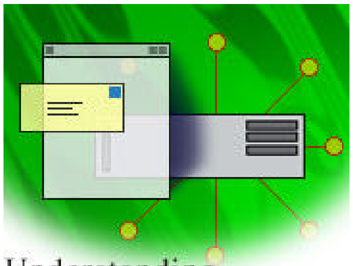
Notes.net

Iris Today



Understanding iNotes Access for Microsoft Outlook

by Jane Hosie-Bounar
(with Margaret Rora)

Level: Advanced
Works with: Domino 5.0
Updated: 06/01/2001

As Lotus unveils the latest updates and improvements to Domino server technology, more administrators find that they want to take advantage of its powerful database and security features, as well as its history of reliable performance. Some of these administrators have already deployed the Microsoft Exchange server, only to discover after the fact that the Domino server provides richer functions, more flexibility, and better network performance.

Because their end users have learned to use Microsoft Outlook and have developed business practices around its features, server administrators have been reluctant to upgrade to Domino either because they fear the loss of some Outlook features in the transition or because of the necessity of migrating Outlook users to Lotus Notes.

With these challenges in mind, Lotus developed iNotes Access for Microsoft Outlook—a deceptively simple solution that lets administrators migrate their mail databases from the Exchange server to the Domino server without impacting Outlook 98 and Outlook 2000 user productivity. End users can continue to use Outlook as they always have, while administrators can enhance and simplify server maintenance and performance using the power of the Domino backend.

This article focuses on the more technical aspects of iNotes Access for Microsoft Outlook. It includes the following sections:

- The building blocks
- Lotus Notes MAPI Services
- MAPI and the Domino server
- The installation process
- Windows messaging subsystem and profiles
- MAPI objects supported by the Lotus Notes MAPI Services

This article assumes an understanding of server administration and server technology.

The building blocks

There are three pieces of the iNotes Access for Microsoft Outlook puzzle that you need to understand: Domino Offline Services (DOLS), the Extended Mail Template, and Messaging Application Programming Interface (MAPI) Services. This section provides a brief overview of the first two pieces. The rest of the article focuses on the Lotus Notes MAPI Services piece of iNotes Access for Microsoft Outlook, as well as the technical details of the installation process.

DOLS

DOLS enables users to have offline access to Domino applications. As a result, an end user's experience using an application offline can be essentially the same as an end user's experience using the same application online. iNotes Access for Microsoft Outlook uses DOLS to allow Outlook users to work offline and later, synchronize their work with the Domino server.

When you install iNotes Access for Microsoft Outlook, you do so using

DOLS. There are numerous files that any DOLS-enabled application needs to download. These files are called “services”; they are selected by default in the Offline Configuration section of the DOLS-enabled database. In the case of iNotes Access for Microsoft Outlook, in addition to the basic services, the list of “Domino services to install offline” includes MAPI enablement.

Note: For more information on DOLS, see the **Domino Offline Services** page of the Lotus Developer Network. For an online demonstration of DOLS, see the **DOLS Demo** on **Notes.net**.

Extended Mail Template

The mail database used with iNotes Access for Microsoft Outlook requires the design elements of the Extended Mail template, mail50ex.ntf, also called Extended Mail (R5.0). This template contains additional views that allow Outlook users to display the Outlook Calendar and Task folders. There are different scenarios for installing the template, depending on a user’s situation:

- The most likely scenario is that the administrator is migrating an existing Exchange mail file. In this case, the migration tool lets you specify which template to use.
- If you are creating a new user with a new mail file, you can specify the template as part of the registration process.
- If there is an existing Notes mail file based on a different template, then you will either need to replace the design on behalf of the end user, using the Notes client or the Domino Administrator client, or the end user will need to perform this step via the Notes client.

Note: If the user has a Notes mail file based on a different template, that user is probably in a mixed environment, where there is already access to a Notes client. This scenario would be true for users of the previous level of MAPI support, which required the Notes client. In this case, the end user can replace the design template using the Notes client, uninstall the Notes client, and then install iNotes Access for Microsoft Outlook.

Lotus Notes MAPI Services

To develop iNotes Access for Microsoft Outlook, Lotus used a middleware technology called Messaging Application Programming Interface, also known as MAPI. Many messaging applications on the market today, as well as many other applications that can send their files through messaging applications, use MAPI as their common interface. The Lotus Notes/Domino version of MAPI is called Lotus Notes MAPI Services.

