made available only after selecting an application to spy on, otherwise it appears dimmed. Using the **OK** button confirms your choice and the name of the selected application will be displayed in the Application Name field of the previous dialog box.

Defining help events (Define menu)

Overview:

Short help messages and **help topics** represent the two ways in which help is made available to the user. Of course, if you add a help **index**, the user has three ways of getting help.

To make help available, you must either define a key or a mouse action.

Defining help events lets you designate the keyboard or mouse action that will be used to trigger the type of help you want to make available in a particular situation: short help messages, help topics, or Help Index.

Procedure:

1. Choose **Help Events** from the **Define menu** (Alt D, H).

WYSI-Help then opens a dialog box that presents the three types of help events that can be defined.

Note that you don't have to define all three types of help events (steps 2, 3 and 4), but only the type(s) that suit(s) the help situation you're actually creating. Steps 2, 3, and 4 can be carried out in any order.

2. Select the event that will give access to the Help Index. This type of help event is defined by a function key.

3. Select the event that will give access to Help Topics. This type of help event is defined by a function key or by pressing on the **right mouse button (the alternate mouse button.)**

4. Select the event that will give access to Short Help Messages. This type of help event can be triggered by using a key, the right mouse button, or by **focusing** (with the mouse pointer) on an **object** for which a short help message is available.
5. To confirm your selection(s), use the **OK** button.

Note that when defining function keys you can also specify that the "Ctrl" or "Shift" or "Alt" keys be used in conjunction with the function keys you have chosen.

Help Index

By default, WYSI-Help proposes using F1 as the key giving access to the Help Index. You can specify an alternative function key by entering any of the other function keys (F1 to F12). You can also define using a combination of keys; a function key used along with one or more of the three keys:
"**Ctrl**" or "**Shift**" or "**Alt**". For example, you could define MAJ, ALT and F5 as the keys to be used to get acess to the Help Index.

Access to the Help Index will always be made available by using the indicated function key. It's not possible to define a function key or combination of keys inhibiting access to the Help Index.

Help topic

By default, Topic Help is made available by pressing simutaneously on the F1 and Ctrl keys. As for the Help Index, you can specify an alternate combination of keys. In addition to a key combination, you may also define using the **alternate mouse button**. By alternate mouse button, we simply mean the button that's not used to communicate with Windows. Since Windows 3 manipulations are normally performed with the left mouse button, theoretically you'll be able to define the **right mouse button** as the one giving access to help topics. However, if you (or the user) switch the way in which the two mouse buttons are used, access to help topics will be made with the left mouse button.

Short Help Messages

By default, access to Short Help messages is obtained by using the **key combination** : Alt F1. Here again an alternative key sequence can be defined.

Defining a function key or combination of keys is not necessary when focusing on a object that triggers a short help message.

Being that a short help message is used to give a brief explanation on a menu choice or a particular option, we conceived WYSI-Help as such that when the mouse cursur focuses on (touches) an object, it makes the corresponding help message appear in the application's title bar. So when you choose **Focus Change**, as soon as an object becomes the current one, the associated help message appears in the title bar, even if you use the keyboard to communicate with the application.

Note:

The mouse cursor can be defined as the only means to get access to help items and short help messages. To do so, the "Key" option must be set to ON (an X appears in the check box)..

For those cases where help is requested and neither a help topic nor a short help message has been associated with the object, you can define a **Default Behavior** option that will be taken into account by WYSI-Help (Define menu).

The same function key (or combination of keys) should not be defined for getting access to help items, short help messages and the Help Index. If you try to do so, WYSI-Help will display an error message once you've confirmed your choices with the OK button.

Warning:

We also advise you not to define as help keys those keys which are main function keys in the application for which help is being built. If you do so, WYSI-Help will always give priority to the help sequence, and the user will never be able to use thefunction key as defined in the application being spied on.

Icon

Clicking the icon from the icon bar gives you direct access to the help events dialog box.

See also:

Index

Define menu

This menu contains the functions allowing for the definition and creation of help applications.

Help Events

The Help Events option lets you define the keys or mouse actions that will be used to trigger the various requests for help made by the user **Help index**, **short help messages**, **help topics**).

Applications to spy on

This command lets you indicate the Windows applications for which you want to build help. Note that WYSI-Help doesn't automatically load the applications listed. If the application is not already loaded, it's your responsability as well as the user's to load it manually.

Applications to Load

This command lets you name the applications for which you want to build help. They can then be automatically loaded and started by WYSI-Help.

One of the main differences between **Applications to Spy on** and **Applications to Load** is the fact that with **Applications to spy on** all the instances of an application will benefit from the same type of help. Whereas with **Applications to Load**, the different instances of the same application can be associated with different help sequences.

Default Behavior

This command lets you define the way in which WYSI-Help should handle a help request when the current object (the object that triggered help) has not been captured.

In this situation, WYSI-Help will try to do one of the following:

- associate an automatic **link** name,
- look for a parent **object**,
- branch to the **help index** if a **help topic** or **message** has not been found.

Icon

This choice lets you use the icon of the spied on application. This makes help applications easier to deliver because the application to spy on will be automatically loaded when starting the help application. Using WYSI-Help thus becomes totally transparent to the final user.

Short Help

This command starts the short help message editor.

Topic Help

This command lets you start the **Topic help editor** and thus define help topics. The Topic editor is one WYSI-Help's basic components.

Each of these commands is presented individually in the "Topic Editor Reference Manaual".

Defining short help (Define menu)

Overview:

As its name implies, the **Short Help** command from the **Define menu** lets you define short messages that are displayed in the title bar of the application for which help is being built, and gives acces to the short help text editor.

