Contents
Contents of HELPFILE help file

Units:

Unit HelpFiles

Unit HelpFiles (in HELPFILE help file)

Unit to define abstract help file object.   Included in SCANH3xx.ZIP as sample of comment formatting and output.

```
This comment will be highlighted as example text in most output formats. The
sample below won't be word-wrapped.
  program HelloWorld;
  begin
    writeln('Hello, world!');
  end.
```

Constants:

ForHelpBuffer

Types:

PTopic
TTopic
PIndexItem
TIndexItem
PIndex
TIndex
PHelpFile
THelpFile

Variables:

NotifyProc

HelpFiles.ForHelpBuffer

Const HelpFiles.ForHelpBuffer: TStreamRanking = (EMSStream, XMSStream, FileStream, NoStream);

HelpFiles.PTopic

HelpFiles.PTopic = ^TTopic;

HelpFiles.TTopic

HelpFiles.TTopic = object(TObject)

An object holding a single topic as part of a THelpFile.

Fields and Methods:

```
Text: PStream;
TopicNum: Longint;
StartofLine: Boolean;
FixedLines: Boolean;
Marked: Boolean;
Highlighting: byte;
constructor Init...
destructor Done; virtual;
function GetLine...   virtual;
function MoreLines...   virtual;
procedure Write...   virtual;
procedure WriteLn...   virtual;
procedure WriteKeyWord...   virtual;
procedure HighLight...   virtual;
procedure ResetHighLight; virtual;
procedure BlankLine; virtual;
procedure StartXrefList...   virtual;
procedure WriteXref...   virtual;
procedure EndXrefList; virtual;
procedure ToggleFixed; virtual;
procedure ToggleMarked; virtual;
```

TTopic.Text

TTopic.Text: PStream;

A stream to which the text of the topic is written.

TTopic.TopicNum

TTopic.TopicNum: Longint;

The topic number in the help file.

TTopic.StartofLine

TTopic.StartofLine: Boolean;

Whether the text is currently at the start of a line.

TTopic.FixedLines

TTopic.FixedLines: Boolean;

Whether lines should be fixed or wrapped.

TTopic.Marked

TTopic.Marked: Boolean;

Whether text is currently being marked.

TTopic.Highlighting

TTopic.Highlighting: byte;

Counts the current highlight level.

TTopic.Init

constructor TTopic.Init(Atopicnum: Longint);

Initialize an empty topic with the given value for TopicNum.

TTopic.Done

destructor TTopic.Done;   virtual;

Dispose of Text and destroy object.

TTopic.GetLine

function TTopic.GetLine(var Buffer; MaxLen: Word): Word;   virtual;

Gets the next line of text, return the length.

TTopic.MoreLines

function TTopic.MoreLines: Boolean;   virtual;

True if there are more lines of text.

TTopic.Write

procedure TTopic.Write(s: String);   virtual;

Writes the string to the help text.

TTopic.WriteLn

procedure TTopic.WriteLn(const s: String);   virtual;

Writes, then adds a newline.

TTopic.WriteKeyWord

procedure TTopic.WriteKeyWord(const s: String; Crossref: Longint);   virtual;

Writes the string with a marker that it's a cross-reference.

TTopic.HighLight

procedure TTopic.HighLight(On: Boolean);   virtual;

Turns highlighting of the text on or off.   If turned on twice, it will need to be turned off twice to return to standard.

TTopic.ResetHighLight

procedure TTopic.ResetHighLight;   virtual;

Turns highlighting off regardless of the initial state.

TTopic.BlankLine

procedure TTopic.BlankLine;   virtual;

Writes a blank line to the help topic, starting a new paragraph afterwards.

TTopic.StartXrefList

procedure TTopic.StartXrefList(const s: String);   virtual;

Starts a list of cross-referenced topics. End the list with EndXrefList.

TTopic.WriteXref

procedure TTopic.WriteXref(const s: String; Len: Word; Crossref: longint); virtual;

Like WriteKeyWord, but writes an entry to a cross-ref list. Len is the length in characters of the longest Xref to come; this may be used to format nicely. Assumes that StartXrefList has been called.

TTopic.EndXrefList

procedure TTopic.EndXrefList;   virtual;

Ends a list of cross-referenced topics started by StartXrefList.

TTopic.ToggleFixed

procedure TTopic.ToggleFixed;   virtual;

Toggles word-wrap mode.   Generally, help files start out word-wrapping; this should turn it off.   The TTopic method just toggles FixedLines.

TTopic.ToggleMarked

procedure TTopic.ToggleMarked;   virtual;

Toggles word marking mode.   Typically marked text would be used for code samples, as in the Borland help files.   This one just toggles Marked.

HelpFiles.PIndexItem

HelpFiles.PIndexItem = ^TIndexItem;

HelpFiles.TIndexItem

HelpFiles.TIndexItem = record

This is an item stored in the index for a help file.

