

# [MS-OXOABKT]: Address Book User Interface Templates Protocol Specification

## Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the protocol documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting [protocol@microsoft.com](mailto:protocol@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** This protocol documentation is intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability.
Microsoft Corporation	June 27, 2008	1.0	Initial Release.
Microsoft Corporation	August 6, 2008	1.01	Revised and edited technical content.
Microsoft Corporation	September 3, 2008	1.02	Revised and edited technical content.

Microsoft Corporation	December 3, 2008	1.03	Revised and edited technical content.
Microsoft Corporation	March 4, 2009	1.04	Revised and edited technical content.

# Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
1.1	Glossary.....	5
1.2	References.....	6
1.2.1	Normative References.....	6
1.2.2	Informative References.....	7
1.3	Protocol Overview.....	7
1.3.1	Creation of New E-Mail Addresses.....	7
1.3.2	Display and Updating of Data.....	8
1.3.3	Collection of Search Data.....	8
1.4	Relationship to Other Protocols.....	8
1.5	Prerequisites/Preconditions.....	8
1.6	Applicability Statement.....	8
1.7	Versioning and Capability Negotiation.....	8
1.8	Vendor-Extensible Fields.....	9
1.9	Standards Assignments.....	9
<b>2</b>	<b>Messages.....</b>	<b>9</b>
2.1	Transport.....	9
2.2	Message Syntax.....	9
2.2.1	NspiGetSpecialTable PropertyRowSet_r format.....	9
2.2.2	NspiGetTemplateInfo PropertyRow_r format.....	10
2.2.2.1	Template Format.....	10
2.2.2.1.1	TRowSet Structure.....	11
2.2.2.1.2	TRow Structure.....	11
2.2.2.1.3	Buffer Format of the CNTRL Structure.....	15
2.2.2.2	Script Format.....	18
<b>3</b>	<b>Protocol Details.....</b>	<b>23</b>
3.1	Client Details.....	23
3.1.1	Abstract Data Model.....	23
3.1.1.1	Dialog Object.....	23
3.1.1.2	Control objects.....	23
3.1.1.3	Address Creation Template Table.....	24
3.1.2	Timers.....	24
3.1.3	Initialization.....	24
3.1.4	Higher-Layer Triggered Events.....	25
3.1.4.1	Creating a new E-Mail Address for a Supported Address Type.....	25
3.1.4.2	Displaying Information about an Address Book Object.....	25
3.1.4.3	Collecting Data to Search the Address Book.....	26
3.1.5	Message Processing Events and Sequencing Rules.....	26
3.1.5.1	Results of NspiGetSpecialTable Call to Retrieve the Address Creation Table	26

3.1.5.2	Results of NspiGetTemplateInfo Call to Retrieve the Creation Template	27
3.1.5.3	Results of NspiGetTemplateInfo Call to Retrieve the Display Template	27
3.1.5.4	Results of NspiGetTemplateInfo Call to Retrieve the Search Template	27
3.1.6	Timer Events .....	27
3.1.7	Other Local Events.....	27
3.2	Server Details.....	28
3.2.1	Abstract Data Model .....	28
3.2.1.1	Template Objects .....	28
3.2.1.2	Table of Supported Address Types and Name of Template to Use to Create Them .....	28
3.2.2	Timers .....	28
3.2.3	Initialization.....	28
3.2.4	Higher-Layer Triggered Events.....	28
3.2.5	Message Processing Events and Sequencing Rules .....	28
3.2.5.1	NspiGetSpecialTable Call from Client .....	28
3.2.5.2	NspiGetTemplateInfo Call from Client .....	29
3.2.6	Timer Events .....	29
3.2.7	Other Local Events.....	29
<b>4</b>	<b><i>Protocol Examples</i></b> .....	<b>29</b>
4.1	Creating a New E-Mail Address for a Supported Address Type .....	29
4.2	Retrieving a Mail User's Template .....	38
<b>5</b>	<b><i>Security</i></b> .....	<b>62</b>
5.1	Security Considerations for Implementers .....	62
5.2	Index of Security Parameters .....	62
<b>6</b>	<b><i>Appendix A: Office/Exchange Behavior</i></b> .....	<b>62</b>
	<b><i>Index</i></b> .....	<b>64</b>

# 1 Introduction

The Address Book User Interface Templates protocol is an extension of the Name Service Provider Interface (NSPI) protocol, as specified in [MS-NSPI]. The Address Book User Interface Templates protocol specifies the following:

- A server-provided **template** for creating specific, single-use e-mail addresses.
- A server-provided layout specification that the client can use for displaying **Address Book object** information.

## 1.1 Glossary

The following terms are defined in [MS-OXGLOS]:

**address book**  
**Address Book object**  
**Address Creation table**  
**address creation template**  
**Augmented Backus-Naur Form (ABNF)**  
**button control**  
**code page**  
**display template**  
**distinguished name (DN)**  
**double-byte character set (DBCS)**  
**mail user**  
**non-Unicode**  
**Name Service Provider Interface (NSPI)**  
**Permanent Entry ID**  
**property**  
**remote procedure call (RPC)**  
**Simple Mail Transfer Protocol (SMTP)**  
**template**

The following data types are defined in [MS-DTYP]:

**Boolean**  
**BYTE**  
**ULONG**

The following terms are specific to this document:

**check box control:** A **dialog control** that displays a static string and a box that can be

checked and unchecked to indicate that an option is selected.

**creation template:** See **Address Creation template**.

**dialog control:** A graphical user interface element that allows a client to interact with a user by displaying information to the user and getting input from the user.

**drop-down list box control:** A **dialog control** that contains a list of possible options of which only the currently selected one is shown. A button on the end of the control can be clicked to display the entire list of available options.

**edit control:** A **dialog control** that displays an editable string to the user.

**group box control:** A **dialog control** that contains other **dialog controls** and around which a box is shown with a static string that labels the box.

**label control:** A **dialog control** that displays a static string to the user.

**list box control:** A **dialog control** that contains a list of possible options of which one can be selected by the user.

**page control:** A **dialog control** that contains other controls, groups them together, and displays a static string that describes them at a tab at the top of the group. These controls can be placed on top of each other and the control for which the tab is selected has its set of controls shown.

**search template:** A **template** that describes a dialog for the user to specify criteria for searching **Address Book objects** that meet the criteria.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

### 1.2.1 Normative References

[MS-DTYP] Microsoft Corporation, "Windows Data Types", March 2007, <http://go.microsoft.com/fwlink/?LinkId=111558>.

[MS-NSPI] Microsoft Corporation, "Name Service Provider Interface (NSPI) Protocol Specification", June 2008.

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", June 2008.

[MS-OXOAB] Microsoft Corporation, "Offline Address Book (OAB) Format and Schema Protocol Specification", June 2008.

[MS-OXOABK] Microsoft Corporation, "Address Book Object Protocol Specification", June 2008.

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List Specification", June 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>.

[RFC4234] Crocker, D., Ed. and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, <http://www.ietf.org/rfc/rfc4234.txt>.

## 1.2.2 Informative References

None.

## 1.3 Protocol Overview

The Address Book User Interface Templates protocol is used for the following three purposes:

- Creation of new e-mail addresses for supported e-mail address types.
- Display and updating of data for different **Address Book objects**.
- Collection of data to perform searches on an **address book**.

### 1.3.1 Creation of New E-Mail Addresses

The creation of new e-mail addresses for supported e-mail address types is the first purpose of the Address Book User Interface Template protocol. By far the most common e-mail address type is the **Simple Mail Transfer Protocol (SMTP)** address type, but the server can support a number of different e-mail address types. This protocol provides a way for **address book** servers to expose the supported address book types to clients and provide a way for the client to create one of these addresses.

The creation of a new e-mail address is a two-step process. First, the client retrieves from the server a list of available address types and the name of the corresponding **creation template** that will be used to create an address of that type. The client can use this list to allow the user to select which address type to create. In the second step, the client requests the creation template that is associated with the selected address type, and uses the **template** to display a dialog to the user and get the necessary information to create the address by using the script that is returned with the template.

### 1.3.2 Display and Updating of Data

The second purpose of the Address Book User Interface Templates protocol is to display and update data for **Address Book objects**. For the purposes of this protocol, the server acts mainly as a database that stores user interface **templates** and then returns them to the client when requested. The client can then use the templates to display and edit data for Address Book objects.

To display and edit data about a particular Address Book object, the client requests a **display template** from the server and uses the returned template along with data that it has retrieved from the Address Book object [MS-OXOABK] to display a dialog to the user. The client can allow the user to change this data and then update the Address Book object to reflect the user's changes.

### 1.3.3 Collection of Search Data

The third purpose of the Address Book User Interface Templates protocol is to collect data that will be used to search the **address book**. For the purposes of this protocol, the server acts mainly as a database that stores user interface **templates** and simply returns them to the client when requested. The client can then use the templates to display a dialog to the user to collect data that it needs to perform search operations on the address book.

To collect data to perform search operations on the address book, the client requests a **search template** from the server and uses the returned search template to display a dialog and collect data to create a filter for the address book to locate **Address Book objects**.

## 1.4 Relationship to Other Protocols

The Address Book User Interface Templates protocol specification relies on an understanding of how to work with **Address Book objects, properties**, and tables (for more details, see [MS-OXOABK]). The specification also relies on an understanding of how the Address Book Object protocol is used to communicate with the server by using the underlying **RPC** transport.

## 1.5 Prerequisites/Preconditions

The Address Book User Interface Templates protocol assumes that the underlying Address Book Object protocol transport has been properly initialized.

## 1.6 Applicability Statement

The Address Book User Interface Templates protocol can be used to assist a user agent with creating e-mail addresses for supported address types and for displaying, creating, or modifying data associated with an **Address Book object**.

## 1.7 Versioning and Capability Negotiation

None.



## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

None.

# 2 Messages

## 2.1 Transport

This protocol uses the Name Service Provider Interface (NSPI) protocol as a transport for communicating between client and server. In particular, the client will use the two **RPC** functions **NspiGetSpecialTable** and **NspiGetTemplateInfo**, as specified in sections 3.1.4.3 and 3.1.4.20 of [MS-NSPI], to retrieve data from the server.

## 2.2 Message Syntax

The following sections specify the format of data that are specific to the Address Book User Interface Templates protocol that are returned from the **NspiGetSpecialTable** and **NspiGetTemplateInfo** function calls.

### 2.2.1 NspiGetSpecialTable PropertyRowSet\_r format

The *dwFlags* parameter that is passed to **NspiGetSpecialTablefunction**, as specified in the Name Service Provider Interface protocol, affects the data that is returned in the *PropertyRowSet\_r* output parameter. The bit flag values for the *dwFlags* parameter for this function are documented in the Name Service Provider Interface protocol specification [MS-NSPI]. The client **MUST** pass the **AB\_ONE\_OFF** flag to retrieve the table of supported address types from the server and **MUST NOT** pass any of the other flags. The **properties** listed in the following table **MUST** be returned by the server in the **PropertyRow\_r** structure that is contained in the *PropertyRowSet\_r* return parameter of the call.

Property	Description
<b>PidTagAddressType</b>	String property that indicates the type of address that is associated with the new recipient created with the <b>template</b> .
<b>PidTagDisplayName</b>	String property that contains a user-readable identification of the address type.
<b>PidTagDisplayType</b>	A <b>LONG</b> property contains a constant that identifies the type of <b>Address Book object</b> that the new recipient will be and therefore what icon the client will display for it. See

	[MS-OXOABK] for values.
<b>PidTagEntryId</b>	Binary property that contains the <b>entry ID</b> of the <b>template</b> to be used to create the new recipient. This identifier can be parsed to get the <b>DN</b> to be passed to <b>NspiGetTemplateInfo</b> to retrieve the template. See [MS-NSPI] for the format of <b>Permanent Entry IDs</b> .
<b>PidTagDepth</b>	<b>LONG</b> property that <b>MUST</b> be ignored.
<b>PidTagSelectable</b>	<b>Boolean</b> property that <b>MUST</b> be ignored.
<b>PidTagInstanceKey</b>	Binary property that contains a unique binary value.

