

===== **MAPI RELEASE NOTES** =====
MAPI SDK Version 1.00 Beta 5

This document contains release notes for the Beta 5 version of the MAPI SDK.

Contents of this file by section:

- 1. System Requirements**
- 2. What's New, Major Changes**
- 3. Known Problems in This Release**
- 4. Provider Descriptions**
- 5. Installing OLE Messaging on Windows 95**
- 6. Installing the 16-bit MAPI PDK from CD**
- 7. Installing the Windows NT MAPI subsystem**
- 8. Product Support**

NOTE: If you find inconsistencies in the documentation, resolve them by using information in the following order:

1. Release notes
2. Header files
3. Sample code
4. Help files
5. Printed documentation (.doc files)

For example, if the release notes contradict a Help file, use the information in the release notes.

===== **1. SYSTEM REQUIREMENTS** =====

The MAPI SDK supports 3 platforms, Windows 3.1 (or Windows for Workgroups 3.11), Windows 95, and Windows NT 3.51 or above. Look below to determine the system requirements specific to your development platform.

Windows 3.1, Windows for Workgroups 3.11

Must have Windows SDK.
Must have Microsoft Visual C++ 1.5.

Windows 95

Must have Win32 SDK.
Must have MSVC 2.0 (or newer)
Can optionally use Microsoft Visual C++ 1.5 if targeting the Windows 3.1 (or WFW 3.11) platform.
Must install "Microsoft Exchange" from the Windows 95 Setup program. Note that "Microsoft Exchange" is not installed or selected by default, so users will have to select these components via the Custom Setup option at the beginning of Windows 95 setup, or it can be added to Windows 95 by activating the "Inbox" icon on your desktop following Windows 95 installation.

Windows NT 3.51

Must have Win32 SDK.
Must have MSVC 2.0 (or newer)
Can optionally use Microsoft Visual C++ 1.5 if targeting the Windows 3.1 (or WFW 3.11)

platforms.

===== 2. KNOWN PROBLEMS IN THIS RELEASE =====

2.1 MAPI APIs frequently are defined to enable use of a MAPI_UNICODE flag value and UNICODE strings, but most of the MAPI 1.0 implementation does not support UNICODE. Methods and functions that are passed MAPI_UNICODE as a flag value and do not support it return MAPI_E_BAD_CHARWIDTH.

2.2 On Windows 95 and NT, notification callbacks can happen on any thread and may occur at a time when it is inconvenient for the originating application to handle them. If your application requires callbacks to happen on a particular thread, use the library function **HrThisThreadAdviseSink** to wrap your AdviseSink before passing it to the MAPI Advise method. You may need careful coding to handle the case where the item of interest exists at the time the original notification arrives but no longer exists at the time it is handled by the internal application window.

2.3 Remoting MAPI interfaces with the CoMarshalInterface and CoUnmarshalInterface functions is implemented and supported but not fully tested yet. Interfaces obtained with the CoUnmarshalInterface function can only be called from the thread on which they were unmarshalled.

2.4 The 32-bit MAPI sample service providers source code is currently only supported on 32-bit Microsoft Visual C++.

2.5 Testing of MAPI sample source code with non-Microsoft compilers has not yet been completed.

2.6 Documentation is part way through an organization revision. Finding individual items using the Search dialog box is often the fastest way to locate documentation on a known item. Also, we plan to release a more up to date version of the help files. Information on the updated release of documentation will be posted on CompuServe.

2.7 Sample providers must be copied to the system directory of your windows installation in order to function correctly.

2.8 Unresolved External error when linking 16bit providers on Windows 95.
If using MSVC 1.5 to build 16 bit apps on Windows 95, you will need to do the following:
-Replace the "windowsx.h" file supplied by MSVC 1.5 with the same file from the Windows 95 SDK.

-Make the following changes to the sdkenv.bat file:

- add the msvc\include" directory to the **beginning** of the line which sets the "include" environmental variable.
- add the msvc\lib directory to the **beginning** of the line which sets the "lib" environmental variable.
- add the msvc\bin directory to the **end** of the line which sets the "path" environmental variable.

