

文章编号: 1000-5692(2001)01-0080-05

# HTML 帮助系统在生物统计分析平台中的应用

吴延熊

(云南省林业科学院 森林培育与开发重点实验室, 云南 昆明 650204)

**摘要:** 论述了基于 HTML 的新型帮助系统的特点, 提出一个可以在生物统计分析平台的开发工具 Delphi 中调用 HTML 帮助系统的单元文件 CallHHelp。该调用单元的核心思想是将 HTML 帮助命令映射到对应的 Win Help 命令上, 从而实现生物统计分析平台对 HTML 帮助系统的正确响应。图 1 参 3

**关键词:** HTML; 帮助系统; 生物统计分析; 调用单元

**中图分类号:** S757.1; Q-0; TP311.52 **文献标识码:** A

## 1 问题的提出

应用软件的帮助系统从 Windows 3X 平台到 Windows 95 平台并没有实质性的变化, 只是从 16 位帮助进化到 32 位帮助, 帮助系统的表现形式和制作方法基本没有改变。这种传统的帮助系统通常称作 Win Help。从 Windows 98 平台开始, 应用软件的帮助系统有了质的飞跃即从传统的 Win Help 向基于 HTML (Hypertext Markup Language, 超文本链接标示语言) 的帮助系统迁移<sup>[1]</sup>。基于 HTML 的帮助系统通常称作 HTML Help。

Windows 98 和 Windows 2000 平台既保留了 Win Help 引擎, 又保留了 HTML Help 引擎, 但是微软基于 Windows 98 和 Windows 2000 平台的最新应用软件都只提供 HTML Help 帮助系统。事实上, 微软从 1997 年就开始陆续向 HTML Help 迁移, 微软的开发工具 Visual Studio 6 就提供了 HTML Help 制作工具, 而将 Win Help 制作工具置于一个可有可无的子目录中。从那时起, 微软不再改进 Win Help, 而是重点发展 HTML Help, 因此 HTML Help 就成了应用软件未来帮助系统的事实标准。定位于商业化通用应用软件的可视化生物统计分析平台即统计大师 (Super SAS), 同时为用户提供了 Win Help 和 HTML Help 两种帮助系统。

## 2 HTML 帮助系统的特点

HTML 帮助系统的特点, 实际上就是 Win Help 固有的局限性。与 Win Help 相比, HTML 帮助系统的特点主要有以下 4 点。

### 2.1 一次编写, 随处运行

HTML Help 帮助系统的最大特点可能就是“一次编写, 随处运行 (write once, use everywhere.)”。

收稿日期: 2000-06-26

基金项目: 云南省自然科学基金资助项目(1999C0025G); 云南省和云南省林业科学院重点实验室学科带头人培养经费联合资助项目(无编号)

作者简介: 吴延熊(1967-), 男, 浙江建德人, 副研究员, 博士, 从事生物统计、可持续经营和预警系统研究。

©1994-2017 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

一旦开发人员编译了某个 HTML Help 帮助文件, 就可以得到一个编译了的应用程序, 因此开发人员可以通过磁盘、因特网或内联网 (Intranet) 来分发 HTML Help 帮助文件。用户并不需要对 HTML Help 帮助文件作任何调整或干预, 就可以在其他平台启动同一应用程序。

Windows 帮助系统由 2 个部分构成: 一是帮助文件, 二是运行器。帮助文件是开发人员为自己的应用程序制作的帮助文档, 运行器则是操作系统的有机组成部分, HTML Help 和 Win Help 都是如此, 但 HTML Help 的运行器已经有了很大的改进。改进后的运行器不仅小巧玲珑, 而且不再直接运行帮助文件, 只是操作一个 Active X 的控件, 从而控制支持 Active X 控件的 Web 浏览器, 比如微软的 Internet Explorer 3.0 以上版本的因特网浏览器。这样处理的好处在于可以实现帮助文件的跨平台运行。只需拥有不同平台的运行器和浏览器, 帮助文件不必重新编译就可以直接运行。