Note: Keep in mind that MAPI is not a versioned protocol like SMTP or IMAP. Changes to MAPI are not controlled by external standards groups. MAPI is also not a data format that lets you include new data elements without breaking existing applications, such as HTML. Compatibility and interoperability between different versions of MAPI-enabled applications must be verified on a case-by-case basis.

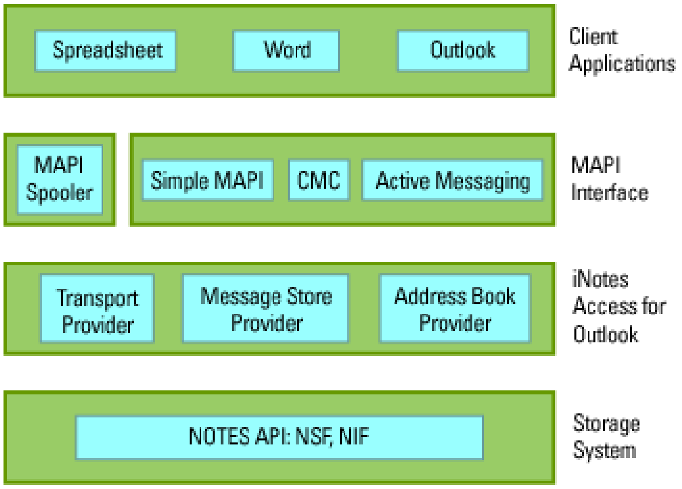
When it was first introduced with Lotus Notes Release 4.0, Lotus Notes MAPI Services supported only mail message delivery and personal address book access for users running Microsoft Outlook on a Domino server. It also required that the Outlook end user have the Notes client installed. This functionality was sufficient for users who merely used Outlook to compose and read messages; however, those users who wanted to use the calendaring and scheduling features offered by Outlook were unable to do so.

Enter iNotes Access for Microsoft Outlook, delivered with Notes Release 5.0.5, which includes both calendaring and scheduling capabilities, and simplified installation and upgrades. iNotes Access for Microsoft Outlook uses Domino Offlines Services (DOLS) to distribute and maintain the code that is needed by the Outlook client, as well as to provide offline support. Notes Release 5.0.5 contains all of the other components required to run Outlook with the Domino server as the messaging system. It also contains the components necessary for users of Notes and users of Microsoft Outlook to communicate with each other

Note: Installation instructions for Lotus Notes Release 5.0.5 and iNotes Access for Microsoft Outlook can be found in the white paper entitled *iNotes Access for Microsoft Outlook Administration*, which is found on the iNotes Access for Microsoft Outlook page off the **iNotes** page of the Lotus Developer Network. This white paper also includes information about Outlook features that are not supported. Also be aware that with Domino Release 5.0.7, delegation and password management was greatly improved; these improvements are described in the **5.0.7 Release Notes**.

MAPI and the Domino server

The following discussion of the MAPI architecture should help you understand how applications interact with the Domino server via the MAPI interface. The diagram below depicts the major components of the MAPI architecture:



Client applications

The top layer of the MAPI architecture includes the MAPI client applications that use MAPI interfaces. These client applications are classified according to the type of message processing they perform: messaging-aware, messaging-enabled, and messaging-based.

Most applications that use MAPI are messaging-enabled and messaging-aware. They are capable of requesting delivery of a message with the application data as the content of the message. Some examples of messaging-enabled and messaging-aware applications are word processors, spreadsheet applications, and document scanners. The File menu in any of these application types usually includes a Send menu option that causes the application to initiate a MAPI request to send a document. This kind of application might have its own user interface to compose the message or may use a messaging-based application to compose and deliver the message. Messaging-enabled applications incorporate messaging as an added feature. For example, Windows Explorer can mail a document, but that is not primarily what Windows Explorer is for.

Messaging-based applications have the ability to exchange information in many formats, as well as the ability to store and organize that information. Electronic mail, scheduling, and document management applications are all messaging-based applications. Obviously, Microsoft Outlook also fits into this category.

