**[MS-OXOABKT]:**

**Address Book User Interface Templates Protocol**

Intellectual Property Rights Notice for Open Specifications Documentation

* **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
* **Copyrights**. This 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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
* **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
* **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](https://go.microsoft.com/fwlink/?LinkId=214445) or the [Microsoft Community Promise](https://go.microsoft.com/fwlink/?LinkId=214448). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
* **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](https://aka.ms/AA9ufj8).
* **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](https://www.microsoft.com/trademarks).
* **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

**Tools**. The Open Specifications documentation 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. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact dochelp@microsoft.com.

**Revision Summary**

| Date | Revision History | Revision Class | Comments |
| --- | --- | --- | --- |
| 4/4/2008 | 0.1 | Major | Initial Availability. |
| 6/27/2008 | 1.0 | Major | Initial Release. |
| 8/6/2008 | 1.0.1 | Editorial | Revised and edited technical content. |
| 9/3/2008 | 1.0.2 | Editorial | Revised and edited technical content. |
| 12/3/2008 | 1.0.3 | Editorial | Revised and edited technical content. |
| 3/4/2009 | 1.0.4 | Editorial | Revised and edited technical content. |
| 4/10/2009 | 2.0 | Major | Updated applicable product releases. |
| 7/15/2009 | 3.0 | Major | Revised and edited for technical content. |
| 11/4/2009 | 3.1.0 | Minor | Updated the technical content. |
| 2/10/2010 | 3.2.0 | Minor | Updated the technical content. |
| 5/5/2010 | 3.3.0 | Minor | Updated the technical content. |
| 8/4/2010 | 3.4 | Minor | Clarified the meaning of the technical content. |
| 11/3/2010 | 3.4 | None | No changes to the meaning, language, or formatting of the technical content. |
| 3/18/2011 | 4.0 | Major | Significantly changed the technical content. |
| 8/5/2011 | 4.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/7/2011 | 4.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 1/20/2012 | 5.0 | Major | Significantly changed the technical content. |
| 4/27/2012 | 6.0 | Major | Significantly changed the technical content. |
| 7/16/2012 | 6.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/8/2012 | 7.0 | Major | Significantly changed the technical content. |
| 2/11/2013 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/26/2013 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 11/18/2013 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/10/2014 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 4/30/2014 | 7.1 | Minor | Clarified the meaning of the technical content. |
| 7/31/2014 | 7.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/30/2014 | 8.0 | Major | Significantly changed the technical content. |
| 3/16/2015 | 9.0 | Major | Significantly changed the technical content. |
| 5/26/2015 | 10.0 | Major | Significantly changed the technical content. |
| 9/14/2015 | 10.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/13/2016 | 10.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 9/14/2016 | 10.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/24/2018 | 11.0 | Major | Significantly changed the technical content. |
| 10/1/2018 | 12.0 | Major | Significantly changed the technical content. |
| 4/22/2021 | 13.0 | Major | Significantly changed the technical content. |
| 8/17/2021 | 14.0 | Major | Significantly changed the technical content. |
| 4/16/2024 | 15.0 | Major | Significantly changed the technical content. |

Table of Contents

[1 Introduction 6](#_Toc163746823)

[1.1 Glossary 6](#_Toc163746824)

[1.2 References 7](#_Toc163746825)

[1.2.1 Normative References 7](#_Toc163746826)

[1.2.2 Informative References 8](#_Toc163746827)

[1.3 Overview 8](#_Toc163746828)

[1.3.1 Creation of New E-Mail Addresses 8](#_Toc163746829)

[1.3.2 Display and Updating of Data 8](#_Toc163746830)

[1.3.3 Collection of Search Data 8](#_Toc163746831)

[1.4 Relationship to Other Protocols 9](#_Toc163746832)

[1.5 Prerequisites/Preconditions 9](#_Toc163746833)

[1.6 Applicability Statement 9](#_Toc163746834)

[1.7 Versioning and Capability Negotiation 9](#_Toc163746835)

[1.8 Vendor-Extensible Fields 9](#_Toc163746836)

[1.9 Standards Assignments 9](#_Toc163746837)

[2 Messages 10](#_Toc163746838)

[2.1 Transport 10](#_Toc163746839)

[2.2 Message Syntax 10](#_Toc163746840)

[2.2.1 NspiGetSpecialTable PropertyRowSet\_r Format 10](#_Toc163746841)

[2.2.2 NspiGetTemplateInfo PropertyRow\_r Format 11](#_Toc163746842)

[2.2.2.1 Template Format 11](#_Toc163746843)

[2.2.2.1.1 TRowSet Structure 11](#_Toc163746844)

[2.2.2.1.2 TRow Structure 12](#_Toc163746845)

[2.2.2.1.3 Buffer Format of the CNTRL Structure 13](#_Toc163746846)

[2.2.2.1.3.1 CNTRL Structure Describing a Label Control 14](#_Toc163746847)

[2.2.2.1.3.2 CNTRL Structure Describing an Edit Control 14](#_Toc163746848)

[2.2.2.1.3.2.1 Expression Syntax for Allowed Characters 14](#_Toc163746849)

[2.2.2.1.3.3 CNTRL Structure Describing a List Box Control 15](#_Toc163746850)

[2.2.2.1.3.4 CNTRL Structure Describing a Check Box Control 15](#_Toc163746851)

[2.2.2.1.3.5 CNTRL Structure Describing a Group Box Control 15](#_Toc163746852)

[2.2.2.1.3.6 CNTRL Structure Describing a Button Control 15](#_Toc163746853)

[2.2.2.1.3.7 CNTRL Structure Describing a Tabbed Page Control 15](#_Toc163746854)

[2.2.2.1.3.8 CNTRL Structure Describing a Multi-Valued List Box Control 15](#_Toc163746855)

[2.2.2.1.3.9 CNTRL Structure Describing a Multi-Valued Drop-Down List Box Control 16](#_Toc163746856)

[2.2.2.2 Script Format 16](#_Toc163746857)

[2.2.2.2.1 Halt Instruction 16](#_Toc163746858)

[2.2.2.2.2 Error Instruction 16](#_Toc163746859)

[2.2.2.2.3 Emit String Instruction 17](#_Toc163746860)

[2.2.2.2.4 Jump Instruction 17](#_Toc163746861)

[2.2.2.2.5 Jump If Not Exists Instruction 17](#_Toc163746862)

[2.2.2.2.6 Jump If Equal Properties Instruction 18](#_Toc163746863)

[2.2.2.2.7 Jump If Equal Values Instruction 18](#_Toc163746864)

[2.2.2.2.8 Emit Property Value Instruction 19](#_Toc163746865)

[2.2.2.2.9 Emit Upper String Instruction 19](#_Toc163746866)

[2.2.2.2.10 Emit Upper Property Instruction 19](#_Toc163746867)

[3 Protocol Details 20](#_Toc163746868)

[3.1 Client Details 20](#_Toc163746869)

[3.1.1 Abstract Data Model 20](#_Toc163746870)

[3.1.1.1 Dialog Object 20](#_Toc163746871)

[3.1.1.2 Control Objects 20](#_Toc163746872)

[3.1.1.3 Address Creation Template Table 21](#_Toc163746873)

[3.1.2 Timers 21](#_Toc163746874)

[3.1.3 Initialization 21](#_Toc163746875)

[3.1.4 Higher-Layer Triggered Events 21](#_Toc163746876)

[3.1.4.1 Creating a New E-Mail Address for a Supported Address Type 21](#_Toc163746877)

[3.1.4.2 Displaying Information about an Address Book Object 22](#_Toc163746878)

[3.1.4.3 Collecting Data to Search the Address Book 22](#_Toc163746879)

[3.1.5 Message Processing Events and Sequencing Rules 22](#_Toc163746880)

[3.1.5.1 Results of NspiGetSpecialTable Call to Retrieve the Address Creation Table 23](#_Toc163746881)

[3.1.5.2 Results of NspiGetTemplateInfo Call to Retrieve the Creation Template 23](#_Toc163746882)

[3.1.5.3 Results of NspiGetTemplateInfo Call to Retrieve the Display Template 23](#_Toc163746883)

[3.1.5.4 Results of NspiGetTemplateInfo Call to Retrieve the Search Template 23](#_Toc163746884)

[3.1.6 Timer Events 24](#_Toc163746885)

[3.1.7 Other Local Events 24](#_Toc163746886)

[3.2 Server Details 24](#_Toc163746887)

[3.2.1 Abstract Data Model 24](#_Toc163746888)

[3.2.1.1 Template Objects 24](#_Toc163746889)

[3.2.1.2 Table of Supported Address Types and Name of Template to Use to Create Them 24](#_Toc163746890)

[3.2.2 Timers 24](#_Toc163746891)

[3.2.3 Initialization 24](#_Toc163746892)

[3.2.4 Higher-Layer Triggered Events 24](#_Toc163746893)

[3.2.5 Message Processing Events and Sequencing Rules 24](#_Toc163746894)

[3.2.5.1 NspiGetSpecialTable Call from Client 24](#_Toc163746895)

[3.2.5.2 NspiGetTemplateInfo Call from Client 25](#_Toc163746896)

[3.2.6 Timer Events 25](#_Toc163746897)

[3.2.7 Other Local Events 25](#_Toc163746898)

[4 Protocol Examples 26](#_Toc163746899)

[4.1 Creating a New E-Mail Address for a Supported Address Type 26](#_Toc163746900)

[4.2 Retrieving a Mail User's Template 32](#_Toc163746901)

[5 Security 48](#_Toc163746902)

[5.1 Security Considerations for Implementers 48](#_Toc163746903)

[5.2 Index of Security Parameters 48](#_Toc163746904)

[6 Appendix A: Product Behavior 49](#_Toc163746905)

[7 Change Tracking 51](#_Toc163746906)

[8 Index 52](#_Toc163746907)

# Introduction

The Address Book User Interface Templates Protocol is an extension of the Name Service Provider Interface (NSPI) Protocol, as described in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543), and the Exchange Server NSPI Protocol, as described in [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136). The Address Book User Interface Templates Protocol provides the following:

* A server-provided template for creating specific, single-use e-mail addresses.
* A server-provided layout specification that the client can use to display [**Address Book object**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633) information.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

## Glossary

This document uses the following terms:

**address book**: A collection of [**Address Book objects**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633), each of which are contained in any number of address lists.

**Address Book object**: An entity in an [**address book**](#gt_d16f7b78-c5a6-48f4-9e0f-3b205b5598b5) that contains a set of attributes, each attribute with a set of associated values.

**address creation template**: A template that describes how to present a dialog to a messaging user along with a script describing how to construct a new email address from the user's response.

**address type**: An identifier for the type of email address, such as [**SMTP**](#gt_0678be67-e739-4e33-97fe-2b03b903a379) and EX.

**Augmented Backus-Naur Form (ABNF)**: A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [[RFC5234]](https://go.microsoft.com/fwlink/?LinkId=123096).

**code page**: An ordered set of characters of a specific script in which a numerical index (code-point value) is associated with each character. Code pages are a means of providing support for character sets and keyboard layouts used in different countries. Devices such as the display and keyboard can be configured to use a specific code page and to switch from one code page (such as the United States) to another (such as Portugal) at the user's request.

**display template**: A template that describes how to display or allow a user to modify information about an [**Address Book object**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633).

**distinguished name (DN)**: (1) A name that uniquely identifies an object by using the relative distinguished name (RDN) for the object, and the names of container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in a tree.

(2) In the Active Directory directory service, the unique identifier of an object in Active Directory, as described in [[MS-ADTS]](%5BMS-ADTS%5D.pdf#Section_d243592709994c628c6d13ba31a52e1a) and [[RFC2251]](https://go.microsoft.com/fwlink/?LinkId=90325).

**double-byte character set (DBCS)**: A character set that can use more than one byte to represent a single character. A DBCS includes some characters that consist of 1 byte and some characters that consist of 2 bytes. Languages such as Chinese, Japanese, and Korean use DBCS.

**entry ID**: See EntryID.

**flags**: A set of values used to configure or report options or settings.

**handle**: Any token that can be used to identify and access an object such as a device, file, or a window.

**language code identifier (LCID)**: A 32-bit number that identifies the user interface human language dialect or variation that is supported by an application or a client computer.

**mail user**: An [**Address Book object**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633) that represents a person or entity that can receive deliverable messages.

**name service provider interface (NSPI)**: A method of performing address-book-related operations on Active Directory.

**non-Unicode**: A character set that has a restricted set of glyphs, such as Shift\_JIS or ISO-2022-JP.

**recipient**: An entity that is in an address list, can receive email messages, and contains a set of attributes. Each attribute has a set of associated values.

**remote procedure call (RPC)**: A communication protocol used primarily between client and server. The term has three definitions that are often used interchangeably: a runtime environment providing for communication facilities between computers (the RPC runtime); a set of request-and-response message exchanges between computers (the RPC exchange); and the single message from an RPC exchange (the RPC message). For more information, see [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824).

**search template**: A template that defines a dialog box which enables users to specify search criteria for [**Address Book objects**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633).