Procedure

1. Choose **Short Help** from the **Define menu** (Alt   D, S)
2. Enter or choose the link name for which you want to build a help message. You'll have to enter this name if the link hasn't already been defined.

To carry out steps 1 and 2 more quickly, simply double-click a link name from the list of existing links.

3. Enter the help message in the appropriate field.

4. Choose the **Add/Modify** button.

5. Choose the **Done** button to effectively create or modify the help message.

To define other help messages with different links without leaving the short help text editor, simply repeat steps 2, 3 and 4 as necessary.

Link Name

This field lets you enter the link name for which you want to create a help message. You can select an existing link name from the corresponding list.

Short Help Text

This editing field lets you enter a new help message or modify an existing one. The length of the help message is not limited. We do recommend, however, that you limit your help messages to the maximum length of a typical Windows title bar, which is 80 characters.

Select Color

This option lets you choose the background and foreground colors for the help message text that will appear in the title bar. For more information on this feature, refer to the section entitled "**Choosing a color** for Short Help Messages".

Add/Modify

A new help message will be added to the list of existing ones.

Delete

When you choose this option, you delete the link name that's located in the "Link Name" list box as well as the associated help message.

Done

This choice closes the Short Help Text Editor window.

Remark:

It is only when you save the help application itself (options: **Save** or **Save As**... from the **File menu**) that help messages are actually saved on hard disk..

Icon

Clicking the ✎ icon from the icon bar gives you direct acces to the Short Help Text Editor window.

Defining topic help (Define menu)

Overview:

The **Topic Help** command from the **Define menu** gives you access to the topic editor. This editor can also be chosen from the window where you installed WYSI-Help by double-clicking the corresponding icon. All of the commands making up the topic editor are documented in the Topic Editor Reference Manual.

Procedure:

Choose **Topic Help** from the **Define menu** (Alt D, T).

You can also click Topic Help from the list of existing links.

Icon :

Clicking the icon from the icon bar gives you direct access to the Topic Editor.

See also:

- **Help events**
- **Link**
- **Short help messages**
- **Help Topic**

Detail button

The lists of non-linked **objects** and existing **links** are presented in WYSI-Help's main window.

You can obtain several levels of information for each object and link.

There are two ways of doing this:

- Use the list commands from the **Edit menu**:

> **Collapse One Level**
> **Expand One Level**
> **Lock to Forced Level**

- Click the button that's located in front of each object or link.

This button is called the detail button because each time you click it, an additional level of complementary information is displayed. When a higher level of information is displayed, clicking on this button will hide the current level of information and bring you back to the previous one.

Note that it's impossible to use the detail button with the keyboard.

Error messages

When you make a manipulation error using WYSI-Help or when a system malfunction occurs (for example, a problem in opening a file), WYSI-Help opens an error message window.

Most of the time, the message will explain the problem encountered and in some cases will tell you how to correct it. To obtain detailed information on a specific error message, refer to the complete list of error messages located in the appendix of the "Context Editor Reference Manual".

See also:

List of Error Messages

Expand one level

Overview

Two kinds of lists are presented in WYSI-Help's window: the list of **non-linked objects** and the list of existing **links**.

These lists can contain more or less information according to the level of details you wish to have displayed on your screen at a given time.

WYSI-Help provides for four different levels of details. These levels are numbered from 0 to 3.

Expanding a level results in increasing the amount of information presented in each list. The current level is displayed in the icon bar and can be identified by a number varying from 0 to 3.

Procedure:

Choose **Expand One Level** from the **Edit menu** (Alt E, E).

By default, when you choose this option, expanding one level takes place for both the non-linked objects and the existing links. However, you may just want to obtain more or less details on a specific link or object.

If this is the case, simply click the **detail button** located in front of the corresponding link or object.

The Expand One Level option has the opposite effect of the **Collapse One Level** option.

Files

As for all programs, WYSI-Help's various components make use of files. With WYSI-Help, all the help files making up a particular application are given the same name. This is the name you indicate when you save or open an application. Since an application is made up of several files, the extension names vary.

Since you can copy all the files having the same name with a single command, saving a help application onto diskette becomes a simple matter.

Example: **COPY MYHELP.* A:**

This command copies all the files making up the help application named MYHELP onto drive A: .

File.SRV

This file represents the main help file. It contains all the links.

File.STR

The .STR file (for STRing) contains all the link names. This file is written using the ANSI character set which should make translating your applications easier. In fact, you can use any standard ANSI text editor to translate the link names (NOTEPAD, for example). However, you should NOT modify the file structure in any way (add or delete lines or change the existing line order).

File.TXT

.TXT files contain the text making up help messages. These files also use the ANSI file format.

FILE.DHF

This file contains the help source code, and is NOT written using the ANSI format.

FILE.DTF

The .DTF file is an index for the .DHF file. These two files should never be separated!

.RTF and .BMP Files

.RTF (Rich Text Format) and .BMP(Bit MaP) files represent Windows 3 help files developed with Microsoft's Software development Kit.

.RTF files can be produced using any standard word processor (Word for Windows, WordPerfect, ...).

Converting .RTF files to WYSI-Help files will be made possible in a future version of WYSI-Help.

File.CTX

The .CTX file is generated by WYSI-Help's topic editor. This file contains the name values for the defined links.

File.HLP

The .HLP file is used by Windows 3's standard help interface, the help engine, to ensure the compatibility between a WYSI-Help help application and Window's 3 help.

Remark:

You should ALWAYS keep your help files at the same level and never mix different file versions of a particular help application.