MAPI interfaces
C++ programmers, C programmers, and Visual Basic programmers can use the simple MAPI interface to write messaging-enabled applications. C++ or C programmers can also use the Common Messaging Calls (CMC) interface. Active Messaging is an object-based client interface that can also be used with Visual Basic or C++.

These MAPI interfaces provide the fundamental functionality to connect to a messaging system, compose messages, and deliver messages. Although these interfaces are powerful enough for simple messaging operations, they are rather limited when it comes to dealing with message organization and flow control.

The MAPI programming interface, once called the MAPI Extensions, is an object-based interface. It is the most direct way to access those features provided by the MAPI service providers. It allows the programmer to manipulate messages, folders, addresses, or any other type of object implemented by the MAPI service provider.

Although MAPI functions for managing simple e-mail messaging are well documented by Microsoft, any MAPI application developer and any MAPI service provider can also define their own private object properties to implement more complex messaging-based applications. It was this ability that allowed Lotus to develop iNotes Access for Microsoft Outlook.

iNotes Access for Microsoft Outlook: a MAPI service provider
As a MAPI service provider, iNotes Access for Microsoft Outlook stores and manages information in an Outlook user's mail file and address books. The message store provider (see the **diagram** above) stores and retrieves messages from the user's mail database. (To understand how iNotes includes calendaring and scheduling features as well as messaging, it's helpful to think of a calendar entry as another class of message.) The address book provider stores and retrieves recipient information from the public name and address book and from the personal address book. The transport provider facilitates message transmission.

The Lotus Notes address book provider and the Lotus Notes message transport provider translate MAPI properties into Notes items. In essence, the mechanism used to store message properties is equivalent to the mechanism used to store Notes items in Notes. This is why a big part of the code in iNotes Access for Microsoft Outlook lies in the implementation of the management of properties. When a user creates a message using Outlook, the Lotus Notes message store provider is called by MAPI to set the properties of the new message. At this time, we translate MAPI properties to Notes items. The opposite happens when a user opens an existing message; we have to map Notes items to MAPI properties.

When you look at the MAPI interface, you can see that a mail message object and an address book object encapsulate properties specific to those objects. These properties are represented with a value pair that contains a tag and its value. For example, to represent a street address entry in the Address Book, the following properties are mapped:

- MAPI property: PR_HOME_ADDRESS_STREET
- Notes item: "StreetAddress"
- Value: 444 High Street

Some properties can be easily mapped and used by both systems. Some properties are used by only one system. Finally, some properties are used by both systems but stored in completely different ways. Since it was our goal to provide full access between users of an Outlook client and a Notes client, it was necessary to also do the following:

- Map all equivalent properties.
- Construct the properties required by Notes that are not provided by Outlook.
- Construct the properties required by Outlook that are not required by Notes.

We also chose to store all data in Domino in the format used by the Notes client to provide the additional ability for users to easily upgrade from Outlook to Notes at some future point without any change to their stored messages. While most items are stored in a format understandable by the Notes client, we also have to store items specific to Outlook. As an illustration, if you look at the document properties of a mail message created by an Outlook user, you will see items with a name that starts with “$Mapi…” followed by a long number. This number is the combination of a property tag and a unique ID.

The MAPI spooler

The MAPI spooler is a separate process in the system, responsible for a number of tasks such as queuing messages for delivery, informing a client application when a new message has been delivered, and generating reports. The spooler is tightly coupled with the transport provider. For example, when a user receives a message, the transport provider notifies the spooler that a new message has arrived. This notification goes to the client application, causing a refresh of the message and allowing the user to read new messages.

iNotes Access for Microsoft Outlook installation process

For a more complete understanding of how iNotes Access for Microsoft Outlook works, it's helpful to understand exactly what happens during the installation process. The installation of iNotes Access for Microsoft Outlook involves the following steps:

1. From a browser, the user enters the URL of the mail file on the server.

2. Once authenticated, the user follows the directions on the Web page and starts the installation.

Install downloads the basic, required DOLS services, including the Sync Manager (which synchronizes online and offline databases), to the client. Because the /inotes page is specified in the URL, the “MAPI enablement” service is also downloaded. Once the download is complete, the Lotus iNotes Sync Manager replicates the mail file from the server to the client.

3. After this replication, the install process starts nmInstall.exe to configure the MAPI service and create the necessary profiles.

