MAPI Sample Address Book

Overview and release notes, 5/13/95

This document gives a brief overview of the sample address book service provider (SMPAB.DLL) that is part of the MAPI SDK.

Architecture

The .SAB file. The sample address book provider is based on a read-only file of fixed-length records. Each record contains the display name and e-mail address of a messaging user. This file supplies the contents of a single, read-only address container, which contains only e-mail users.

A separate utility program, SAB.EXE, converts a text file to an .SAB file. Each line of input text must be formatted as follows:

display name<tab>*e-mail address*

The e-mail address must not contain white space, and the input file must be sorted on the display name; the conversion utility does not sort.

Address type. The sample address book supports one e-mail address type; it is hard-coded to **MSPEER**. If you plan to use the sample address book and sample transport together, configure the sample transport provider to use **MSPEER** also.

Entry identifiers. The sample address book registers one MAPIUID with **SetProviderUID()**; it is also hard-coded. The unique part of a messaging user entry identifier is the e-mail address. Messaging user entry identifiers use the hard-coded UID, so they are the same across all instances of the sample address book.

The sample address book chooses a UID at random the first time it is configured for use in its top-level container entry identifier. This enables multiple instances of the provider, each with its own top-level container, to coexist in a single profile. **NOTE: IF YOU USE THIS SOURCE CODE AS THE BASIS FOR YOUR OWN PROVIDER, SAVE YOURSELF A LOT OF TROUBLE AND CHANGE THE HARDCODED UID IN ABP.H.**

The sample address book does not distinguish between short- and long-term entry identifiers.

Table implementation. The sample address book largely delegates the implementation of its hierarchy table to the **IMAPITableData** interface in MAPI32.DLL.

For its contents table, the sample address book implements the **IMAPITable** interface directly on the .SAB file rather than delegating to MAPI32.DLL. The contents table is a good example of a caching table implementation; it defers reading records from disk until absolutely necessary, and can handle quite large files with limited memory. The contents table also supports prefix searching and ambiguous name resolution (ANR).

The contents table implementation closes the .SAB file if the table interface is not accessed for a while. This makes it possible to update the .SAB file while it is in use. If the .SAB file changes, the sample address book notices at the next file access and issues a notification on the contents table so that the user interface may be updated. It uses the idle task engine in MAPI32.DLL to do the closing.

Supported features

Table restrictions. The sample address book implements the two most commonly used types of restrictions for an address book: prefix match and ANR. It does not implement the full MAPI restriction language, and restrictions are supported only on the display name.

Prefix matching enables incremental searching through an address book container in which the user types the initial characters of a display name. ANR, on the other hand, matches one or more prefix strings against the entire container, regardless of first or last name order, and returns all matching entries; this enables checking of partial names typed into a message.

Details. The sample address book implements its own details display table for messaging users. There are two sets of properties divided into two pages. The first page has the standard properties: display name, e-mail address, and address type. The second page has an example of almost every control type implemented in MAPI display tables (The data is hard-coded, not drawn from the address book.). There is a template identifier for messaging user details, which enables all details to be supported for users to be copied from the sample address book to the personal address book.

Details are also supported for the top-level directory, including the ability to change the .SAB file from the details pane. The directory details view uses a notification to update the display when the .SAB file is changed.

Custom recipients (one-offs). The sample address book implements a one-off table with a set of properties different from the native details; these properties combine to form the e-mail address. The template identifier enables messaging users to be added to the personal address book or directly to a message using a special dialog box; processing is routed through the sample address book, which has the ability to generate the e-mail address from the properties entered in the dialog box.

Advanced search dialog box. The sample address book implements an advanced search dialog box, accessible by choosing the Advanced button on the MAPI addressing dialog box. In reality, the advanced search is only slightly different from the standard search, which uses an ANR restriction.

Status interface. The **IMAPIStatus** interface is partially supported; its **IMAPIProp** methods are delegated to the **IMAPIPropData** interface in MAPI32.DLL.

Configuration interfaces. The sample address book supports both interactive and programmatic configuration.

Cross-platform support. The sample address book source code generates executables for Windows 3.*X*, Windows 95, and Windows NT 3.51 and later from a single set of source files.

Unsupported features

Sorting (The table is permanently sorted on the display name.)

Distribution lists

Creating, deleting, and modifying entries

Multi-valued properties

Named properties

Distinguishing first and last name in display names

Configuration

The only configurable property supported by the sample address book is the path to the .SAB file. The path can be set interactively either through the service configuration dialog box or at provider logon. It can also be set programmatically through the **ServiceEntry()** function. For the definitions needed to set the path programmatically, see the SMPAB.H header file.