See also:

Converting files

Function keys (help keys)

There are three different ways of obtaining help when a user presses on a key (or a combination of keys) that has been defined with WYSI-Help:

According to the request made, the user will obtain:
- a **help message**,
- a **help topic**,
- the **help index**.

In these three cases, keys or combination of keys are used to obtain the kind of help that you defined. You cannot inhibit the use of keys for getting access to the Help Index. You can, on the other hand, define the mouse as the exclusive means for getting access to help messages and help topics.

To define the key or combinations of keys that will give the user access to help, choose the **Help Events** option from the **Define menu**.

By default, WYSI-Help proposes the following keys (or key combinations):

- **F1** for the Help Index,

- **Alt** and **F1** for help messages,

- **Ctrl** and **F1** for help topics.

See also:
Defining help events

File menu

This menu contains all the commands needed to manage the various files making up a help application; **files** handled by the Context Help Editor, as well as files handled by the Topic Help Editor, when loaded.

New

creates a new help application. As long as the new application has not been saved, WYSI-Help assigns it the name UNTITLED.SRV. This name is also displayed in WYSI-Help's title bar.

Open

opens an existing help application allowing you to modifiy, consult or print it.

Save

saves the application using its current name. If the application has not been named (UNTITLED.SRV), WYSI-Help will automatically select by default the Save As... choice from this same menu .

Save As...

opens a dialog box that allows you to change the filename as well as the drive and pathname for the current help application. The corresponding .SRV , .TXT, and .STR files are also put up to date.

Print

Prints the **object** and **link** lists for the current application. The level of details concerning the information to be printed should be selected before choosing this command.

Select Printer

displays the list of printer driver programs available for each type of printer. This allows you to choose the printer corresponding to your configuration.

Exit

lets you quit WYSI-Help without quitting Windows. This command is the same as the **Close** command located in WYSI-Help's System **Control menu**.

General help

General help can be defined in opposition to **context sensitive help**. When using general help, **the user** must search by himself for the help that interests him. This type of help is usually obtained by choosing a topic from an index or table of contents.

It is important to note that WYSI-Help will also activate general help automatically when a request for **context sentive help** has been made and the corresponding object has not been defined. This latter way of functioning is particularly interesting (**default behavior**) as much for the **author** as for the **user**. It corresponds to an approach which is much more practical and user-friendly than that normally used in Windows applications.

See also:

Author
Context sensitive help
Default behavior
Index
Object
Help Topic

Group

A group corresponds to a set of **objects** (from 1 to n) that have been associated with the same help topic.

Just as a **link** associates an object to a particular help topic, WYSI-Help includes a mechanism that provides for using the same **help topic** without having to duplicate it for each help topic that makes reference to it.

This is obtained by associating a group of objects with the same **link** (as if it were a single object).
The link will then give access to the same **help topic** from the different **objects** contained in the group.

Surveillance motor (HMRUN.EXE)

Overview:

WYSI-Help's help motor is the component that makes help applications run once they are created.

You'll have to deliver this file (HMRUN.EXE) as well as the different files making up your help application to the final users.

What follows is a complete discussion concerning the help motor.

Procedure:

We assume that your help application has been developed and is running correctly.

1. Copy the desired icons into WYSI-Help's help driver file (HMRUN.EXE) by choosing **Icon** from the **Define** menu.

You have two ways of starting your help application. For the purposes of this discussion, we'll suppose that your help application is contained in a file named A(.SRV).

a. Start HMRUN with a parameter:

```
HMRUN A
```

b. Rename HMRUN so that it uses the same name as your help application. You can then start it directly:

```
REN HMRUN.EXE A.EXE

A
```

The second approach offers an advantage to the final user in that there is virtually no difference between the original application and the help application; the help motor uses the same name and is thus transparent to the user.

Remark:

- If you decide to rename the HMRUN.EXE file, you should recopy the application icons into it BEFORE renaming it.

- You can also pass parameters to the original program by respecting the following rules:

The parameter(s) must be typed in after the application's filename in the list of applications to be loaded. There can only be one application in this list (there can, however, be other applications in the list of applications to spy on).

If these conditions are met, all the parameters following the application's name will be communicated to the application.

The programs to be loaded by WYSI-Help should be located in the same directory as WYSI-Help, or in a directory specified in the MS-DOS PATH command. None of the directories that you may have defined for your working environment while developing help applications will be taken into account.

Files to deliver

Assuming that the application's name is A, you would provide the user with the following files:

HMRUN.EXE
A.SRV
A.STR
A.TXT

and if the help application contains help topics:

A.CTX
A.HLP

We also recommend you to include the HMAKER.MSG file that contains the list of complete **error messages**. If this file is not available, the error messages displayed on the screen while executing the help application will only contain one word and not the complete message.

Since a help application shouldn't contain any errors, providing the user with this file is left to your discretion.

WYSI-Help

The WYSI-Help System allows for the development and delivery of   help applications for programs running under Windows 3. The WYSI-Help System is made up of the following set of software programs:

- The **context editor**. This is the very heart of the WYSI-Help System. It allows you to **capture objects** and associate them with **links**. It also includes a **short help message editor**.

- The **topic editor**. This editor allows you to create help topics and **hypertext links**. Help topics can be displayed in either a **help window** or a **pop-up window**.

- The **surveillance motor** interprets help applications. It's the program (HMRUN.EXE) that's used to run help applications that spy on one or several Windows applications.

Help menu

Of course, WYSI-Help provides you with a help menu.

And WYSI-Help's help was built with... WYSI-Help!

Help topics

Help topics are usually more informative than **short help messages** in that they provide the user with a complete explanation on a particular object or procedure.

