```
/*
 * Contacts.h
 */

#import <AppKit/AppKit.h>
#import <ScriptingBridge/ScriptingBridge.h>


@class ContactsApplication, ContactsDocument, ContactsWindow, ContactsAddress, ContactsContactInfo, ContactsAIMHandle, ContactsCustomDate, ContactsEmail, ContactsEntry, ContactsGroup, ContactsICQHandle, ContactsInstantMessage, ContactsJabberHandle, ContactsMSNHandle, ContactsPerson, ContactsPhone, ContactsRelatedName, ContactsSocialProfile, ContactsUrl, ContactsYahooHandle;

enum ContactsSaveOptions {
	ContactsSaveOptionsYes = 'yes ' /* Save the file. */,
	ContactsSaveOptionsNo = 'no  ' /* Do not save the file. */,
	ContactsSaveOptionsAsk = 'ask ' /* Ask the user whether or not to save the file. */
};
typedef enum ContactsSaveOptions ContactsSaveOptions;

enum ContactsPrintingErrorHandling {
	ContactsPrintingErrorHandlingStandard = 'lwst' /* Standard PostScript error handling */,
	ContactsPrintingErrorHandlingDetailed = 'lwdt' /* print a detailed report of PostScript errors */
};
typedef enum ContactsPrintingErrorHandling ContactsPrintingErrorHandling;

enum ContactsSaveableFileFormat {
	ContactsSaveableFileFormatArchive = 'abbu' /* The native Contacts file format */
};
typedef enum ContactsSaveableFileFormat ContactsSaveableFileFormat;

enum ContactsInstantMessageServiceType {
	ContactsInstantMessageServiceTypeAIM = 'az85',
	ContactsInstantMessageServiceTypeFacebook = 'az94',
	ContactsInstantMessageServiceTypeGaduGadu = 'az86',
	ContactsInstantMessageServiceTypeGoogleTalk = 'az87',
	ContactsInstantMessageServiceTypeICQ = 'az88',
	ContactsInstantMessageServiceTypeJabber = 'az89',
	ContactsInstantMessageServiceTypeMSN = 'az90',
	ContactsInstantMessageServiceTypeQQ = 'az91',
	ContactsInstantMessageServiceTypeSkype = 'az92',
	ContactsInstantMessageServiceTypeYahoo = 'az93'
};
typedef enum ContactsInstantMessageServiceType ContactsInstantMessageServiceType;

@protocol ContactsGenericMethods

- (void) closeSaving:(ContactsSaveOptions)saving savingIn:(NSURL *)savingIn;  // Close a document.
- (void) saveIn:(NSURL *)in_ as:(ContactsSaveableFileFormat)as;  // Save a document.
- (void) printWithProperties:(NSDictionary *)withProperties printDialog:(BOOL)printDialog;  // Print a document.
```

```
- (void) delete;  // Delete an object.
- (void) duplicateTo:(SBObject *)to withProperties:(NSDictionary *)withProperties;  //
Copy an object.
- (void) moveTo:(SBObject *)to;  // Move an object to a new location.

@end



/*
 * Standard Suite
 */

// The application's top-level scripting object.
@interface ContactsApplication : SBApplication

- (SBElementArray<ContactsDocument *> *) documents;
- (SBElementArray<ContactsWindow *> *) windows;

@property (copy, readonly) NSString *name;  // The name of the application.
@property (readonly) BOOL frontmost;  // Is this the active application?
@property (copy, readonly) NSString *version;  // The version number of the application.

- (id) open:(id)x;  // Open a document.
- (void) print:(id)x withProperties:(NSDictionary *)withProperties printDialog:
(BOOL)printDialog;  // Print a document.
- (void) quitSaving:(ContactsSaveOptions)saving;  // Quit the application.
- (BOOL) exists:(id)x;  // Verify that an object exists.
- (id) save;  // Save all Contacts changes. Also see the unsaved property for the
application class.

@end

// A document.
@interface ContactsDocument : SBObject <ContactsGenericMethods>

@property (copy, readonly) NSString *name;  // Its name.
@property (readonly) BOOL modified;  // Has it been modified since the last save?
@property (copy, readonly) NSURL *file;  // Its location on disk, if it has one.


@end

// A window.
@interface ContactsWindow : SBObject <ContactsGenericMethods>

@property (copy, readonly) NSString *name;  // The title of the window.
- (NSInteger) id;  // The unique identifier of the window.
@property NSInteger index;  // The index of the window, ordered front to back.
@property NSRect bounds;  // The bounding rectangle of the window.
@property (readonly) BOOL closeable;  // Does the window have a close button?
@property (readonly) BOOL miniaturizable;  // Does the window have a minimize button?
@property BOOL miniaturized;  // Is the window minimized right now?
@property (readonly) BOOL resizable;  // Can the window be resized?
@property BOOL visible;  // Is the window visible right now?
```

```
@property (readonly) BOOL zoomable;  // Does the window have a zoom button?
@property BOOL zoomed;  // Is the window zoomed right now?
@property (copy, readonly) ContactsDocument *document;  // The document whose contents are displayed in the window.


@end



/*
 * Contacts Script Suite
 */

@interface ContactsApplication (ContactsScriptSuite)

- (SBElementArray<ContactsGroup *> *) groups;
- (SBElementArray<ContactsPerson *> *) people;

@property (copy) ContactsPerson *myCard;  // Returns my Contacts card.