Fields:

```
 Context,
 Inserted: longint;
 Subtitle,
 Token: TToken;
```

TIndexItem.Context

TIndexItem.Context,

The context or topic number.

Inserted: longint;

The count in the index when this item was inserted; allows a stable sort of the index.

TIndexItem.Subtitle

TIndexItem.Subtitle,

The token number of the subtitle string.

Token: TToken;

The token number of the name of index entry.

HelpFiles.PIndex

HelpFiles.PIndex = ^TIndex;

HelpFiles.TIndex

HelpFiles.TIndex = object(TBigSortedCollection)

This is an index for a help file, meant to hold TIndexItem records.

Fields and Methods:

Sortby...
procedure FreeItem...   virtual;
function Compare...   virtual;
procedure Insert...   virtual;
procedure AddItem...
procedure AddTokens...

TIndex.Sortby

TIndex.Sortby: (ByToken, BySubTitle, ByContext);

Marks which sort order should be used.

TIndex.FreeItem

procedure TIndex.FreeItem(Item: Pointer);   virtual;

Disposes of a TIndexItem.

TIndex.Compare

function TIndex.Compare(Item1, Item2: Pointer): Integer;   virtual;

Compares two index items according to the Sortby field.

TIndex.Insert

procedure TIndex.Insert(Item: Pointer);   virtual;

This inserts duplicates after existing values.

TIndex.AddItem

procedure TIndex.AddItem(const ATitle, ASubtitle: String; Atopicnum: Longint);

Add a new index entry by specifying the strings to use.

TIndex.AddTokens

procedure TIndex.AddTokens(ATitle, ASubtitle: TToken; Atopicnum: Longint);

Add a new index entry by specifying the token numbers.

HelpFiles.PHelpFile

HelpFiles.PHelpFile = ^THelpFile;

HelpFiles.THelpFile

HelpFiles.THelpFile = object(TObject)

This is the main abstract object representing a help file. It serves as a container for THelpTopics.

Fields and Methods:

```
Index: PIndex;
MultiIndexed: Boolean;
constructor Init;
destructor Done; virtual;
function NumTopics...   virtual;
function GetTitle...   virtual;
function GetSubTitle...   virtual;
function GetTopic...   virtual;
function NewTopic...   virtual;
procedure AddTopic...   virtual;
procedure DisplayTopic...   virtual;
procedure SetMainTopic...   virtual;
procedure Rewrite...   virtual;
procedure RewriteNotify...   virtual;
function TextSize...   virtual;
```

THelpFile.Index

THelpFile.Index: PIndex;

This is a TIndex maintained by the help file.

THelpFile.MultiIndexed

THelpFile.MultiIndexed: Boolean;

Indicates whether a topic can have more than one index entry in this help file type.

THelpFile.Init

constructor THelpFile.Init;

Construct an empty help file, and initialize Index to nil and MultiIndexed to false.

THelpFile.Done

destructor THelpFile.Done;   virtual;

Destroy the object and dispose of the Index.

THelpFile.NumTopics

function THelpFile.NumTopics: Longint;   virtual;

Return the number of topics in this file.

THelpFile.GetTitle

function THelpFile.GetTitle(TopicNum: Longint): String;   virtual;

Constructs a topic title for the given topic number.

THelpFile.GetSubTitle

function THelpFile.GetSubTitle(TopicNum: Longint): String;   virtual;

Constructs a topic subtitle.

THelpFile.GetTopic

function THelpFile.GetTopic(Context: Longint): PTopic;   virtual;

Extracts the given topic from the help file.

THelpFile.NewTopic

function THelpFile.NewTopic(Context: Longint; Someinfo: Pointer): PTopic; virtual;

Constructs a new topic of the appropriate type. Someinfo might be used by a descendant type.

THelpFile.AddTopic

procedure THelpFile.AddTopic(ATopic: PTopic);   virtual;

Writes the topic at the end of the base file, and records it with the appropriate topic number. If a topic with that number existed previously, it'll effectively be deleted. Atopic is disposed after adding it.

THelpFile.DisplayTopic

procedure THelpFile.DisplayTopic(var Where: Text; TopicNum: Longint);   virtual;

Displays the given topic number.

THelpFile.SetMainTopic

procedure THelpFile.SetMainTopic(TopicNum: Longint); virtual;

Defines which Topic is the main contents topic.

THelpFile.Rewrite

procedure THelpFile.Rewrite(s: PStream);   virtual;

Rewrites the help file to the given stream.

THelpFile.RewriteNotify

procedure THelpFile.RewriteNotify(num: longint);   virtual;

Should be called for each topic as it's written to the output file by ReWrite. The default version calls NotifyProc. This could be used to show a status report; num will run from 0 to pred(NumTopics), possibly skipping some values, but doesn't correspond to the topic number.

THelpFile.TextSize

function THelpFile.TextSize: Longint;   virtual;

Returns the total size of text so far.

HelpFiles.NotifyProc

HelpFiles.NotifyProc: procedure (num: Longint);

A procedure to be called by THelpfile.Rewrite.