A help topic is always displayed in a **help window** and usually corresponds to an explicit request made by the user (a choice made from the help menu or by pressing on a pre-defined help key) .

Usually, help topics are organized in relation to other help topics in such a way that certain topics may give access to others.

Help window

When a request for help is made, the corresponding **help topic** is displayed in a help window. This is simply a window that has been standardized by the Windows 3 interface.
You can browse through help by using the Help menu which in itself offers various ways of getting around in help.

You can also choose words or expressions that are written in **green**; these words indicate **hypertext links**.

Designating other words or expressions will open up **pop-up** windows.

Hypertext link

This mechanism allows for linking **help topics** together in such a way that accessing one can give you access to another.

Using the example of a dictionary will help you to better understand this concept.

When defining a word, a dictionary may make reference to other words. So, in order to understand what one word means, you may have to look up the definition of several words.

In the same way, a **help topic** may refer to other help topics. If this is the case, the word(s) or expressions will be written in **green and underlined**. Clicking on them with the mouse will bring you directly to the corresponding help topic.

The Back feature (**help menu window**) takes you back through the chain of hypertext links that you followed until you reach the beginning.

Icon (Define menu)

Overview:

One of the main objectives you'll want to achieve in developing your help application is that using it be as simple as possible for the final user.

With WYSI-Help you can make the help application you've developed completely transparent to the user.

This is accomplished by embedding the help you've developed in the spied-on application and in copying the application's icon into WYSI-Help's "motor" program (HMRUN.EXE). Double-clicking the application's icon will first start the help application and then the actual application for which help was built.
The help application is actually running above the application itself, thus making help completely transparent to the user.

Let's take a closer look at what actually happens by studying the following example:

Suppose you've built a help application for Lotus 1-2-3. As usual, the user locates the Lotus icon and double-clicks it to start it. In this instance, however, the user has actually started the **Surveillance motor**, which in turn starts Lotus 1-2-3. The user didn't have to perform any extra steps to start Lotus 1-2-3, and probably isn't even aware that he (she) also started a help application!

Of course, if you want the user to to be able to modify the existing help mechansim, he'll have to be informed.

Procedure:

The following discussion presents the procedure to use when you want to embed the help application you've developed in the actual application itself. This corresponds to the creation of a help application that runs above the application named A (A.). The name, A.EXE, is only used for the purposes of this discussion, and should be replaced by the actual name of the application for which help is being built. For simplicity, we'll assume that the A application can be located in the current directory of your hard disk. We'll also include the steps needed to build a help application (steps 2 to 5).

1. Copy A.EXE using another name (A2.EXE, for example). To do this, you can either use the Windows File Manager option or the MS-DOS COPY command.

2. Start WYSI-Help.

3. Choose **New** from the **File menu**.

4. Choose **Save As...** from the **File** menu, and name your help application. Take heed to use the same name as the one used for the original application (for example, A.SRV).

5. Choose **Applications to Load** from the **Define menu**, and add A2.EXE to the list.

6. **Capture your help objects**, associate them with **links**, and define the corresponding help messages and help topics.

7. Choose **Icon** from the **Define menu**. This opens the Select Icon dialog box.

8. Specify looking for icons in *.EXE files and not in *.ICO files ("Filename" field).

9. Locate the A2.EXE file by browsing through the different drives and directories (Directories list box).

10. Choose the **Copy** button. This copies the original application's icon into the WYSI-Help motor program (HMRUN.EXE which is located in the WYSI-Help directory).

11. To finish, choose the **OK** button.

12. Quit WYSI-Help in choosing to save your files.

13. Using either Windows File Manager or MS-DOS, copy HMRUN.EXE into A.EXE . This will copy over the existing A.EXE file which was copied into the A2.EXE file (step 1 of this procedure).

14. Install your new program to Windows Program Manager by using "New" from the "File" menu.

15. To start your application, simply double-click the new program's icon.

16. Your help application is now made up of the following files: A.*   (SRV, TXT, HLP, CTX, STR, EXE and the A2.EXE file). For the final user, A.EXE represents the same file as A2.EXE, which is, in fact, WYSI-Help's motor program.

Icon bar

Overview

WYSI-Help's icon bar lets you perform menu selections without having to select them from the corresponding menu. This makes repetitive or frequent tasks much easier to handle. The icon bar is located just below WYSI-Help's menu bar and is displayed asfollows.

Each icon represents a particular function:

The first icon allows for managing **non-linked objects** and existing **links**.

This icon corresponds to the **Open** choice from the **File menu** and can be used to open a help application.

This icon corresponds to the **Save** choice from the **File menu** and lets you save your work on hard disk.

This icon corresponds to the **New** choice from the **File menu**. It lets you cancel your current work and start a new help application.

This icon corresponds to the Short Help choice from the **Define menu** and allows for starting the **short help text editor**.

This icon corresponds to the **Topic Help** choice from the **Define menu**. It lets you start the **topic editor.**

This icon corresponds to the **Link** choice from the **Edit menu**. It lets you group objects together and associate them with a single link.

This icon lets you switch to **test mode** and thus verify the help sequence currently being built.

This icon has the opposite effect of the preceding one. It interrupts test mode. In this example, it appears in non-test mode (non-highlighted).

This icon corresponds to the **Help events** item from the **Define menu**.

Choosing this icon will let you get access to WYSI-Help's help index.

To select one of the icon functions, you simply click it.

Remark:

The icon's bar presentation is pre-determined and can not be changed. In this way, you'll be able to identify the available options more quickly and thus increase your working productivity.

Implicit link

When you omit **capturing an object** and thus associating a **link** with it, WYSI-Help will assist you. Note that **Context sensitive help** can exit for the object without it being linked.