@property (readonly) BOOL unsaved;  // Does Contacts have any unsaved changes?
@property (copy) NSArray<ContactsPerson *> *selection;  // Currently selected entries
@property (copy, readonly) id defaultCountryCode;  // Returns the default country code for addresses.

@end

// Address for the given record.
@interface ContactsAddress : SBObject <ContactsGenericMethods>

@property (copy) id city;  // City part of the address.
@property (copy, readonly) id formattedAddress;  // properly formatted string for this address.
@property (copy) id street;  // Street part of the address, multiple lines separated by carriage returns.
- (NSString *) id;  // unique identifier for this address.
- (void) setId: (NSString *) id;
@property (copy) id zip;  // Zip or postal code of the address.
@property (copy) id country;  // Country part of the address.
@property (copy) id label;  // Label.
@property (copy) id countryCode;  // Country code part of the address (should be a two character iso country code).
@property (copy) id state;  // State, Province, or Region part of the address.


@end

// Container object in the database, holds a key and a value
@interface ContactsContactInfo : SBObject <ContactsGenericMethods>

@property (copy) id label;  // Label is the label associated with value like "work", "home", etc.
@property (copy) id value;  // Value.
- (NSString *) id;  // unique identifier for this entry, this is persistent, and stays with the record.
```

```
@interface ConfigPerson : ConfigEntry
\\ A person in the address book database.

@end


@property (copy) id userName;  \\ The user name of this instance message address.
@property (copy) id serviceType;  \\ The service type of this instance message address.
@property (copy, readonly) id serviceName;  \\ The service name of this instance message address.

@interface ConfigInstanceMessage : ConfigInfo
\\ Address for instance messaging.

@end


@property (copy) NSString *name;  \\ The name of this group.

- (CBEnumerator<ConfigPerson *> *) people;
- (CBEnumerator<ConfigGroup *> *) groups;

@interface ConfigGroup : ConfigEntry
\\ A Group Record in the address book database

@end


- (ConfigPerson *) removeFrom:(CBOject *)from;  \\ Remove a child object.
- (ConfigPerson *) addTo:(CBOject *)to;  \\ Add a child object.

@property BOOL selected;  \\ Is the entry selected?
- (NSString *) id;  \\ Unique and persistent identifier for this record.
@property (copy, readonly) NSDate *creationDate;  \\ When the contact was created.
@property (copy, readonly) NSDate *modification;
@end


@interface ConfigEntry : ConfigInfo
\\ An entry in the address book database

@end


@interface ConfigEmail : ConfigInfo
\\ Email address for a person.

@end


@interface ConfigUsfomDate : ConfigInfo
\\ Arbitrary date associated with this person.

@end
```

```
- (NSString *) id;  // The persistent unique identifier for this profile.

@interface ContactsSocialProfile : SBObject <ContactsGenericMethods>
// Profile for social networks.

@end


@interface ContactsRelatedName : ContactsContactInfo
// Other names related to this person.

@end


@interface ContactsPhone : ContactsContactInfo
// Phone number for a person.

@end


@property (copy) id firstName;  // The First name of this person.
@property (copy) id lastName;  // The Last name of this person.
@property (copy) id jobTitle;  // The job title of this person.
person.
@property (copy) id phoneticFirstName;  // The phonetic version of the First name of this
@property (copy) id middleName;  // The Middle name of this person.
@property BOOL company;  // Is the current record a company or a person.
@property (copy) id note;  // Notes for this person.
name display order preference setting in Contacts.
@property (copy, readonly) NSString *name;  // First/Last name of the person, uses the
@property (copy) id image;  // Image for person.
@property (copy) id department;  // Department that this person works for.
this person.
@property (copy) id phoneticMiddleName;  // The Phonetic version of the Middle name of
@property (copy) id title;  // The title of this person.
person.
@property (copy) id phoneticLastName;  // The phonetic version of the Last name of this
@property (copy) id birthDate;  // The birth date of this person.
@property (copy) id homePage;  // The home page of this person.
returns a card in version 3.0 format.
@property (copy, readonly) id vcard;  // Person information in vCard format, this always
@property (copy) id suffix;  // The Suffix of this person.
@property (copy) id maidenName;  // The Maiden name of this person.
@property (copy) id organization;  // Organization that employs this person.
@property (copy) id nickname;  // The Nickname of this person.

- (SBElementArray<ContactsEmail *> *) emails;
- (SBElementArray<ContactsRelatedName *> *) relatedNames;
- (SBElementArray<ContactsSocialProfile *> *) socialProfiles;
- (SBElementArray<ContactsInstantMessage *> *) instantMessages;
- (SBElementArray<ContactsCustomDate *> *) customDates;
- (SBElementArray<ContactsGroup *> *) groups;
- (SBElementArray<ContactsPhone *> *) phones;
- (SBElementArray<ContactsAddress *> *) addresses;
- (SBElementArray<ContactsUrl *> *) urls;
```

```
@property (copy) id serviceName;  \\ The service name of this social profile.
@property (copy) id userName;  \\ The username used with this social profile.
@property (copy) id serviceIdentifier;  \\ A service-specific identifier used with this
social profile.
@property (copy) id url;  \\ The URL of this social profile.

@end


@interface ConfigcfsUrl : ConfigcfsConfigcfInto
\\ URLs for this person.

@end
```