### 2.2.2 NspiGetTemplateInfo PropertyRow\_r format

The *dwFlags* parameter, which is passed to the **NspiGetTemplateInfo** function, as specified in [MS-OXOAB], affect what properties are returned in the *PropertyRow\_r* return parameter. The flags for this function are specified in [MS-NSPI].

The following table lists the flags that are used by this protocol that can be passed in the *dwFlags* parameter of **NspiGetTemplateInfo** and the corresponding properties that are returned in the **PropertyRow\_r** return parameter.

Flag	Property Added to <i>PropertyRow_r</i>	Description of contents of property
TI_TEMPLATE 0x00000001	<b>PidTagTemplateData</b> 0x00010102	Binary <b>property</b> that contains a <b>TRowSet</b> structure followed by data that is pointed to in the <b>TRowSet</b> structure (the format specified in section 2.2.2.1).
TI_SCRIPT 0x00000004	<b>PidTagScriptData</b> 0x00040102	Binary property that contains script instructions and data (the format specified in section 2.2.2.2).

#### 2.2.2.1 Template Format

The dialog **template** consists of a set of rows that are represented by a **TRowSet** structure (see section 2.2.2.1.1) with each **TRow** structure describing one control in the dialog. To create the dialog from the template, each control that is described in a **TRow** structure **MUST** be added to a dialog in the location and with the size specified.

The **control flags** field, as specified in section 2.2.2.1.2, indicates additional information about the control, including whether it is editable. The control structure will indicate which static strings are to be used for the control, and the **property** that can be used to initialize the control and can be updated if the user edits the value in the control. When a **page control** is encountered, a new tabbed page is added to the dialog, and the controls that follow the page control are placed on that page.

### 2.2.2.1.1 TRowSet Structure

A **TRowSet** structure is defined in the following table.

Name	Type	Size in Bytes	Description
<b>Type</b>	<b>ULONG</b>	4	Type of the template. This <b>MUST</b> be 0x00000001.
<b>cRows</b>	<b>ULONG</b>	4	Count of <b>TRows</b> that are defined in this structure. This field <b>MUST</b> be followed by exactly <i>cRows</i> <b>TRow</b> structures.
<b>Row1</b>	<b>TRow</b> structure	36	<b>TRow</b> structure that contains data about a control.
<b>Row2</b>	<b>TRow</b> structure	36	<b>TRow</b> structure that contains data about a control.
...			
<b>RowN</b>	<b>TRow</b> structure	36	Last of <i>cRows</i> <b>TRow</b> structures.

### 2.2.2.1.2 TRow Structure

Each **TRow** structure describes a control that **MUST** be presented to the user in a **display area**. The display area is measured in pixels.

A **TRow** structure is defined in the following table.

Name	Type	Size in Bytes	Description
<b>XPos</b>	<b>ULONG</b>	4	X coordinate of the upper-left corner of the control. For more details, see the following text.
<b>DeltaX</b>	<b>ULONG</b>	4	Width of the control. For more details, see the following text.
<b>YPos</b>	<b>ULONG</b>	4	Y coordinate of the upper-left corner of the control. For more details, see the following text.
<b>DeltaY</b>	<b>ULONG</b>	4	Height of the control. For more details, see the following text.
<b>ControlType</b>	<b>ULONG</b>	4	Type of the control. For more details, see the following text.
<b>ControlFlags</b>	<b>ULONG</b>	4	Flags that describe the control's attributes. For more details, see the following text.
<b>ControlStructure</b>	<b>CNTRL</b> structure	12	Structure that contains data that is relevant to a particular control type.  For more details, see section 2.2.2.1.3.

**XPos** and **YPos** specify the X and Y coordinates of the upper-left corner of the control in pixels in the display area.

**DeltaX** and **DeltaY** specify the width and height of the control in pixels. The values are relative to the **XPos** and **YPos** of the control.

The other three properties describe various characteristics of the control.

**ControlType** indicates the type of control. **ControlType** MUST be one of the values listed in the following table.

Value	Description
0x00000000	A <b>label control</b> .
0x00000001	An <b>edit text box control</b> .
0x00000002	A <b>list box control</b> .
0x00000005	A <b>check box control</b> .
0x00000006	A <b>group box control</b> .
0x00000007	A <b>button control</b> .
0x00000008	A tabbed <b>page control</b> .
0x0000000B	A multi-valued list box control that is populated by a multi-valued <b>property</b> .
0x0000000C	A multi-valued <b>drop-down list box control</b> that is populated by a multi-valued property of type string.

**ControlFlags** is a bit field that describes the attributes of the control and MUST contain any combination of the bits that are specified in the following table for all values of **ControlType**, except for 0x00000008 (tabbed page control), as specified in the Description column. If the value of **ControlType** is 0x00000008, then the value of **ControlFlags** can be any value and MUST be ignored.

Value	Description
0x00000001	This flag indicates that the control can contain multiple lines. This means that a 0x0D and 0x0A can be entered within the control. This flag SHOULD NOT be set if the value of the <b>ControlType</b> field is any other value except 0x00000001 ( <b>edit box control</b> ). If it is set and the value of <b>ControlType</b> is not 0x00000001, then this flag MUST be ignored.
0x00000002	This flag indicates that the control can be edited, and the value that is associated with the control can be changed. When this flag is not set, the control is read-only. This value is ignored on the label, group box, button, multi-valued drop-down list box, and list box controls.
0x00000004	This flag indicates that if the control allows changes (0x00000002 attribute set), then it MUST have a value before the dialog can be dismissed.
0x00000008	This flag enables immediate setting of a value. As soon as a value in the control changes, that data MUST be updated in the property that is associated with that control.
0x00000010	This flag indicates the control is treated like a password entry control. The value MUST NOT be displayed by using the actual characters entered. This flag MUST only be set if the value of the <b>ControlType</b> field is 0x00000001 (edit box control).
0x00000020	If this flag is set, then the <b>edit control</b> MUST allow <b>double-byte character set (DBCS)</b> characters. This flag MUST NOT be set if the value of the <b>ControlType</b> field is any anything except 0x00000001 (edit box control).
0x00000040	This flag indicates that when a selection is made within the list box, the index column of that list box is set as a property. This flag MUST only be set if the 0x00000008 <b>ControlFlag</b> is also set.

**ControlStructure** is a **CNTRL** structure that contains information that is relevant to the particular type of control (see section 2.2.2.1.3).

### 2.2.2.1.3 Buffer Format of the CNTRL Structure

The base **CNTRL** structure is as follows, with each entry taking a different meaning, depending on the type of control, as shown in the following table.

Name	Type	Size	Description
<b>dwType</b>	<b>DWORD</b>	4	Varies depending on the control. See the following subsections for details.
<b>ulSize</b>	<b>ULONG</b>	4	Varies depending on the control. See the following subsections for details.
<b>ulString</b>	<b>ULONG</b>	4	The offset in <b>BYTES</b> from the base of the <b>TRowSet</b> structure to a null-terminated <b>non-Unicode</b> string. This string <b>MUST</b> be in the <b>code page</b> indicated by <i>ulTemplateCodePage</i> parameter of the <b>NspiGetTemplateInfo</b> call and <b>MUST</b> be terminated by a NULL character. In these strings, the "&" (ampersand) has special meaning and indicates that the character that immediately follows it <b>MUST</b> be used as an easy way to select this control. If the control cannot be selected, then the control that follows it is selected. If an "&" needs to be in the string and it <b>SHOULD NOT</b> have any special meaning, then a sequence of "&&" can be used to indicate this.  For more details about string values, usage, and limitatitons, see the following subsections.

#### 2.2.2.1.3.1 CNTRL Structure Describing a Label Control

**dwType** - **MUST** be 0x00000000 and **MUST** be ignored.

**ulSize** - **SHOULD** be 0x00000000 and **MUST** be ignored.

**ulString** - String that contains label text of **label control**. The string **MUST NOT** be over 128 characters long, including the NULL-terminating character.

#### 2.2.2.1.3.2 CNTRL Structure Describing a Edit Control

**dwType** – **Property** of data entered into the edit box control.

**ulSize** – Number of characters allowed to be entered into the edit box control.

**ulString** – String that contains a regular expression that describes the allowed characters that can be entered into the **edit control** (see the following subsection). The string **MUST NOT** be over 15 characters long, including the NULL-terminating character.

#### 2.2.2.1.3.2.1 Expression Syntax for Allowed Characters

The filter string has two possible expressions. The first expression allows any character to be entered into the **edit control**, and this expression is simply a string that contains only the "\*" (asterisk) character. The second expression lists the characters that are valid to be entered or that are invalid to be entered into the edit control. This expression is shown in **ABNF** [RFC4234] in the following format:

```
"[" *1("~") 1*(char-val / char-val "-" char-val) "]"
```

The expression **MUST** be included in square brackets ("[]"). When the first character inside the brackets is the tilde ("~") character, the expression represents characters that are not allowed in the edit control; otherwise, it represents only the characters that are allowed in the edit control. The rest of the characters inside the brackets are characters or ranges of characters to be allowed or disallowed from the edit control.

To represent any character that is a special character in this expression syntax with a backslash character ("\\"), the backslash character can be placed in front of the character. The backslash character will be ignored, and the character that follows it will be treated as a normal character and not as a special character. To represent a single character to allow/disallow, the character (with the leading backslash if necessary) is put in the string. To represent a range of characters to allow/disallow, the first character in the range is put in the string, followed by a dash ("-") character, followed by the final character in the range. The combination of all individual characters and character ranges is the set of characters that will be allowed or disallowed.

For example, if only the characters A, F, and T through Z are allowed to be entered into the control, then the expression is:

```
[AFT-Z]
```

If you want to allow all characters except the "[" (which will need the backslash character) and Z characters, then the expression is:

```
[~\[Z]
```

#### 2.2.2.1.3.3 CNTRL Structure Describing a List Box Control

**dwType** – **Property** of the table to populate this **list box control** from and to which the data from this list box control **SHOULD** be saved.

**ulSize** – **SHOULD** be 0x00000000 and **MUST** be ignored.

**ulString** – **MUST** be a string that contains only the character "\*" and **MUST** be ignored.



#### 2.2.2.1.3.4 CNTRL Structure Describing a Check Box Control

**dwType** – **Property** of data that is represented by this **check box control**.

**ulSize** – SHOULD be 0x00000000 and MUST be ignored.

**ulString** – String that contains the label text of check box control. The string MUST NOT be over 128 bytes long, including the NULL terminating character.

#### 2.2.2.1.3.5 CNTRL Structure Describing a Group Box Control

**dwType** – SHOULD be 0x00000000 and MUST be ignored.

**ulSize** – SHOULD be 0x00000000 and MUST be ignored.

**ulString** – String that contains the label text of the **group box control**. The string MUST NOT be over 128 bytes long, including the NULL-terminating character.

#### 2.2.2.1.3.6 CNTRL Structure Describing a Button Control

**dwType** – **Property** that is used to perform an action. This value MUST be **PidTagAddressBookManageDistributionList** (see [MS-OXOABK]). If this value is anything else, it MUST be ignored.

**ulSize** – MUST be 0x00000000 and MUST be ignored.

**ulString** – String that contains the label text of the **button control**. The string MUST NOT be greater than 128 bytes in length, including the NULL-terminating character.

#### 2.2.2.1.3.7 CNTRL Structure Describing a Tabbed Page Control

**dwType** – SHOULD be 0x00000000 and all other values MUST be ignored.

**ulSize** – SHOULD be 0x00000000 and MUST be ignored.

**ulString** – String that contains the label text of the tabbed **page control**. The string MUST NOT be greater than 32 bytes in length, including the NULL-terminating character.