**Simple Mail Transfer Protocol (SMTP)**: A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [[RFC5321]](https://go.microsoft.com/fwlink/?LinkId=144740).

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [[RFC2119]](https://go.microsoft.com/fwlink/?LinkId=90317). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](https://go.microsoft.com/fwlink/?linkid=850906).

### Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[MS-LCID] Microsoft Corporation, "[Windows Language Code Identifier (LCID) Reference](%5BMS-LCID%5D.pdf#Section_70feba9f294e491eb6eb56532684c37f)".

[MS-NSPI] Microsoft Corporation, "[Name Service Provider Interface (NSPI) Protocol](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543)".

[MS-OXNSPI] Microsoft Corporation, "[Exchange Server Name Service Provider Interface (NSPI) Protocol](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136)".

[MS-OXOABK] Microsoft Corporation, "[Address Book Object Protocol](%5BMS-OXOABK%5D.pdf#Section_f4cf9b4c923245069e712270de217614)".

[MS-OXPROPS] Microsoft Corporation, "[Exchange Server Protocols Master Property List](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148)".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, [https://www.rfc-editor.org/info/rfc2119](https://go.microsoft.com/fwlink/?LinkId=90317)

### Informative References

[MS-OXOAB] Microsoft Corporation, "[Offline Address Book (OAB) File Format and Schema](%5BMS-OXOAB%5D.pdf#Section_b475038666ec4e69abb6208dd131c7de)".

[MS-OXPROTO] Microsoft Corporation, "[Exchange Server Protocols System Overview](%5BMS-OXPROTO%5D.pdf#Section_734ab967e43e425babe1974af56c0283)".

## Overview

The Address Book User Interface Templates Protocol is used for the following:

* Creation of new e-mail addresses for supported e-mail [**address types**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba).
* Display and updating of data for different [**Address Book objects**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633).
* Collection of data to perform searches on an [**address book**](#gt_d16f7b78-c5a6-48f4-9e0f-3b205b5598b5).

### Creation of New E-Mail Addresses

The Address Book User Interface Templates Protocol enables the creation of new e-mail addresses for supported e-mail [**address types**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba). By far the most common e-mail address type is the [**Simple Mail Transfer Protocol (SMTP)**](#gt_0678be67-e739-4e33-97fe-2b03b903a379) address type, but the server can support a number of different e-mail address types. This protocol provides a way for [**address book**](#gt_d16f7b78-c5a6-48f4-9e0f-3b205b5598b5) 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. In the first step, 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.

### Display and Updating of Data

The Address Book User Interface Templates Protocol displays and updates data for [**Address Book objects**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633). 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**](#gt_13610349-044b-4a1a-a342-8c400a854546) from the server and uses the returned template along with data that it has retrieved from the Address Book object, as described in [[MS-OXOABK]](%5BMS-OXOABK%5D.pdf#Section_f4cf9b4c923245069e712270de217614), 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.

### Collection of Search Data

The Address Book User Interface Templates Protocol enables the collection of data that will be used to search the [**address book**](#gt_d16f7b78-c5a6-48f4-9e0f-3b205b5598b5). 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**](#gt_cda912f5-be6a-438b-8b08-263ecfbbd3ad) 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**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633).

## Relationship to Other Protocols

The Address Book User Interface Templates Protocol relies on the protocols that work with [**Address Book objects**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633), properties, and tables, as described in [[MS-OXOAB]](%5BMS-OXOAB%5D.pdf#Section_b475038666ec4e69abb6208dd131c7de), [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543), and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136). This protocol also relies on the Address Book Object Protocol, as described in [[MS-OXOABK]](%5BMS-OXOABK%5D.pdf#Section_f4cf9b4c923245069e712270de217614), which is used to communicate with the server by using the underlying [**remote procedure call (RPC)**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) transport.

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [[MS-OXPROTO]](%5BMS-OXPROTO%5D.pdf#Section_734ab967e43e425babe1974af56c0283).

## Prerequisites/Preconditions

The Address Book User Interface Templates Protocol assumes that the underlying Address Book Object Protocol transport, as described in [[MS-OXOABK]](%5BMS-OXOABK%5D.pdf#Section_f4cf9b4c923245069e712270de217614), has been properly initialized.

## Applicability Statement

The Address Book User Interface Templates Protocol can be used to enable a user agent to create e-mail addresses for supported [**address types**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba) and to display, create, modify data associated with an [**Address Book object**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633).

## Versioning and Capability Negotiation

None.

## Vendor-Extensible Fields

None.

## Standards Assignments

None.

# Messages

## Transport

This protocol SHOULD[<1>](#Appendix_A_1" \o "Product behavior note 1) use the Exchange Server NSPI Protocol, as specified in [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136), and MAY[<2>](#Appendix_A_2" \o "Product behavior note 2) use the Name Service Provider interface (NSPI) Protocol, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543), as a transport for communicating between client and server.

## 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. The **NspiGetSpecialTable** function is specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.3. The **NspiGetTemplateInfo** function is specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.18.

### NspiGetSpecialTable PropertyRowSet\_r Format

The *dwFlags* parameter that is passed to the **NspiGetSpecialTable** function, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.3, affects the data that is returned in the *PropertyRowSet\_r* output parameter. The bit [**flag**](#gt_425bcab9-7911-4eae-b414-624b7a51eb5f) values for the *dwFlags* parameter for this function are specified in [MS-OXNSPI] section 2.2.1.12. The client MUST pass the **NspiAddressCreationTemplates** flag to retrieve the table of supported [**address types**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba) 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, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.3, that is contained in the *PropertyRowSet\_r* return parameter of the call.

| Property name | Description |
| --- | --- |
| **PidTagAddressType** ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.576) | **STRING** property that indicates the type of address that is associated with the new [**recipient**](#gt_53dfe4f3-05d0-41aa-8217-ecd1962b340b) created with the template. |
| **PidTagDisplayName** ([MS-OXPROPS] section 2.677) | **STRING** property that contains a user-readable identification of the address type. |
| **PidTagDisplayType**([MS-OXPROPS] section 2.680) | A **LONG** property that contains a constant that identifies the type of [**Address Book object**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633) that the new recipient will be and therefore what icon the client will display for it. The values are specified in [[MS-OXOABK]](%5BMS-OXOABK%5D.pdf#Section_f4cf9b4c923245069e712270de217614) section 2.2.3.11. |
| **PidTagEntryId** [MS-OXPROPS] section 2.684) | A **BINARY** property that contains the [**entry ID**](#gt_50f3e9cf-a07f-403a-9ae9-c5ec21b2edaf) of the template to be used to create the new recipient. This identifier can be parsed to get the [**distinguished name (DN) (2)**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) to be passed to **NspiGetTemplateInfo** function, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.18 to retrieve the template. For details about the format of permanent entry IDs, see [MS-NSPI] and [MS-OXNSPI] section 2.2.9.3.  |
| **PidTagDepth** ([MS-OXPROPS] section 2.674) | A **LONG** property that MUST be ignored. |
| **PidTagSelectable** ([MS-OXPROPS] section 2.1000) | A **BOOLEAN** property that MUST be ignored. |
| **PidTagInstanceKey** ([MS-OXPROPS] section 2.744) | A **BINARY** property that contains a unique binary value.  |

### NspiGetTemplateInfo PropertyRow\_r Format

The *dwFlags* parameter, which is passed to the **NspiGetTemplateInfo** function, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.18, affects what properties are returned in the *PropertyRow\_r* return parameter.

The following table lists the [**flags**](#gt_425bcab9-7911-4eae-b414-624b7a51eb5f) that are used by this protocol that can be passed in the *dwFlags* parameter of the **NspiGetTemplateInfo** function and the corresponding properties that are returned in the *PropertyRow\_r* return parameter.

| *dwFlags* parameter flag name | Property added to *PropertyRow\_r* parameter | Description of contents of property |
| --- | --- | --- |
| **TI\_TEMPLATE**0x00000001 | **PidTagTemplateData**0x00010102 | Binary property that contains a **TRowSet** structure followed by data that is pointed to in the **TRowSet** structure (the Template format is specified in section [2.2.2.1](#Section_da2c176808b64e9a938b7f8cfd82d164)). |
| **TI\_SCRIPT**0x00000004 | **PidTagScriptData**0x00040102 | Binary property that contains script instructions and data (the Script format is specified in section [2.2.2.2](#Section_20bb0c7ab2e641f68e49b3bc5c217368)). |

#### Template Format

The dialog template consists of a set of rows that are represented by a **TRowSet** structure, as specified in section [2.2.2.1.1](#Section_51697ef9acff4f1fb6100c2d93f8bc3e), with each **TRow** structure, as specified in section [2.2.2.1.2](#Section_01ebc1b3566e4e5f8986fb4578595114), 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 **ControlFlags** field, as specified in section 2.2.2.1.2, indicates additional information about the control, including whether it is editable. The CNTRL structure, as specified in section [2.2.2.1.3](#Section_484207b2abb844a2981bde0c723f98dd), 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.

##### TRowSet Structure

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

| Field name | Type | Size in Bytes | Description |
| --- | --- | --- | --- |
| **Type**  | **ULONG** | 4 | Type of the template. This MUST be 0x00000001. |
| **cRows**  | **ULONG**  | 4 | Count of **TRow** structures that are defined in this structure. This field MUST be followed by exactly **cRows** **TRow** structures. |
| **Row1**  | **TRow** structure | 36 | **TRow** structure that contains data about a control. |
| **Row2**  | **TRow** structure | 36 | **TRow** structure that contains data about a control. |
| ... |
| **RowN**  | **TRow** structure | 36 | Last of **cRows** **TRow** structures. |

##### 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.

| Field name | Type | Size in Bytes | Description |
| --- | --- | --- | --- |
| **XPos**  | **ULONG** | 4 | X coordinate of the upper-left corner of the control. For more details, see the text that follows this table. |
| **DeltaX**  | **ULONG**  | 4 | Width of the control. For more details, see the text that follows this table. |
| **YPos**  | **ULONG** | 4 | Y coordinate of the upper-left corner of the control. For more details, see the text that follows this table. |
| **DeltaY**  | **ULONG**  | 4 | Height of the control. For more details, see the text that follows this table. |
| **ControlType**  | **ULONG**  | 4 | Type of the control. For more details, see the text that follows this table. |
| **ControlFlags**  | **ULONG**  | 4 | [**Flags**](#gt_425bcab9-7911-4eae-b414-624b7a51eb5f) that describe the control's attributes. For more details, see the text that follows this table. |
| **ControlStructure**  | **CNTRL** structure | 12 | Structure that contains data that is relevant to a particular control type.For more details, see section [2.2.2.1.3](#Section_484207b2abb844a2981bde0c723f98dd). |

**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.

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

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

The **ControlFlags** field 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 the **ControlType** field, except for 0x00000008 (tabbed page control), as specified in the Description column. If the value of **ControlType** is 0x00000008, then the value of the **ControlFlags** field can be any value and MUST be ignored.

| Value | Meaning |
| --- | --- |
| 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 **ControlType** field is any other value except 0x00000001 (edit text box control). If it is set and the value of the **ControlType** field is not 0x00000001, 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 when the **ControlType** field is set to one of the following values: 0x00000000 (label control), 0x00000002 (list box control), 0x00000006 (group box control), 0x00000007 (button control), or 0x0000000C (multi-valued drop-down list box control). |
| 0x00000004 | This flag indicates that if the control allows changes (0x00000002 attribute set), 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 that 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 **ControlType** field is 0x00000001 (edit text box control).  |
| 0x00000020 | If this flag is set, the edit control MUST allow [**double-byte character set (DBCS)**](#gt_f4852631-44aa-45e8-b1c8-34bb299a10e1) characters. This flag MUST NOT be set if the value of the **ControlType** field is anything except 0x00000001 (edit text 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 **ControlFlags** bit is also set. |

The **ControlStructure** field is a **CNTRL** structure that contains information that is relevant to the particular type of control. For details, see section 2.2.2.1.3.

##### Buffer Format of the CNTRL Structure

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

| Field name | Type | Size | Description |
| --- | --- | --- | --- |
| **dwType**  | **DWORD** | 4 | Varies depending on the control. For details, see sections [2.2.2.1.3.1](#Section_d0b957214ef946de804759af22c071e1) through [2.2.2.1.3.9](#Section_2a0a705417764c32a3d70a6043a031d3). |
| **ulSize**  | **ULONG**  | 4 | Varies depending on the control. For details, see sections 2.2.2.1.3.1 through 2.2.2.1.3.9. |
| **ulString**  | **ULONG** | 4 | The offset in **BYTEs** from the base of the **TRowSet** structure to a null-terminated [**non-Unicode**](#gt_809bd024-a87c-4f5a-9c8c-63051bb33cb4) string. This string MUST be in the [**code page**](#gt_210637d9-9634-4652-a935-ded3cd434f38) indicated by *dwCodePage* parameter of the **NspiGetTemplateInfo** function call, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.18, and MUST be terminated by a NULL character. In these strings, the "&" (ampersand) has special meaning and indicates that the character that immediately follows it MUST be used as a shortcut key to select this control. If the control cannot be selected, the control that follows it is selected. If an "&" needs to be in the string and it SHOULD NOT have any special meaning, a sequence of "&&" can be used to indicate this.For more details about string values, usage and limitations, see sections 2.2.2.1.3.1 through 2.2.2.1.3.9. |

###### 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 the label text of a label control. The string MUST NOT be over 128 characters long, including the NULL-terminating character.