Because the iNotes Access for Microsoft Outlook installation uses the /inotes page, an agent detects whether or not the mail file has an offline configuration profile. For a first time installation, the profile doesn’t exist, so the agent creates a profile that has the necessary services selected. In this case, whenever there is a subscription refresh, the profile exists and is correct. However, in the case where a user has edited the configuration profile with the Notes client, install will use whatever services have been explicitly selected, which may or may not be the correct services.

Windows messaging subsystem and profiles

MAPI programs like Outlook, Word, and Internet Explorer use profiles to define how to access available message storage and submission services. Using multiple profiles lets you define services and configuration information specific to particular environments. A profile is a repository of information that describes each of these services; it includes the service name, the location of the DLL file that facilitates the program interface, and other configuration information specific to the provider service.

The iNotes Access for Microsoft Outlook user needs a profile in order to use Lotus Notes MAPI Services. iNotes creates two profiles automatically, without direct user intervention, during the installation: the Local Mail profile and the Mail on *server-name* profile. The Local Mail profile uses a local replica of the user's mail file, and the Offline Address Book. This profile can be used whether or not the user is currently connected to the network. When connected to the network, the Lotus iNotes Sync Manager replicates the local mail file with the mail file that resides on the server. The Mail on *server-name* profile uses the copy of the user's mail file that resides on the server, and has access to the server's Address Book as well as the Offline Address Book. This profile can only be used when the user is connected to the Domino server.

When the Web-based (DOLS) iNotes installation creates these profiles, it uses nmInstall.exe with five parameters—each of which represents important information about the environment.

Most administrators will never use nmInstall.exe manually. However, administrators might be interested to know exactly what goes on when this program runs. Should you, as an administrator, ever encounter profile problems, you can use this information to help solve them. For example:

- The most common installation problem is usually related to the location of mapisvc.inf on the client machine, or other symptoms of conflicts between multiple MAPI-enabled applications that are installed on the same machine. There are **5.0.5 Release Notes** on this topic, and it is also covered in the **Domino R5 Administration Help**.
- Another problem is that Outlook may not have been properly configured after installation, and the install fails because it can't find where to write the profiles.
- During the development of iNotes Access for Microsoft Outlook, nmInstall was also used in a testing environment to install multiple users on the same machine. nmInstall was needed because the installation of a new user will overwrite the default profiles created for the previous user. Although this is not a suggested or supported use of nmInstall, an administrator could use it in this way to test multiple users on a single machine prior to deployment.

To run nmInstall, type the following command from the MS DOS command prompt:

```
nmInstall inipath homeserver password serverpath localpath
```

where:

- *inipath* specifies the full path to the notes.ini file, including the file name. When nmInstall is run automatically during installation, this parameter is the mapinotes.ini file in the user's data directory. When it is run manually, it is the notes.ini file for the user.
- *homeserver* specifies the name of the Domino server that contains the mail database. For example, if the server is *servername/domainname*, then if you are running the command in the same domain, you would just enter *servername*. If you were running it in a different domain, you would use *servername/domainname.*
- *password* specifies the password.
- *serverpath* specifies the path to the server mail database relative to the Notes data directory. You also need to include the mail database file name.

- *localpath* specifies the path to the local mail database relative to the iNotes directory. You also need to include the mail database file name.

Note: Do not use commas to separate the arguments; only spaces are allowed. For the password (argument 3), you can use an empty string ("") in the command line, but this will cause the program to prompt you for the password.

The program nmInstall performs six major steps to configure iNotes Access for Microsoft Outlook. The following is a brief discussion of these steps. Steps 1 and 5 are related to the iNotes environment; steps 2, 3, 4, and 6 are related to MAPI.

Step 1: Verify the password
This step verifies that the user is the correct user for the MAPI profile. nmInstall checks the password by using it to open the remote mail database. If the open operation fails, the program stops. In other words, if the server is down, if the network is inaccessible, or if the mail file is missing or damaged, you cannot run nmInstall.