#### 2.2.2.1.3.8 CNTRL Structure Describing a Multi-Valued List Box Control

**dwType** – **Property** for multi-valued data that is displayed in this **list box control**.

**ulSize** – SHOULD be 0x00000000 and MUST be ignored.

**ulString** – MUST be a string that contains only the character "\*" and MUST be ignored.

#### 2.2.2.1.3.9 CNTRL Structure Describing a Multi-Valued Drop-Down List Box Control

**dwType** – **Property** for the multi-valued data that is displayed in this **list box control**.

**ulSize** – SHOULD be 0x00000000 and MUST be ignored.

**ulString** – MUST be a string that contains only the character "\*" and MUST be ignored.

### 2.2.2.2 Script Format

A script is a set of instructions that are processed by using data collected by the **template** to produce a new e-mail address. The **PidTagScriptData property** in the PropertyRow\_r is a binary property that contains the information listed in the following table.

Name	Type	Size	Description
<i>Size</i>	DWORD	4	Specifies the number of <b>DWORDs</b> of script data that follow.
<i>ScriptData</i>	DWORD Array	Varies	Specifies a series of instructions and the data that accompanies them, as specified in the following subsections.

This binary script data contains a series of instructions (as specified in the following subsections) that can be executed to format an address and the data that is needed to execute those instructions. The first **DWORD** contains the number of **DWORDs** of instructions, "N". The next *N* **DWORDs** are the instructions. The data that is referenced by the instructions immediately follows the instructions.

The script is used to create a string that contains the e-mail address from the data gathered from the dialog that was created from the **template**. To process the script, begin at the first **DWORD** of *ScriptData* and process each instruction in turn. The result of the script is the result string. The result string MUST initially be empty and various instructions will append data to it. This string is the object's e-mail address and MUST only be used if the script doesn't end in error.

The instructions are specified in the following sections.

#### 2.2.2.2.1.1 Halt Instruction

Halt instruction is one **DWORD**, as shown in the following table.

Name	Type	Size	Value
<i>Halt</i>	<b>DWORD</b>	4	0x00000000

When this instruction is encountered, the script has finished and was successful. Processing **MUST** be halted and the current value of the result string is the e-mail address.

#### 2.2.2.2.1.2 Error Instruction

Error instruction is one **DWORD**, as shown in the following table.

Name	Type	Size	Value
<i>Error</i>	<b>DWORD</b>	4	0x00000001

When this instruction is encountered, the script is over and has ended in an error. Processing **MUST** be halted and the result string **MUST NOT** be used.

#### 2.2.2.2.1.3 Emit String Instruction

Emit String instruction is a 2-**DWORD** instruction and is as shown in the following table.

Name	Type	Size	Value
<i>Emit String</i>	<b>DWORD</b>	4	0x80000002
<i>First Operand</i>	<b>DWORD</b>	4	See the following text.

The *First Operand* is an offset in **BYTES** from the start of the *ScriptData* in the **PidTagScriptData property's** binary data to a **non-Unicode** null-terminated string, which is used as the operand for this instruction. When this instruction is encountered, the script **MUST** append the operand string to the result string and advance to the next instruction.

#### 2.2.2.2.1.4 Jump Instruction

Jump instruction is a 2-**DWORD** instruction, as shown in the following table.

Name	Type	Size	Value
<i>Jump</i>	<b>DWORD</b>	4	0x00000003
<i>Jump Offset</i>	<b>DWORD</b>	4	See the following text.

The *Jump Offset* parameter is an offset in **BYTES** from the start of the *ScriptData* in the **PidTagScriptData property's** binary data where the next instruction to execute is located.

When this instruction is encountered, the script MUST continue its execution from the instruction at the offset indicated.

#### 2.2.2.2.1.5 Jump If Not Exists Instruction

Jump If Not Exists instruction is a 3-DWORD instruction, as shown in the following table.

Name	Type	Size	Value
<i>Jump If Not Exists</i>	<b>DWORD</b>	4	0x00000004
<i>First Operand</i>	<b>DWORD</b>	4	See the following text.
<i>Jump Offset</i>	<b>DWORD</b>	4	See the following text.

The *First Operand* is a parameter that indicates a property that SHOULD be retrieved from the data collected by using the **template**.

The *Jump Offset* is an offset in **BYTES** from the start of the *ScriptData* in the **PidTagScriptData** property's binary data where the next instruction to execute is located.

When this operation is encountered, an attempt MUST be made to retrieve the property from the data collected by using the template. If the property was successfully retrieved, then the script is advanced over this instruction and execution continues. If the property fails to be retrieved, then the script will continue execution from the instruction at the offset indicated in *Jump Offset*.

#### 2.2.2.2.1.6 Jump If Equal Properties Instruction

Jump If Equal Properties instruction is a 4-DWORD instruction, as shown in the following table.

Name	Type	Size	Value
<i>Jump If Equal Properties</i>	<b>DWORD</b>	4	0x00000005
<i>First Operand</i>	<b>DWORD</b>	4	See the following text.
<i>SecondOperand</i>	<b>DWORD</b>	4	See the following text.
<i>Jump Offset</i>	<b>DWORD</b>	4	See the following text.

The *First Operand* is a parameter that indicates a **property** that will be retrieved from the data collected by using the **template**. The value of the property MUST be either a **non-Unicode**

null-terminated string or a **Boolean**. The value retrieved from the data collected by using the **template** is used as the first operand for the instruction.

The *Second Operand* is a parameter that indicates a property that will be retrieved from the data collected by using the template. The value of the property **MUST** be either a **non-Unicode** string or a Boolean and its type **MUST** match that of the *First Operand*. This value retrieved from the data that is collected by the template is used as the second operand for the instruction.

The *Jump Offset* is an offset in **BYTES** from the start of the *ScriptData* in the **PidTagScriptData** property's binary data where the next instruction to execute is located.

When this operation is encountered, the first two operands are compared, and if they are not equal, the script is advanced over this instruction and execution continues. If they are equal, the script will continue execution with the instruction at the offset indicated in the *Jump Offset*.

#### 2.2.2.2.1.7 Jump If Equal Values Instruction

Jump If Equal Values instruction is a 4-**DWORD** instruction and is shown in the following table.

Name	Type	Size	Value
<i>Jump If Equal Values</i>	<b>DWORD</b>	4	0x40000005
<i>First Operand</i>	<b>DWORD</b>	4	See the following text.
<i>Second Operand</i>	<b>DWORD</b>	4	See the following text.
<i>Jump Offset</i>	<b>DWORD</b>	4	See the following text.

The *First Operand* is a parameter that indicates a **property** that will be retrieved from the data collected by using the **template**. The value of the property **MUST** be either a **non-Unicode** string or a **Boolean**. The value retrieved from the object is used as the first operand for the instruction.

The *Second Operand* is an offset in bytes from the start of the *ScriptData* in the **PidTagScriptData** property's binary data where data is located, which is used as the second operand for this instruction. The type of the second operand is determined by the type of the first operand. Specifically, if the first operand is a Boolean, then the second operand is also treated as a Boolean, and if the first operand is a non-Unicode null-terminated string, then the second operand is also treated as a non-Unicode null-terminated string.

The *Jump Offset* is an offset in bytes from the start of the *ScriptData* in the **PidTagScriptData** property's binary data where the next instruction to execute is located.

When this operation is encountered, the values of the first two operands are compared, and if they are not equal, the script is advanced over this instruction and execution continues. If they are equal, the script will continue its execution with the instruction at the offset indicated in the *Jump Offset*.

#### 2.2.2.2.1.8 Emit Property Value Instruction

Emit Property value instruction is a 2-**DWORD** instruction, as shown in the following table.

Name	Type	Size	Value
<i>Emit Property Value</i>	<b>DWORD</b>	4	0x00000002
<i>First Operand</i>	<b>DWORD</b>	4	See the following text.

The *First Operand* is a parameter that **MUST** be retrieved from the data collected by using the **template**. The value of the **property** **MUST** be a **non-Unicode** string and **MUST** be terminated by a NULL character. The value will be used as the operand for this instruction. When this instruction is encountered, the script **MUST** append the operand string to the result string and advance to the next instruction.

#### 2.2.2.2.1.9 Emit Upper String Instruction

Emit Upper String instruction is a 2-**DWORD** instruction, as shown in the following table.

Name	Type	Size	Value
<i>Emit Upper String</i>	<b>DWORD</b>	4	0x80000006
<i>First Operand</i>	<b>DWORD</b>	4	See the following text.

The *First Operand* is an offset in **BYTES** from the start of the *ScriptData* in the **PidTagScriptData property's** binary data to a **non-Unicode** null-terminated string, which is used as the operand for this instruction. When this instruction is encountered, the script **MUST** first convert the operand string to all uppercase letters and then append the string to the script's result string and advance to the next instruction.

#### 2.2.2.2.1.10 Emit Upper Property Instruction

Emit Upper Property instruction is a 2-**DWORD** instruction, as shown in the following table.

Name	Type	Size	Value
<i>Emit Upper Property</i>	<b>DWORD</b>	4	0x00000006

Name	Type	Size	Value
<i>First Operand</i>	<b>DWORD</b>	4	<b>Property</b> of property to fetch and use as an operand.

The *First Operand* is a parameter that will be retrieved from the data collected by using the **template**. The value of the property **MUST** be a **non-Unicode** null-terminated string, and it is used as the operand for this instruction. When this instruction is encountered, the script **MUST** first convert the operand string to all uppercase letters, and then append the string to the script's result string and advance to the next instruction.

## 3 Protocol Details

### 3.1 Client Details

#### 3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that a client implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

##### 3.1.1.1 Dialog Object

A dialog object is an object that can be displayed to the user and onto which control objects can be placed in specified locations in order to display information and allow the user to update that information.

##### 3.1.1.2 Control objects

Control objects are user interface objects that can be used to display to and receive information from the user in various forms. There are eight types of control objects. These are described in the following table.

Control Type	Description
Label	Control used to display a string to the user. This control is not editable.
Edit	Control used to display a simple string to the user and to allow that string to be edited.

List Box	Control that contains a list of possible options of which one is selected. The user can change the selection.
Check Box	Control that displays a string that cannot be changed by the user and a box that can be checked and unchecked to indicate whether the option described by the string is selected.
Group Box	Control that contains other controls and around which is shown a box and a string that is the label for this group of controls.
Button	Control that displays a string to the user that performs a specified action when clicked.
Drop-Down List Box	Control similar to the <b>list box control</b> , except that only the currently selected item in the list is displayed to the user, and an arrow that is displayed on the end of the control causes the entire list to be displayed to the user so that a new item can be selected.
Page	Control that contains other controls. This control groups the other controls together and displays a string that describes them as a tab on the group. These controls can be placed on top of each other and the group for which the tab is selected determines the set of controls that will be shown.

### 3.1.1.3 Address Creation Template Table

An **Address Creation template** table is a table that contains a row for each supported address type, the address type's name as it will be displayed to the user, and what **creation template** is associated with it. It will also contain data that describes how to display it to the user.

### 3.1.2 Timers

None.

### 3.1.3 Initialization

The underlying Address Book Object protocol **MUST** be initialized. There is no initialization specific to the Address Book User Interface Templates protocol.



### 3.1.4 Higher-Layer Triggered Events

#### 3.1.4.1 Creating a new E-Mail Address for a Supported Address Type

When the client needs to use an e-mail address that does not exist on the **address book** server, it can create a new e-mail address for a supported address type and use this address to identify a recipient.

When the client wants to create a new e-mail address, first the **address creation table** MUST be retrieved by calling **NspiGetSpecialTable** with **AB\_ONE\_OFF** as the *dwFlags* parameter. The function returns a **PropertyRowSet\_r** structure that contains the data needed to create a list of **creation templates** for the supported address types. This list is then used to select an address type, possibly by displaying this list to the user or by selecting a type programmatically. When the address type has been selected, the data from the corresponding **PropertyRow\_r** can be used to determine the **distinguished name (DN)** for the creation template that will be used to create the new e-mail address and its address type. **PidTagAddressType** is the **property** in the **PropertyRow\_r** that indicates the e-mail address type. **PidTagEntryId** is the **property** in the **PropertyRow\_r** that can be parsed to get the DN. The **PidTagEntryId** is a **Permanent Entry ID**, and its format is specified in [MS-NSPI].