###### CNTRL Structure Describing an Edit Control

**dwType**: Property of data entered into the edit control.

**ulSize**: Number of characters allowed to be entered into the edit 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.

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 [**Augmented Backus-Naur Form (ABNF)**](#gt_24ddbbb4-b79e-4419-96ec-0fdd229c9ebf) 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, the expression is:

[AFT-Z]

If all characters except the"[" (which will need the backslash character) and Z characters are allowed, the expression is:

[~\[Z]

###### 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.

###### 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.

###### 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.

###### CNTRL Structure Describing a Button Control

**dwType**: Property that is used to perform an action. This value MUST be the **PidTagAddressBookManageDistributionList** property, as specified in [[MS-OXOABK]](%5BMS-OXOABK%5D.pdf#Section_f4cf9b4c923245069e712270de217614) section 2.2.10.2. 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.

###### 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.

###### 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.

###### 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**: MUST be ignored.

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

#### 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 ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.988) in the **PropertyRow\_r** structure, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 2.2.3, is a binary property that contains the information listed in the following table.

| Parameter name | Type | Size | Description |
| --- | --- | --- | --- |
| *Size* | **DWORD** | 4 | This parameter SHOULD[<3>](#Appendix_A_3" \o "Product behavior note 3) be included. Specifies the number of **DWORD** types of script data that follow.  |
| *ScriptData* | **DWORD** Array | Varies | Specifies a series of instructions and the data that accompanies them, as specified in sections [2.2.2.2.1](#Section_797f514a61c748de8414208596629028) through [2.2.2.2.10](#Section_1dc84fb92d684427a7fc8132d4f23157).  |

This binary script data contains a series of instructions that can be executed to format an address and the data that is needed to execute those instructions. The first **DWORD** type contains the number of **DWORD** types of instructions, "N". The next N **DWORD** types 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** type 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 does not end in error.

The instructions are specified in the following sections.

##### Halt Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Halt* | **DWORD** | 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.

##### Error Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Error* | **DWORD** | 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.

##### Emit String Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Emit String* | **DWORD** | 4 | 0x80000002 |
| *First Operand* | **DWORD** | 4 | See the text that follows this table. |

The *First Operand* is an offset in BYTEs from the start of the *ScriptData* in the **PidTagScriptData** ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.988) property's binary data to a [**non-Unicode**](#gt_809bd024-a87c-4f5a-9c8c-63051bb33cb4) 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.

##### Jump Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Jump* | **DWORD** | 4 | 0x00000003 |
| *Jump Offset* | **DWORD** | 4 | See the text that follows this table. |

The *Jump Offset* parameter is an offset in BYTEs from the start of the *ScriptData* in the **PidTagScriptData** ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.988) 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.

##### Jump If Not Exists Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Jump If Not Exists* | **DWORD** | 4 | 0x00000004 |
| *First Operand* | **DWORD** | 4 | See the text that follows this table. |
| *Jump Offset* | **DWORD** | 4 | See the text that follows this table. |

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* parameter is an offset in BYTEs from the start of the ScriptData in the **PidTagScriptData** ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.988) 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, the script is advanced over this instruction and execution continues. If the property fails to be retrieved, the script will continue execution from the instruction at the offset indicated in the *Jump Offset* parameter.

##### Jump If Equal Properties Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Jump If Equal Properties* | **DWORD** | 4 | 0x00000005 |
| *First Operand* | **DWORD** | 4 | See the text that follows this table. |
| *Second Operand* | **DWORD** | 4 | See the text that follows this table. |
| *Jump Offset* | **DWORD** | 4 | See the text that follows this table. |

The *First Operand* parameter 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**](#gt_809bd024-a87c-4f5a-9c8c-63051bb33cb4) 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* parameter 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* parameter. 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** ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.988) 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* parameter.

##### Jump If Equal Values Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Jump If Equal Values* | **DWORD** | 4 | 0x40000005 |
| *First Operand* | **DWORD** | 4 | See the text that follows this table. |
| *Second Operand* | **DWORD** | 4 | See the text that follows this table. |
| *Jump Offset* | **DWORD** | 4 | See the text that follows this table. |

The *First Operand* parameter 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**](#gt_809bd024-a87c-4f5a-9c8c-63051bb33cb4) 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** ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.988) 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*.

##### Emit Property Value Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Emit Property Value* | **DWORD** | 4 | 0x00000002 |
| *First Operand* | **DWORD** | 4 | See the text that follows this table. |

The *First Operand* parameter MUST be retrieved from the data collected by using the template. The value of the property MUST be a [**non-Unicode**](#gt_809bd024-a87c-4f5a-9c8c-63051bb33cb4) 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.

##### Emit Upper String Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Emit Upper String* | **DWORD** | 4 | 0x80000006 |
| *First Operand* | **DWORD** | 4 | See the following text. |

The *First Operand* parameter is an offset in BYTEs from the start of the ScriptData in the **PidTagScriptData** ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.988) property's binary data to a [**non-Unicode**](#gt_809bd024-a87c-4f5a-9c8c-63051bb33cb4) 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.

##### Emit Upper Property Instruction

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

| Parameter name | Type | Size | Value |
| --- | --- | --- | --- |
| *Emit Upper Property* | **DWORD** | 4 | 0x00000006 |
| *First Operand* | **DWORD** | 4 | Property of property to fetch and use as an operand. |

The *First Operand* parameter will be retrieved from the data collected by using the template. The value of the property MUST be a [**non-Unicode**](#gt_809bd024-a87c-4f5a-9c8c-63051bb33cb4) 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.

# Protocol Details

## Client Details

### 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.

#### 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.

#### 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 nine types of control objects, as described in the following table.

| Control type name | 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 list box control, 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. |
| Multi-Valued List Box | Control similar to the list box control, except that multiple items in the list can be selected simultaneously. |

#### Address Creation Template Table

An [**address creation template**](#gt_500f0788-fb8b-4b09-bd01-15e879894b94) table is a table that contains a row for each supported [**address type**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba), 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.

### Timers

None.

### Initialization

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

### Higher-Layer Triggered Events

The following section specifies the higher-layer triggered events and corresponding processing that the client MUST perform when those events take place.

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

When the client has 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**](#gt_53dfe4f3-05d0-41aa-8217-ecd1962b340b).

When the client creates a new e-mail address, first the address creation table MUST be retrieved by calling the **NspiGetSpecialTable** function, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.3, where the value of the *dwFlags* parameter is set to **NspiAddressCreationTemplates** (0x00000002) and the rest of the input parameters are initialized as specified in[MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.3. The function returns a **PropertyRowSet\_r** structure that contains the data needed to create a list of creation templates for the supported [**address types**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba). 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** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.3, can be used to determine the [**distinguished name (DN)**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) for the creation template that will be used to create the new e-mail address and its address type. The **PidTagAddressType** property([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.576) is the property in the **PropertyRow\_r** structure that indicates the e-mail address type. **PidTagEntryId** ([MS-OXPROPS] section 2.684) is the property in the **PropertyRow\_r** that can be parsed to get the DN. The **PidTagEntryId** property is a **PermanentEntryID** structure, and its format is specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.9.3.

Next, the creation dialog template that will be used to create a new e-mail address MUST be retrieved by using the **NspiGetTemplateInfo** function call, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.18, passing in the DN (2) value for the creation template 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 (0x0000004) 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\_CONTENTS (0x00000040) for a value of 0x00000065. [<4>](#Appendix_A_4" \o "Product behavior note 4)

The function's *dwCodePage* input parameter is the [**code page**](#gt_210637d9-9634-4652-a935-ded3cd434f38) in which the strings in the template are stored, and in which the *ppData* return parameter is a pointer to a **PropertyRow\_r** structure 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 ([MS-OXPROPS] section 2.682) and the address type retrieved from the selected **PropertyRow\_r** structure 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.

#### 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**](#gt_13610349-044b-4a1a-a342-8c400a854546) for the [**address book**](#gt_d16f7b78-c5a6-48f4-9e0f-3b205b5598b5) 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**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633), the **NspiGetTemplateInfo** function, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.18, 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 the **TI\_TEMPLATE** (0x00000001) flag set so the template will be retrieved and MAY contain the bits for the **TI\_HELPFILE\_NAME** (0x00000020) and **TI\_HELPFILE\_CONTENTS** (0x00000040) flags for a value of 0x00000061.[<5>](#Appendix_A_5" \o "Product behavior note 5) The *dwLocaleID* parameter contains the [**LCID**](#gt_c7f99c66-592f-4053-b62a-878c189653b6) value, as specified in [[MS-LCID]](%5BMS-LCID%5D.pdf#Section_70feba9f294e491eb6eb56532684c37f), of the template. The function's *dwCodePage* input parameter is the [**code page**](#gt_210637d9-9634-4652-a935-ded3cd434f38) in which the strings in the template are stored and the *ppData* return parameter contains a pointer to a **PropertyRow\_r** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.3, 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** ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.1043) property of the **PropertyRow\_r** structure. If the dialog is updated, 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 by using the **NspiModProps** and **NspiModLinkAtt** functions, depending on the property type. The **NspiModLinkAtt** function is used to update the distribution list membership, and **NspiModProps** is used to update the other properties. Only the changed properties SHOULD be sent to the server. The **NspiModProps** function is specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.14. The **NspiModLinkAtt** function is specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.1.15.

#### 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**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633), the client MUST retrieve the [**search template**](#gt_cda912f5-be6a-438b-8b08-263ecfbbd3ad) for the [**address book**](#gt_d16f7b78-c5a6-48f4-9e0f-3b205b5598b5) 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, the **NspiGetTemplateInfo** function, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.18, MUST be called with the *ulType* parameter set to the DT\_SEARCH (as defined in [MS-NSPI] and [MS-OXNSPI] section 2.2.1.3) and the *pDN* parameter set to 0x00000000. The *dwFlags* parameter contains a bitwise combination that MUST include the bit for the **TI\_TEMPLATE** (0x00000001) flag set so the template will be retrieved and MAY contain the bits for the **TI\_HELPFILE\_NAME** (0x00000020) and **TI\_HELPFILE\_CONTENTS** (0x00000040) flags for a value of 0x00000061. [<6>](#Appendix_A_6" \o "Product behavior note 6) The function's *dwCodePage* input parameter is the [**code page**](#gt_210637d9-9634-4652-a935-ded3cd434f38) in which the strings in the template are stored and the *ppData* return parameter contains a pointer to a **PropertyRow\_r** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.3, 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** structure, as defined in [MS-NSPI] and [MS-OXNSPI] section 2.2.5.7, from the controls that have been filled in. This **Restriction\_r** structure can be passed to **NspiGetMatches** function in the *Filter* input parameter to locate an Address Book object, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.10.

### 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.

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

The results of the **NspiGetSpecialTable** function call, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.3, with its *dwFlags* parameter set to **NspiAddressCreationTemplates** (0x00000002) is a **PropertyRowSet\_r** structure that contains the address creation table information, as specified in section [2.2.1](#Section_4ac5435ad3b448fe9c69ce85e00cf39a). 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** structure, the **PidTagDisplayName** property([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.677) can be used as the user-visible string in the list. After an address type has been selected, the **PidTagEntryId** property([MS-OXPROPS] section 2.684) in the selected **PropertyRow\_r** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.3, MUST be parsed and the [**distinguished name (DN)**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) found. This DN value MUST be used as the *pDN* parameter in a call to **NspiGetTemplateInfo** function, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.18, to retrieve the creation template and finish the creation of the e-mail address.

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

The results of the **NspiGetTemplateInfo** function call, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.18, when the [**distinguished name (DN)**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) for the creation template is passed in as the *pDN* parameter of a **PropertyRow\_r** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.3, 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. A Property Bag object is an object used to save and retrieve property values. The Property Bag object is provided as a standard interface for saving property values, independent of the data format the container uses to save its source data. 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 object. Then, by using the Property Bag object to retrieve these properties when needed, the creation script MUST be executed as specified in section [2.2.2.2](#Section_20bb0c7ab2e641f68e49b3bc5c217368) to create the new e-mail address. This e-mail address MUST be used to set the **PidTagEmailAddress** property([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.682) and the [**address type**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba) from the address creation table MUST be used to set the **PidTagAddressType** property([MS-OXPROPS] section 2.576) to create a new address.

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

The results of the **NspiGetTemplateInfo** function call, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.18, when the display type of an object is passed in as the *ulType* parameter is a **PropertyRow\_r** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.3, 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.