Assuming you install MSVC to the c:\msvc directory, the lines might look like the following:
set include

```
=c:\msvc\include;c:\msvc\mfc\include;%sdkroot%\inc16;%_include%;%sdkroot%\inc32
set lib=c:\msvc\lib;c:\msvc\mfc\lib;%sdkroot%\lib16;%lib%
path=%sdkroot%\binw16;%sdkroot%\bin;c:\msvc\bin;c:%_path%
```

2.9 MDB Viewer (MDBVU*.exe) crashes if user exits from session before completing logon sequence. If a user tries to logon to the MDBVU*.exe app and "OK's" the first dialog, then

ignores the second MAPILogon dialog and instead selects the Exit command from the menu of the opened session, the viewer will crash or GP Fault depending on the platform.

2.10 Compiler warnings: compiling the sample code may result in compiler warnings. These warnings can be safely ignored.

2.11 There is NO interop between previous versions of MAPI and the current version of MAPI. Also, there is no interop between Windows 95 M7 and Windows 95 M8 releases of MAPI or of Microsoft Exchange clients, stores, or address books. You cannot send/receive messages to/from mismatched Windows 95 releases.

2.12 When you install Windows 95, or use Microsoft Exchange for the first time, the Microsoft Exchange Setup Wizard will be run. The Wizard is also run each time you Add a new profile in the Microsoft Exchange Profiles control panel. It does not run if you add a service to an existing profile.

2.13 In the sample address book provider, saving changes after setting properties on an address entry returns SUCCESS when it should correctly return MAPI_E_NO_ACCESS. An example of this problem follows:

```
ABRoot->OpenEntry( SAB entry, &SabUser )
SabUser->SetProps( PR_COMMENT="...", PR_LOCATION="..." )
SabUser->SaveChanges(KEEP_OPEN_READWRITE)
```

2.14 SZFindSZ(), which is an exported function in MAPI32.DLL, does not work as expected. It only finds a substring if that substring is at the end of the search string. Impact on the sample code is limited to IFLDR_CopyTo() and IFLDR_CopyProps() in the sample message store. These functions will not detect circular links in folder hierarchies as they are copied.

2.15 There are numerous problems building the 16-bit sample applications and service providers from the 16-bit MAPI PDK under Windows 3.1x. Here are workarounds for those problems:

SAMPLE.AB: Delete the line
DEFSOURCE=SMPAB.DES
from the makefile, or build using the following command line:

```
nmake DEFSOURCE=
SAMPLE.XP: Delete the line
DEFSOURCE=SMPXP.DES
from the makefile, or build using the following command line:
nmake DEFSOURCE=
```

CMCCLI: Copy PVALLOC.H from the DEV\H subdirectory into the DEV\CMCCLI subdirectory. Copy PVALLOC.C from the DEV\MISC subdirectory into the DEV\CMCCLI subdirectory. In the makefile, change the line

```
H=$(MAPI)\$(SRCROOT)\smppcmcli
to
```

```
H=$(MAPI)\$(SRCROOT)\cmcli
SAMPLE.MS:
```

2.16 Most of the sample MAPI applications and providers, when built on Windows NT or Windows 95 and targeted for Windows NT, link to CTL3D32.DLL. If so built, they will not run on Windows 95, where CTL3D32.DLL is not normally present. To work around this problem, target Windows 95 specifically, using the following command line:

```
nmake TARGETOS=WIN95
You may add the line
TARGETOS=WIN95
```

to the makefile if you always target Windows 95. The default specification is TARGETOS=BOTH, which does not cause _WIN95 to be defined by the C compiler. Future versions of the sample

applications and providers will be enhanced to detect the operating system version at runtime and load CTL3D32.DLL dynamically if necessary.