Next, the creation dialog **template** that will be used to create a new e-mail address MUST be retrieved by using the **NspiGetTemplateInfo** call, passing in the **DN** value for the **creation template** that you want as the *pDN* parameter and 0x00000000 as the *ulType* parameter. The *dwFlags* parameter contains a bitwise combination that MUST include the bits for **TI\_TEMPLATE** (0x00000001) set so that the template will be retrieved, and **TI\_SCRIPT** (0x00000004) set so the script to format the e-mail address is retrieved, and MAY contain the bits for **TI\_HELPFILE\_NAME** (0x00000020) and **TI\_HELPFILE** (0x00000040) for a value of 0x00000065.

The function's *dwCodePage* input parameter is the **code page** in which the strings in the template are stored, and in which the *ppData* return parameter is a pointer to a **PropertyRow\_r** that contains the data needed to create and display a dialog to create the new e-mail address. When the dialog is completed and dismissed, the data from the dialog MUST be used to run the script and create the e-mail address for this entry. This e-mail address MUST be used to populate the **PidTagEmailAddress** property and the address type retrieved from the selected **PropertyRow\_r** MUST be used to populate the **PidTagAddressType** property. These two properties comprise the e-mail address that can be used as an e-mail recipient.

#### 3.1.4.2 Displaying Information about an Address Book Object

When a client or user agent wants to view or change the information contained in an **address book** entry, the client MUST retrieve the **display template** for the address book entry's display type and display the data to the user. To retrieve the display dialog template that is used to display information about a particular **Address Book object**, **NspiGetTemplateInfo** MUST be called with the *ulType* parameter set to the display type of the object and the *pDN*

parameter set to 0x00000000. The *dwFlags* parameter contains a bitwise combination that MUST include the bit for TI\_TEMPLATE (0x00000001) set so the **template** will be retrieved and MAY contain the bits for TI\_HELPFILE\_NAME (0x00000020) and TI\_HELPFILE (0x00000040) for a value of 0x00000061. The function's *dwCodePage* input parameter is the **code page** in which the strings in the template are stored and the *ppData* return parameter contains a pointer to a **PropertyRow\_r** that contains the data needed to create and display the dialog. Data to initialize the dialog MUST be retrieved from the Address Book object by using the properties specified for each control in the dialog in the **PidTagTemplateData property** of the **PropertyRow\_r**, as specified in [MS-NSPI]. If the dialog is updated, then the data from the dialog MUST be used to update the properties that are associated with the controls and these properties MUST be updated on the Address Book object.

### 3.1.4.3 Collecting Data to Search the Address Book

When a client or user agent wants to search the **address book** for a particular **Address Book object**, the client MUST retrieve the **search template** for the address book and display the **template** to the user to collect the data to use to search the address book. To retrieve the search template that is used to collect information to use to search the address book, **NspiGetTemplateInfo** MUST be called with the *ulType* parameter set to the DT\_SEARCH (see definition in [MS-NSPI]) and the *pDN* parameter set to 0x00000000. The *dwFlags* parameter contains a bitwise combination that MUST include the bit for TI\_TEMPLATE (0x00000001) set so the template will be retrieved and MAY contain the bits for TI\_HELPFILE\_NAME (0x00000020) and TI\_HELPFILE (0x00000040) for a value of 0x00000061. The function's *dwCodePage* input parameter is the **code page** in which the strings in the template are stored and the *ppData* return parameter contains a pointer to a **PropertyRow\_r** that contains the data needed to create and display the dialog. When the dialog is completed and dismissed, the data from the dialog can be used to create a **Restriction\_r**, as defined in [MS-NSPI], from the controls that have been filled in. This **Restriction\_r** can be passed to **NspiGetMatches** in the *Filter* input parameter to locate an Address Book object, as specified in [MS-NSPI].

### 3.1.5 Message Processing Events and Sequencing Rules

The following events MUST be processed by a client that implements the Address Book User Interface Templates protocol. Note that no particular sequence is required for the message processing.

#### 3.1.5.1 Results of NspiGetSpecialTable Call to Retrieve the Address Creation Table

The results of the **NspiGetSpecialTable** call when its *dwFlag* parameter is AB\_ONE\_OFF is a **PropertyRowSet\_r** that contains the **address creation table** information (see section 2.2.1). These rows can be displayed as a list to show to users so that they can select the type of address to create. For each row in the **PropertyRowSet\_r**, the **PidTagDisplayName** can be used as the user-visible string in the list. After an address type has been selected, the **PidTagEntryId** in the selected **PropertyRow\_r** MUST be parsed (for details about the

format, see [MS-NSPI]) and the **distinguished name (DN)** found. This DN value MUST be used as the *pDN* parameter in a call to **NspiGetTemplateInfo** to retrieve the **creation template** and finish the creation of the e-mail address.

### 3.1.5.2 Results of NspiGetTemplateInfo Call to Retrieve the Creation Template

The results of the **NspiGetTemplateInfo** call when the **distinguished name (DN)** for the **creation template** is passed in as the *pDN* parameter of a **PropertyRow\_r** that contains the **template** for the dialog to display and the creation script. The client MUST use the dialog template to create a dialog and display it to the user. The client can create a new Property Bag object that is empty and use it to initialize the dialog so that it is blank. After the user has provided values for all controls that are marked as required, and closes the dialog, the properties that are associated with the controls can be updated in the Property Bag. Then, by using the Property Bag to retrieve these properties when needed, the creation script MUST be executed as specified in section 2.2.2.2 to create the new e-mail address. This e-mail address MUST be used to set the **PidTagEmailAddress property** and the address type from the **address creation table** MUST be used to set the **PidTagAddressType** property to create a new address.

### 3.1.5.3 Results of NspiGetTemplateInfo Call to Retrieve the Display Template

The results of the **NspiGetTemplateInfo** call when the display type of an object is passed in as the *ulType* parameter is a **PropertyRow\_r** that contains the **template** for the dialog to display. The client MUST use the dialog template to create a dialog and display it to the user. The client MUST use the object the type for which was passed in to initialize the dialog. If the user updates any information in the dialog and closes the dialog, the **properties** that are associated with the updated controls MUST be updated in the object.

### 3.1.5.4 Results of NspiGetTemplateInfo Call to Retrieve the Search Template

The results of the **NspiGetTemplateInfo** call when the display type of DT\_SEARCH is passed in as the *ulType* parameter is a **PropertyRow\_r** that contains the **search template** for the dialog to display. The client MUST use the search template to create a dialog and display it to the user for input. If the user inputs any information into the dialog and closes the dialog, the properties associated with the controls SHOULD be used to create a **Restriction\_r** to be used as the Filter input parameter in a call to **NspiGetMatches**. **NspiGetMatches** SHOULD handle filters containing properties in the search template.

## 3.1.6 Timer Events

None.

## 3.1.7 Other Local Events

None.

## 3.2 Server Details

### 3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that a server implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model, as long as their external behavior is consistent with that described in this document.

#### 3.2.1.1 Template Objects

The server keeps a Template object for each display type and for the **creation template** for each supported address type in the **address creation table**. These objects contain the **template**, and if needed, the script that will be returned from **NspiGetTemplateInfo** in the **PropertyRow\_r**.

#### 3.2.1.2 Table of Supported Address Types and Name of Template to Use to Create Them

The server can keep a Table object that contains the list of supported address types, the **creation templates** that are associated with the address types, and any other data that is needed to construct and return the **PropertyRowSet\_r** when it receives a call from **NspiGetSpecialTable**.

#### 3.2.2 Timers

None.

#### 3.2.3 Initialization

None.

#### 3.2.4 Higher-Layer Triggered Events

None.

#### 3.2.5 Message Processing Events and Sequencing Rules

The following events **MUST** be processed by a server that implements the Address Book User Interface Templates protocol. Note that no particular sequence is required for the message processing.

##### 3.2.5.1 NspiGetSpecialTable Call from Client

The client calls in to the server by using the **NspiGetSpecialTable** RPC function with the **AB\_ONE\_OFF** flag set in the *dwFlags* parameter to retrieve the table of supported address types. The server's handling of any other flags that can be passed to the **NspiGetSpecialTable**

function are specified in [MS-NSPI]. The server retrieves the table of support address types and MUST format the table into a **PropertyRow\_r** before returning this data to the client.

### 3.2.5.2 NspiGetTemplateInfo Call from Client

The client calls in to the server by using the **NspiGetTemplateInfo** RPC function with the *dwFlags* parameter that contains some bitwise combination of the bit flags **TI\_TEMPLATE** (0x00000001) so the **template** will be retrieved, and **TI\_SCRIPT** (0x00000004) so the script to format the e-mail address is retrieved (see section 2.2.2). The server MUST use the display type specified in the *ulType* input parameter or the template **DN** specified in the *pDN* pinput parameter to retrieve the Template object. Finally, the server MUST create the *PropertyRow\_r* return parameter by using the Template object and return this data to the client.

### 3.2.6 Timer Events

None.

### 3.2.7 Other Local Events

None.

## 4 Protocol Examples

Starting with a connection bound to the server, the following sections include sample structures that would be returned by the **NSPI** function call.

### 4.1 Creating a New E-Mail Address for a Supported Address Type

To create a new e-mail address for one of the supported address types, the client has to first request the list of supported address types from the server by calling the **NspiGetSpecialTable** RPC function. The first step is to bind to the server by using the **NspiBind** call to retrieve an RPC context handle for the server.

**NspiGetSpecialTable** is then called, passing a **AB\_ONE\_OFF** flag in the *dwFlags* parameter.

The following are the input parameters for the **NspiGetSpecialTable** call:

```
dwFlags: 0x00000002
STAT: hIndex=0x0,
      ContainerID=0xcccccccc,
      CurrentRec=0x00000000,
      Delta=0x00000000,
      NumPos=0x00000000,
      TotalRecs=0xcccccccc
      CodePage =0x4e4
      TemplateLocale=0x409
      SortLocale=0x409
```

lpVersion: Not used - 0xcccccccc

The call returns a **PropertyRowSet\_r** in the *HierTabrows* return parameter. The following is an example of the *HierTabRows* that can be returned:

```
HierTabRows:      cRows - 0x00000005
  Row0
    cValues - 0x00000007
    ulAdrEntryPad - not used - 0x00000000
      Prop0
        ulPropTag - PidTagDisplayName (0x3001001e)
        ulReserved - not used - 0x00000000
        Value - cc:Mail Address
      Prop1
        ulPropTag - PidTagAddressType (0x3002001e)
        ulReserved - not used - 0x00000000
        Value - CCMAIL
      Prop2
        ulPropTag - PidTagDisplayType (0x39000003)
        ulReserved - not used - 0x00000000
        Value - 0x00000000
      Prop3
        ulPropTag - PidTagDepth (0x30050003)
        ulReserved - not used - 0x00000000
        Value - 0x00000000
      Prop4
        ulPropTag - PidTagSelectable (0x3609000b)
        ulReserved - not used - 0x00000000
        Value - 0x00000001
      Prop5
        ulPropTag - PidTagInstanceKey (0x0ff60102)
        ulReserved - not used - 0x00000000
        Value - 0x02957c9c
0000 d6 23 00 00                                     .#..
      Prop6
        ulPropTag - PidTagEntryId (0x0fff0102)
        ulReserved - not used - 0x00000000
        Value - 0x02957ca0
0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 .....@..B.....
0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/...../o=N
0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 T5/ou=0000000000
0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 0000000000000000
0040 30 30 30 30 30 30 2f 63-6e 3d 34 33 33 34 34 43 000000/cn=43344C
0050 30 37 44 34 43 45 41 36-34 46 42 45 39 34 32 37 07D4CEA64FBE9427
0060 43 44 31 36 41 31 33 43-44 34 00                                CD16A13CD4.
  Row1
    cValues - 0x00000007
    ulAdrEntryPad - not used - 0x00000000
```

```

Prop0
  ulPropTag - PidTagDisplayName (0x3001001e)
  ulReserved - not used - 0x00000000
  Value - Microsoft Mail Address
Prop1
  ulPropTag - PidTagAddressType (0x3002001e)
  ulReserved - not used - 0x00000000
  Value - MS
Prop2
  ulPropTag - PidTagDisplayType (0x39000003)
  ulReserved - not used - 0x00000000
  Value - 0x00000000
Prop3
  ulPropTag - PidTagDepth (0x30050003)
  ulReserved - not used - 0x00000000
  Value - 0x00000000
Prop4
  ulPropTag - PidTagSelectable (0x3609000b)
  ulReserved - not used - 0x00000000
  Value - 0x00000001
Prop5
  ulPropTag - PidTagInstanceKey (0x0fff60102)
  ulReserved - not used - 0x00000000
  Value - 0x02957df0
0000 d3 23 00 00                                     .#..
Prop6
  ulPropTag - PidTagEntryId (0x0fff0102)
  ulReserved - not used - 0x00000000
  Value - 0x02957df4
0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 .....@..B.....
0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/...../o=N
0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 30 T5/ou=0000000000
0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30 0000000000000000
0040 30 30 30 30 30 30 2f 63-6e 3d 37 46 32 36 33 44 000000/cn=7F263D
0050 42 37 42 39 35 31 41 32-34 33 38 38 45 43 42 39 B7B951A24388ECB9
0060 37 39 34 36 38 42 43 42-45 45 00                    79468BCBEE.