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

The results of the **NspiGetTemplateInfo** function call, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.18, when the display type of DT\_SEARCH is passed in as the *ulType* parameter is a **PropertyRow\_r** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.3, that contains the [**search template**](#gt_cda912f5-be6a-438b-8b08-263ecfbbd3ad) 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** structure to be used as the *Filter* input parameter in a call to the **NspiGetMatches** function, as specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.10. The **NspiGetMatches** function SHOULD handle filters that contain properties in the search template.

### Timer Events

None.

### Other Local Events

None.

## Server Details

### 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.

#### Template Objects

The server keeps a template object for each display type and for the creation template for each supported [**address type**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba) in the address creation table. These objects contain the template, and if needed, the script that will be returned from **NspiGetTemplateInfo** function in the **PropertyRow\_r** structure.

#### 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**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba), the creation templates that are associated with the address types, and any other data that is needed to construct and return the **PropertyRowSet\_r** structure when it receives a call from the **NspiGetSpecialTable** function, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.3.

### Timers

None.

### Initialization

None.

### Higher-Layer Triggered Events

None.

### Message Processing Events and Sequencing Rules

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

#### NspiGetSpecialTable Call from Client

The client calls in to the server by using the **NspiGetSpecialTable** function, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.3, with the **NspiAddressCreationTemplates** flag (0x00000002) set in the *dwFlags* parameter to retrieve the table of supported address types. The server's handling of any other [**flags**](#gt_425bcab9-7911-4eae-b414-624b7a51eb5f) that can be passed to the **NspiGetSpecialTable** function are specified in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.3. The server retrieves the table of supported address types and MUST format the table into a **PropertyRow\_r** structure, as specified in [MS-NSPI] and [MS-OXNSPI] section 2.2.3, before returning this data to the client.

#### NspiGetTemplateInfo Call from Client

The client calls in to the server by using the **NspiGetTemplateInfo** function, as specified in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.18, with the *dwFlags* parameter that contains some bitwise combination of the bit [**flags**](#gt_425bcab9-7911-4eae-b414-624b7a51eb5f) **TI\_TEMPLATE** (0x00000001) so the template will be retrieved, and **TI\_SCRIPT** (0x0000004) so the script to format the e-mail address is retrieved, as specified in section [2.2.2](#Section_68d4dd27ac444fbc9a0e20fbbf906389)). The server MUST use the display type specified in the *ulType* input parameter or the template [**DN**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) specified in the *pDN* input parameter to retrieve the template object. Finally, the server MUST create the **PropertyRow\_r** structure return parameter by using the template object and return this data to the client.

### Timer Events

None.

### Other Local Events

None.

# Protocol Examples

Starting with a connection bound to the server, the following sections include sample structures that would be returned by the [**NSPI**](#gt_e63aea5b-046b-4176-9359-fde82613a406) function call.

## 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**](#gt_435e947e-e78d-42d8-bee0-4598c959f0ba) from the server by calling the **NspiGetSpecialTable** function, as described in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.3. The first step is to bind to the server by using the **NspiBind** function, as described in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.1, call to retrieve an [**RPC**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) context [**handle**](#gt_5044babb-08e3-4bb9-bc12-fe8f542b05ee) for the server.

The **NspiGetSpecialTable** function, as described in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.3, is then called, passing the **NspiAddressCreationTemplates** flag (0x00000002) in the *dwFlags* parameter.

The following are the input parameters for the **NspiGetSpecialTable** function call.

**Note**Not all parameters are shown, only relevant information. For more information about the parameters, see [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.3.

1. dwFlags: 0x00000002
2. pStat: hIndex=0x00000000,
3. ContainerID=0xcccccccc,
4. CurrentRec=0x00000000,
5. Delta=0x00000000,
6. NumPos=0x00000000,
7. TotalRecs=0xcccccccc,
8. CodePage =0x000004e4,
9. TemplateLocale=0x00000409,
10. SortLocale=0x00000409
11. lpVersion: Not used - 0xcccccccc

The call returns a **PropertyRowSet\_r** structure in the *ppRows* parameter. The following is an example of the *ppRows* parameter that can be returned.

1. ppRows: cRows - 0x00000005
2. Row0
3. cValues - 0x00000007
4. ulAdrEntryPad - not used - 0x00000000
5. Prop0
6. ulPropTag - PidTagDisplayName (0x3001001e)
7. ulReserved - not used - 0x00000000
8. Value - cc:Mail Address
9. Prop1
10. ulPropTag - PidTagAddressType (0x3002001e)
11. ulReserved - not used - 0x00000000
12. Value - CCMAIL
13. Prop2
14. ulPropTag - PidTagDisplayType (0x39000003)
15. ulReserved - not used - 0x00000000
16. Value - 0x00000000
17. Prop3
18. ulPropTag - PidTagDepth (0x30050003)
19. ulReserved - not used - 0x00000000
20. Value - 0x00000000
21. Prop4
22. ulPropTag - PidTagSelectable (0x3609000b)
23. ulReserved - not used - 0x00000000
24. Value - 0x00000001
25. Prop5
26. ulPropTag - PidTagInstanceKey (0x0ff60102)
27. ulReserved - not used - 0x00000000
28. Value - 0x02957c9c
29. 0000 d6 23 00 00 .#..
30. Prop6
31. ulPropTag - PidTagEntryId (0x0fff0102)
32. ulReserved - not used - 0x00000000
33. Value - 0x02957ca0
34. 0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 ......@..B......
35. 0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/........../o=N
36. 0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 30 T5/ou=0000000000
37. 0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30 0000000000000000
38. 0040 30 30 30 30 30 30 2f 63-6e 3d 34 33 33 34 34 43 000000/cn=43344C
39. 0050 30 37 44 34 43 45 41 36-34 46 42 45 39 34 32 37 07D4CEA64FBE9427
40. 0060 43 44 31 36 41 31 33 43-44 34 00 CD16A13CD4.
41. Row1
42. cValues - 0x00000007
43. ulAdrEntryPad - not used - 0x00000000
44. Prop0
45. ulPropTag - PidTagDisplayName (0x3001001e)
46. ulReserved - not used - 0x00000000
47. Value - Microsoft Mail Address
48. Prop1
49. ulPropTag - PidTagAddressType (0x3002001e)
50. ulReserved - not used - 0x00000000
51. Value - MS
52. Prop2
53. ulPropTag - PidTagDisplayType (0x39000003)
54. ulReserved - not used - 0x00000000
55. Value - 0x00000000
56. Prop3
57. ulPropTag - PidTagDepth (0x30050003)
58. ulReserved - not used - 0x00000000
59. Value - 0x00000000
60. Prop4
61. ulPropTag - PidTagSelectable (0x3609000b)
62. ulReserved - not used - 0x00000000
63. Value - 0x00000001
64. Prop5
65. ulPropTag - PidTagInstanceKey (0x0ff60102)
66. ulReserved - not used - 0x00000000
67. Value - 0x02957df0
68. 0000 d3 23 00 00 .#..
69. Prop6
70. ulPropTag - PidTagEntryId (0x0fff0102)
71. ulReserved - not used - 0x00000000
72. Value - 0x02957df4
73. 0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 ......@..B......
74. 0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/........../o=N
75. 0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 30 T5/ou=0000000000
76. 0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30 0000000000000000
77. 0040 30 30 30 30 30 30 2f 63-6e 3d 37 46 32 36 33 44 000000/cn=7F263D
78. 0050 42 37 42 39 35 31 41 32-34 33 38 38 45 43 42 39 B7B951A24388ECB9
79. 0060 37 39 34 36 38 42 43 42-45 45 00 79468BCBEE.
80. Row2
81. cValues - 0x00000007
82. ulAdrEntryPad - not used - 0x00000000
83. Prop0
84. ulPropTag - PidTagDisplayName (0x3001001e)
85. ulReserved - not used - 0x00000000
86. Value - MacMail Address
87. Prop1
88. ulPropTag - PidTagAddressType (0x3002001e)
89. ulReserved - not used - 0x00000000
90. Value - MSA
91. Prop2
92. ulPropTag - PidTagDisplayType (0x39000003)
93. ulReserved - not used - 0x00000000
94. Value - 0x00000000
95. Prop3
96. ulPropTag - PidTagDepth (0x30050003)
97. ulReserved - not used - 0x00000000
98. Value - 0x00000000
99. Prop4
100. ulPropTag - PidTagSelectable (0x3609000b)
101. ulReserved - not used - 0x00000000
102. Value - 0x00000001
103. Prop5
104. ulPropTag - PidTagInstanceKey (0x0ff60102)
105. ulReserved - not used - 0x00000000
106. Value - 0x02957f40
107. 0000 d5 23 00 00 .#..
108. Prop6
109. ulPropTag - PidTagEntryId (0x0fff0102)
110. ulReserved - not used - 0x00000000
111. Value - 0x02957f44
112. 0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 ......@..B......
113. 0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/........../o=N
114. 0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 30 T5/ou=0000000000
115. 0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30 0000000000000000
116. 0040 30 30 30 30 30 30 2f 63-6e 3d 37 42 35 30 35 30 000000/cn=7B5050
117. 0050 37 33 41 44 44 41 44 33-34 39 38 33 30 42 32 43 73ADDAD349830B2C
118. 0060 35 46 41 39 38 32 36 33-44 46 00 5FA98263DF.
119. Row3
120. cValues - 0x00000007
121. ulAdrEntryPad - not used - 0x00000000
122. Prop0
123. ulPropTag - PidTagDisplayName (0x3001001e)
124. ulReserved - not used - 0x00000000
125. Value - Internet Address
126. Prop1
127. ulPropTag - PidTagAddressType (0x3002001e)
128. ulReserved - not used - 0x00000000
129. Value - SMTP
130. Prop2
131. ulPropTag - PidTagDisplayType (0x39000003)
132. ulReserved - not used - 0x00000000
133. Value - 0x00000000
134. Prop3
135. ulPropTag - PidTagDepth (0x30050003)
136. ulReserved - not used - 0x00000000
137. Value - 0x00000000
138. Prop4
139. ulPropTag - PidTagSelectable (0x3609000b)
140. ulReserved - not used - 0x00000000
141. Value - 0x00000001
142. Prop5
143. ulPropTag - PidTagInstanceKey (0x0ff60102)
144. ulReserved - not used - 0x00000000
145. Value - 0x02956320
146. 0000 d4 23 00 00 .#..
147. Prop6
148. ulPropTag - PidTagEntryId (0x0fff0102)
149. ulReserved - not used - 0x00000000
150. Value - 0x02956324
151. 0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 ......@..B......
152. 0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/........../o=N
153. 0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 30 T5/ou=0000000000
154. 0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30 0000000000000000
155. 0040 30 30 30 30 30 30 2f 63-6e 3d 41 39 36 30 39 33 000000/cn=A96093
156. 0050 42 30 45 33 34 45 43 46-34 37 38 42 38 38 42 36 B0E34ECF478B88B6
157. 0060 41 43 36 36 41 36 32 35-42 43 00 AC66A625BC.
158. Row4
159. cValues - 0x00000007
160. ulAdrEntryPad - not used - 0x00000000
161. Prop0
162. ulPropTag - PidTagDisplayName (0x3001001e)
163. ulReserved - not used - 0x00000000
164. Value - X.400 Address
165. Prop1
166. ulPropTag - PidTagAddressType (0x3002001e)
167. ulReserved - not used - 0x00000000
168. Value - X400
169. Prop2
170. ulPropTag - PidTagDisplayType (0x39000003)
171. ulReserved - not used - 0x00000000
172. Value - 0x00000000
173. Prop3
174. ulPropTag - PidTagDepth (0x30050003)
175. ulReserved - not used - 0x00000000
176. Value - 0x00000000
177. Prop4
178. ulPropTag - PidTagSelectable (0x3609000b)
179. ulReserved - not used - 0x00000000
180. Value - 0x00000001
181. Prop5
182. ulPropTag - PidTagInstanceKey (0x0ff60102)
183. ulReserved - not used - 0x00000000
184. Value - 0x02956474
185. 0000 d2 23 00 00 .#..
186. Prop6
187. ulPropTag - PidTagEntryId (0x0fff0102)
188. ulReserved - not used - 0x00000000
189. Value - 0x02956478
190. 0000 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00 ......@..B......
191. 0010 2b 2f e1 82 01 00 00 00-02 01 00 00 2f 6f 3d 4e +/........../o=N
192. 0020 54 35 2f 6f 75 3d 30 30-30 30 30 30 30 30 30 30 T5/ou=0000000000
193. 0030 30 30 30 30 30 30 30 30-30 30 30 30 30 30 30 30 0000000000000000
194. 0040 30 30 30 30 30 30 2f 63-6e 3d 34 45 38 30 41 46 000000/cn=4E80AF
195. 0050 33 41 34 37 34 44 38 46-34 45 38 46 45 39 31 41 3A474D8F4E8FE91A
196. 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** ([[MS-OXPROPS]](%5BMS-OXPROPS%5D.pdf#Section_f6ab1613aefe447da49c18217230b148) section 2.576) and **PidTagEntryId** ([MS-OXPROPS] section 2.684) property values are extracted. The **PidTagEntryId** property is parsed and the [**DN**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) is determined to be the following.

