

Context-sensitive Help with compiled Help systems

By Carol Johnston. Monday, 07 July 2003

What is context-sensitive Help?

Context-sensitive Help is assistance that is appropriate to where the user is in the software application, and what they are trying to do.

Who is involved in creating it?

Context-sensitive Help requires that the Help author and the programmer join forces.

- The programmer creates a means of calling the Help from each dialog of the application. This could be via a Help button or a right-click menu option.
- The Help author creates appropriate topics within the Help file.

It is strongly recommended that both parties meet up early on in the project to double-check that both are clear on the process and role responsibilities.

Who does what?

To implement context-sensitive Help we need to map Help-related objects on the application interface to appropriate Help topics. This mapping is achieved by way of a **header file**.

There are two ways for the Help author and the programmer to work together to achieve context-sensitive Help:

1. The programmer creates a header file and sends it to the Help author. (This is the more common situation.)
2. The Help author creates a header file and sends it to the programmer.

So what exactly is a header file?

Each object on the application interface has a unique number associated with it (the **context number**). This number is what will be used by the application to find the correct Help topic.

However, it is much easier for humans to deal with descriptive names, rather than numbers. For example, the **context name** CTX_SALES_DIALOG is much easier to understand and recognise than the context number 2384. However, the application and the compiled HTML Help ultimately require a number.

The **header file** is simply a list of all the context names and their corresponding context numbers. It is a simple text file whose layout depends on the programming language being used to create the application.

Language	File extension	Syntax	Example
C	.H	#define <i>string number</i>	#define CTX_SALES_DIALOG 1000
Pascal	.PAS	<i>string</i> = <i>number</i> ;	CTX_SALES_DIALOG = 1000;
Basic	.BAS	<i>string</i> = <i>number</i> or GLOBAL CONST <i>string</i> = <i>number</i>	CTX_SALES_DIALOG = 1000 GLOBAL CONST CTX_SALES_DIALOG = 1000

The header file is not used at runtime. The information it contains is included in the Help system on compilation.

When the programmer creates the header file...

... the **programmer** must:

1. Produce a list of the context names and numbers for all dialogs and fields that require context-sensitive Help– the header file.
2. Send the header file to the author.

... the **author** must then:

1. Import it into the Help project.
2. Set up the context names in the header file to be **aliases** for the corresponding topic names. This literally means giving the topics alternative names.

When the Help author creates the header file...

... the **author** must:

1. Create context name aliases and context numbers for all dialog level context-sensitive topics.
2. Create context numbers for all field level context-sensitive topics.
3. Produce a list of the context names and numbers – the header file.
4. Send the header file to the programmer.

...the **programmer** must then:

1. Import it into the application.
2. Associate all the context names in the header file with the corresponding Help-related objects in the application.

Writing the content

It is important to decide what sort of information to include in the context-sensitive topics.

Dialog level

Dialog level Help usually provides one or more of the following:

- A general description of the dialog.
- A description of the various fields within it.
- Links to topics that provide Help on the tasks relevant to the dialog.
- If the dialog has only one task associated with it, step-by-step instructions on how to complete the task.

Field level

Field level Help usually provides a short topic with information about the selected field. This is usually in a popup window and is known as ‘What’s This?’ Help.

When providing popup field level Help using Microsoft HTML Help

In this situation, the topics must be written in a **text file**. HTML Help does not support the use of graphics and formatting in popups. Even if you are using an authoring tool that has a method for producing formatted popups, this functionality is not applicable in a context-sensitive environment. Write your text-only topics in a text file, using the syntax:

.topic context name

topic text

You can use more than one text file, but all the topics for any particular dialog must reside in the same text file. Since the context names are hard-coded into the text files, there is no need for aliasing. The text file(s) must be included in the compilation of the Help project.

Example

A text-only topics file for the sales dialog in an application might start of like this:

```
.topic CTX_SALES_DIALOG_ADD_BUTTON 1010
Click this button to add a new sales item to the list.

.topic CTX_SALES_DIALOG_ADD_SALENAME 1020
Type in the product name using the correct capitalisation.

.topic CTX_SALES_DIALOG_ADD_SALEDATE 1020
Type in the date that the sale was confirmed in the format DD/MM/YYYY.
```

Help authoring tools and context-sensitive Help

Different authoring tools have different interfaces for dealing with header files and text-only topics.

Importing a programmer's header file

- Some HATs allow an automated import of the programmer's header file.
- Others require you to (laboriously) type in the context names and context numbers by hand for every context-sensitive topic.

Associating the context names in the programmer's header file with the corresponding Help-related objects in the application

- Some HATs have a simple point-and-click interface for setting-up context name aliases for topics – thus keeping the aliasing process hidden behind the scenes.
- Others require you to know about aliases.

Creating a header file for export to the programmer

- Some HATs hide the aliasing process from the author.
- Some HAT's can automate the context numbering.
- Some cannot directly create the header file. Authors must open the fundamental project text file (.HPJ for WinHelp, .HHP for HTML Help) and copy the relevant information into a new file!

Creating text-only topics for HTML Help

Although these can be created in any text editor, some HATs provide a simple interface for writing them so that the text files are automatically added to the Help project and compiled. Some produce the required text files behind the scenes as they compile HTML Help.

Help authoring tools and context-sensitive Help

Problems often arise in communication between the Help author and the programmer due to there being inconsistent terminology for the elements involved. Confusion also arises within the Help community due to different authoring tools using different terminology.

The table below shows some of the names that programmers, Help authors and authoring tools use to refer to the elements of context-sensitive Help.

Term	Synonyms
Header file	<ul style="list-style-type: none"> • map file • mapping file
Context number	<ul style="list-style-type: none"> • context ID • map ID • map number • mapping number • UID (unique ID) • GUID (globally unique ID) • Help ID
Context name	<ul style="list-style-type: none"> • context string • symbolic constant • constant • define • macro • topic ID

About Cherryleaf

Cherryleaf is the trading name of Cherryleaf Limited, a company based in the United Kingdom. We have locations in or near Birmingham, Brighton, Heathrow (London) and Oxford, and we carry out work throughout the UK and Europe.

Cherryleaf helps companies improve their performance by making information easy to access and use. We can help you create, develop, resource, manage and maintain your documentation: online Help for software, corporate policy, performance support or IT Systems documents.

Cherryleaf has a unique blend of information design, writing and teaching skills, as well as technological expertise. We combine these skills to develop documents, information systems and affordable content management solutions that bring people, business processes, content and technology together. We care about documentation and the ways you can turn it into a productive resource