Step 2: Configure the message services configuration file, MAPISVC.INF
The message services configuration file contains information about the various MAPI service providers. It looks like a conventional Windows INI file. It contains a [Services] section that enumerates all registered services. For each registered service, there is a section that enumerates those properties required by MAPI to successfully initialize the service. Here is a sample of what might be in the file MAPISVC.INF:

```
[NOTES]
PR_SERVICE_DLL_NAME=nwnsp.dll   ;this is the base dll
PR_SERVICE_SUPPORT_FILES=nwnsp.dll   ;the real name is
nwnsp32.dll
PR_SERVICE_DELETE_FILES=nwnsp.dll
PR_SERVICE_ENTRY_NAME=ServiceEntry  ;this is the entry point in the
dll
WIZARD_ENTRY_NAME=ServiceWizard   ;this is the entry point for the
Service Wizard
PR_RESOURCE_FLAGS=SERVICE_SINGLE_COPY|
SERVICE_PRIMARY_IDENTITY
Providers=NAB,NMS,NXP   ;this is the list of providers that we support

[NAB]   ;the Name and Address Book service provider
PR_PROVIDER_DLL_NAME=nwnsp.dll
PR_RESOURCE_TYPE=MAPI_AB_PROVIDER
PR_DISPLAY_NAME=Lotus Notes Address Book
PR_PROVIDER_DISPLAY=Lotus Notes Address Book
PR_RESOURCE_FLAGS=STATUS_PRIMARY_IDENTITY

[NMS]   ;the message store service provider
PR_PROVIDER_DLL_NAME=nwnsp.dll
PR_RESOURCE_TYPE=MAPI_STORE_PROVIDER
PR_RESOURCE_FLAGS=STATUS_PRIMARY_IDENTITY|
STATUS_DEFAULT_STORE
PR_PROVIDER_DISPLAY=Lotus Notes Message Store
PR_DISPLAY_NAME=Lotus Notes Message Store
PR_MDB_PROVIDER=395d475fecf1cd1193dc5aab3C478538

[NXP]   ;the transport provider
PR_PROVIDER_DLL_NAME=nwnsp.dll
PR_RESOURCE_TYPE=MAPI_TRANSPORT_PROVIDER
PR_RESOURCE_FLAGS=STATUS_PRIMARY_IDENTITY
PR_PROVIDER_DISPLAY=Lotus Notes Transport
```

```
PR_DISPLAY_NAME=Lotus Notes Transport
```

Notice that the names of the keys are the same as the names of the MAPI properties. You can find these properties documented in the *MSDN Platform SDK* available from Microsoft.

Step 3: Register the message store class and the interface for free busy support
Registering the message store class is necessary for MAPI to be able to instantiate our message store objects. The two items that are registered are:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Current Version\SharedDLLS]
nwnsp32.dll
```

and

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\9.0\Outlook\SchedulingInformation\FreeBusySupport]
"NOTES"="{B2892222-7C18-11D3-98DC-00A0C9A342AA}"
```

Outlook obtains this CLSID from the registry. With this information, Outlook also obtains the path to the Lotus message store DLL – nmsp32.dll, which contains the interface for free busy support.

Step 4: Copy the file nwnsp32.dll to the Windows/System directory
This DLL file must be in the Windows/System directory. It is the first DLL that MAPI loads when the user selects a profile that contains the Lotus Notes MAPI Services. MAPI loads this DLL file to get the interfaces of the Lotus Notes service providers. It is discussed in more detail in the **Base DLL** section.

Step 5: Create the Location documents in the user's Personal Address Book
The installation procedure for iNotes Access for Microsoft Outlook creates the empty Personal Address Book and downloads a replica of the user mail database (including its current data). nmInstall creates a Location document in the Personal Address Book for access to the Local Mail database and another Location document for access to the Mail on *server-name* database. The names of the Location documents are the same as the profile names.

If the database is replicated to other servers within a cluster, nmInstall creates a Location document for each replica of the database.

Step 6: Create the profiles
For each Location created in step 5, nmInstall creates a MAPI profile. The profiles are configured using the MAPI IMsgServiceAdmin and IProfAdmin interfaces. A profile is a series of Windows registry entries that are used by the MAPI interface layer and by the provider interface. If interested, you may find these registry entries under HKEY_CURRENT_USER/Software/Microsoft/WindowsNT/Current Version/Windows Messaging Subsystem/Profiles/.

Here you will find a key for each profile created by nmInsall. Each key contains a number of unique IDs that describe a set of properties for that profile. Open the key that contains the Notes UID {704808520df8ce11a2e102608cda47ee} and you will be in familiar terrain. You will recognize some of the data values like the path to the NOTES.INI file, the path to the mail database, the service name, and others. In fact, this key shows you the configuration data for the Lotus Notes MAPI message store service. Most of these values are exactly the same as, or are derived from, the values of the arguments used when you run nmInstall.exe.