In fact, if you wish to do so, WYSI-Help will exploit the information provided for by Windows to find an **implicit link**. For example, an implicit link could be a menu label.

This simply means that if a request for help is made on a non-linked object and that an implicit link exists, the user will obtain help even though you did not explicitly define an associated link with it. This powerful feature is just one of the options you can choose using WYSI-Help's **Default Behavior** command (**Define menu**).

While capturing objects, the **control window** lets you know which objects have implicit links (by a **blue** control window).

Index

The index corresponds to the table of contents listing **help topics**. In general, each help topic will be listed in the help index and getting acces to them will also be obtained from the index.

However, nothing prevents you from structuring your different help topics into a hierarchical organization. In this case, only those help topics having the highest rank or level will be listed in the index. Access to the others will be made from the "mother"   topic according to the concept of **hypertext links**.

Lock to forced level (Edit menu)

Overview:

Two kinds of lists are presented in WYSI-Help's window: the list of non-linked objects and the list of existing links.

These lists can contain more or less information according to the level of details you wish to have displayed on your screen at a given time.

WYSI-Help provides for four different levels of details. These levels are numbered from 0 to 3.
If you so desire, you can, at any time, change the current level of information so as to display more or less details on the complete list of objects and links. You can also click the **detail button** located in front of each object and link so as to increase or reduce the level of   information displayed for an individual object or link, without affecting the overall level.

In fact, the lists have two representations: a representation in which all of the objects or all of the links are presented with the same level of details, and a "free" representation that displays for each object and each link the level of details that you have specifically defined using the detail button.

The main interest in free mode is that it lets you accurately define what you see on the screen and what will be printed.

Using the command **Lock to force level** lets you define a reference level for the current level
Once you've defined the reference level, using the **Force Detail Level** command lets you switch back and forth between the reference level and the current "free" level.

To illustrate this point, let's suppose that you do the following:

You define level 1 as the reference level. You then expand the overall   level to 2 and finish by clicking on the detail button located in front of the "Cut" link, so as to set it to level 3.

Using the "Force Detail Level" command will then let you return to the reference level 1 or fixed level of this example.

Note that when a checkmark appears in front of the **Force Detail Level** command (**Edit** menu) you know that the reference level (fixed level) is the one currently displayed, otherwise you're in "free" level.

Icon

The icon from the icon bar gives you direct access to the two functions allowing you to define the reference level (fixed level) and toggle between displaying the "fixed" and "free" level lists.

The lower left square of this icon gives direct access to the "Forced Detail Level" command, and is also used to display the number corresponding to the current level of   information (0 to3).

Launching applications manually (Surveillance menu)

Overview:

By selecting **Automatic Launch** **(Preferences** command from the **Edit menu)**, you can specify that WYSI-Help automatically load all of the applications that have been defined as applications to load.

If you've disabled the "Automatic Launch" option , you will have to explicitly request that WYSI-Help start the applications to load.

Procedure:

1. Select one or several applications to load by choosing **Applications to Load** from the **Define menu**.

2. Choose **Launch Application(s)** from the **Surveillance menu** (Alt S, L).

The applications are launched automatically.

Remark:

- This command is particularly useful for starting applications that take a long time to load.

- Even if you've set the Automatic Launch option to the ON position, using this command can prove to be useful after quitting an application that WYSI-Help started automatically. In fact, in this instance, manual loading is the only sure way of knowing that the application has been reloaded and that WYSI-Help is spying on it.

- The Launch application(s) command can be used repeatedly. For example, you indicate that application A should be loaded. Then, you choose Application(s) to Launch . You add application B to the list of applications to load, and request, once again, that the applications be loaded. In this instance, application A, that was previously loaded, will not be reloaded.

- The applications that you defined to be spied on (and not loaded) will not be started by this command.

Link

A link is the keyword that you'll use to identify a **help topic** or **short help message** in a unique way. Each help topic should have a link name associated with it.

Link names can contain any uppercase or lowercase characters (in fact, uppercase and lowercase characters are equivalent). You may also use accents, the underline character '_', the period '.', and the space character ' '.

A link name can be any expression made up of one or several words. In this case, a space and the underline character are equivalent.

The link name is used to establish hypertext links. It's from the link name that you will be able to indicate which topic ought to be accessed by the portion of text written in the hypertext color (**green**) in the current topic.

These same links will be used to get direct access to any topic. As it name implies, the "Search" command from the **help window** will let the user search through the complete list of existing links. Choosing one of the links lets the **user** get direct acces to the corresponding help topic.

Link properties (Edit menu)

Overview:

This command lets you specify the type of help that will be associated with a **link** : **short help message** or **help topic**. In fact, a link can only be associated with either a short help message or a help topic.

Note that a link that isn't associated with a help message or help topic will only hang the associated help event.

Procedure:

1. Select a link from the existing links list.

2. Choose **Properties** from the **Edit menu** (Alt, E, P)

To get direct access to the "Properties" command, you can also double-click a link name appearing in the "Existing Links" list (equivalent of steps 1 and 2).

3. Select the box corresponding to the type(s) of help you want to associate with the link: help message and/or help topic. Clicking a box that already contains an X will cancel a choice (the X disappears).

4. You can then choose to directly edit the corresponding help text(s) by selecting **Edit Short Help** or **Edit Topic Help**.

5. To finish, choose the **OK** button.

Link Name

lets you change the link name.

Short Help

lets you specify that a short help message will be associated with the link (an X appears in the corresponding check box).

Topic Help

lets you specify that a help topic will be associated with the link (an X appears in the corresponding check box).

