=================== MAPI RELEASE NOTES ===================
MAPI SDK Version 1.00 <u>Windows NT 4.0</u>

This document contains release notes for the Windows NT SUR version of the MAPI SDK. Changes are relative to the MAPI SDK Beta 5 version, which shipped with Windows 95.

Contents of this file by section:

> **1. System Requirements**
> **2. Known Problems in This Release**
> **3. What's New, Major Changes**
> **4. Component Descriptions**
> **5. Installing OLE Messaging on Windows 95**
> **6. Building 16-bit MAPI Components**
> **7. Installing the MAPI subsystem on Windows NT 3.51**
> **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. Online documentation

For example, if the release notes contradict the online documentation, 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.

Windows NT 4.0

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 NT 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 NT setup, or it can be added to Windows NT by activating the "Inbox" icon on your desktop following Windows NT installation.

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

2.1   Many of the MAPI samples require some sort of installation procedure, in addition to being compiled, in order to be used with MAPI or Microsoft Exchange. Sample service provider DLLs, for instance, must be copied to the system directory of your Windows installation and added to a MAPI profile. Similar steps are necessary for forms and Exchange client extensions. Each sample directory contains a readme file that describes in detail how to prepare the sample for use.

2.2   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%

2.3   Compiling the sample code may result in compiler warnings. These warnings can be safely ignored.

2.4   Current versions of MAPI are interoperable with the Windows 95 version. There is NO interoperability between current versions of MAPI and prerelease Windows 95 versions of MAPI.

2.5   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)

2.6   In OLE Messaging, the Add method of the Recipients collection does not validate its Type parameter. The legal values are 1, 2, or 3 (mapiTo, mapiCc, mapiBcc). Passing other values does not result in a runtime error as expected, but adds an invalid recipient to the collection instead. Also, the following syntax for adding a recipient does not work. Assume that objRecipColl is

already a message's recipient collection object:
```
        set objRecip = objRecipColl.Add
        objRecip.Name = "test name"
        objRecip.Type = mapiTo
        objRecip.Address = "smtp:ThisWillBounce@nowhere.org"
```
Use the following syntax instead:
```
        set objRecip = objRecipColl.Add(Name:="test name",
                Type:=mapiTo, Address:="smtp:ThisWillBounce@nowhere.org")
```

2.7   The parameter validation functions __ValidateParameters and __CPPValidateParameters published with the Win95 version of MAPI do not work on Windows NT for RISC CPUs. The original functions are still supported for Intel CPUs, and new functions have been added for both Intel and RISC CPUs. The docfile sample store, flatfile sample address book, and remote sample transport have changed to use the new functions, and consequently they no longer work with the Win95 MAPI subsystem. Refer to MAPIVAL.H for details.

2.8   By default, the makefiles for most of the MAPI sample programs bind to MSVCRT40.LIB when used with Microsoft Visual C++ 4.0. This yields the smallest executables, but may be inconvenient if you plan to distribute a component widely for testing or deployment, since you will have to distribute MSVCRT40.DLL as well. Changing the library macro $(guilibsdll) to $(guilibsmt) in the makefile will replace MSVCRT40.DLL with statically linked C runtime code from LIBCMT.LIB.

2.9   Microsoft Visual C++ versions 1.5, 2.2, 4.0, and 4.1 all contain MAPI header files that are older than the versions in the current Win32 SDK. Make sure \mstools\include precedes \msvc20\include or \msdev\include in your INCLUDE environment variable.
        The simple MAPI sample application, SMPCLI.EXE, does not build with the MAPI.H which shipped in MSVC 1.5 and other, earlier releases of the simple MAPI header. It uses function prototypes defined in the current version of MAPI.H. To build this sample you must move or rename \msvc\include\mapi.h.

2.10   If you want to build a 16-bit extension to the Exchange client, be aware that the header file EXCHEXT.H requires that you also include COMMCTRL.H. This file can simply be copied from the Win32 SDK include directory into your Win16 build environment. Note that none of the sample Exchange extensions in the SDK currently support Win16.