## 2.2 帮助文件的制作简单

HTML Help 帮助文件的制作相对简便。许多开发人员不愿意迁移到基于 HTML 的帮助系统, 表面上看起来有 2 个原因。一是自己已经习惯了 Win Help, 长期以来 Win Help 的制作方法和表现形式都没有什么变化, 制作起来轻车熟路; 二是制作 HTML Help 帮助文件比较困难。事实上, 前一个理由可以成立, 后一个理由却完全错误。Win Help 在很大程度上依赖 RTF (rich text format, 丰富文本格式) 文件, 而 RTF 文件拥有许多令人费解的格式控制字符, 编辑 HTML 文件要比编辑 RTF 文件简单得多。

微软已经宣布将继续支持基于 RTF 的 Win Help, 但不要指望微软会开发新的 Win Help 制作工具。微软继续支持 Win Help 的目的是提供能将 Win Help 帮助文件转换成新帮助文件的 HTML Help 编译和开发工具, 保护开发人员的前期工作。

## 2.3 帮助内容的表现力强

HTML Help 帮助文件不仅要比 Win Help 帮助文件更具鲁棒性 (robustness), 而且还具有更强的帮助内容表现力。HTML Help 帮助文件不再局限于文本和静态图像, 可以插入 GIF 动画, 也可以利用浏览器插件在文档中播放声音文件、动画影片或其他格式的多媒体文件, 使得帮助文档的表现形式更加丰富多彩。另外, HTML Help 帮助文件还可以加入 URL (uniform resource locator) 地址, 所谓 URL 就是在因特网的 WWW 服务程序上用于指定信息位置的表示方法, 这样帮助文档的内容不再局限于包装好的文档内容, 从而可以扩展到更加广泛、更加灵活和更加实时的因特网帮助资源。

## 2.4 信息搜索的功能强大

HTML Help 的信息搜索功能强大, 方式独特。开发人员可以将帮助主题归结为某个给定的类型, 这就如同数学中的集合概念一样。例如, 可以将帮助主题标记为概念 (conceptual)、教程 (tutorial)、步骤 (procedural) 或其他类型。假如开发人员正在为 2 个非常接近的产品撰写帮助文件, 通过使用这些主题标记 (tags), 就可以为 2 个产品撰写一个帮助文件。用户为其中一个产品搜索帮助, 仅显示与它相关的帮助主题。以这种方式切分帮助文件, 使得帮助系统的维护变得非常简单。

不过, HTML Help 也存在一些不尽人意之处。比如每个帮助主题都需要自己的 HTML 文件, 这就使得帮助工程变得有些复杂。HTML Help 支持弹出窗口, 但弹出窗口不如 Win Help 美观, 而且执行起来比较困难等。

# 3 HTML 帮助系统的调用单元

在 Delphi 应用程序中调用 HTML 帮助系统并非易事, 因为现行的 Delphi 开发工具, 并不支持 HTML 帮助系统<sup>[2,3]</sup>。在开发可视化生物统计分析平台时, 我们编写了一个能够正确调用 HTML 帮助系统的单元文件 CallHHelp。

CallHHelp 单元的工作原理是将自己的事件处理函数赋予 Application.OnHelp 事件。这就意味着开发人员不能再将自己的事件处理函数赋予 Application.OnHelp, 否则该单元将不起作用。另外 CallHHelp 单元忽略了任何窗体的 HelpFile 属性。因此 HelpFile 属性在调用 HTML Help 时不再有用。

CallHHelp 单元的使用方法是将它加入应用程序中, 并在任何需要使用 HTML Help 的单元文件的 Uses 语句中引用该单元, 那么任何 Win Help 帮助请求将被自动转换成相应的 HTML Help 调用。