1. /o=NT5/ou=00000000000000000000000000000000/cn=43344C07D4CEA64FBE9427CD16A13CD4

This value is passed to **NSPIGetTemplateInfo** function, as described in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.18, as the *pDN* parameter to retrieve the creation template. The following are the input parameters that are passed to **NspiGetTemplateInfo** function.

1. dwFlags: 0x00000065
2. ulType: 0x00000000
3. pDN: /o=NT5/ou=00000000000000000000000000000000/cn=43344C07D4CEA64FBE9427CD16A13CD4
4. dwCodePage: 0x000004e4
5. ulLocaleID: 0x00000409

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

1. ppData
2. cValues - 0x00000002
3. ulAdrEntryPad - not used - 0x00000000
4. Prop0
5. ulPropTag - PidTagTemplateData (0x00010102)
6. ulReserved - not used - 0x00000000
7. TRowSet - Type - 0x00000001
8. cRows - 0x00000007
9. Row0
10. XPos - 0x00000000
11. XDelta - 0x00000000
12. YPos - 0x00000000
13. YDelta- 0x00000000
14. ControlType - 0x00000008
15. ControlFlags - 0x00000d70
16. ControlStructure
17. dwType - 0x00000000
18. ulSize - 0x00000000
19. ulString - 0x00000104
20. General
21. Row1
22. XPos - 0x00000006
23. XDelta - 0x00000064
24. YPos - 0x0000000c
25. YDelta- 0x00000014
26. ControlType - 0x00000000
27. ControlFlags - 0x00000000
28. ControlStructure
29. dwType - 0x00000000
30. ulSize - 0x00000000
31. ulString - 0x0000010c
32. &Display name:
33. Row2
34. XPos - 0x0000006b
35. XDelta - 0x000000fa
36. YPos - 0x0000000c
37. YDelta- 0x0000000c
38. ControlType - 0x00000001
39. ControlFlags - 0x00000026
40. ControlStructure
41. dwType - 0x3001001e
42. ulSize - 0x00000100
43. ulString - 0x0000011b
44. \*
45. Row3
46. XPos - 0x00000006
47. XDelta - 0x00000064
48. YPos - 0x00000023
49. YDelta- 0x00000014
50. ControlType - 0x00000000
51. ControlFlags - 0x00000000
52. ControlStructure
53. dwType - 0x00000000
54. ulSize - 0x00000000
55. ulString - 0x0000011d
56. &Mailbox:
57. Row4
58. XPos - 0x0000006b
59. XDelta - 0x000000fa
60. YPos - 0x00000023
61. YDelta- 0x0000000c
62. ControlType - 0x00000001
63. ControlFlags - 0x00000006
64. ControlStructure
65. dwType - 0x6701001e
66. ulSize - 0x00000100
67. ulString - 0x00000127
68. \*
69. Row5
70. XPos - 0x00000006
71. XDelta - 0x00000064
72. YPos - 0x0000003a
73. YDelta- 0x00000014
74. ControlType - 0x00000000
75. ControlFlags - 0x00000000
76. ControlStructure
77. dwType - 0x00000000
78. ulSize - 0x00000000
79. ulString - 0x00000129
80. &Post Office:
81. Row6
82. XPos - 0x0000006b
83. XDelta - 0x000000fa
84. YPos - 0x0000003a
85. YDelta- 0x0000000c
86. ControlType - 0x00000001
87. ControlFlags - 0x00000006
88. ControlStructure
89. dwType - 0x6702001e
90. ulSize - 0x00000100
91. ulString - 0x00000137
92. \*
93. 0000 01 00 00 00 07 00 00 00-00 00 00 00 00 00 00 00 ................
94. 0010 00 00 00 00 00 00 00 00-08 00 00 00 70 0d 00 00 ............p...
95. 0020 00 00 00 00 00 00 00 00-04 01 00 00 06 00 00 00 ................
96. 0030 64 00 00 00 0c 00 00 00-14 00 00 00 00 00 00 00 d...............
97. 0040 00 00 00 00 00 00 00 00-00 00 00 00 0c 01 00 00 ................
98. 0050 6b 00 00 00 fa 00 00 00-0c 00 00 00 0c 00 00 00 k...............
99. 0060 01 00 00 00 26 00 00 00-1e 00 01 30 00 01 00 00 ....&......0....
100. 0070 1b 01 00 00 06 00 00 00-64 00 00 00 23 00 00 00 ........d...#...
101. 0080 14 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
102. 0090 00 00 00 00 1d 01 00 00-6b 00 00 00 fa 00 00 00 ........k.......
103. 00a0 23 00 00 00 0c 00 00 00-01 00 00 00 06 00 00 00 #...............
104. 00b0 1e 00 01 67 00 01 00 00-27 01 00 00 06 00 00 00 ...g....'.......
105. 00c0 64 00 00 00 3a 00 00 00-14 00 00 00 00 00 00 00 d...:...........
106. 00d0 00 00 00 00 00 00 00 00-00 00 00 00 29 01 00 00 ............)...
107. 00e0 6b 00 00 00 fa 00 00 00-3a 00 00 00 0c 00 00 00 k.......:.......
108. 00f0 01 00 00 00 06 00 00 00-1e 00 02 67 00 01 00 00 ...........g....
109. 0100 37 01 00 00 47 65 6e 65-72 61 6c 00 26 44 69 73 7...General.&Dis
110. 0110 70 6c 61 79 20 6e 61 6d-65 3a 00 2a 00 26 4d 61 play name:.\*.&Ma
111. 0120 69 6c 62 6f 78 3a 00 2a-00 26 50 6f 73 74 20 4f ilbox:.\*.&Post O
112. 0130 66 66 69 63 65 3a 00 2a-00 ffice:.\*.
113. Prop1
114. ulPropTag - PidTagScriptData (0x00040102)
115. ulReserved - not used - 0x00000000
116. Size – 0x0000000F
117. Operation Jump Not Exists - 0x00000004
118. PropTag - 0x6701001e
119. Offset - 0x00000014
120. Operation Emit - 0x00000002
121. PropTag - 0x6701001e
122. Operation Emit String - 0x80000002
123. Offset - 0x00000034
124. Operation Jump Not Exists - 0x00000004
125. PropTag - 0x6702001e
126. Offset - 0x00000030
127. Operation Emit - 0x00000002
128. PropTag - 0x6702001e
129. Operation Halt - 0x00000000
130. 0000 0f 00 00 00 04 00 00 00-1e 00 01 67 14 00 00 00 ...........g....
131. 0010 02 00 00 00 1e 00 01 67-02 00 00 80 34 00 00 00 .......g....4...
132. 0020 04 00 00 00 1e 00 02 67-30 00 00 00 02 00 00 00 .......g0.......
133. 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 box similar to the one shown in the following figure.



Figure 1: Address creation dialog box

The following data is then entered into the dialog box:

Display name: Bob

Mailbox: BobsMailbox

Post office: GeneralPostOffice

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

BobsMailbox at GeneralPostOffice

Therefore, the **PidTagEmailAddress** property([MS-OXPROPS] section 2.682) that represents this user is "BobsMailbox at GeneralPostOffice" and the **PidTagAddressType** property value is "CCMAIL."

## Retrieving a Mail User's Template

To display information about an [**Address Book object**](#gt_4792b6d3-b01a-43f6-aca4-42fc4e79a633) to the user, the client has to first request the [**display template**](#gt_13610349-044b-4a1a-a342-8c400a854546) from the server by calling the **NspiGetTemplateInfo** function, as described in [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 3.1.4.1.18. The first step is to bind to the server by using the **NspiBind** function call, as described in [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.1, to retrieve an [**RPC**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) context [**handle**](#gt_5044babb-08e3-4bb9-bc12-fe8f542b05ee) for the server.

For example, the **NspiGetTemplateInfo** function is called to get the template to display data about a [**mail user**](#gt_70581e4e-e0b5-4f4e-ab02-6a0d29f6fccf) by passing the *ulType* parameter with the mail user display type (**DT\_MAILUSER**).

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

**Note** Only relevant information, andnot all parameters, is shown. For more information about the parameters, see [MS-NSPI] and [MS-OXNSPI] section 3.1.4.1.18.

1. dwFlags: 0x00000061
2. ulType: 0x00000000
3. pDN: (null)
4. dwCodePage: 0x000004e4
5. dwLocaleID: 0x00000409