Edit Short Help

gives you direct access to the Short Help Text Editor.

Edit Topic Help

gives you direct access to the Topic Help editor.

See also:

- **Defining short help**
- **Defining topic help**
- **Help topics**
- **Link**
- **Short help messages**

Testing a help application

Overview:

A **help application** is a computer program. Even though a help application is a simple program, as for all programs, it should be tested with care.

This section discusses the different methods included in the WYSI-Help System to switch to test mode while developing your help application.

Procedure:

You have three different ways to switch to test mode:

- choose **Test Mode** from the **Surveillance** menu (Alt S, T),

- click the TEST icon in the **icon bar**,

- click the left side of WYSI-Help's **control window**.

For each of the methods allowing you to switch to test mode, there exists a corresponding method to stop it:

- choose **Test Mode** once again from the **Surveillance** menu (Alt S, T),

- click the "end of test" icon (the word "TEST is crossed-out) located directly to the right of the TEST icon in the icon bar .

- click on the right side of the control window.

Remark:

To test your help application, you'll have to put yourself in the position of your future help users and then verify that they'll obtain all of the help you've defined .

Verifying your help application will thus consist in browsing through the entire spied-on application and for each **objec**t (whether it be associated with a **link** or not), you'll make a request for help according to how you've defined access to it (**help events**).

As you can imagine, it isn't an easy job and you'll have to carry it out methodically and with much determination so as to track down any flaws that may exist.

We would advise you as follows:

- Adopt a systematic approach by periodically testing your application as you develop.

- When you retest, test not only the new modifications but the entire application. This includes what you previously tested!

You may think that this laborious, but in fact, with experience you'll see that it's the only feasible method that exists.

Any computer programmer will tell you that modifying a single point in a program will often times generate modifications that weren't foreseen.

When you switch to test mode and if you've selected the **Save Before Testing** option from the **Preferences** menu, WYSI-Help will save all the modifications that you may have made since last saving your file. According to the size of your help application, this can be a lengthy operation as the saving process also includes saving your topic files (.DHF and .DTF). So don't worry!

Icons

The icon from the icon bar gives you direct access to testing your help application.

The icon from the icon bar stops test mode. This icon is only available when you're in test mode.

Menus

WYSI-Help is a menu-driven program and its main editing window contains five menus (six if you include the **system control menu**).

What follows is a presentation of WYSI-Help's five menus :

The **File menu** allows for managing the different files making up a help application: saving it, opening it, printing it...

The **Edit menu** lets you manage the **object** and **link** lists that appear just below the menu bar.

The **Define menu** includes the functions allowing for the creation and definition of the various elements making up a help application.

The **Surveillance menu** lets you try out your help application so as to test that it's properly functioning.

The **Help menu** provides you with the information needed to work with WYSI-Help.

Choosing an option from a menu is carried out like in any other Windows application; by using the mouse or the keyboard.

WYSI-Help's **icon bar** gives you direct acces to some of the menu commands, without having to choose them from the corresponding menu. Selecting an icon from the icon bar can only be carried out with the mouse.

New help application

Overview:

From one work session to the next, WYSI-Help memorizes the name of the last help application you were working on when you quit WYSI-Help so that when you begin a new work session, it will automatically load the application you were working on when youquit. This only applies, of course, to applications that haven't been deleted from your hard disk.

When you want to create a new help application, you'll have to explicitly make the request.

Procedure:

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

The objects and links that were previously listed disappear from WYSI-Help's window. In other words, WYSI-Help's window is emptied.

As long as you haven't saved your new application, WYSI-Help will consider it as being "**untitled**", and it will appear as such in WYSI-Help's title bar .

When you choose the **New** command, if the current help application hasn't been put up to date and saved on hard disk, WYSI-Help will automatically prompt you to do so.

WYSI-Help also stops the applications that you expressly designated to load (Applications to load command).

Icon

gives you direct access to this command.

See also:

Save
Save As
Opening help applications
Files

Non-linked objects

A non-linked object is an object that has been captured, but has not been associated with a **link**.   As long as they're not associated with a link or help topic, non-linked objects serve no purpose.

Non-linked objects are grouped together in the Non Linked Objects list in WYSI-Help's window. According to how you've set the "Immediate definition" option (Preference command), you'll be prompted (or not) to define the **link** as soon as the object has been captured.

See also
:

Capturing objects
Creating help
Link properties
Preferences

Object

A type of help that you will be able to build with WYSI-Help is **Context sensitive help**. This means that the type of help obtained by the **user** will depend on what he or she was actually doing when help was requested.

The element used by WYSI-Help to determine the context that immediately preceded the user's request for help is called an object, and corresponds to a menu, dialog box, button, etc...
With WYSI-Help, each object that triggers help will have to be identifed in a unique manner. To do so, you'll use a link which is simply the connection existing between an object and a **help topic**. In sum, a **link** must be used to associate an object with a **help topic**.

Object properties (Edit menu)

Overview

WYSI-Help lets you change the way in which help can be obtained for a particular object.

For example, you could designate that help not be made available by a help event (key press, combination of keys or mouse action), but by the normal selection of an object. In this way, you can have help take the place of   the action provided for in the application. For example, you will use this method to replace old help menus in an application.

Only using this feature will make you understand what a powerful mechanism it is.

Procedure:

1. **Select** an object from the list.

2. Choose **Properties** from the **Edit menu**.

Note that if you've selected several objects, or if you've selected an object and a link, you will not be able to get access to this feature.

Steps 1 and 2 are equivalent to double-clicking on the object.

3. Define if you wish to maintain getting access to help on defined **help events**.

4. If desired, you can specify that **selecting the object** will let the user obtain help. To do so, choose the corresponding check box.