Note: Keep in mind that this information is provided for troubleshooting. In general, it is a good idea not to alter the registry in any way. However, if something isn’t working correctly, sometimes the registry contents can provide a clue about what went wrong. In general, the end user should perform all profile administration using the Control panel tool (accessed by clicking Show Profiles in the Properties panel) or the Services tool (accessed by clicking Services from the Outlook Tools menu).

We expect that most users will be using iNotes Access for Microsoft Outlook with only two profiles: one profile to access the local mail file and one profile to access the server-based mail file. However, they might also need additional profiles, such as a profile to access each cluster replica and a profile to access each mail file belonging to another person or group to which the user has been granted access. We also anticipate that there will be a small percentage of users who want to have the flexibility of changing the profile configuration or adding profiles.

MAPI objects supported by the Lotus Notes MAPI Services

Each of the services that iNotes Access for Microsoft Outlook provides are contained in a set of Dynamic Link Libraries (DLLs) that correspond to the type of MAPI objects that it supports. Here is the list of libraries and a description of the services they provide:

- nwnsp32.dll is the base DLL. This is the DLL that MAPI loads to get the interfaces of the Lotus Notes service providers.
- nabp32.dll contains the interfaces for the address book objects.
- nmsp32.dll contains the interfaces for the storage objects. This is where most of the work to read and write messages takes place.
- nxpp32.dll contains the interfaces for the transport provider.

The file nmStrings.dll does not provide MAPI services. This DLL contains all the translatable strings and was developed for ease of localization. Each locale has a different nmStrings.dll.

Base DLL

nwnsp32.dll is the wrapper for the initialization of all the providers. If you look at the file mapisvc.inf, you will see that only one DLL is registered—nwnsp.dll. To understand the role of this DLL, it might help to step through the logon process that occurs when the Outlook user selects a profile to log on.

MAPI needs to create a session object for the user, which is controlled by the address book provider. This means that MAPI must get access to the address book provider interface before it can request a session. Here is how this works:

- MAPI loads the library nwnsp32.dll and calls the function ABProviderInit. This function should return a success code, the pointer to the address book provider interface, and the version number.
- The function ABProviderInit loads the module nabp32.dll and calls the function WrapABProviderInit—a private function responsible for instantiating the interface and performing other initialization tasks.
- MAPI uses the provider interface to query the specific interfaces required to create a session and log the user on to Domino.

For each provider, there is a corresponding initialization function that MAPI calls to get to the interface of the provider.

Nwnsp32.dll also provides the entry points to the interfaces of the Administration Services that allow a user to configure the MAPI profiles.

Message store provider
The module nmsp32.dll contains the message store provider, which is responsible for composing, sending, and receiving messages.

The Lotus Mail MAPI message store provider is a tightly coupled message store. This means that this provider takes much of the responsibility of the MAPI spooler, for example, sending and receiving messages. For more information, see the **Transport provider** section, below.

The message store provider implementation uses Domino to store the messages. This means that all messages are stored as notes (in Notes format) in the user's mail database. Each note contains items that correspond to the MAPI properties of the message.

The objects stored in the mail database are folders, messages, and attachments. There is no difference between the basic mechanism used to store a mail message and the basic mechanism used to store a calendar entry: they are simply messages of a different class. In fact, it is the class property that causes Outlook to select the proper form and icon for display because iNotes maps each class to a form name. The MAPI message classes supported by the Lotus Mail message store provider are:

Appointments

IPM.Appointment	*Meeting*
IPM.Schedule.Meeting.Request	*Meeting request*
IPM.Schedule.Meeting.Resp.Pos	*Invitation acceptance*
IPM.Schedule.Meeting.Resp.Tent	*Tentative invitation acceptance*
IPM.Schedule.Meeting.Resp.Neg	*Invitation refusal*
IPM.Schedule.Meeting.Canceled	*Canceled meeting*

Journal
IPF.Journal

Contacts
IPM.Contact

Tasks
IPM.Task
IPM.Activity

Notes
IPM.StickyNote

It is important to note that iNotes Access for Microsoft Outlook does not provide Notes forms for the Journal, Contacts, or Sticky Notes classes, so a user migrating to a Notes client would not be able to use those forms by default. However, these classes work as they should for a user using Outlook as a mail client.

