

```
//
// FILENAME: eTDocInfo.h
// SUMMARY: Required persistent interface for ALL eText documents
// SUPERCLASS: Object:HashTable:NXStringTable:eTDocInfo
// PROTOCOLS: <ComponentWriting,ComponentReading,InspectableTarget>
// INTERFACE: None
// AUTHOR: Rohit Khare
// COPYRIGHT: ©1993,94 California Institute of Technology, eText Project
//
// Description
// eTDocInfo is supposed to encapsulate the essential
// persistent properties of a document; it's a document's
// proxy, or shadow, representation. The required fields are
// delineated in the main class, and these fields are designed
// to be pertinent to all eText documents. Fields pertinent
// only to textual substrates and UI information are in an informal
// category, eTextDoc. To contrast, a future Draw substrate would
// add an eDrawDoc category. Both categories are informal because
// they might add instance variables, but the spirit is the same.
//
// eTDocInfo can also output itself as a file, as HTML, and as LaTeX
// The functions associated with textual and graphic substrates have been
// placed in separate categories, eText and eDraw.
//
// HISTORY
// 10/30/94: Fixed tag font default registration, added +aliases for tags.
// 10/30/94: Modified to support <InspectableTarget>
// 10/27/94: -isVirgin added to public API.
// 10/04/94: -free bug; NXStrTab was free()ing NXAtoms. (was in init/rcfpth)
// 10/04/94: Added eTDoc ivar by analogy to path[] and eTComponents.
// 10/01/94: Reorganized into categories: eText and eDraw.
// 09/30/94: Revamped for eText5; renamed eTDocInfo
// 07/20/94: Added LaTeX support.
// 07/10/94: Added support for HTML header/trailer writing
// 01/11/94: Added Agent field.
// 01/11/94: Added flexible "bindery" methods for extension of navinfo data
// 01/10/94: Created. See Actors/eTNavinfo
//
// Imported Interfaces
//
// #import "eTextKernel.h"
//
// Field Definitions
//
// Datafile Names
// #define DOCINFOFILE NXUniqueString(".", "DOCI EXT")
```



```
//  
+ initialize;  
+ (HashTable *)docInfoTable;  
+ findDocInfo:(long) aDocID;  
- init;  
- free;  
  
//DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD  
// <ComponentReading, ComponentWriting>  
//  
- writeComponentToPath:(NXAtom)path inFormat:(int) theFormat;  
- readComponentFromPath:(NXAtom)path;  
  
//DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD  
// Accessor Methods  
//  
- setDoc: newDoc;  
- etDoc;  
  
- (long) docID;  
- (NXAtom) docIDStr;  
  
- (NXAtom) docTitle;  
- setDocTitle:(const char *) newTitle;  
  
- (NXAtom) docType;  
- setType:(const char *) newType;  
  
- (NXAtom) author;  
- setAuthor:(const char *) newAuthor;  
  
- (NXAtom) agent;  
- setAgent:(const char *) newAgent;  
  
- (long) parentId;  
- (NXAtom) parent;  
- setParent:(const char *) newParent;  
- setParentID:(long) newParentID;  
  
- (NXAtom) peers; // A space-delimited list of ASCII %x docIDs  
- setPeers:(const char *) newPeers;  
- addPeerID:(long) newPeerID;  
  
- (NXAtom) keywords;  
- setKeywords:(const char *) newKeywords;  
  
- (NXAtom) rtfComments; // RTF source of comment field
```


[illegible]