5. Choose **Ok** to complete this operation.

Title

The title for the object as defined by Windows is displayed.

Class

The type of object as defined by Windows (button, menu choice, etc.) is also displayed.

Help on Defined Events

If you leave an X in this check box, access to help will be made available as defined in the Help Events dialog box.

If   you clear this choice, all of the defined keys, combination of keys or mouse actions that normally give access to help will be inhibited when the associated object is the application's current one.

Help on Selection

By selecting this check box, you modify the way in which the spied-on application normally functions by triggering help when the object is selected, rather than giving access to the function that is normally provided for in the spied-on application. This feature is particularly useful for modifying old help topics from a help menu.

See also:

Control window
Defining help events

Printing

Overview

Since it's much easier to read a listing of objects and links than to scroll through them on the screen, WYSI-Help's printing facilities will make developing your help applications easier to do.
This is particularly true if your application contains numerous **objects** and **links**.

Procedure:

Choose **Print** from the **File menu** (Alt F, P).

Printing is carried out as for any Windows application.

The listing that you obtain will contain the level of information that's currently displayed in WYSI-Help's window for the non-linked objects list and the existing links list.

See also:

Choosing a printer

Pop up help

Pop-up help refers to topic help information that is made available in pop-up windows.

The term pop-up is used here because this type of window only stays open as long as the user holds down the mouse button.

This type of help is used to display the lowest level of   help information.

Access from one pop-up topic to another isn't possible.

See also:
Help window

Preferences

Overview

Choosing this option lets you customize WYSI-Help's editing functions to suit your particular needs. You can choose a different setting for each application being developed. These settings concern:

- **hiding the** **control window** (Hide Live Info),
- immediately defining the help associated with each **link** or **object**,
- minimizing WYSI-Help's main window when testing a help application,
- automatic saving of the help application before testing,
- hiding the **icon bar**,
- automatic loading of applications to be loaded (when defined and when loading a WYSI-Help application).

Procedure:

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

2. Checkmark (or clear) the options corresponding to your preferences/needs.

3. To confirm your settings, choose the **OK** button.

Hide Live Info

By default this setting is not active. The display window is visible.

Immediate Definition

This setting allows for immediately defining the **link** name, **short help message**, and the **help topic** as you capture **objects**. Each time you capture an object,a dialog box appears that allows for defining the associated help. By default this setting is active.

Minimize Main Window

This allows for shrinking WYSI-Help's window into an icon as soon as one of the applications to be spied on is loaded or when in test mode. By default, this setting is inactive.

Save BeforeTesting

Testing an application usually doesn't cause any particular problem, but it's always good practice to take precautionary measures so as to avoid losing long hours of development.

By default, this setting is active.

Hide Icon Bar

The **icon bar** lets you carry out certain menu command more quickly, and thus increases your productivity. It may, however, at times hinder the overall view you have concerning the list of objects and links. When this is the case, you can activate this setting so as to obtain the greatest amount of available screen space. If you don't use a mouse, you will also probably want to activate this option.

By default, the icon bar is visible.

Automatic Launch

When you start WYSI-Help, it automatically loads the last help application you were working on, as well as the applications that you designated. If this setting (Automatic Launch) has been activated, this is also the case when you add another application to load. By default, this option is active.

From one session to the next, WYSI-Help saves your current settings.These setting apply to all help applications, and not to specific ones .

Quitting WYSI-Help (File Menu)

Overview:

When you quit WYSI-Help, it checks for any modifications that you may have performed during the current session, and prompts you to save the file if any are found (if so desired, you can also choose not to save the file). It also lets you cancel your request to leave.

Procedures:

Choose **Quit** from the **File menu** (Alt   F, Q)

or

Choose **Close** from WYSI-Help's **System** Control-menu (Alt   Spacebar, C)

Leaving Windows automatically makes you leave WYSI-Help.

All of the following commands let you quit WYSI-Help and display a message if the application has been changed since last saving it:

- Close from Windows' System Control-menu

- File, Quit

- Close from WYSI-Help's System Control-menu

Save (File menu)

As its name implies, this command saves all the files (.SRV, .TXT, .STR, .DTF, .DHF) making up the help application you're currently working on using the name you assigned it.

The last two types of files (.DTF et .DHF) are only saved if the topic help editor is loaded.

If you haven't assigned a name to a new application (**New** command from the **File menu**), and you choose the **Save** command to save it, by default, the **Save as**... command will be automatically chosen for you.

Icon

The icon from the icon bar gives you direct access to file saving functions without having to go by the File menu. Here again, if you haven't assigned a name to a new file, the "Save As..." command will be used instead of the "Save" command.

Save as... (File menu)

Overview

Help applications that you create will have to be saved on hard disk for future use.

Remember that as long as your application hasn't been explicitly saved, what you're currently working on isn't saved either. Furthermore, if a system malfunction occurs or there's a power failure, all of the modifications that you've made since you last saved your application may be lost.

The **Save As**... command from the **File** menu will let you assign a name to a new file, or duplicate the current application by giving it another name.

Procedure:

1. Choose **Save As**... from the **File menu** (Alt F, A).

2. Enter the name you want to assign to the help application without specifying a filename extension (WYSI-Help will automatically add the .SRV extension for you). If the application isn't to be saved in the current directory, you must also include acomplete pathname, before the application name.

3. To start the saving process, choose **OK**.

If you want to change the directory in which the application will be saved, you can select the one to use by looking through the ones listed in the "Directories" list box.

Filename

lets you enter the name you want to assign to the help application. If the name you enter corresponds to an application that already exists in the current directory, a message will be displayed after you've confirmed your choice with the OK button.