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

1. ppData
2. cValues - 0x00000001
3. ulAdrEntryPad - not used - 0x00000000
4. Prop0
5. ulPropTag - PidTagTemplateData (0x00010102)
6. ulReserved - not used - 0x00000000
7. TRowSet - Type - 0x00000001
8. cRows - 0x00000041
9. Row0
10. XPos - 0x00000000
11. XDelta - 0x00000000
12. YPos - 0x00000000
13. YDelta- 0x00000000
14. ControlType - 0x00000008
15. ControlFlags - 0x00000ce4
16. ControlStructure
17. dwType - 0x00000000
18. ulSize - 0x00000000
19. ulString - 0x0000092c
20. General
21. Row1
22. XPos - 0x00000006
23. XDelta - 0x00000167
24. YPos - 0x00000003
25. YDelta- 0x00000029
26. ControlType - 0x00000006
27. ControlFlags - 0x00000000
28. ControlStructure
29. dwType - 0x00000000
30. ulSize - 0x00000000
31. ulString - 0x00000934
32. Name
33. Row2
34. XPos - 0x0000000c
35. XDelta - 0x00000046
36. YPos - 0x0000000f
37. YDelta- 0x00000008
38. ControlType - 0x00000000
39. ControlFlags - 0x00000000
40. ControlStructure
41. dwType - 0x00000000
42. ulSize - 0x00000000
43. ulString - 0x00000939
44. &First:
45. Row3
46. XPos - 0x00000053
47. XDelta - 0x00000025
48. YPos - 0x0000000d
49. YDelta- 0x0000000c
50. ControlType - 0x00000001
51. ControlFlags - 0x00000000
52. ControlStructure
53. dwType - 0x3a06001e
54. ulSize - 0x00000040
55. ulString - 0x00000941
56. \*
57. Row4
58. XPos - 0x0000007b
59. XDelta - 0x0000002c
60. YPos - 0x0000000f
61. YDelta- 0x00000008
62. ControlType - 0x00000000
63. ControlFlags - 0x00000000
64. ControlStructure
65. dwType - 0x00000000
66. ulSize - 0x00000000
67. ulString - 0x00000943
68. Ini&tials:
69. Row5
70. XPos - 0x000000a8
71. XDelta - 0x0000000f
72. YPos - 0x0000000d
73. YDelta- 0x0000000c
74. ControlType - 0x00000001
75. ControlFlags - 0x00000000
76. ControlStructure
77. dwType - 0x3a0a001e
78. ulSize - 0x00000006
79. ulString - 0x0000094e
80. \*
81. Row6
82. XPos - 0x000000bd
83. XDelta - 0x00000046
84. YPos - 0x0000000f
85. YDelta- 0x00000008
86. ControlType - 0x00000000
87. ControlFlags - 0x00000000
88. ControlStructure
89. dwType - 0x00000000
90. ulSize - 0x00000000
91. ulString - 0x00000950
92. &Last:
93. Row7
94. XPos - 0x00000103
95. XDelta - 0x00000064
96. YPos - 0x0000000d
97. YDelta- 0x0000000c
98. ControlType - 0x00000001
99. ControlFlags - 0x00000000
100. ControlStructure
101. dwType - 0x3a11001e
102. ulSize - 0x00000040
103. ulString - 0x00000957
104. \*
105. Row8
106. XPos - 0x0000000c
107. XDelta - 0x00000046
108. YPos - 0x0000001e
109. YDelta- 0x00000008
110. ControlType - 0x00000000
111. ControlFlags - 0x00000000
112. ControlStructure
113. dwType - 0x00000000
114. ulSize - 0x00000000
115. ulString - 0x00000959
116. Display&:
117. Row9
118. XPos - 0x00000053
119. XDelta - 0x00000064
120. YPos - 0x0000001c
121. YDelta- 0x0000000c
122. ControlType - 0x00000001
123. ControlFlags - 0x00000000
124. ControlStructure
125. dwType - 0x3001001e
126. ulSize - 0x00000100
127. ulString - 0x00000963
128. \*
129. Row10
130. XPos - 0x000000bd
131. XDelta - 0x00000046
132. YPos - 0x0000001e
133. YDelta- 0x00000008
134. ControlType - 0x00000000
135. ControlFlags - 0x00000000
136. ControlStructure
137. dwType - 0x00000000
138. ulSize - 0x00000000
139. ulString - 0x00000965
140. Al&ias:
141. Row11
142. XPos - 0x00000103
143. XDelta - 0x00000064
144. YPos - 0x0000001c
145. YDelta- 0x0000000c
146. ControlType - 0x00000001
147. ControlFlags - 0x00000000
148. ControlStructure
149. dwType - 0x3a00001e
150. ulSize - 0x00000040
151. ulString - 0x0000096d
152. \*
153. Row12
154. XPos - 0x0000000c
155. XDelta - 0x00000046
156. YPos - 0x00000032
157. YDelta- 0x00000008
158. ControlType - 0x00000000
159. ControlFlags - 0x00000000
160. ControlStructure
161. dwType - 0x00000000
162. ulSize - 0x00000000
163. ulString - 0x0000096f
164. Add&ress:
165. Row13
166. XPos - 0x00000053
167. XDelta - 0x00000064
168. YPos - 0x00000030
169. YDelta- 0x0000001b
170. ControlType - 0x00000001
171. ControlFlags - 0x00000001
172. ControlStructure
173. dwType - 0x3a29001e
174. ulSize - 0x00000400
175. ulString - 0x00000979
176. \*
177. Row14
178. XPos - 0x0000000c
179. XDelta - 0x00000046
180. YPos - 0x00000050
181. YDelta- 0x00000008
182. ControlType - 0x00000000
183. ControlFlags - 0x00000000
184. ControlStructure
185. dwType - 0x00000000
186. ulSize - 0x00000000
187. ulString - 0x0000097b
188. Cit&y:
189. Row15
190. XPos - 0x00000053
191. XDelta - 0x00000064
192. YPos - 0x0000004e
193. YDelta- 0x0000000c
194. ControlType - 0x00000001
195. ControlFlags - 0x00000000
196. ControlStructure
197. dwType - 0x3a27001e
198. ulSize - 0x00000080
199. ulString - 0x00000982
200. \*
201. Row16
202. XPos - 0x0000000c
203. XDelta - 0x00000046
204. YPos - 0x0000005f
205. YDelta- 0x00000008
206. ControlType - 0x00000000
207. ControlFlags - 0x00000000
208. ControlStructure
209. dwType - 0x00000000
210. ulSize - 0x00000000
211. ulString - 0x00000984
212. &State:
213. Row17
214. XPos - 0x00000053
215. XDelta - 0x00000064
216. YPos - 0x0000005d
217. YDelta- 0x0000000c
218. ControlType - 0x00000001
219. ControlFlags - 0x00000000
220. ControlStructure
221. dwType - 0x3a28001e
222. ulSize - 0x00000080
223. ulString - 0x0000098c
224. \*
225. Row18
226. XPos - 0x0000000c
227. XDelta - 0x00000046
228. YPos - 0x0000006e
229. YDelta- 0x00000008
230. ControlType - 0x00000000
231. ControlFlags - 0x00000000
232. ControlStructure
233. dwType - 0x00000000
234. ulSize - 0x00000000
235. ulString - 0x0000098e
236. &Zip code:
237. Row19
238. XPos - 0x00000053
239. XDelta - 0x00000064
240. YPos - 0x0000006c
241. YDelta- 0x0000000c
242. ControlType - 0x00000001
243. ControlFlags - 0x00000000
244. ControlStructure
245. dwType - 0x3a2a001e
246. ulSize - 0x00000028
247. ulString - 0x00000999
248. \*
249. Row20
250. XPos - 0x0000000c
251. XDelta - 0x00000046
252. YPos - 0x0000007d
253. YDelta- 0x00000008
254. ControlType - 0x00000000
255. ControlFlags - 0x00000000
256. ControlStructure
257. dwType - 0x00000000
258. ulSize - 0x00000000
259. ulString - 0x0000099b
260. Co&untry/Region:
261. Row21
262. XPos - 0x00000053
263. XDelta - 0x00000064
264. YPos - 0x0000007b
265. YDelta- 0x0000000c
266. ControlType - 0x00000001
267. ControlFlags - 0x00000000
268. ControlStructure
269. dwType - 0x3a26001e
270. ulSize - 0x00000003
271. ulString - 0x000009ac
272. \*
273. Row22
274. XPos - 0x000000bd
275. XDelta - 0x00000046
276. YPos - 0x00000032
277. YDelta- 0x00000008
278. ControlType - 0x00000000
279. ControlFlags - 0x00000000
280. ControlStructure
281. dwType - 0x00000000
282. ulSize - 0x00000000
283. ulString - 0x000009ae
284. Titl&e:
285. Row23
286. XPos - 0x00000103
287. XDelta - 0x00000064
288. YPos - 0x00000030
289. YDelta- 0x0000000c
290. ControlType - 0x00000001
291. ControlFlags - 0x00000000
292. ControlStructure
293. dwType - 0x3a17001e
294. ulSize - 0x00000040
295. ulString - 0x000009b6
296. \*
297. Row24
298. XPos - 0x000000bd
299. XDelta - 0x00000046
300. YPos - 0x00000041
301. YDelta- 0x00000008
302. ControlType - 0x00000000
303. ControlFlags - 0x00000000
304. ControlStructure
305. dwType - 0x00000000
306. ulSize - 0x00000000
307. ulString - 0x000009b8
308. Co&mpany:
309. Row25
310. XPos - 0x00000103
311. XDelta - 0x00000064
312. YPos - 0x0000003f
313. YDelta- 0x0000000c
314. ControlType - 0x00000001
315. ControlFlags - 0x00000000
316. ControlStructure
317. dwType - 0x3a16001e
318. ulSize - 0x00000040
319. ulString - 0x000009c2
320. \*
321. Row26
322. XPos - 0x000000bd
323. XDelta - 0x00000046
324. YPos - 0x00000050
325. YDelta- 0x00000008
326. ControlType - 0x00000000
327. ControlFlags - 0x00000000
328. ControlStructure
329. dwType - 0x00000000
330. ulSize - 0x00000000
331. ulString - 0x000009c4
332. &Department:
333. Row27
334. XPos - 0x00000103
335. XDelta - 0x00000064
336. YPos - 0x0000004e
337. YDelta- 0x0000000c
338. ControlType - 0x00000001
339. ControlFlags - 0x00000000
340. ControlStructure
341. dwType - 0x3a18001e
342. ulSize - 0x00000040
343. ulString - 0x000009d1
344. \*
345. Row28
346. XPos - 0x000000bd
347. XDelta - 0x00000046
348. YPos - 0x0000005f
349. YDelta- 0x00000008
350. ControlType - 0x00000000
351. ControlFlags - 0x00000000
352. ControlStructure
353. dwType - 0x00000000
354. ulSize - 0x00000000
355. ulString - 0x000009d3
356. &Office:
357. Row29
358. XPos - 0x00000103
359. XDelta - 0x00000064
360. YPos - 0x0000005d
361. YDelta- 0x0000000c
362. ControlType - 0x00000001
363. ControlFlags - 0x00000000
364. ControlStructure
365. dwType - 0x3a19001e
366. ulSize - 0x00000080
367. ulString - 0x000009dc
368. \*
369. Row30
370. XPos - 0x000000bd
371. XDelta - 0x00000046
372. YPos - 0x0000006e
373. YDelta- 0x00000008
374. ControlType - 0x00000000
375. ControlFlags - 0x00000000
376. ControlStructure
377. dwType - 0x00000000
378. ulSize - 0x00000000
379. ulString - 0x000009de
380. Assista&nt:
381. Row31
382. XPos - 0x00000103
383. XDelta - 0x00000064
384. YPos - 0x0000006c
385. YDelta- 0x0000000c
386. ControlType - 0x00000001
387. ControlFlags - 0x00000000
388. ControlStructure
389. dwType - 0x3a30001e
390. ulSize - 0x00000100
391. ulString - 0x000009ea
392. \*
393. Row32
394. XPos - 0x000000bd
395. XDelta - 0x00000046
396. YPos - 0x0000007d
397. YDelta- 0x00000008
398. ControlType - 0x00000000
399. ControlFlags - 0x00000000
400. ControlStructure
401. dwType - 0x00000000
402. ulSize - 0x00000000
403. ulString - 0x000009ec
404. P&hone:
405. Row33
406. XPos - 0x00000103
407. XDelta - 0x00000064
408. YPos - 0x0000007b
409. YDelta- 0x0000000c
410. ControlType - 0x00000001
411. ControlFlags - 0x00000000
412. ControlStructure
413. dwType - 0x3a08001e
414. ulSize - 0x00000040
415. ulString - 0x000009f4
416. \*
417. Row34
418. XPos - 0x00000000
419. XDelta - 0x00000000
420. YPos - 0x00000000
421. YDelta- 0x00000000
422. ControlType - 0x00000008
423. ControlFlags - 0x00000ce5
424. ControlStructure
425. dwType - 0x00000000
426. ulSize - 0x00000000
427. ulString - 0x000009f6
428. Organization
429. Row35
430. XPos - 0x00000006
431. XDelta - 0x00000167
432. YPos - 0x00000004
433. YDelta- 0x00000008
434. ControlType - 0x00000000
435. ControlFlags - 0x00000000
436. ControlStructure
437. dwType - 0x00000000
438. ulSize - 0x00000000
439. ulString - 0x00000a03
440. &Manager:
441. Row36
442. XPos - 0x00000006
443. XDelta - 0x00000167
444. YPos - 0x0000000f
445. YDelta- 0x00000014
446. ControlType - 0x00000002
447. ControlFlags - 0x00000002
448. ControlStructure
449. dwType - 0x8005000d
450. ulSize - 0x00000000
451. ulString - 0x00000a0d
452. \*
453. Row37
454. XPos - 0x00000006
455. XDelta - 0x00000167
456. YPos - 0x00000025
457. YDelta- 0x00000008
458. ControlType - 0x00000000
459. ControlFlags - 0x00000000
460. ControlStructure
461. dwType - 0x00000000
462. ulSize - 0x00000000
463. ulString - 0x00000a0f
464. &Direct reports:
465. Row38
466. XPos - 0x00000006
467. XDelta - 0x00000167
468. YPos - 0x00000030
469. YDelta- 0x00000060
470. ControlType - 0x00000002
471. ControlFlags - 0x00000000
472. ControlStructure
473. dwType - 0x800e000d
474. ulSize - 0x00000000
475. ulString - 0x00000a20
476. \*
477. Row39
478. XPos - 0x00000000
479. XDelta - 0x00000000
480. YPos - 0x00000000
481. YDelta- 0x00000000
482. ControlType - 0x00000008
483. ControlFlags - 0x00000ce6
484. ControlStructure
485. dwType - 0x00000000
486. ulSize - 0x00000000
487. ulString - 0x00000a22
488. Phone/Notes
489. Row40
490. XPos - 0x00000006
491. XDelta - 0x00000167
492. YPos - 0x00000003
493. YDelta- 0x00000050
494. ControlType - 0x00000006
495. ControlFlags - 0x00000000
496. ControlStructure
497. dwType - 0x00000000
498. ulSize - 0x00000000
499. ulString - 0x00000a2e
500. Phone numbers
501. Row41
502. XPos - 0x0000000c
503. XDelta - 0x00000046
504. YPos - 0x00000012
505. YDelta- 0x00000008
506. ControlType - 0x00000000
507. ControlFlags - 0x00000000
508. ControlStructure
509. dwType - 0x00000000
510. ulSize - 0x00000000
511. ulString - 0x00000a3c
512. Bu&siness:
513. Row42
514. XPos - 0x00000053
515. XDelta - 0x00000064
516. YPos - 0x00000010
517. YDelta- 0x0000000c
518. ControlType - 0x00000001
519. ControlFlags - 0x00000000
520. ControlStructure
521. dwType - 0x3a08001e
522. ulSize - 0x00000040
523. ulString - 0x00000a47
524. \*
525. Row43
526. XPos - 0x000000bd
527. XDelta - 0x00000046
528. YPos - 0x00000012
529. YDelta- 0x00000008
530. ControlType - 0x00000000
531. ControlFlags - 0x00000000
532. ControlStructure
533. dwType - 0x00000000
534. ulSize - 0x00000000
535. ulString - 0x00000a49
536. &Home:
537. Row44
538. XPos - 0x00000103
539. XDelta - 0x00000064
540. YPos - 0x00000010
541. YDelta- 0x0000000c
542. ControlType - 0x00000001
543. ControlFlags - 0x00000000
544. ControlStructure
545. dwType - 0x3a09001e
546. ulSize - 0x00000040
547. ulString - 0x00000a50
548. \*
549. Row45
550. XPos - 0x0000000c
551. XDelta - 0x00000046
552. YPos - 0x00000022
553. YDelta- 0x00000008
554. ControlType - 0x00000000
555. ControlFlags - 0x00000000
556. ControlStructure
557. dwType - 0x00000000
558. ulSize - 0x00000000
559. ulString - 0x00000a52
560. Busi&ness 2:
561. Row46
562. XPos - 0x00000053
563. XDelta - 0x00000064
564. YPos - 0x00000020
565. YDelta- 0x00000060
566. ControlType - 0x0000000c
567. ControlFlags - 0x00000001
568. ControlStructure
569. dwType - 0x3a1b101e
570. ulSize - 0x00000040
571. ulString - 0x00000a5f
572. \*
573. Row47
574. XPos - 0x000000bd
575. XDelta - 0x00000046
576. YPos - 0x00000022
577. YDelta- 0x00000008
578. ControlType - 0x00000000
579. ControlFlags - 0x00000000
580. ControlStructure
581. dwType - 0x00000000
582. ulSize - 0x00000000
583. ulString - 0x00000a61
584. H&ome 2:
585. Row48
586. XPos - 0x00000103
587. XDelta - 0x00000064
588. YPos - 0x00000020
589. YDelta- 0x00000060
590. ControlType - 0x0000000c
591. ControlFlags - 0x00000001
592. ControlStructure
593. dwType - 0x3a2f101e
594. ulSize - 0x00000040
595. ulString - 0x00000a6a
596. \*
597. Row49
598. XPos - 0x0000000c
599. XDelta - 0x00000046
600. YPos - 0x00000032
601. YDelta- 0x00000008
602. ControlType - 0x00000000
603. ControlFlags - 0x00000000
604. ControlStructure
605. dwType - 0x00000000
606. ulSize - 0x00000000
607. ulString - 0x00000a6c
608. &Fax:
609. Row50
610. XPos - 0x00000053
611. XDelta - 0x00000064
612. YPos - 0x00000030
613. YDelta- 0x0000000c
614. ControlType - 0x00000001
615. ControlFlags - 0x00000000
616. ControlStructure
617. dwType - 0x3a23001e
618. ulSize - 0x00000040
619. ulString - 0x00000a72
620. \*
621. Row51
622. XPos - 0x000000bd
623. XDelta - 0x00000046
624. YPos - 0x00000032
625. YDelta- 0x00000008
626. ControlType - 0x00000000
627. ControlFlags - 0x00000000
628. ControlStructure
629. dwType - 0x00000000
630. ulSize - 0x00000000
631. ulString - 0x00000a74
632. &Mobile:
633. Row52
634. XPos - 0x00000103
635. XDelta - 0x00000064
636. YPos - 0x00000030
637. YDelta- 0x0000000c
638. ControlType - 0x00000001
639. ControlFlags - 0x00000000
640. ControlStructure
641. dwType - 0x3a1c001e
642. ulSize - 0x00000040
643. ulString - 0x00000a7d
644. \*
645. Row53
646. XPos - 0x0000000c
647. XDelta - 0x00000046
648. YPos - 0x00000042
649. YDelta- 0x00000008
650. ControlType - 0x00000000
651. ControlFlags - 0x00000000
652. ControlStructure
653. dwType - 0x00000000
654. ulSize - 0x00000000
655. ulString - 0x00000a7f
656. Ass&istant:
657. Row54
658. XPos - 0x00000053
659. XDelta - 0x00000064
660. YPos - 0x00000040
661. YDelta- 0x0000000c
662. ControlType - 0x00000001
663. ControlFlags - 0x00000000
664. ControlStructure
665. dwType - 0x3a2e001e
666. ulSize - 0x00000040
667. ulString - 0x00000a8b
668. \*
669. Row55
670. XPos - 0x000000bd
671. XDelta - 0x00000046
672. YPos - 0x00000042
673. YDelta- 0x00000008
674. ControlType - 0x00000000
675. ControlFlags - 0x00000000
676. ControlStructure
677. dwType - 0x00000000
678. ulSize - 0x00000000