Transport provider
The module nxpp32.dll contains the transport provider. This provider works with the MAPI spooler, which is a separate process responsible for receiving and sending messages. In the **diagram** above, the spooler is shown at the MAPI interface level, with the providers shown at the next level down.

When the Outlook client needs to use the services provided by the transport provider, it always accesses those services through the MAPI spooler. The MAPI spooler is a separate process running on a separate thread, so it is running asynchronously.

When the Outlook client needs to access the address book provider or the message store provider, it uses the other pieces of the interface shown in the diagram.

To provide its services, the transport provider interacts with the message store provider to gain access to the messages. There are two main uses of the transport provider within iNotes Access for Microsoft Outlook: communicating with the MAPI spooler and using the Outbox.

Communicating with the MAPI spooler

The transport provider notifies the Outlook client of alarms, such as meetings and to-dos, and also notifies the Outlook client when new messages arrive. On request from the MAPI spooler, the transport provider determines if there are any alarms or new messages and sends this information back to the MAPI spooler, which notifies the Outlook client.

Using the Outbox

The second use of the transport provider is message delivery. When the Outlook client is using its local mail file (that is, the Local Mail profile), the message store provider places the message in the Outbox for the MAPI spooler to deliver. The MAPI spooler uses the transport provider to accomplish this, but obviously the transport provider can't deliver the message until the client is connected to the network.

Note: In the case where the Outlook client is using its mail file on the server (that is, the Mail on *servername* profile), the message store provider delivers the message directly to the server. The message does not go to the client's Outbox, and the MAPI spooler and transport provider are not involved.

There are two cases that cause message delivery to happen once the message is in the Outbox.

In the first case, the client is using the Local Mail profile, is connected to the network, and clicks the Send/Receive button or exits Outlook with mail pending in the Outbox. In this case, the request goes to the MAPI spooler, which accesses the transport provider, which initiates the iNotes Sync Manager to replicate the mail file with the server, causing the message to be delivered. (In addition, the Sync Manager replicates all other changes that have occurred in the local mail file. There is no method of delivering just the mail in the Outbox without replicating all changes that have occurred to the mail file.)

In the second case, the client is using the Local Mail profile, is not connected to the network, and exits Outlook with mail pending in the Outbox. The message cannot be delivered, since there is no connection to the network. The user can connect to the network at a later time and do one of two things:

- Restart Outlook and use the Send/Receive button as above.
- Use the iNotes Sync Manager to initiate replication, causing the message to be delivered. If the user initiates the iNotes Sync Manager, then the MAPI spooler and the transport provider are not at all involved in the message delivery.

Address book

The module nabp32.dll contains the Lotus Notes address book service provider. This service provider is responsible for the logon/logoff functions, and for creating, modifying, deleting, and accessing the address book entries for users and distribution lists.

Each address book entry contains corresponding MAPI properties. iNotes Access for Microsoft Outlook supports all of the properties required to address a mail recipient, as well as all the properties that are visible to the user in the Outlook address book user interface.

It is important to understand that Contacts in Outlook are not address book entries. The Contacts folder in Outlook is used to store a specific type of message—IPM.Contact—that type of message is used in the Contact Management application within Outlook. The only relation to the address book is at the user interface level, where Outlook provides the option to include the Contacts folder in the list of name and address books. An Outlook user may create multiple Contacts folders. Since Contact entries are messages, these are stored in the user mail database and are handled by the message store provider, not the address book provider.

Conclusion

The implementation of iNotes Access for Microsoft Outlook is based on the MAPI architecture. The three MAPI service providers—address book, message store, and transport—offer all of the functions required to store different classes of messages in a Domino database. As a result, iNotes Access for Microsoft Outlook offers a hassle-free migration path with no additional training required. Furthermore, Outlook users can work together with Notes users without difficulty, and any Outlook user can upgrade to Notes without any change to their mail or local user directory.

ABOUT MAGARET RORA

Margaret Rora is development manager for iNotes Access for Microsoft Outlook, Domino Messaging, and Domino Directory, and QE manager for Domino Messaging, Directory, and Networking at Iris.