2.11   The sample message store DOCFILE.MS does not build with Microsoft Visual C++ version 4.1. This is due to a preprocessor bug in that version of the compiler only; earlier versions do not have the bug, and it will be fixed in version 4.2 and later.

2.12   Most MAPI samples link with the C runtime library LIBCMT.LIB. If you install Microsoft Visual C++ with the Minimal option, it does not install this library and you will not be able to build the MAPI samples.

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

The Developer's Guide portion of the online documentation has been enhanced considerably. Microsoft will continue to release updated versions of the online documentation. Information on the updated release of documentation will be posted on Microsoft's FTP site. The address is ftp.microsoft.com/developr/mapi/docs/NEW_DOCS.EXE. OLE Messaging documentation is included in the MAPI SDK online documentation.

Note also that while Microsoft no longer distributes Microsoft Word files for producing printed documentation, the browser used to display the online documentation is now capable of printing entire chapters.

Additional technical articles and abstracts of sample code are now available on the World Wide Web, at http://www.microsoft.com/win32dev/mapi/.

All MAPI samples now build a single Win32 executable for Intel CPUs.It will run on Windows 95 and on Windows NT for Intel CPUs.

The MAPI SDK now includes MAPI NT runtime support for all CPU types supported by Windows NT 3.51: Intel, Alpha, MIPS, and PowerPC. The samples also support all these platforms.

Following are changes made to public MAPI header files between Win95 RTM and Windows NT 4.0 RTM.

EXCHEXT.H
New command ID EECMDID_FilePropertiesRecipients added for recipient properties.

EXCHEXTX.H
New file, lists extensions specific to the enhanced MS Exchange client.

EXCHFORM.H
New file, lists standard verb identifiers for MAPI forms and values for creating the standard verb operation map in the form .CFG file.

MAPI.H
Corrected type definition for MAPIFREEBUFFER.
Removed several flags which had been defined for MAPILogon, but in fact should only be used with MAPILogonEx.

MAPICODE.H
Added two new codes for character set errors.

MAPIDEFS.H
Correct conditional for including WINERROR.H on Win32 systems.
Define MAPI_NT_SERVICE here rather than in MAPIX.H. Its usage has changed: this flag is now passed to service providers on their XXProviderInit entry points, and it is no longer required either on MAPIInitialize or on MAPILogonEx.
Do not define certain simple data types when compiling with MIDL (Microsoft Interface Declaration Language compiler).
Define several values for the new PR_DELIVERY_POINT property.
Add STORE_PUBLIC_FOLDERS and STORE_UNCOMPRESSED_RTF bits for the PR_STORE_SUPPORT_MASK property.
Add MSGFLAG_RN_PENDING and MSGFLAG_NRN_PENDING bits for the PR_MESSAGE_FLAGS property.
Add CLEAR_RN_PENDING and CLEAR_NRN_PENDING flags to IMAPIFolder::SetReadFlags and IMessage::SetReadFlag.
Add DT_FOLDER_SPECIAL value for the PR_DISPLAY_TYPE property.

MAPIFORM.H
Add platform definition for Windows NT SUR.

MAPIGUID.H
Add MUID_PROFILE_INSTANCE. It identifies a profile section with a single property, PR_INSTANCE_KEY, which is guaranteed to be different for every profile created on a particular workstation.

MAPINLS.H
Add MAPI_NOWIDECHAR tag so that Windows NT programs which include MAPI

headers but do not load MAPI32.DLL can turn off redefinition of certain Unicode-related Win32 APIs. MAPI redefines those APIs because they are not implemented on Windows 95.

MAPIOID.H

Add OID_MIMETAG value for the PR_ATTACH_TAG property.