例如, 首先可以在可视化生物统计分析平台的开发工具 Delphi 中, 定义生物统计分析平台的帮助文件“SuperSAS. chm”, 然后通过下述语句即可打开如图 1 所示的 HTML Help 帮助文件对话框。

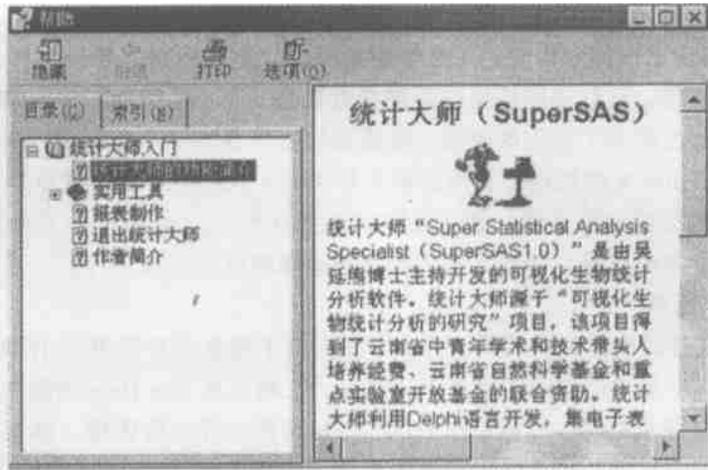


图 1 HTML 帮助对话框

Figure 1 HTML help dialog box

Application. HelpCommand (HELP\_CONTENTS, 0)。

CallHHelp 单元完整的源代码如下:

```
unit CallHHelp;
interface
uses Windows, Messages, SysUtils, Forms;

const
  HH_DISPLAY_TOPIC           = $0000;
  HH_HELP_FINDER             = $0000; //WinHelp 的等价值
  HH_DISPLAY_TOC             = $0001;
  HH_DISPLAY_INDEX           = $0002;
  HH_DISPLAY_SEARCH          = $0003;
  HH_SET_WIN_TYPE            = $0004;
  HH_GET_WIN_TYPE            = $0005;
  HH_GET_WIN_HANDLE          = $0006;
  HH_ENUM_INFO_TYPE          = $0007;
  HH_SET_INFO_TYPE           = $0008;
  HH_SYNC                     = $0009;
  HH_RESERVED1               = $000A;
  HH_RESERVED2               = $000B;
  HH_RESERVED3               = $000C;
  HH_KEYWORD_LOOKUP          = $000D;
  HH_DISPLAY_TEXT_POPUP      = $000E; //弹出窗口显示字符串资源 ID 或文本
  HH_HELP_CONTEXT            = $000F;
  HH_TP_HELP_CONTEXTMENU     = $0010; //与 HELP_CONTEXTMENU 相当
  HH_TP_HELP_WM_HELP         = $0011; //与 HELP_WM_HELP 相当
  HH_CLOSE_ALL                = $0012; //关闭所有直接或间接调用所打开的窗口
```

```

HH_ALINK_LOOKUP           = $0013;
HH_GET_LAST_ERROR        = $0014;
HH_ENUM_CATEGORY         = $0015;
HH_ENUM_CATEGORY_IT      = $0016;
HH_RESET_IT_FILTER       = $0017;
HH_SET_INCLUSIVE_FILTER  = $0018;
HH_SET_EXCLUSIVE_FILTER  = $0019;
HH_INITIALIZE             = $001C; //初始化帮助系统
HH_UNINITIALIZE          = $001D; //非初始化帮助系统
HH_PRETRANSLATEMESSAGE   = $00FD;
HH_SET_GLOBAL_PROPERTY   = $00FC;

```

```

function HtmlHelp (hwndCaller; THandle; pszFile; PChar; uCommand; cardinal; dwData; longint);
THandle; stdcall;

```

implementation

```

function HtmlHelp (hwndCaller: THandle; pszFile: PChar; uCommand: cardinal; dwData: longint):
THandle; stdcall; external 'hhctrl. ocx' name 'HtmlHelpA';

