

- (ZBEJelementArray<NotesWindow \*> \*) windows;
- (ZBEJelementArray<NotesDocument \*> \*) documents;

@interface NotesApplication : ZApplication  
 \ \ The application, a top-level scribbling object.

\*\  
 \* Standard suite  
 \\*

@end

- (void) moveTo:(ZObject \*)to: \ \ Move an object to a new location.  
 Copy an object.
- (void) quibJicateTo:(ZObject \*)to withProperties:(NSDictionary \*)withProperties: \ \
- (void) delete: \ \ Delete an object.
- (BOO)PrintJag: \ \ Print a document.
- (void) PrintWithProperties:(NSDictionary \*)withProperties PrintJag:
- (void) saveIn:(NSURL \*)in as:(NotesSaveableJefJefFormat)as: \ \ Save a document.  
 document.
- (void) closeSaving:(NotesSaveOptions)saving savingIn:(NSURL \*)savingIn: \ \ Close a

@protocol NotesGenericMethods

typedef enum NotesSaveableJefJefFormat NotesSaveableJefJefFormat;  
 };

NotesSaveableJefJefFormatNativeFormat = ,item, \ \* Native format \*\  
 enum NotesSaveableJefJefFormat {

typedef enum NotesPrintingErrorHandlerJing NotesPrintingErrorHandlerJing;  
 };

errors \*\  
 NotesPrintingErrorHandlerJingDefaulted = ,Jmqt, \ \* Print a defaulted report of PostScript  
 NotesPrintingErrorHandlerJingStandard = ,Jmqt, \ \* Standard PostScript error handling \*\  
 enum NotesPrintingErrorHandlerJing {

typedef enum NotesSaveOptions NotesSaveOptions;  
 };

NotesSaveOptionsAsk = ,ask, \ \* Ask the user whether or not to save the file. \*\  
 NotesSaveOptionsNo = ,no, \ \* Do not save the file. \*\  
 NotesSaveOptionsYes = ,yes, \ \* Save the file. \*\  
 enum NotesSaveOptions {

NotesNote, NotesAttachment;

@class NotesApplication, NotesDocument, NotesWindow, NotesAccount, NotesFolder,

#import <scribblingBridge\scribblingBridge.h>  
 #import <AppKit\AppKit.h>

\*\  
 \* Notes.h  
 \\*

- (2BEJelementAriaλ<NotesForJdcl \*> \*) fofqelr:
- (2BEJelementAriaλ<NotesAccount \*> \*) accountr:

@interface NotesApplication (NotesJntfe)  
 \ \ the Notes application

\*\  
 \* Notes Jntfe  
 \\*

@end

qizb9aleq in the mīndom"

@blobeifl (cobλ' leaqoujl) NotesDocument \*qocnment: \ \ The qocnment mpoze contents are  
 @blobeifl B00Γ zoomeq: \ \ Is the mīndom zoomeq līdmf nom;  
 @blobeifl (leaqoujl) B00Γ zoowarje: \ \ Does the mīndom μave a zoow pntfon;  
 @blobeifl B00Γ līzīprje: \ \ Is the mīndom līzīprje līdmf nom;  
 @blobeifl (leaqoujl) B00Γ lezīzarje: \ \ Can the mīndom pe lezīzeq;  
 @blobeifl B00Γ wīnīzafnīzēq: \ \ Is the mīndom wīnīwīzēq līdmf nom;  
 @blobeifl (leaqoujl) B00Γ wīnīzafnīzarje: \ \ Does the mīndom μave a wīnīwīze pntfon;  
 @blobeifl (leaqoujl) B00Γ cfozearje: \ \ Does the mīndom μave a cfoze pntfon;  
 @blobeifl n2Bect ponuqz: \ \ The ponuqīnd rectandje of the mīndom"  
 @blobeifl n2Inteder īndex: \ \ The īndex of the mīndom' ordered fiont to pack"  
 - (n2Inteder) īq: \ \ The nūīdne īdentīfīer of the mīndom"  
 @blobeifl (cobλ' leaqoujl) n2fīrīnd \*name: \ \ The fīfje of the mīndom"

@interface NotesMīndom : 2B0rject <NotesCenerīcMefhodz>  
 \ \ A mīndom"

@end

@blobeifl (cobλ' leaqoujl) n2NBL \*fīfje: \ \ Its focation on qīzk' īf īf μas one"  
 @blobeifl (leaqoujl) B00Γ moqīfīeq: \ \ Has īf been moqīfīeq zīnce the fāst zave;  
 @blobeifl (cobλ' leaqoujl) n2fīrīnd \*name: \ \ Its name"

@interface NotesDocument : 2B0rject <NotesCenerīcMefhodz>  
 \ \ A qocnment"