MAPITAGS.H

Add several new properties.
PR_ORIGINAL_SENSITIVITY for propagating the sensitivity of the original message in a thread.
PR_DELEGATION for mapping a Schedule+ 1.0 message property
PR_DELIVERY_POINT for mapping the X.400 MH_T_DELIVERY_POINT attribute
PR_ATTACH_LONG_PATHNAME for specifying the path to an attached file that includes long directory or file names. PR_ATTACH_PATHNAME is restricted to 8.3 directory and file names.
PR_ATTACH_MIME_TAG is a copy of the MIME Content-type header for an attached file or message.
PR_7BIT_DISPLAY_NAME is a copy of PR_DISPLAY_NAME in a more restricted character set, required by some MTAs.

MAPIUTIL.H

Add two new utility functions:
HrDispatchNotifications. This function
ScCreateConversationIndex takes the original value of PR_CONVERSATION_INDEX in cbParent/lpbParent (0/NULL if the message is not a reply or forward), and returns a new value for PR_CONVERSATION_INDEX in lpcbIndex/lppbIndex.
Document the MAPI_MODIFY and STORE_UNCOMPRESSED_RTF flags to WrapCompressedRTFStream().

MAPIVAL.H

Restrict existing definitions of the parameter validation macros ValidateParameters, UlValidateParameters, and CheckParameters to Intel platforms only.
Add new, RISC-capable macros, using new MAPI32.DLL entry points, for RISC platrforms.

MAPIWIN.H

Add Win16 equivalent to the Win32 SetFileTime function.
Add FIsTask, a safe substitute for the Win16 IsTask API which can crash.

MAPIWZ.H

Add Boolean property PR_WIZARD_NO_PST_PAGE to enable providers to suppress this page in the wizard.

MAPIX.H

Move definition of MAPI_NT_SERVICE to MAPIDEFS.H.

===================== 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. Its source is available via FTP, at ftp://ftp.microsoft.com/developr/mapi/samples/mfcapps/mfcapps.zip. Written

in C++.

CMCCLI is a executable program used to demonstrate CMC calls. It implements a small E-mail client. Its source is in the mstools\samples\mapi\cmc.cli directory. Written in C.

CMDEXT is an Exchange extension that adds a command to Exchange's menu. Its source is in mstools\samples\mapi\command.ext. Written in C++.

EVEXT is an Exchange extension that responds to outgoing and incoming mail events by computing a checksum on the message and comparing the value computed on the incoming message with the value stored on the outgoing message. Written in C++.

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.

MAPISP 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. Its source is available via FTP, at ftp://ftp.microsoft.com/developr/mapi/samples/mfcapps/mfcapps.zip. Written in C++.

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 post offices (versions 3.0, 3.2, and 3.5, also Windows for Workgroups and Windows 95 workgroup 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.

PROPSH is an Exchange extension that displays a property sheet with summary properties for certain types of messages containing documents. Its source is in the mstools\samples\mapi\propsh.ext directory. Written in C++.

ROUTE is a small extended MAPI client application. It demonstrates how to store email address properties (in this case, a routing list) in a message in a form that allows the email addresses to be updated by MAPI-aware gateways when the message moves across a gateway into a different messaging domain. Its source is in the mstools\samples\mapi\route.cli directory. Written in C.

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 Simple MAPI.   Its source is available through Compuserve. Written in C++.

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\manager.sh. Written in C.

SMPAB is a dynamic library which implements the Sample Address Book. This is an address book provider that uses .SAB files, which are produced by the **SAB** executable program. See **ADDRBOOK.TXT** for a sample input file to the SAB executable program. The source is in mstools\samples\mapi\flatfile.ab. Written in C.

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

SMPFRM is a simple custom form for use with Microsoft Exchange, the sample routing client, and other clients that support MAPI form interfaces. Its source is in the mstools\samples\mapi\simple.frm directory. Written in C++.

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\docfile.ms. The Sample Store uses one directory in the file system for each folder, and one file in the file system for each message. Written in C.

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\peer.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. Written in C.

XPWDSR is a transport provider that supports the remote client UI for marking and downloading messages based on summary information. It uses server programs **WINDSRV** and **WINDSADM**.