The filename extension can be omitted. If you want to include it, you should enter the filename extension: .SRV. However, if you enter an extension other than .SRV (.ANY for example), WYSI-Help will ignore the extension you've specified and replace itby .SRV. WYSI-Help will also create and save the other files making up the help application (.TXT, .STR as well as the .HLP, .DHF and .DTF files, if the help topic editor is loaded).

Directories

lets you select the drive and directory where you want to save your help application.

Surveillance menu

The Surveillance menu contains two menu selections:

Test Mode

is used to start or stop a test on the current application. This lets you try out your application as you're developping it and thus control its behavior (error-free and running as you would like it.

Launch Application(s)

starts all the applications that you defined to load (Applications to Load from the **Define menu**), even if these applications have been stopped.

System menu (Control-menu)

The system menu lets you manage WYSI-Help's window.

This menu can be opened by clicking on the square containing a "dash" in the upper left corner of WYSI-Help's window, or by pressing on ALT SPACEBAR.

The system menu includes the following commands:

Restore

restores WYSI-Help's window to its previous size.

Move

lets you move WYSI-Help's window to another position on the screen.

Size

lets you change the size of WYSI-Help's window.

Minimize

shrinks WYSI-Help's window to an icon.

Maximize

enlarges WYSI-Help's window so that it takes up the entire screen (maximum size). This will be useful when you'll need to get a better view of the objects and links contained in the two lists.

Close

stops WYSI-Help.

Switch To...

lets you switch to another running application or return to Program Manager's window.

Since the system menu is a standard Window feature and often used in Windows applications, its use will not be discussed any further here. To obtain detailed information concerning the system menu, you can refer to the Windows User's Guide.

Short help messages

Short help messages appear in the title bar of the help application and are used to provide the user with a brief comment or explanation on a menu choice, command or option button, etc...

In general, a short help message is provided without the user having to make an explicit request for help.

See also:

Defining short help

Selecting a link

Overview:

This command lets you select a **link** to delete by highlighting it in the list of existing links.

Procedure:

1. Use the the **detail button** to make the existing links appear on the screen.

2. Highlight the link to delete by clicking it. Its name appears in inverse video (clicking it once again will bring it back to normal).

You can only select one link at a time.

Remark:

If you double-click a **link** instead of clicking it, you'll get access to the option allowing you to define its properties.

Selecting an object

Overview:

This command lets you select (highlight) one or several objects in the list of **non-linked objects** so as to carry out the different procedures allowing you to delete an object, group together several objects, etc...

Procedure:

1. Use the the detail button associated with the non-linked objects or the **detail button** of a **link**, so as to make the corresponding objects appear on the screen.

2. Click the object to select (highlight). Its name appears in reverse video (clicking it once again will bring it back to normal).

Remark:

If you double-click an object instead of clicking it, you'll get access to the option allowing you to define its properties.

Link object(s) (Edit menu)

Overview:

Normally, objects that are **captured** should be associated with **links**.

You can attach an object to the corresponding **help topic** by designating a link, or by grouping together several objects under the same link name.

Procedure:

1. **Select** one or several objects from the list.
2 Choose **Link** from the **Edit menu** (Alt E, L).
3. Enter the destination link name in the corresponding text box.
4. You can then choose to directly edit the corresponding help text(s) by selecting **Edit Short Help** or **Edit Topic Help**.
5. Complete the operation by choosing the **OK** button.

Destination Link Name

lets you change the link name. By default, WYSI-Help proposes using the name of the selected object name. When linking several objects, WYSI-Help displays the last object name of the objects you selected in the list (step 1).

Existing Links

In this zone, WYSI-Help displays the lists of links that already exist. You can also select one of the links from this list as the destination link name. In this case, the selected objects will be added to the list of objects already associated with the link.

Edit short Help

gives you direct acces to the Short Help Text Editor.

Edit Topic Help

gives you direct access to the Topic Help text editor.

Icon

The icon gives direct acces to the **Link** function.

See also:

Selecting objects
Short help messages

Opening help applications

Overview:

Once created, a help application is saved on hard disk for future use: modification, execution, or printing. If you want to perform any of these operations on a particular application, you'll have to open it.

Procedure:

1. Choose **Open** from the **File menu** (Alt F, O).

2. Type or select the name of the help application to open. If the application is not located in the current directory, you'll have to enter the exact pathname.

3. To confirm your selection, choose the **OK** button.

If you wish to look for an application located on hard disk, you may do so by using two different list boxes: Files (lists the help files for the current directory) and Directories (lists the drives and directories on hard disk).

To find one of your help applications, search for the files having the extension: .SRV (all of the files making up your application will also be loaded).

File name

lets you enter the name of the file to open. If the name you enter doesn't correspond to an application in the current directory, after you choose the OK button, an appropriate error message is displayed.

You must choose a help application with the extension .SRV. You cannot save or open help applications from files not using this extension.

Files

lets you select a file from a list of files.

Directories

lets you select the drive and directory where the help application is located.

When you choose the directories list box, the .SRV extension is automatically displayed in the "File" textbox and all the files having the .SRV extension are automatically displayed in the "Files" list box. If the file you're looking for isn't located in the designated drive and directory, you can select another directory and all of the help applications existing in that directory will be displayed.

Remark:

When you open a help application and the current application hasn't been saved, WYSI-Help will prompt you to do so before proceeding.

Restoring the topic editor's files is automatic, if already loaded.

Icon

The icon from the icon bar gives you direct access to opening help applications.

See also:

- **Save**
- **Save as**...
- **Files**