679. ulString - 0x00000a8d
680. Pa&ger:
681. Row56
682. XPos - 0x00000103
683. XDelta - 0x00000064
684. YPos - 0x00000040
685. YDelta- 0x0000000c
686. ControlType - 0x00000001
687. ControlFlags - 0x00000000
688. ControlStructure
689. dwType - 0x3a21001e
690. ulSize - 0x00000040
691. ulString - 0x00000a95
692. \*
693. Row57
694. XPos - 0x00000006
695. XDelta - 0x00000167
696. YPos - 0x0000005a
697. YDelta- 0x00000008
698. ControlType - 0x00000000
699. ControlFlags - 0x00000000
700. ControlStructure
701. dwType - 0x00000000
702. ulSize - 0x00000000
703. ulString - 0x00000a97
704. No&tes:
705. Row58
706. XPos - 0x00000006
707. XDelta - 0x00000167
708. YPos - 0x00000064
709. YDelta- 0x0000002b
710. ControlType - 0x00000001
711. ControlFlags - 0x00000001
712. ControlStructure
713. dwType - 0x3004001e
714. ulSize - 0x00000400
715. ulString - 0x00000a9f
716. \*
717. Row59
718. XPos - 0x00000000
719. XDelta - 0x00000000
720. YPos - 0x00000000
721. YDelta- 0x00000000
722. ControlType - 0x00000008
723. ControlFlags - 0x00000ce7
724. ControlStructure
725. dwType - 0x00000000
726. ulSize - 0x00000000
727. ulString - 0x00000aa1
728. Member Of
729. Row60
730. XPos - 0x00000006
731. XDelta - 0x00000167
732. YPos - 0x00000004
733. YDelta- 0x00000008
734. ControlType - 0x00000000
735. ControlFlags - 0x00000000
736. ControlStructure
737. dwType - 0x00000000
738. ulSize - 0x00000000
739. ulString - 0x00000aab
740. &Group membership:
741. Row61
742. XPos - 0x00000006
743. XDelta - 0x00000167
744. YPos - 0x0000000e
745. YDelta- 0x00000084
746. ControlType - 0x00000002
747. ControlFlags - 0x00000000
748. ControlStructure
749. dwType - 0x8008000d
750. ulSize - 0x00000000
751. ulString - 0x00000abe
752. \*
753. Row62
754. XPos - 0x00000000
755. XDelta - 0x00000000
756. YPos - 0x00000000
757. YDelta- 0x00000000
758. ControlType - 0x00000008
759. ControlFlags - 0x00000ce8
760. ControlStructure
761. dwType - 0x00000000
762. ulSize - 0x00000000
763. ulString - 0x00000ac0
764. E-mail Addresses
765. Row63
766. XPos - 0x00000006
767. XDelta - 0x00000167
768. YPos - 0x00000004
769. YDelta- 0x00000008
770. ControlType - 0x00000000
771. ControlFlags - 0x00000000
772. ControlStructure
773. dwType - 0x00000000
774. ulSize - 0x00000000
775. ulString - 0x00000ad1
776. &E-mail addresses:
777. Row64
778. XPos - 0x00000006
779. XDelta - 0x00000167
780. YPos - 0x0000000e
781. YDelta- 0x00000084
782. ControlType - 0x0000000b
783. ControlFlags - 0x00000000
784. ControlStructure
785. dwType - 0x800f101e
786. ulSize - 0x00000000
787. ulString - 0x00000ae4
788. \*
789. 0000 01 00 00 00 41 00 00 00-00 00 00 00 00 00 00 00 ...A...........
790. 0010 00 00 00 00 00 00 00 00-08 00 00 00 e4 0c 00 00 ...............
791. 0020 00 00 00 00 00 00 00 00-2c 09 00 00 06 00 00 00 .......,.......
792. 0030 67 01 00 00 03 00 00 00-29 00 00 00 06 00 00 00 .......).......
793. 0040 00 00 00 00 00 00 00 00-00 00 00 00 34 09 00 00 ...........4...
794. 0050 0c 00 00 00 46 00 00 00-0f 00 00 00 08 00 00 00 ...F...........
795. 0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ...............
796. 0070 39 09 00 00 53 00 00 00-25 00 00 00 0d 00 00 00 9...S.........
797. 0080 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 06 3a ..............:
798. 0090 40 00 00 00 41 09 00 00-7b 00 00 00 2c 00 00 00 ...A...{...,...
799. 00a0 0f 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 ...............
800. 00b0 00 00 00 00 00 00 00 00-43 09 00 00 a8 00 00 00 .......C.......
801. 00c0 0f 00 00 00 0d 00 00 00-0c 00 00 00 01 00 00 00 ...............
802. 00d0 00 00 00 00 1e 00 0a 3a-06 00 00 00 4e 09 00 00 ......:....N...
803. 00e0 bd 00 00 00 46 00 00 00-0f 00 00 00 08 00 00 00 ...F...........
804. 00f0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ...............
805. 0100 50 09 00 00 03 01 00 00-64 00 00 00 0d 00 00 00 .......d.......
806. 0110 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 11 3a ..............:
807. 0120 40 00 00 00 57 09 00 00-0c 00 00 00 46 00 00 00 ...W.......F...
808. 0130 1e 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 ................
809. 0140 00 00 00 00 00 00 00 00-59 09 00 00 53 00 00 00 ........Y...S...
810. 0150 64 00 00 00 1c 00 00 00-0c 00 00 00 01 00 00 00 d...............
811. 0160 00 00 00 00 1e 00 01 30-00 01 00 00 63 09 00 00 .......0....c...
812. 0170 bd 00 00 00 46 00 00 00-1e 00 00 00 08 00 00 00 ....F...........
813. 0180 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
814. 0190 65 09 00 00 03 01 00 00-64 00 00 00 1c 00 00 00 e.......d.......
815. 01a0 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 00 3a ...............:
816. 01b0 40 00 00 00 6d 09 00 00-0c 00 00 00 46 00 00 00 @...m.......F...
817. 01c0 32 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 2...............
818. 01d0 00 00 00 00 00 00 00 00-6f 09 00 00 53 00 00 00 ........o...S...
819. 01e0 64 00 00 00 30 00 00 00-1b 00 00 00 01 00 00 00 d...0...........
820. 01f0 01 00 00 00 1e 00 29 3a-00 04 00 00 79 09 00 00 ......):....y...
821. 0200 0c 00 00 00 46 00 00 00-50 00 00 00 08 00 00 00 ....F...P.......
822. 0210 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00................
823. 0220 7b 09 00 00 53 00 00 00-64 00 00 00 4e 00 00 00 {...S...d...N...
824. 0230 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 27 3a ..............':
825. 0240 80 00 00 00 82 09 00 00-0c 00 00 00 46 00 00 00 ............F...
826. 0250 5f 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 \_...............
827. 0260 00 00 00 00 00 00 00 00-84 09 00 00 53 00 00 00 ............S...
828. 0270 64 00 00 00 5d 00 00 00-0c 00 00 00 01 00 00 00 d...]...........
829. 0280 00 00 00 00 1e 00 28 3a-80 00 00 00 8c 09 00 00 ......(:........
830. 0290 0c 00 00 00 46 00 00 00-6e 00 00 00 08 00 00 00 ....F...n.......
831. 02a0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
832. 02b0 8e 09 00 00 53 00 00 00-64 00 00 00 6c 00 00 00 ....S...d...l...
833. 02c0 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 2a 3a ..............\*:
834. 02d0 28 00 00 00 99 09 00 00-0c 00 00 00 46 00 00 00 (...........F...
835. 02e0 7d 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 }...............
836. 02f0 00 00 00 00 00 00 00 00-9b 09 00 00 53 00 00 00 ............S...
837. 0300 64 00 00 00 7b 00 00 00-0c 00 00 00 01 00 00 00 d...{...........
838. 0310 00 00 00 00 1e 00 26 3a-03 00 00 00 ac 09 00 00 ......&:........
839. 0320 bd 00 00 00 46 00 00 00-32 00 00 00 08 00 00 00 ....F...2.......
840. 0330 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
841. 0340 ae 09 00 00 03 01 00 00-64 00 00 00 30 00 00 00 ........d...0...
842. 0350 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 17 3a ...............:
843. 0360 40 00 00 00 b6 09 00 00-bd 00 00 00 46 00 00 00 @...........F...
844. 0370 41 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 A...............
845. 0380 00 00 00 00 00 00 00 00-b8 09 00 00 03 01 00 00 ................
846. 0390 64 00 00 00 3f 00 00 00-0c 00 00 00 01 00 00 00 d...?...........
847. 03a0 00 00 00 00 1e 00 16 3a-40 00 00 00 c2 09 00 00 .......:@.......
848. 03b0 bd 00 00 00 46 00 00 00-50 00 00 00 08 00 00 00 ....F...P.......
849. 03c0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
850. 03d0 c4 09 00 00 03 01 00 00-64 00 00 00 4e 00 00 00 ........d...N...
851. 03e0 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 18 3a ...............:
852. 03f0 40 00 00 00 d1 09 00 00-bd 00 00 00 46 00 00 00 @...........F...
853. 0400 5f 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 \_...............
854. 0410 00 00 00 00 00 00 00 00-d3 09 00 00 03 01 00 00 ................
855. 0420 64 00 00 00 5d 00 00 00-0c 00 00 00 01 00 00 00 d...]...........
856. 0430 00 00 00 00 1e 00 19 3a-80 00 00 00 dc 09 00 00 .......:........
857. 0440 bd 00 00 00 46 00 00 00-6e 00 00 00 08 00 00 00 ....F...n.......
858. 0450 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
859. 0460 de 09 00 00 03 01 00 00-64 00 00 00 6c 00 00 00 ........d...l...
860. 0470 0c 00 00 00 01 00 00 00-00 00 00 00 1e 00 30 3a ..............0:
861. 0480 00 01 00 00 ea 09 00 00-bd 00 00 00 46 00 00 00 ............F...
862. 0490 7d 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 }...............
863. 04a0 00 00 00 00 00 00 00 00-ec 09 00 00 03 01 00 00 ................
864. 04b0 64 00 00 00 7b 00 00 00-0c 00 00 00 01 00 00 00 d...{...........
865. 04c0 00 00 00 00 1e 00 08 3a-40 00 00 00 f4 09 00 00 .......:@.......
866. 04d0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
867. 04e0 08 00 00 00 e5 0c 00 00-00 00 00 00 00 00 00 00 ................
868. 04f0 f6 09 00 00 06 00 00 00-67 01 00 00 04 00 00 00 ........g.......
869. 0500 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
870. 0510 00 00 00 00 03 0a 00 00-06 00 00 00 67 01 00 00 ............g...
871. 0520 0f 00 00 00 14 00 00 00-02 00 00 00 02 00 00 00 ................
872. 0530 0d 00 05 80 00 00 00 00-0d 0a 00 00 06 00 00 00 ................
873. 0540 67 01 00 00 25 00 00 00-08 00 00 00 00 00 00 00 g.............
874. 0550 00 00 00 00 00 00 00 00-00 00 00 00 0f 0a 00 00 ................
875. 0560 06 00 00 00 67 01 00 00-30 00 00 00 60 00 00 00 ....g...0...`...
876. 0570 02 00 00 00 00 00 00 00-0d 00 0e 80 00 00 00 00 ................
877. 0580 20 0a 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ...............
878. 0590 00 00 00 00 08 00 00 00-e6 0c 00 00 00 00 00 00 ................
879. 05a0 00 00 00 00 22 0a 00 00-06 00 00 00 67 01 00 00 ....".......g...
880. 05b0 03 00 00 00 50 00 00 00-06 00 00 00 00 00 00 00 ....P...........
881. 05c0 00 00 00 00 00 00 00 00-2e 0a 00 00 0c 00 00 00 ................
882. 05d0 46 00 00 00 12 00 00 00-08 00 00 00 00 00 00 00 F...............
883. 05e0 00 00 00 00 00 00 00 00-00 00 00 00 3c 0a 00 00 ............<...
884. 05f0 53 00 00 00 64 00 00 00-10 00 00 00 0c 00 00 00 S...d...........
885. 0600 01 00 00 00 00 00 00 00-1e 00 08 3a 40 00 00 00 ...........:@...
886. 0610 47 0a 00 00 bd 00 00 00-46 00 00 00 12 00 00 00 G.......F.......
887. 0620 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
888. 0630 00 00 00 00 49 0a 00 00-03 01 00 00 64 00 00 00 ....I.......d...
889. 0640 10 00 00 00 0c 00 00 00-01 00 00 00 00 00 00 00 ................
890. 0650 1e 00 09 3a 40 00 00 00-50 0a 00 00 0c 00 00 00 ...:@...P.......
891. 0660 46 00 00 00 22 00 00 00-08 00 00 00 00 00 00 00 F..."...........
892. 0670 00 00 00 00 00 00 00 00-00 00 00 00 52 0a 00 00 ............R...
893. 0680 53 00 00 00 64 00 00 00-20 00 00 00 60 00 00 00 S...d... ...`...
894. 0690 0c 00 00 00 01 00 00 00-1e 10 1b 3a 40 00 00 00 ...........:@...
895. 06a0 5f 0a 00 00 bd 00 00 00-46 00 00 00 22 00 00 00 \_.......F..."...
896. 06b0 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
897. 06c0 00 00 00 00 61 0a 00 00-03 01 00 00 64 00 00 00 ....a.......d...
898. 06d0 20 00 00 00 60 00 00 00-0c 00 00 00 01 00 00 00 ...`...........
899. 06e0 1e 10 2f 3a 40 00 00 00-6a 0a 00 00 0c 00 00 00 ../:@...j.......
900. 06f0 46 00 00 00 32 00 00 00-08 00 00 00 00 00 00 00 F...2...........
901. 0700 00 00 00 00 00 00 00 00-00 00 00 00 6c 0a 00 00 ............l...
902. 0710 53 00 00 00 64 00 00 00-30 00 00 00 0c 00 00 00 S...d...0.......
903. 0720 01 00 00 00 00 00 00 00-1e 00 23 3a 40 00 00 00 ..........#:@...
904. 0730 72 0a 00 00 bd 00 00 00-46 00 00 00 32 00 00 00 r.......F...2...
905. 0740 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
906. 0750 00 00 00 00 74 0a 00 00-03 01 00 00 64 00 00 00 ....t.......d...
907. 0760 30 00 00 00 0c 00 00 00-01 00 00 00 00 00 00 00 0...............
908. 0770 1e 00 1c 3a 40 00 00 00-7d 0a 00 00 0c 00 00 00 ...:@...}.......
909. 0780 46 00 00 00 42 00 00 00-08 00 00 00 00 00 00 00 F...B...........
910. 0790 00 00 00 00 00 00 00 00-00 00 00 00 7f 0a 00 00 ................
911. 07a0 53 00 00 00 64 00 00 00-40 00 00 00 0c 00 00 00 S...d...@.......
912. 07b0 01 00 00 00 00 00 00 00-1e 00 2e 3a 40 00 00 00 ...........:@...
913. 07c0 8b 0a 00 00 bd 00 00 00-46 00 00 00 42 00 00 00 ........F...B...
914. 07d0 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
915. 07e0 00 00 00 00 8d 0a 00 00-03 01 00 00 64 00 00 00 ............d...
916. 07f0 40 00 00 00 0c 00 00 00-01 00 00 00 00 00 00 00 @...............
917. 0800 1e 00 21 3a 40 00 00 00-95 0a 00 00 06 00 00 00 ..!:@...........
918. 0810 67 01 00 00 5a 00 00 00-08 00 00 00 00 00 00 00 g...Z...........
919. 0820 00 00 00 00 00 00 00 00-00 00 00 00 97 0a 00 00 ................
920. 0830 06 00 00 00 67 01 00 00-64 00 00 00 2b 00 00 00 ....g...d...+...
921. 0840 01 00 00 00 01 00 00 00-1e 00 04 30 00 04 00 00 ...........0....
922. 0850 9f 0a 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
923. 0860 00 00 00 00 08 00 00 00-e7 0c 00 00 00 00 00 00 ................
924. 0870 00 00 00 00 a1 0a 00 00-06 00 00 00 67 01 00 00 ............g...
925. 0880 04 00 00 00 08 00 00 00-00 00 00 00 00 00 00 00 ................
926. 0890 00 00 00 00 00 00 00 00-ab 0a 00 00 06 00 00 00 ................
927. 08a0 67 01 00 00 0e 00 00 00-84 00 00 00 02 00 00 00 g...............
928. 08b0 00 00 00 00 0d 00 08 80-00 00 00 00 be 0a 00 00 ................
929. 08c0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
930. 08d0 08 00 00 00 e8 0c 00 00-00 00 00 00 00 00 00 00 ................
931. 08e0 c0 0a 00 00 06 00 00 00-67 01 00 00 04 00 00 00 ........g.......
932. 08f0 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
933. 0900 00 00 00 00 d1 0a 00 00-06 00 00 00 67 01 00 00 ............g...
934. 0910 0e 00 00 00 84 00 00 00-0b 00 00 00 00 00 00 00 ................
935. 0920 1e 10 0f 80 00 00 00 00-e4 0a 00 00 47 65 6e 65 ............Gene
936. 0930 72 61 6c 00 4e 61 6d 65-00 26 46 69 72 73 74 3a ral.Name.&First:
937. 0940 00 2a 00 49 6e 69 26 74-69 61 6c 73 3a 00 2a 00 .\*.Ini&tials:.\*.
938. 0950 26 4c 61 73 74 3a 00 2a-00 44 69 73 70 6c 61 79 &Last:.\*.Display
939. 0960 26 3a 00 2a 00 41 6c 26-69 61 73 3a 00 2a 00 41 &:.\*.Al&ias:.\*.A
940. 0970 64 64 26 72 65 73 73 3a-00 2a 00 43 69 74 26 79 dd&ress:.\*.Cit&y
941. 0980 3a 00 2a 00 26 53 74 61-74 65 3a 00 2a 00 26 5a :.\*.&State:.\*.&Z
942. 0990 69 70 20 63 6f 64 65 3a-00 2a 00 43 6f 26 75 6e ip code:.\*.Co&un
943. 09a0 74 72 79 2f 52 65 67 69-6f 6e 3a 00 2a 00 54 69 try/Region:.\*.Ti
944. 09b0 74 6c 26 65 3a 00 2a 00-43 6f 26 6d 70 61 6e 79 tl&e:.\*.Co&mpany
945. 09c0 3a 00 2a 00 26 44 65 70-61 72 74 6d 65 6e 74 3a :.\*.&Department:
946. 09d0 00 2a 00 26 4f 66 66 69-63 65 3a 00 2a 00 41 73 .\*.&Office:.\*.As
947. 09e0 73 69 73 74 61 26 6e 74-3a 00 2a 00 50 26 68 6f sista&nt:.\*.P&ho
948. 09f0 6e 65 3a 00 2a 00 4f 72-67 61 6e 69 7a 61 74 69 ne:.\*.Organizati
949. 0a00 6f 6e 00 26 4d 61 6e 61-67 65 72 3a 00 2a 00 26 on.&Manager:.\*.&
950. 0a10 44 69 72 65 63 74 20 72-65 70 6f 72 74 73 3a 00 Direct reports:.
951. 0a20 2a 00 50 68 6f 6e 65 2f-4e 6f 74 65 73 00 50 68 \*.Phone/Notes.Ph
952. 0a30 6f 6e 65 20 6e 75 6d 62-65 72 73 00 42 75 26 73 one numbers.Bu&s
953. 0a40 69 6e 65 73 73 3a 00 2a-00 26 48 6f 6d 65 3a 00 iness:.\*.&Home:.
954. 0a50 2a 00 42 75 73 69 26 6e-65 73 73 20 32 3a 00 2a \*.Busi&ness 2:.\*
955. 0a60 00 48 26 6f 6d 65 20 32-3a 00 2a 00 26 46 61 78 .H&ome 2:.\*.&Fax
956. 0a70 3a 00 2a 00 26 4d 6f 62-69 6c 65 3a 00 2a 00 41 :.\*.&Mobile:.\*.A
957. 0a80 73 73 26 69 73 74 61 6e-74 3a 00 2a 00 50 61 26 ss&istant:.\*.Pa&
958. 0a90 67 65 72 3a 00 2a 00 4e-6f 26 74 65 73 3a 00 2a ger:.\*.No&tes:.\*
959. 0aa0 00 4d 65 6d 62 65 72 20-4f 66 00 26 47 72 6f 75 .Member Of.&Grou
960. 0ab0 70 20 6d 65 6d 62 65 72-73 68 69 70 3a 00 2a 00 p membership:.\*.
961. 0ac0 45 2d 6d 61 69 6c 20 41-64 64 72 65 73 73 65 73 E-mail Addresses
962. 0ad0 00 26 45 2d 6d 61 69 6c-20 61 64 64 72 65 73 73 .&E-mail address
963. 0ae0 65 73 3a 00 2a 00 es:.\*.