```

Row2

```

cValues - 0x00000007
ulAdrEntryPad - not used - 0x00000000
Prop0
  ulPropTag - PidTagDisplayName (0x3001001e)
  ulReserved - not used - 0x00000000
  Value - MacMail Address
Prop1
  ulPropTag - PidTagAddressType (0x3002001e)
  ulReserved - not used - 0x00000000
  Value - MSA
Prop2
  ulPropTag - PidTagDisplayType (0x39000003)
  ulReserved - not used - 0x00000000

```

```

        Value - 0x00000000
    Prop3
        ulPropTag - PidTagDepth (0x30050003)
        ulReserved - not used - 0x00000000
        Value - 0x00000000
    Prop4
        ulPropTag - PidTagSelectable (0x3609000b)
        ulReserved - not used - 0x00000000
        Value - 0x00000001
    Prop5
        ulPropTag - PidTagInstanceKey (0x0ff60102)
        ulReserved - not used - 0x00000000
        Value - 0x02957f40
0000 d5 23 00 00                                     .#..
    Prop6
        ulPropTag - PidTagEntryId (0x0fff0102)
        ulReserved - not used - 0x00000000
        Value - 0x02957f44
0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 .....@..B.....
0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/...../o=N
0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 30 T5/ou=0000000000
0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30 0000000000000000
0040 30 30 30 30 30 30 2f 63-6e 3d 37 42 35 30 35 30 000000/cn=7B5050
0050 37 33 41 44 44 41 44 33-34 39 38 33 30 42 32 43 73ADDAD349830B2C
0060 35 46 41 39 38 32 36 33-44 46 00                    5FA98263DF.
    Row3
        cValues - 0x00000007
        ulAdrEntryPad - not used - 0x00000000
    Prop0
        ulPropTag - PidTagDisplayName (0x3001001e)
        ulReserved - not used - 0x00000000
        Value - Internet Address
    Prop1
        ulPropTag - PidTagAddressType (0x3002001e)
        ulReserved - not used - 0x00000000
        Value - SMTP
    Prop2
        ulPropTag - PidTagDisplayType (0x39000003)
        ulReserved - not used - 0x00000000
        Value - 0x00000000
    Prop3
        ulPropTag - PidTagDepth (0x30050003)
        ulReserved - not used - 0x00000000
        Value - 0x00000000
    Prop4
        ulPropTag - PidTagSelectable (0x3609000b)
        ulReserved - not used - 0x00000000
        Value - 0x00000001
    Prop5
        ulPropTag - PidTagInstanceKey (0x0ff60102)

```



```

        ulReserved - not used - 0x00000000
        Value - 0x02956320
0000 d4 23 00 00                                     .#..
        Prop6
            ulPropTag - PidTagEntryId (0xffff0102)
            ulReserved - not used - 0x00000000
            Value - 0x02956324
0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 .....@..B.....
0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/...../o=N
0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 30 T5/ou=0000000000
0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30 0000000000000000
0040 30 30 30 30 30 30 2f 63-6e 3d 41 39 36 30 39 33 000000/cn=A96093
0050 42 30 45 33 34 45 43 46-34 37 38 42 38 38 42 36 B0E34ECF478B88B6
0060 41 43 36 36 41 36 32 35-42 43 00                               AC66A625BC.
        Row4
            cValues - 0x00000007
            ulAdrEntryPad - not used - 0x00000000
                Prop0
                    ulPropTag - PidTagDisplayName (0x3001001e)
                    ulReserved - not used - 0x00000000
                    Value - X.400 Address
                Prop1
                    ulPropTag - PidTagAddressType (0x3002001e)
                    ulReserved - not used - 0x00000000
                    Value - X400
                Prop2
                    ulPropTag - PidTagDisplayType (0x39000003)
                    ulReserved - not used - 0x00000000
                    Value - 0x00000000
                Prop3
                    ulPropTag - PidTagDepth (0x30050003)
                    ulReserved - not used - 0x00000000
                    Value - 0x00000000
                Prop4
                    ulPropTag - PidTagSelectable (0x3609000b)
                    ulReserved - not used - 0x00000000
                    Value - 0x00000001
                Prop5
                    ulPropTag - PidTagInstanceKey (0x0ff60102)
                    ulReserved - not used - 0x00000000
                    Value - 0x02956474
0000 d2 23 00 00                                     .#..
        Prop6
            ulPropTag - PidTagEntryId (0xffff0102)
            ulReserved - not used - 0x00000000
            Value - 0x02956478
0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 .....@..B.....
0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/...../o=N
0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 30 T5/ou=0000000000
0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30 0000000000000000

```

```

0040 30 30 30 30 30 30 2f 63-6e 3d 34 45 38 30 41 46 000000/cn=4E80AF
0050 33 41 34 37 34 44 38 46-34 45 38 46 45 39 31 41 3A474D8F4E8FE91A
0060 32 43 41 43 42 46 39 38-44 43 00 2CACBF98DC.

```

These rows are then used to create a list of address types, and this list is displayed so that the user can choose which type to create. The "cc:Mail Address" row is selected and examined, and the **PidTagAddressType** and **PidTagEntryId** values are extracted. The **PidTagEntryId** is parsed and the DN is determined to be the following:

```

/o=NT5/ou=00000000000000000000000000000000/cn=43344C07D4CEA64FBE9427CD1
6A13CD4

```

This value is passed to **NSPIGetTemplateInfo** as the *pDN* parameter to retrieve the **creation template**. The following are the input parameters that are passed to **NspiGetTemplateInfo**:

```

dwFlags:                0x00000065
dwType:                 0x00000000
pDN:
/o=NT5/ou=00000000000000000000000000000000/cn=43344C07D4CEA64FBE9427CD1
6A13CD4
ulTemplateCodePage:    0x000004e4
ulServerTemplateLocale: 0x00000409

```

**NspiGetTemplateInfo** will return a **PropertyRow\_r** in the *ppData* output parameter, and this **PropertyRow\_r** will contain the template and script data. **NspiGetTemplateInfo** returns the following:

```

ppData
  cValues - 0x00000002
  ulAdrEntryPad - not used - 0x00000000
  Prop0
    ulPropTag - PidTagTemplateData (0x00010102)
    ulReserved - not used - 0x00000000
    TRowSet - Type - 0x00000001
      cRows - 0x00000007
      Row0
        XPos - 0x00000000
        XDelta - 0x00000000
        YPos - 0x00000000
        YDelta- 0x00000000
        ControlType - 0x00000008
        ControlFlags - 0x00000d70
        ControlStructure
          dwType - 0x00000000
          ulSize - 0x00000000
          ulString - 0x00000104
          General
      Row1

```

XPos - 0x00000006  
XDelta - 0x00000064  
YPos - 0x0000000c  
YDelta- 0x00000014  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x0000010c  
    &Display name:

Row2

XPos - 0x0000006b  
XDelta - 0x000000fa  
YPos - 0x0000000c  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000026  
ControlStructure  
    dwType - 0x3001001e  
    ulSize - 0x00000100  
    ulString - 0x0000011b  
    \*

Row3

XPos - 0x00000006  
XDelta - 0x00000064  
YPos - 0x00000023  
YDelta- 0x00000014  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x0000011d  
    &Mailbox:

Row4

XPos - 0x0000006b  
XDelta - 0x000000fa  
YPos - 0x00000023  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000006  
ControlStructure  
    dwType - 0x6701001e  
    ulSize - 0x00000100  
    ulString - 0x00000127  
    \*

Row5

XPos - 0x00000006  
XDelta - 0x00000064

```

YPos - 0x0000003a
YDelta- 0x00000014
ControlType - 0x00000000
ControlFlags - 0x00000000
ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x00000129
    &Post Office:

```

Row6

```

XPos - 0x0000006b
XDelta - 0x000000fa
YPos - 0x0000003a
YDelta- 0x0000000c
ControlType - 0x00000001
ControlFlags - 0x00000006
ControlStructure
    dwType - 0x6702001e
    ulSize - 0x00000100
    ulString - 0x00000137

```

\*

```

0000 01 00 00 00 07 00 00 00-00 00 00 00 00 00 00 .....
0010 00 00 00 00 00 00 00 00-08 00 00 00 70 0d 00 00 .....p...
0020 00 00 00 00 00 00 00 00-04 01 00 00 06 00 00 00 .....
0030 64 00 00 00 0c 00 00 00-14 00 00 00 00 00 00 00 d.....
0040 00 00 00 00 00 00 00 00-00 00 00 00 0c 01 00 00 .....
0050 6b 00 00 00 fa 00 00 00-0c 00 00 00 0c 00 00 00 k.....
0060 01 00 00 00 26 00 00 00-1e 00 01 30 00 01 00 00 ....&.....0....
0070 1b 01 00 00 06 00 00 00-64 00 00 00 23 00 00 00 .....d...#...
0080 14 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0090 00 00 00 00 1d 01 00 00-6b 00 00 00 fa 00 00 00 .....k.....
00a0 23 00 00 00 0c 00 00 00-01 00 00 00 06 00 00 00 #.....
00b0 1e 00 01 67 00 01 00 00-27 01 00 00 06 00 00 00 ...g....'.....
00c0 64 00 00 00 3a 00 00 00-14 00 00 00 00 00 00 00 d...:.....
00d0 00 00 00 00 00 00 00 00-00 00 00 00 29 01 00 00 .....)....
00e0 6b 00 00 00 fa 00 00 00-3a 00 00 00 0c 00 00 00 k.....:.....
00f0 01 00 00 00 06 00 00 00-1e 00 02 67 00 01 00 00 .....g....
0100 37 01 00 00 47 65 6e 65-72 61 6c 00 26 44 69 73 7...General.&Dis
0110 70 6c 61 79 20 6e 61 6d-65 3a 00 2a 00 26 4d 61 play name:.*.&Ma
0120 69 6c 62 6f 78 3a 00 2a-00 26 50 6f 73 74 20 4f ilbox:.*.&Post O
0130 66 66 69 63 65 3a 00 2a-00 ffice:.*.