===== 3. WHAT'S NEW , MAJOR CHANGES=====

- All the MAPI support dll's have now been merged into MAPI32.DLL. You should relink your application and/or provider with MAPI32.LIB in order to take this change into account.
- Services.ini has been renamed to mapisvc.inf.
- Some documentation is not available in this release. Some of the textual descriptions of MAPI 1.0 are undergoing extensive reorganization. As a result, some textual descriptions that were previously available are not available in the current set of documents. Please use the available documentation along with the sample apps and providers to answer most of your questions. When the re-organization of the documentation is more complete, we will make those documents available via Compuserve. Also, regular updates of the Help file(s) will be posted to Compuserve. See section 8 of the document for support specifics..
- A new PST recovery tool, SCANPST, is now available.
- OLE Messaging now available on all platforms. See documentation.
- A Sample Mail Handler (SMH.dll) has been added as a sample of message hook processing. See the readme in the SMH directory for details.
- Substantive changes to forms API's and a new sample Checkers form application is included in the SDK.
- Handling of leaked objects by providers has changed, as has timing of Service Provider DLL unload.

We now **defer unloading of providers** until the final MAPIUninitialize. This keeps the objects alive (unless the provider forcibly deleted them when its logon object is released, which it should not!) long enough for us to force OLE to disconnect the objects without crashing. At provider logoff time, providers should completely destroy only objects with no references remaining. Any object with a positive refcount (*i.e.* a leaked object) should stay alive, although it needs to support only IUnknown methods after this. Providers can safely invalidate the rest of the v-table. They should also invalidate/free all data associated with the object that is not used in any IUnknown method. The IUnknown methods on all such (leaked) objects can get called at any time until the provider DLL gets unloaded, and they should work.
- The MAPI Spooler now exits when the last profile logs off. This may disclose leaks or other problems in your providers which were not apparent until now.
- Exchange Client Extensibility docs and help files have been added.
- OLE Messaging documentation and help files have had a major update.
- Additional sample source code emphasizing C++ is available and will be posted to Compuserve.

===== 4. COMPONENT DESCRIPTIONS =====

The MAPI SDK contains the following components:

(Note: although not explicitly stated, the filename for the 32 bit version of components is the same as the 16 bit filename + "32" added to the end of the 8 character file name (e.g. abview.exe is the 16 bit version, abview32.exe is the 32 bit version)).

ABVIEW is an executable program which allows browsing of Address Books.

CMCCLI is a executable program used to demonstrate CMC calls. It implements a small E-mail client. Its source is in mstools\samples\mapicmccli.

MAPI is a dynamic library which implements 16-bit access to all MAPI32 functionality. On 32-bit platforms this is accomplished through thunking and/or interface remoting to the native 32 bit messaging subsystem.

MAPI32 is a dynamic library which implements the Simple MAPI and CMC calls, as well as providing the core of the Extended MAPI implementation.

MAPI32 is the "message spooler," an executable program which is also a core part of MAPI.

MDBVU is an executable program which allows browsing the detail of a message store. It uses the **PROPVU**, **STATVU**, and **TBLVU** dynamic libraries.

MDISP.exe and MDISP.tlb. The .exe is the OLE Automation server for OLE Messaging, and the .tlb file is the Type Library for OLE Messaging.

MERGEINI.EXE is an executable program, used as part of the install and setup of additional service providers into an already configured MAPI system. It merges configuration about the new service provider into the MAPISVC.INF file in the Windows SYSTEM or SYSTEM32 subdirectory.

MERGEINI.EXE takes a subset of a complete MAPISVC.INF, describing just the services providers to be installed, and merges this with the existing MAPISVC.INF without destroying existing service provider information. MERGEINI.EXE optionally prompts the user while performing its task and takes the following command line options:

-m	Merge the stub with the existing MAPISVC.INF and prompt the user to confirm the merge.
-c	Copy this stub over the existing MAPISVC.INF, destroying the other MAPISVC.INF.
-q	Do not prompt the user at any time.
-h	Display a short Help dialog box with available parameters.
<stub-path-name>	Full path to the stub MAPISVC.INF to merge from.

MMFMIG is a dynamic library which provides Microsoft Exchange the ability to import messages from MS Mail 3.x files (.mmf). It is accessed under File.Import Mail Messages. This will allow the user to migrate messages contained in the .mmf file into a Microsoft Exchange Personal Information Store (.pst). The user can then access the messages as if they were created by the Microsoft Exchange client. However, due to a bug, the user may not be able to automatically "reply" or "reply all", without manually resolving the addresses.