The source is in the mstools\samples\mapi\remote.xp and mstools\samples\mapi\remote.srv directories. Written in C++.

WCHECK is a sample form used for playing checkers by mail. Its source is in mstools\samples\ mapi\checkers.frm. Written in C++.

=============== **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. Building 16-bit MAPI Components** =========================

In this version of the Win32 SDK, you can build Win16 and Win32 targets from a single SDK installation on a system running Windows NT or Windows 95. At the same time, the standalone SDK setup for Win16 systems has been discontinued.

To build 16-bit targets on a Win32 system, in addition to installing the Win32 SDK, you must install Microsoft Visual C++ 1.5. The following notes assume you have installed VC 1.5 in the directory **C:\MSVC15**, and the Win32 SDK in **C:\MSTOOLS**:
1. Modify your PATH environment variable to include the 16-bit tools and 32-bit tools, in that order.
        set PATH=c:\msvc15\bin;c:\mstools\bin;...
2. Modify your INCLUDE environment variable to include the 16-bit headers and the MAPI headers, in that order.
        set INCLUDE=c:\msvc15\include;c:\msvc15\mfc\include;c:\mstools\include;
        c:\mstools\mfc\include;...
3. Modify your LIB environment variable to include the 16-bit libraries and MAPI.LIB, in that order.
        set LIB=c:\msvc15\lib;c:\msvc15\mfc\lib;c:\mstools\lib;...
4. Build any samples of interest using MAKEFILE.16 instead of MAKEFILE. The former is included for all samples that can build 16-bit targets. All 16-bit makefiles use a common include file, \mstools\samples\mapi\common\WIN16.MAK; if you need global changes to the build process, do them in WIN16.MAK.

Note: this MAPI subsystem is supplied for development purposes only; it contains components that are not redistributable. Redistributable MAPI components for Windows 3.1 are available from Microsoft's FTP site, at ftp.microsoft.com/developr/mapi/WMS_RDST/.

=============== **7. Installing the MAPI subsystem on Windows NT 3.51** ===========

To install the MAPI subsystem for Windows NT 3.51, run the setup program found in the subdirectory of MSTOOLS\MAPI that is appropriate for your CPU type:
        MSTOOLS\MAPI\i386
        MSTOOLS\MAPI\ALPHA
        MSTOOLS\MAPI\MIPS
        MSTOOLS\MAPI\PPC

Note: this MAPI subsystem does not run on any version of Windows NT other than 3.51. It cannot run on earlier versions, and on later versions (Windows NT SUR beta 1 and later) the MAPI subsystem is included as an operating system component.

Note: this MAPI subsystem is supplied for development purposes only; it contains components

that are not redistributable. Redistributable MAPI components for Windows NT 3.51 are available from Microsoft's FTP site, at ftp.microsoft.com/developr/mapi/WMS_RDST/. The Intel, Alpha, Power PC, and MIPS versions are all shipping code.

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

MAPI is no longer in beta test on any platform, and beta support has been withdrawn. Support for MAPI development efforts is available from Microsoft AnswerPoint through normal support channels. In addition, there are numerous online resources where you can find information and answers to your MAPI questions.

Online documentation, redistributable MAPI binaries for Windows NT and Windows 3.x, and sample source code can be downloaded via FTP from
        ftp.microsoft.com/developr/mapi/

Technical articles, overviews of sample source code, and other materials can be found on the World Wide Web at
        http://www.microsoft.com/win32dev/mapi/

The Internet news forum **microsoft.public.win32.programmer.mapi** carries discussions and Q&A with MAPI developers worldwide. This forum is sponsored by Microsoft, but is not monitored by Microsoft support engineers.

There is an Internet mailing list which carries considerable discussion among MAPI developers. Send e-mail to LISTSERV@SEARN.SUNET.SE with "subscribe MAPI-L" in the message body to join the list, or put "help" in the message body to receive more information.