```

Prop1

```

ulPropTag - PidTagScriptData (0x00040102)
ulReserved - not used - 0x00000000
Size - 0x0000000F

```

```

Operation Jump Not Exists - 0x00000004
    PropTag - 0x6701001e
    Offset - 0x00000014
Operation Emit - 0x00000002

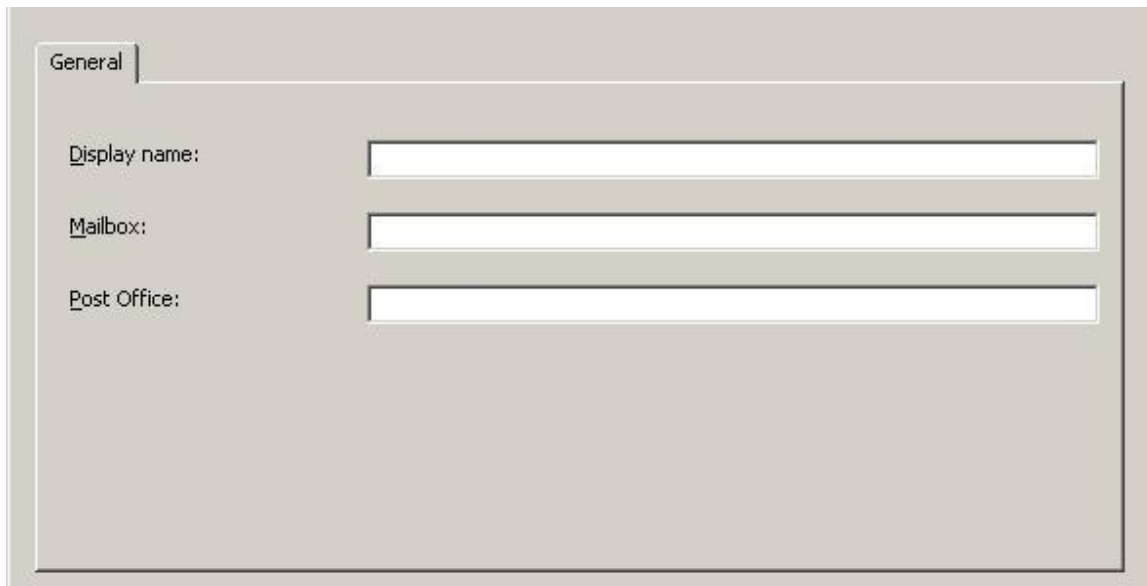
```

```

        PropTag - 0x6701001e
        Operation Emit String - 0x80000002
        Offset - 0x00000034
        at
        Operation Jump Not Exists - 0x00000004
        PropTag - 0x6702001e
        Offset - 0x00000030
        Operation Emit - 0x00000002
        PropTag - 0x6702001e
        Operation Halt - 0x00000000
0000 0f 00 00 00 04 00 00 00-1e 00 01 67 14 00 00 00 .....g....
0010 02 00 00 00 1e 00 01 67-02 00 00 80 34 00 00 00 .....g....4...
0020 04 00 00 00 1e 00 02 67-30 00 00 00 02 00 00 00 .....g0.....
0030 1e 00 02 67 00 00 00 00-20 61 74 20 00 00 00 00 ...g.... at ....

```

This **template** can be processed to create a dialog similar to the one shown in Figure 1.



**Figure 1: Address creation dialog**

The following data is then entered into the dialog:

Display name: Bob

Mailbox: BobsMailbox

Post office: GeneralPostOffice

The script processes the above data, and produces the following e-mail address:

BobsMailbox at GeneralPostOffice

Therefore, the **PidTagEmailAddress** that represents this user is "BobsMailbox at GeneralPostOffice" and the **PidTagAddressType** is "CCMAIL."

## 4.2 Retrieving a Mail User's Template

To display information about an **Address Book object** to the user, the client has to first request the **display template** from the server by calling the **NspiGetTemplateInfo** RPC function. The first step is to bind to the server by using the **NspiBind** call to retrieve an RPC context handle for the server.

For example, **NspiGetTemplateInfo** is called to get the template to display data about a **mail user** by passing *ulType* with the mail user display type (DT\_MAILUSER).

The following are the input parameters for an example of a call to **NspiGetTemplateInfo**:

```
dwFlags:                0x00000061
dwType:                 0x00000000
pDN:                    (null)
ulTemplateCodePage:     0x000004e4
ulServerTemplateLocale: 0x00000409
```

The call returns a **PropertyRow\_r** in the *ppData* return parameter. The following is an example of the *ppData* that can be returned:

```
ppData
  cValues - 0x00000001
  ulAdrEntryPad - not used - 0x00000000
  Prop0
    ulPropTag - PidTagTemplateData (0x00010102)
    ulReserved - not used - 0x00000000
    TRowSet - Type - 0x00000001
      cRows - 0x00000041
      Row0
        XPos - 0x00000000
        XDelta - 0x00000000
        YPos - 0x00000000
        YDelta- 0x00000000
        ControlType - 0x00000008
        ControlFlags - 0x00000ce4
        ControlStructure
          dwType - 0x00000000
          ulSize - 0x00000000
          ulString - 0x0000092c
          General
      Row1
        XPos - 0x00000006
        XDelta - 0x00000167
        YPos - 0x00000003
        YDelta- 0x00000029
```

ControlType - 0x00000006  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000934  
Name

Row2

XPos - 0x0000000c  
XDelta - 0x00000046  
YPos - 0x0000000f  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000939  
    &First:

Row3

XPos - 0x00000053  
XDelta - 0x00000025  
YPos - 0x0000000d  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a06001e  
    ulSize - 0x00000040  
    ulString - 0x00000941  
    \*

Row4

XPos - 0x0000007b  
XDelta - 0x0000002c  
YPos - 0x0000000f  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000943

Initials:

Row5

XPos - 0x000000a8  
XDelta - 0x0000000f  
YPos - 0x0000000d  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a0a001e  
    ulSize - 0x00000006  
    ulString - 0x0000094e  
\*

Row6

XPos - 0x000000bd  
XDelta - 0x00000046  
YPos - 0x0000000f  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000950  
&Last:

Row7

XPos - 0x00000103  
XDelta - 0x00000064  
YPos - 0x0000000d  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a11001e  
    ulSize - 0x00000040  
    ulString - 0x00000957  
\*

Row8

XPos - 0x0000000c  
XDelta - 0x00000046  
YPos - 0x0000001e  
YDelta- 0x00000008



ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000959  
Display&:

Row9

XPos - 0x00000053  
XDelta - 0x00000064  
YPos - 0x0000001c  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3001001e  
    ulSize - 0x00000100  
    ulString - 0x00000963  
\*

Row10

XPos - 0x000000bd  
XDelta - 0x00000046  
YPos - 0x0000001e  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000965  
Al&ias:

Row11

XPos - 0x00000103  
XDelta - 0x00000064  
YPos - 0x0000001c  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a00001e  
    ulSize - 0x00000040  
    ulString - 0x0000096d

\*

Row12

```

XPos - 0x0000000c
XDelta - 0x00000046
YPos - 0x00000032
YDelta- 0x00000008
ControlType - 0x00000000
ControlFlags - 0x00000000
ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x0000096f
Add&ress:

```

Row13

```

XPos - 0x00000053
XDelta - 0x00000064
YPos - 0x00000030
YDelta- 0x0000001b
ControlType - 0x00000001
ControlFlags - 0x00000001
ControlStructure
    dwType - 0x3a29001e
    ulSize - 0x00000400
    ulString - 0x00000979

```

\*

Row14

```

XPos - 0x0000000c
XDelta - 0x00000046
YPos - 0x00000050
YDelta- 0x00000008
ControlType - 0x00000000
ControlFlags - 0x00000000
ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x0000097b
Cit&y:

```

Row15

```

XPos - 0x00000053
XDelta - 0x00000064
YPos - 0x0000004e
YDelta- 0x0000000c

```

ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a27001e  
    ulSize - 0x00000080  
    ulString - 0x00000982  
    \*

Row16

XPos - 0x0000000c  
XDelta - 0x00000046  
YPos - 0x0000005f  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000984  
    &State:

Row17

XPos - 0x00000053  
XDelta - 0x00000064  
YPos - 0x0000005d  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a28001e  
    ulSize - 0x00000080  
    ulString - 0x0000098c  
    \*

Row18

XPos - 0x0000000c  
XDelta - 0x00000046  
YPos - 0x0000006e  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x0000098e

```

&Zip code:
Row19
  XPos - 0x00000053
  XDelta - 0x00000064
  YPos - 0x0000006c
  YDelta- 0x0000000c
  ControlType - 0x00000001
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x3a2a001e
    ulSize - 0x00000028
    ulString - 0x00000999
  *
Row20
  XPos - 0x0000000c
  XDelta - 0x00000046
  YPos - 0x0000007d
  YDelta- 0x00000008
  ControlType - 0x00000000
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x0000099b
  Co&untry/Region:
Row21
  XPos - 0x00000053
  XDelta - 0x00000064
  YPos - 0x0000007b
  YDelta- 0x0000000c
  ControlType - 0x00000001
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x3a26001e
    ulSize - 0x00000003
    ulString - 0x000009ac
  *
Row22
  XPos - 0x000000bd
  XDelta - 0x00000046
  YPos - 0x00000032
  YDelta- 0x00000008

```

ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x000009ae  
Titl&e:

Row23

XPos - 0x00000103  
XDelta - 0x00000064  
YPos - 0x00000030  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a17001e  
    ulSize - 0x00000040  
    ulString - 0x000009b6  
\*

Row24

XPos - 0x000000bd  
XDelta - 0x00000046  
YPos - 0x00000041  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x000009b8  
Co&mpany:

Row25

XPos - 0x00000103  
XDelta - 0x00000064  
YPos - 0x0000003f  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a16001e  
    ulSize - 0x00000040  
    ulString - 0x000009c2

```

*
Row26
  XPos - 0x000000bd
  XDelta - 0x00000046
  YPos - 0x00000050
  YDelta- 0x00000008
  ControlType - 0x00000000
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x000009c4
    &Department:
Row27
  XPos - 0x00000103
  XDelta - 0x00000064
  YPos - 0x0000004e
  YDelta- 0x0000000c
  ControlType - 0x00000001
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x3a18001e
    ulSize - 0x00000040
    ulString - 0x000009d1
*
Row28
  XPos - 0x000000bd
  XDelta - 0x00000046
  YPos - 0x0000005f
  YDelta- 0x00000008
  ControlType - 0x00000000
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x000009d3
    &Office:
Row29
  XPos - 0x00000103
  XDelta - 0x00000064
  YPos - 0x0000005d
  YDelta- 0x0000000c

```

ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a19001e  
    ulSize - 0x00000080  
    ulString - 0x000009dc  
\*

Row30

XPos - 0x000000bd  
XDelta - 0x00000046  
YPos - 0x0000006e  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x000009de  
Assista&nt:

Row31

XPos - 0x00000103  
XDelta - 0x00000064  
YPos - 0x0000006c  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a30001e  
    ulSize - 0x00000100  
    ulString - 0x000009ea  
\*

Row32

XPos - 0x000000bd  
XDelta - 0x00000046  
YPos - 0x0000007d  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x000009ec

P&hone:

Row33

XPos - 0x00000103  
XDelta - 0x00000064  
YPos - 0x0000007b  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a08001e  
    ulSize - 0x00000040  
    ulString - 0x000009f4  
\*

Row34

XPos - 0x00000000  
XDelta - 0x00000000  
YPos - 0x00000000  
YDelta- 0x00000000  
ControlType - 0x00000008  
ControlFlags - 0x00000ce5  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x000009f6  
Organization

Row35

XPos - 0x00000006  
XDelta - 0x00000167  
YPos - 0x00000004  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a03  
&Manager:

Row36

XPos - 0x00000006  
XDelta - 0x00000167  
YPos - 0x0000000f  
YDelta- 0x00000014



ControlType - 0x00000002  
ControlFlags - 0x00000002  
ControlStructure  
    dwType - 0x8005000d  
    ulSize - 0x00000000  
    ulString - 0x00000a0d  
\*

Row37

XPos - 0x00000006  
XDelta - 0x00000167  
YPos - 0x00000025  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a0f  
    &Direct reports:

Row38

XPos - 0x00000006  
XDelta - 0x00000167  
YPos - 0x00000030  
YDelta- 0x00000060  
ControlType - 0x00000002  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x800e000d  
    ulSize - 0x00000000  
    ulString - 0x00000a20  
\*

Row39

XPos - 0x00000000  
XDelta - 0x00000000  
YPos - 0x00000000  
YDelta- 0x00000000  
ControlType - 0x00000008  
ControlFlags - 0x00000ce6  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a22

Phone/Notes

Row40

XPos - 0x00000006  
XDelta - 0x00000167  
YPos - 0x00000003  
YDelta- 0x00000050  
ControlType - 0x00000006  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a2e  
Phone numbers

Row41

XPos - 0x0000000c  
XDelta - 0x00000046  
YPos - 0x00000012  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a3c  
Bu&siness:

Row42

XPos - 0x00000053  
XDelta - 0x00000064  
YPos - 0x00000010  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a08001e  
    ulSize - 0x00000040  
    ulString - 0x00000a47  
\*

Row43

XPos - 0x000000bd  
XDelta - 0x00000046  
YPos - 0x00000012  
YDelta- 0x00000008

ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a49  
    &Home:

Row44

XPos - 0x00000103  
XDelta - 0x00000064  
YPos - 0x00000010  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a09001e  
    ulSize - 0x00000040  
    ulString - 0x00000a50  
    \*

Row45

XPos - 0x0000000c  
XDelta - 0x00000046  
YPos - 0x00000022  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a52  
    Busi&ness 2:

Row46

XPos - 0x00000053  
XDelta - 0x00000064  
YPos - 0x00000020  
YDelta- 0x00000060  
ControlType - 0x0000000c  
ControlFlags - 0x00000001  
ControlStructure  
    dwType - 0x3alb101e  
    ulSize - 0x00000040  
    ulString - 0x00000a5f

```

*
Row47
  XPos - 0x000000bd
  XDelta - 0x00000046
  YPos - 0x00000022
  YDelta- 0x00000008
  ControlType - 0x00000000
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x00000a61
    H&ome 2:
Row48
  XPos - 0x00000103
  XDelta - 0x00000064
  YPos - 0x00000020
  YDelta- 0x00000060
  ControlType - 0x0000000c
  ControlFlags - 0x00000001
  ControlStructure
    dwType - 0x3a2f101e
    ulSize - 0x00000040
    ulString - 0x00000a6a
*
Row49
  XPos - 0x0000000c
  XDelta - 0x00000046
  YPos - 0x00000032
  YDelta- 0x00000008
  ControlType - 0x00000000
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x00000a6c
    &Fax:
Row50
  XPos - 0x00000053
  XDelta - 0x00000064
  YPos - 0x00000030
  YDelta- 0x0000000c

```

ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a23001e  
    ulSize - 0x00000040  
    ulString - 0x00000a72  
    \*

Row51

XPos - 0x000000bd  
XDelta - 0x00000046  
YPos - 0x00000032  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a74  
    &Mobile:

Row52

XPos - 0x00000103  
XDelta - 0x00000064  
YPos - 0x00000030  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a1c001e  
    ulSize - 0x00000040  
    ulString - 0x00000a7d  
    \*

Row53

XPos - 0x0000000c  
XDelta - 0x00000046  
YPos - 0x00000042  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a7f

Ass&istant:

Row54

XPos - 0x00000053  
XDelta - 0x00000064  
YPos - 0x00000040  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a2e001e  
    ulSize - 0x00000040  
    ulString - 0x00000a8b  
\*

Row55

XPos - 0x000000bd  
XDelta - 0x00000046  
YPos - 0x00000042  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a8d  
Pa&ger:

Row56

XPos - 0x00000103  
XDelta - 0x00000064  
YPos - 0x00000040  
YDelta- 0x0000000c  
ControlType - 0x00000001  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x3a21001e  
    ulSize - 0x00000040  
    ulString - 0x00000a95  
\*

Row57

XPos - 0x00000006  
XDelta - 0x00000167  
YPos - 0x0000005a  
YDelta- 0x00000008

ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000a97  
Notes:

Row58

XPos - 0x00000006  
XDelta - 0x00000167  
YPos - 0x00000064  
YDelta- 0x0000002b  
ControlType - 0x00000001  
ControlFlags - 0x00000001  
ControlStructure  
    dwType - 0x3004001e  
    ulSize - 0x00000400  
    ulString - 0x00000a9f  
\*

Row59

XPos - 0x00000000  
XDelta - 0x00000000  
YPos - 0x00000000  
YDelta- 0x00000000  
ControlType - 0x00000008  
ControlFlags - 0x00000ce7  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000aa1  
Member Of

Row60

XPos - 0x00000006  
XDelta - 0x00000167  
YPos - 0x00000004  
YDelta- 0x00000008  
ControlType - 0x00000000  
ControlFlags - 0x00000000  
ControlStructure  
    dwType - 0x00000000  
    ulSize - 0x00000000  
    ulString - 0x00000aab

```

&Group membership:
Row61
  XPos - 0x00000006
  XDelta - 0x00000167
  YPos - 0x0000000e
  YDelta- 0x00000084
  ControlType - 0x00000002
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x8008000d
    ulSize - 0x00000000
    ulString - 0x00000abe
    *
Row62
  XPos - 0x00000000
  XDelta - 0x00000000
  YPos - 0x00000000
  YDelta- 0x00000000
  ControlType - 0x00000008
  ControlFlags - 0x00000ce8
  ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x00000ac0
    E-mail Addresses
Row63
  XPos - 0x00000006
  XDelta - 0x00000167
  YPos - 0x00000004
  YDelta- 0x00000008
  ControlType - 0x00000000
  ControlFlags - 0x00000000
  ControlStructure
    dwType - 0x00000000
    ulSize - 0x00000000
    ulString - 0x00000ad1
    &E-mail addresses:
Row64
  XPos - 0x00000006
  XDelta - 0x00000167
  YPos - 0x0000000e
  YDelta- 0x00000084

```



```

ControlType - 0x0000000b
ControlFlags - 0x00000000
ControlStructure
    dwType - 0x800f101e
    ulSize - 0x00000000
    ulString - 0x00000ae4

```

\*

```

0000 01 00 00 00 41 00 00 00-00 00 00 00 00 00 00 ...A.....
0010 00 00 00 00 00 00 00 00-08 00 00 00 e4 0c 00 00 .....
0020 00 00 00 00 00 00 00 00-2c 09 00 00 06 00 00 00 .....),.....
0030 67 01 00 00 03 00 00 00-29 00 00 00 06 00 00 00 .....).
0040 00 00 00 00 00 00 00 00-00 00 00 00 34 09 00 00 .....4...
0050 0c 00 00 00 46 00 00 00-0f 00 00 00 08 00 00 00 ...F.....
0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0070 39 09 00 00 53 00 00 00-25 00 00 00 0d 00 00 00 9...S.....
0080 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 06 3a .....:
0090 40 00 00 00 41 09 00 00-7b 00 00 00 2c 00 00 00 ...A...{...,...
00a0 0f 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 .....
00b0 00 00 00 00 00 00 00 00-43 09 00 00 a8 00 00 00 .....C.....
00c0 0f 00 00 00 0d 00 00 00-0c 00 00 00 01 00 00 00 .....
00d0 00 00 00 00 1e 00 0a 3a-06 00 00 00 4e 09 00 00 .....:....N...
00e0 bd 00 00 00 46 00 00 00-0f 00 00 00 08 00 00 00 ...F.....
00f0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0100 50 09 00 00 03 01 00 00-64 00 00 00 0d 00 00 00 .....d.....
0110 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 11 3a .....:
0120 40 00 00 00 57 09 00 00-0c 00 00 00 46 00 00 00 ...W.....F...
0130 1e 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 .....
0140 00 00 00 00 00 00 00 00-59 09 00 00 53 00 00 00 .....Y...S...
0150 64 00 00 00 1c 00 00 00-0c 00 00 00 01 00 00 00 d.....
0160 00 00 00 00 1e 00 01 30-00 01 00 00 63 09 00 00 .....0....c...
0170 bd 00 00 00 46 00 00 00-1e 00 00 00 08 00 00 00 ...F.....
0180 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0190 65 09 00 00 03 01 00 00-64 00 00 00 1c 00 00 00 e.....d.....
01a0 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 00 3a .....:
01b0 40 00 00 00 6d 09 00 00-0c 00 00 00 46 00 00 00 @...m.....F...
01c0 32 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 2.....
01d0 00 00 00 00 00 00 00 00-6f 09 00 00 53 00 00 00 .....o...S...
01e0 64 00 00 00 30 00 00 00-1b 00 00 00 01 00 00 00 d...0.....
01f0 01 00 00 00 1e 00 29 3a-00 04 00 00 79 09 00 00 .....):...y...
0200 0c 00 00 00 46 00 00 00-50 00 00 00 08 00 00 00 ...F...P.....
0210 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0220 7b 09 00 00 53 00 00 00-64 00 00 00 4e 00 00 00 {...S...d...N...

```

```

0230 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 27 3a .....':
0240 80 00 00 00 82 09 00 00-0c 00 00 00 46 00 00 00 .....F...
0250 5f 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 _.....
0260 00 00 00 00 00 00 00 00-84 09 00 00 53 00 00 00 .....S...
0270 64 00 00 00 5d 00 00 00-0c 00 00 00 01 00 00 00 d...].....
0280 00 00 00 00 1e 00 28 3a-80 00 00 00 8c 09 00 00 .....(:.....
0290 0c 00 00 00 46 00 00 00-6e 00 00 00 08 00 00 00 ....F...n.....
02a0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
02b0 8e 09 00 00 53 00 00 00-64 00 00 00 6c 00 00 00 ....S...d...l...
02c0 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 2a 3a .....*:
02d0 28 00 00 00 99 09 00 00-0c 00 00 00 46 00 00 00 (. ....F...
02e0 7d 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 }.....
02f0 00 00 00 00 00 00 00 00-9b 09 00 00 53 00 00 00 .....S...
0300 64 00 00 00 7b 00 00 00-0c 00 00 00 01 00 00 00 d...{.....
0310 00 00 00 00 1e 00 26 3a-03 00 00 00 ac 09 00 00 .....&:.....
0320 bd 00 00 00 46 00 00 00-32 00 00 00 08 00 00 00 ....F...2.....
0330 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0340 ae 09 00 00 03 01 00 00-64 00 00 00 30 00 00 00 .....d...0...
0350 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 17 3a .....:
0360 40 00 00 00 b6 09 00 00-bd 00 00 00 46 00 00 00 @.....F...
0370 41 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 A.....
0380 00 00 00 00 00 00 00 00-b8 09 00 00 03 01 00 00 .....
0390 64 00 00 00 3f 00 00 00-0c 00 00 00 01 00 00 00 d...?.....
03a0 00 00 00 00 1e 00 16 3a-40 00 00 00 c2 09 00 00 .....:@.....
03b0 bd 00 00 00 46 00 00 00-50 00 00 00 08 00 00 00 ....F...P.....
03c0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
03d0 c4 09 00 00 03 01 00 00-64 00 00 00 4e 00 00 00 .....d...N...
03e0 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 18 3a .....:
03f0 40 00 00 00 d1 09 00 00-bd 00 00 00 46 00 00 00 @.....F...
0400 5f 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 _.....
0410 00 00 00 00 00 00 00 00-d3 09 00 00 03 01 00 00 .....
0420 64 00 00 00 5d 00 00 00-0c 00 00 00 01 00 00 00 d...].....
0430 00 00 00 00 1e 00 19 3a-80 00 00 00 dc 09 00 00 .....:.....
0440 bd 00 00 00 46 00 00 00-6e 00 00 00 08 00 00 00 ....F...n.....
0450 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0460 de 09 00 00 03 01 00 00-64 00 00 00 6c 00 00 00 .....d...l...
0470 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 30 3a .....0:
0480 00 01 00 00 ea 09 00 00-bd 00 00 00 46 00 00 00 .....F...
0490 7d 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 }.....
04a0 00 00 00 00 00 00 00 00-ec 09 00 00 03 01 00 00 .....
04b0 64 00 00 00 7b 00 00 00-0c 00 00 00 01 00 00 00 d...{.....
04c0 00 00 00 00 1e 00 08 3a-40 00 00 00 f4 09 00 00 .....:@.....