```

type

```

TUseHTMLHelp=class (TObject)
    function ApplicationHelp (Command: Word; Dard; Longint; vat CallHelp; Boolean): Boolean;
end;

```

```

function TUseHTMLHelp. ApplicationHelp (Command: Word; Data: Longint; var CallHelp: Boolean):
Boolean;

```

begin

```

//确保 VCL 不再激活 Win Help

```

```

CallHelp: =False;

```

```

Result: = True;

```

```

if Command in [HELP_CONTEXT, HELP_CONTEXTPOPUP] then

```

```

//上下文跳转

```

```

    HtmlHelp (Application. MainForm. Handle, PChar (Application. HelpFile),

```

```

HH_HELP_CONTEXT, Data)

```

```

else if Command=HELP_KEY then

```

```

//搜索关键词

```

```

    HtmlHelp (Application. MainForm. Handle, PChar (Application. HelpFile),

```

```

HH_DISPLAY_INDEX, Data)

```

```

else if Command=HELP_QUIT then

```

```

//如果应用程序退出, 关闭所有帮助文件

```

```

    HtmlHelp (Application. MainForm. Handle, nil, HH_CLOSE_ALL, 0)

```

```

else

```

```

//否则显示帮助目录

```

```

    HtmlHelp (Application. MainForm. Handle,           PChar (Application. HelpFile),
HH_HELP_FINDER, 0)
end;
var
    HTMLHelpUser: TuseHTMLHelp;

initialization
    HTMLHelpUser := TuseHTMLHelp. Create;           //创建
    Application. OnHelp := HTMLHelpUser. ApplicationHelp;
finalization
    Application. OnHelp := nil;
    HTMLHelpUser. Free;                             //释放
end.

```

#### 4 结论与讨论

应用软件的使用既简单,又复杂。说它简单,因为软件设计者总是企图设计出傻瓜式的软件。“聪明人使用傻瓜”是软件设计者的追求和理念,也是用户的企盼和要求。说它复杂,因为真正傻瓜式的应用软件离我们还很远。软件的傻瓜性究竟体现在何处,我们认为除了人性化的界面之外,最主要的还是完善、美观、方便和智能化的帮助系统,以便用户随时随地能够得到所需的帮助。

当前,国产应用软件经常简化帮助系统,很少提供HTML帮助系统,有的甚至忽视帮助系统的制作。我们认为,这是国产应用软件很少流行的一个重要原因。相比之下,占据全球绝大多数市场份额的办公套件Office,不仅提供了内容非常丰富的HTML帮助系统,而且还提供智能化的Office助手,使得帮助系统向更加智能化的方向发展。其实,通过上述调用单元,Delphi应用程序也可以轻松实现对HTML帮助系统的响应。

#### 参考文献:

- [1] 杨伦,白玉祺,童恒建. 怎样在VC 6.0和VB 6.0中使用新一代帮助系统HTML[J]. 电脑编程技巧与维护, 2000, (1): 67-69.
- [2] 吴延熊,周国模,郭仁鉴. Active X在区域森林资源预警技术系统中的应用[J]. 浙江林学院学报, 1999, 16(1): 100-106.
- [3] 张威,卢庆龄. Delphi4.0编程与技巧[M]. 北京: 科学出版社, 1999. 384-422.

## Application of HTML help system in biological statistic analysis platform

WU Yan-xiong

(Key Laboratory of Forest Cultivation and Development, Yunnan Academy of Forestry, Kunming 650204, Yunnan, China)

**Abstract:** This paper discusses characteristics of the new help system based on HTML, and puts forward unit file "CallHHHelp" which can be used to call HTML help by Delphi that is developing tool of biological statistic analysis platform. The core method of this unit file is to map HTML help commands to corresponding Win help commands. Finally, the biological statistic analysis platform can correctly call HTML help system by the unit file.

**Key words:** HTML; help system; biological statistic analysis; call unit