MSFS is a dynamic library which implements a transport provider and address book provider to Microsoft Mail for PC Networks 3.0 and 3.2 post offices.

MSPST is a dynamic library which implements the Microsoft Personal Information Store (a.k.a. PST), and the Microsoft Personal Address Book (a.k.a. PAB). The Microsoft Personal Information Store is a message store provider which you use to store messages in a .PST file, typically on your own disk. A .PST file can only be accessed by one computer at a time. The Microsoft Personal Address Book is an address book provider which stores information about recipients in a .PAB file, typically on your own disk. You can copy recipients from other address books into the PAB, and you can have access to the PAB even when your computer is disconnected from the network. A .PAB file can only be accessed by one computer at a time. The PAB is typically configured to be first in the search order of address books.

SCANPST is an executable program used to recover the contents of a damaged PST file.

SEND is an executable program used to exercise the address books, a transport, a store, ... via SMAIL. The source is in mstools\samples\mapi\sendapp.

SMH is a sample message handler provider that can be used to move incoming messages based on some MAPI properties (user configurable). The source is in mstools\samples\mapi\smh

SMPAB is a dynamic library which implements the Sample Address Book. This is an address book provider whose source is distributed as part of the SDK in mstools\samples\mapi\sample.ab. The Sample Address Book provider reads .SAB files, which are produced by the **SAB** executable program. See **ADDRBOOK.TXT** for a sample input file to the SAB executable program.

SMPCLI is a executable program used to demonstrate Simple MAPI calls. It implements a small E-mail client. Its source is in mstools\samples\mapi\smplecli directory.

SMPMS is a dynamic library which implements the Sample Message Store. This is a message store provider whose source is distributed as part of the SDK in mstools\samples\mapi\sample.ms. The Sample Store uses one directory in the file system for each folder, and one file in the file system for each message.

SMPXP is a dynamic library which implements the Sample Transport. This is a transport provider whose source is distributed as part of the SDK in mstools\samples\mapi\sample.xp. The Sample Transport can be configured to store all outgoing messages in a single directory, simply for testing by one user, or can be configured to make a separate copy for each recipient of the message. When configured properly the Sample Transport can be used to implement a low-performance messaging system with multiple users, for testing purposes. The Sample Transport uses TNEF for formatting its message files.

===== 5. Installing OLE Messaging on Windows 95 =====

The MAPI subsystem is installed as part of Windows 95 installation, in particular the "Microsoft Exchange" option (or the setup wizard which is run when you click the "Inbox" icon on the desktop) but OLE Messaging is not included in the initial release of Windows 95.

To install OLE Messaging on Windows 95, run OLEMESS.BAT from the MSTOOLS\BIN subdirectory.

===== 6. Installing the 16-bit MAPI PDK from CD =====

To install the MAPI SDK for use with Windows 3.1 (or WFW 3.11), first install Microsoft Visual C++ 1.5, then install the MAPI SDK by running the setup program found in the MSTOOLS\MAPI\WIN16 directory.

===== 7. Installing the Windows NT MAPI subsystem =====

To install the MAPI subsystem for Windows NT 3.51, run the setup program found in the MSTOOLS\MAPI\WINNT directory.

===== 8. PRODUCT SUPPORT =====

Beta support of MAPI is carried out in the **winext** forum on CompuServe. You will be able to post questions/bugs as well as receive updated files and documentation **INCLUDING additional sample code using C++ (e.g. a C++ Sample Transport and a Sample Client are currently available)**.

To access the support forum, Type "go winext" from any Compuserve prompt. MAPI has it's own private section in the winext forum. If you do not already have access to this private section, log onto Compuserve and send email to "cisbeta@microsoft.com" with your name, company name and your MAPI Beta ID number. You should receive access within 48 hours.

The **winext** forum is divided into different sections where ongoing conversations are carried out

among beta testers. Note that this forum contains both public and private sections. Please limit your discussion of the MAPI beta to the private MAPI section. Many of your operational problems and questions may be covered in forum discussions with other beta testers.

In addition to being able to post questions to other beta testers, Microsoft Support Engineers will also be monitoring the MAPI section.