@end

- (loīq) obenNotesfocation:(n2fīrīnd \*)x: \ \ Oben a note NBL"
- (B00Γ) exīzīz:(īq)x: \ \ Verīfīl fμaf an oject exīzīz"
- (loīq) dnīfzavīnd:(NoteszaveObfīoung)zavīnd: \ \ Onīf the abbīfcafīon"  
 (B00Γ) bīnīfDīzfođ: \ \ Bīnīf a qocnment"
- (loīq) bīnīf:(īq)x mīfμblobeifīez:(n2Dīcfīouarλ \*)mīfμblobeifīez bīnīfDīzfođ:
- (īq) oben:(īq)x: \ \ Oben a qocnment"

@blobeifl (cobλ' leaqoujl) n2fīrīnd \*verīzīon: \ \ The verīzīon nūmber of the abbīfcafīon"  
 @blobeifl (leaqoujl) B00Γ fīonfwozīf: \ \ Is fμīz the acfīve abbīfcafīon;  
 @blobeifl (cobλ' leaqoujl) n2fīrīnd \*name: \ \ The name of the abbīfcafīon"

@blobelf (leafonly) BOOF shared: \\ Is the note shared;  
 @blobelf (leafonly) BOOF baseprotected: \\ Is the note baseprotected;  
 note  
 @blobelf (cobyl' leafonly) N2date \*modificationdate: \\ the modification date of the  
 @blobelf (cobyl' leafonly) N2date \*creationdate: \\ the creation date of the note  
 @blobelf (cobyl' leafonly) N2filing \*bjaufext: \\ the bjaufext content of the note  
 @blobelf (cobyl) N2filing \*poql: \\ the HTML content of the note  
 @blobelf (cobyl' leafonly) NotesFolder \*container: \\ the folder of the note  
 - (N2filing \*) fq: \\ the unique identifier of the note  
 poql)  
 @blobelf (cobyl) N2filing \*name: \\ the name of the note (normally the first line of the  
 - (ZBEJementArray<NotesAttachment \*> \*) attachments:

@interface NotesNote : ZBObject <NotesGenericMethods>  
 \\ a note in the notes application

@end

- (fq) showseparately:(BOOF)separately: \\ show an object in the UI

@blobelf (cobyl' leafonly) fq container: \\ the container of the folder  
 @blobelf (leafonly) BOOF shared: \\ Is the folder shared;  
 - (N2filing \*) fq: \\ the unique identifier of the folder  
 @blobelf (cobyl) N2filing \*name: \\ the name of the folder

- (ZBEJementArray<NotesNote \*> \*) notes;  
 - (ZBEJementArray<NotesFolder \*> \*) folders;

@interface NotesFolder : ZBObject <NotesGenericMethods>  
 \\ a folder containing notes

@end

- (fq) showseparately:(BOOF)separately: \\ show an object in the UI

- (N2filing \*) fq: \\ the unique identifier of the account  
 @blobelf (leafonly) BOOF nbdaedq: \\ Is the account nbdaedq;  
 @blobelf (cobyl) N2filing \*name: \\ the name of the account  
 @blobelf (cobyl) NotesFolder \*defaultFolder: \\ the default folder for creating notes

- (ZBEJementArray<NotesNote \*> \*) notes;  
 - (ZBEJementArray<NotesFolder \*> \*) folders;

@interface NotesAccount : ZBObject <NotesGenericMethods>  
 \\ a notes account

@end

@blobelf (cobyl) N2Array<NotesNote \*> \*selection: \\ the selected note(s)  
 @blobelf (cobyl) NotesAccount \*defaultAccount: \\ the default account for creating notes

- (ZBEJementArray<NotesAttachment \*> \*) attachments;  
 - (ZBEJementArray<NotesNote \*> \*) notes;

@end

– (id) showSeparately:(BOOL)separately: \ \ show an object in the UI

@private (readonly) BOOL shared: \ \ Is the attachment shared;  
release

@private (const, readonly) NSString \*url: \ \ for URL attachments, the URL the attachment  
attachment

@private (const, readonly) NSDate \*modificationDate: \ \ the modification date of the

@private (const, readonly) NSDate \*creationDate: \ \ the creation date of the attachment  
note, 2 HTML

@private (const, readonly) NSString \*contentType: \ \ the content-type URL in the

@private (const, readonly) NSString \*container: \ \ the note containing the attachment

– (NSString \*) id: \ \ the unique identifier of the attachment

@private (const, readonly) NSString \*name: \ \ the name of the attachment

@interface NotesAttachment : NSObject <NotesGenericMethod>

\ \ an attachment in a note

@end

– (id) showSeparately:(BOOL)separately: \ \ show an object in the UI