```

```

04d0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
04e0 08 00 00 00 e5 0c 00 00-00 00 00 00 00 00 00 .....
04f0 f6 09 00 00 06 00 00 00-67 01 00 00 04 00 00 .....g.....
0500 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0510 00 00 00 00 03 0a 00 00-06 00 00 00 67 01 00 .....g...
0520 0f 00 00 00 14 00 00 00-02 00 00 00 02 00 00 .....
0530 0d 00 05 80 00 00 00 00-0d 0a 00 00 06 00 00 .....
0540 67 01 00 00 25 00 00 00-08 00 00 00 00 00 00 g.....
0550 00 00 00 00 00 00 00 00-00 00 00 00 0f 0a 00 .....
0560 06 00 00 00 67 01 00 00-30 00 00 00 60 00 00 ...g...0...`...
0570 02 00 00 00 00 00 00 00-0d 00 0e 80 00 00 00 .....
0580 20 0a 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0590 00 00 00 00 08 00 00 00-e6 0c 00 00 00 00 00 .....
05a0 00 00 00 00 22 0a 00 00-06 00 00 00 67 01 00 ...".g...
05b0 03 00 00 00 50 00 00 00-06 00 00 00 00 00 00 ...P.....
05c0 00 00 00 00 00 00 00 00-2e 0a 00 00 0c 00 00 .....
05d0 46 00 00 00 12 00 00 00-08 00 00 00 00 00 00 F.....
05e0 00 00 00 00 00 00 00 00-00 00 00 00 3c 0a 00 .....<...
05f0 53 00 00 00 64 00 00 00-10 00 00 00 0c 00 00 S...d.....
0600 01 00 00 00 00 00 00 00-1e 00 08 3a 40 00 00 .....:@...
0610 47 0a 00 00 bd 00 00 00-46 00 00 00 12 00 00 G.....F.....
0620 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0630 00 00 00 00 49 0a 00 00-03 01 00 00 64 00 00 ...I.....d...
0640 10 00 00 00 0c 00 00 00-01 00 00 00 00 00 00 .....
0650 1e 00 09 3a 40 00 00 00-50 0a 00 00 0c 00 00 ...:@...P.....
0660 46 00 00 00 22 00 00 00-08 00 00 00 00 00 00 F...".....
0670 00 00 00 00 00 00 00 00-00 00 00 00 52 0a 00 .....R...
0680 53 00 00 00 64 00 00 00-20 00 00 00 60 00 00 S...d...`...
0690 0c 00 00 00 01 00 00 00-1e 10 1b 3a 40 00 00 .....:@...
06a0 5f 0a 00 00 bd 00 00 00-46 00 00 00 22 00 00 _.....F..."...
06b0 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
06c0 00 00 00 00 61 0a 00 00-03 01 00 00 64 00 00 ...a.....d...
06d0 20 00 00 00 60 00 00 00-0c 00 00 00 01 00 00 ...`.....
06e0 1e 10 2f 3a 40 00 00 00-6a 0a 00 00 0c 00 00 ../:@...j.....
06f0 46 00 00 00 32 00 00 00-08 00 00 00 00 00 00 F...2.....
0700 00 00 00 00 00 00 00 00-00 00 00 00 6c 0a 00 .....l...
0710 53 00 00 00 64 00 00 00-30 00 00 00 0c 00 00 S...d...0.....
0720 01 00 00 00 00 00 00 00-1e 00 23 3a 40 00 00 .....#:@...
0730 72 0a 00 00 bd 00 00 00-46 00 00 00 32 00 00 r.....F...2...
0740 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0750 00 00 00 00 74 0a 00 00-03 01 00 00 64 00 00 ...t.....d...
0760 30 00 00 00 0c 00 00 00-01 00 00 00 00 00 00 0.....

```

```

0770 1e 00 1c 3a 40 00 00 00-7d 0a 00 00 0c 00 00 00 ...:@...}.....
0780 46 00 00 00 42 00 00 00-08 00 00 00 00 00 00 00 F...B.....
0790 00 00 00 00 00 00 00 00-00 00 00 00 7f 0a 00 00 .....
07a0 53 00 00 00 64 00 00 00-40 00 00 00 0c 00 00 00 S...d...@.....
07b0 01 00 00 00 00 00 00 00-1e 00 2e 3a 40 00 00 00 .....:@...
07c0 8b 0a 00 00 bd 00 00 00-46 00 00 00 42 00 00 00 .....F...B...
07d0 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
07e0 00 00 00 00 8d 0a 00 00-03 01 00 00 64 00 00 00 .....d...
07f0 40 00 00 00 0c 00 00 00-01 00 00 00 00 00 00 00 @.....
0800 1e 00 21 3a 40 00 00 00-95 0a 00 00 06 00 00 00 ..!:@.....
0810 67 01 00 00 5a 00 00 00-08 00 00 00 00 00 00 00 g...Z.....
0820 00 00 00 00 00 00 00 00-00 00 00 00 97 0a 00 00 .....
0830 06 00 00 00 67 01 00 00-64 00 00 00 2b 00 00 00 ...g...d...+...
0840 01 00 00 00 01 00 00 00-1e 00 04 30 00 04 00 00 .....0....
0850 9f 0a 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0860 00 00 00 00 08 00 00 00-e7 0c 00 00 00 00 00 00 .....
0870 00 00 00 00 a1 0a 00 00-06 00 00 00 67 01 00 00 .....g...
0880 04 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 .....
0890 00 00 00 00 00 00 00 00-ab 0a 00 00 06 00 00 00 .....
08a0 67 01 00 00 0e 00 00 00-84 00 00 00 02 00 00 00 g.....
08b0 00 00 00 00 0d 00 08 80-00 00 00 00 be 0a 00 00 .....
08c0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
08d0 08 00 00 00 e8 0c 00 00-00 00 00 00 00 00 00 00 .....
08e0 c0 0a 00 00 06 00 00 00-67 01 00 00 04 00 00 00 .....g.....
08f0 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0900 00 00 00 00 d1 0a 00 00-06 00 00 00 67 01 00 00 .....g...
0910 0e 00 00 00 84 00 00 00-0b 00 00 00 00 00 00 00 .....
0920 1e 10 0f 80 00 00 00 00-e4 0a 00 00 47 65 6e 65 .....Gene
0930 72 61 6c 00 4e 61 6d 65-00 26 46 69 72 73 74 3a ral.Name.&First:
0940 00 2a 00 49 6e 69 26 74-69 61 6c 73 3a 00 2a 00 .*&Initials:.*.
0950 26 4c 61 73 74 3a 00 2a-00 44 69 73 70 6c 61 79 &Last:.*.Display
0960 26 3a 00 2a 00 41 6c 26-69 61 73 3a 00 2a 00 41 &:.*.Alias:.*.A
0970 64 64 26 72 65 73 73 3a-00 2a 00 43 69 74 26 79 dd&ress:.*.Cit&y
0980 3a 00 2a 00 26 53 74 61-74 65 3a 00 2a 00 26 5a :.*.&State:.*.&Z
0990 69 70 20 63 6f 64 65 3a-00 2a 00 43 6f 26 75 6e ip code:.*.Co&un
09a0 74 72 79 2f 52 65 67 69-6f 6e 3a 00 2a 00 54 69 try/Region:.*.Ti
09b0 74 6c 26 65 3a 00 2a 00-43 6f 26 6d 70 61 6e 79 tl&e:.*.Co&pany
09c0 3a 00 2a 00 26 44 65 70-61 72 74 6d 65 6e 74 3a :.*.&Department:
09d0 00 2a 00 26 4f 66 66 69-63 65 3a 00 2a 00 41 73 :.*.&Office:.*.As
09e0 73 69 73 74 61 26 6e 74-3a 00 2a 00 50 26 68 6f sista&nt:.*.P&ho
09f0 6e 65 3a 00 2a 00 4f 72-67 61 6e 69 7a 61 74 69 ne:.*.Organizati
0a00 6f 6e 00 26 4d 61 6e 61-67 65 72 3a 00 2a 00 26 on.&Manager:.*.&

```

```

0a10 44 69 72 65 63 74 20 72-65 70 6f 72 74 73 3a 00 Direct reports:.
0a20 2a 00 50 68 6f 6e 65 2f-4e 6f 74 65 73 00 50 68 *.Phone/Notes.Ph
0a30 6f 6e 65 20 6e 75 6d 62-65 72 73 00 42 75 26 73 one numbers.Bu&s
0a40 69 6e 65 73 73 3a 00 2a-00 26 48 6f 6d 65 3a 00 iness:.*.&Home:.
0a50 2a 00 42 75 73 69 26 6e-65 73 73 20 32 3a 00 2a *.Busi&ness 2:.*
0a60 00 48 26 6f 6d 65 20 32-3a 00 2a 00 26 46 61 78 .H&ome 2:.*.&Fax
0a70 3a 00 2a 00 26 4d 6f 62-69 6c 65 3a 00 2a 00 41 :.*.&Mobile:.*.A
0a80 73 73 26 69 73 74 61 6e-74 3a 00 2a 00 50 61 26 ss&istant:.*.Pa&
0a90 67 65 72 3a 00 2a 00 4e-6f 26 74 65 73 3a 00 2a ger:.*.No&tes:.*
0aa0 00 4d 65 6d 62 65 72 20-4f 66 00 26 47 72 6f 75 .Member Of.&Grou
0ab0 70 20 6d 65 6d 62 65 72-73 68 69 70 3a 00 2a 00 p membership:.*.
0ac0 45 2d 6d 61 69 6c 20 41-64 64 72 65 73 73 65 73 E-mail Addresses
0ad0 00 26 45 2d 6d 61 69 6c-20 61 64 64 72 65 73 73 .&E-mail address
0ae0 65 73 3a 00 2a 00 es:.*.

```

By processing the **template** in this **PropertyRow\_r**, the dialog shown in Figure 2 is created.

**Figure 2: Address Book object display dialog**

The client then retrieves the properties specified in the **template** from the requested Address Book object to populate the various **dialog controls**.

## 5 Security

### 5.1 Security Considerations for Implementers

The execution of scripts in this protocol has to be implemented in a secure manner. The script execution checks for valid scripts, but it is also important to be aware of the possibility of infinite loops and other potential security considerations.

General security considerations that pertain to the underlying **NSPI RPC**-based transport also apply (see [MS-NSPI]).

### 5.2 Index of Security Parameters

None.

## 6 Appendix A: Office/Exchange Behavior

The information in this specification is applicable to the following versions of Office/Exchange:

- Microsoft Office 2003
- Microsoft Exchange Server 2003
- Microsoft Office 2007
- Microsoft Exchange Server 2007

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms **SHOULD** or **SHOULD NOT** implies Office/Exchange behavior in accordance with the **SHOULD** or **SHOULD NOT** prescription. Unless otherwise specified, the term **MAY** implies Office/Exchange does not follow the prescription.

---

<1> Section 3.1.4.1: Exchange 2003 and Exchange 2007 ignore the **TI\_HELPFILE\_NAME** and **TI\_HELPFILE** flags and Microsoft Office Outlook 2003 and Microsoft Office Outlook 2007 currently pass these flags but they have no effect on the results of the **NSPIGetTemplateInfo** function.

<2> Section 3.1.4.2: Exchange 2003 and Exchange 2007 ignore the **TI\_HELPFILE\_NAME** and **TI\_HELPFILE** flags and Outlook 2003 and Outlook 2007 currently pass these flags but they have no effect on the results of the **NspiGetTemplateInfo** function.

<3> Section 3.1.4.3: Exchange 2003 and Exchange 2007 currently ignore the **TI\_HELPFILE\_NAME** and **TI\_HELPFILE** flags and Outlook 2003 and Outlook 2007

---

currently pass these flags but they have no effect on the results of the **NSPIGetTemplateInfo** function.

# Index

- Applicability statement, 8
- Client details, 23
- Examples, 29
- Fields, vendor-extensible, 9
- Glossary, 5
- Index of security parameters, 62
- Informative references, 7
- Introduction, 5
- Message syntax, 9
- Messages, 9
  - Message syntax, 9
  - Transport, 9
- Normative references, 6
- Office/Exchange behavior, 62
- Overview, 7
- Preconditions, 8
- Prerequisites, 8
- Protocol details, 23
  - Client details, 23
  - Server details, 28
- References, 6
  - Informative references, 7
  - Normative references, 6
- Relationship to other protocols, 8
- Security
  - Index of security parameters, 62
  - Security considerations for implementers, 62
- Security considerations for implementers, 62
- Server details, 28
- Standards assignments, 9
- Transport, 9
- Vendor-extensible fields, 9
- Versioning and capability negotiation, 8