By processing the template in this **PropertyRow\_r** structure, the dialog box shown in the following figure is created.



Figure 2: Address Book object display dialog box

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

# Security

## 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**](#gt_e63aea5b-046b-4176-9359-fde82613a406) [**RPC**](#gt_8a7f6700-8311-45bc-af10-82e10accd331)-based transport also apply. For more information, see [[MS-NSPI]](%5BMS-NSPI%5D.pdf#Section_6dd0a3eab4d44a73a857add03a89a543) and [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136) section 5.1.

## Index of Security Parameters

None.

# Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

* Microsoft Exchange Server 2003
* Microsoft Exchange Server 2007
* Microsoft Exchange Server 2010
* Microsoft Exchange Server 2013
* Microsoft Exchange Server 2016
* Microsoft Exchange Server 2019
* Microsoft Office Outlook 2003
* Microsoft Office Outlook 2007
* Microsoft Outlook 2010
* Microsoft Outlook 2013
* Microsoft Outlook 2016
* Microsoft Outlook 2019
* Microsoft Outlook 2021
* Microsoft Outlook 2024 Preview

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

[<1> Section 2.1](#Appendix_A_Target_1): Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 point the client to the Exchange [**NSPI**](#gt_e63aea5b-046b-4176-9359-fde82613a406) server, which implements the NSPI methods as described in [[MS-OXNSPI]](%5BMS-OXNSPI%5D.pdf#Section_63662a26c8fc4493a41afbcbb7e43136). The only exception is when Exchange 2010, Exchange 2013, Exchange 2016, or Exchange 2019 is installed on an Active Directory® global catalog server, in which case the server points the client to Active Directory Domain Services (AD DS).

[<2> Section 2.1](#Appendix_A_Target_2): Exchange 2003 and Exchange 2007 point the client to AD DS, which implements the NSPI methods as described in [MS-NSPI].

[<3> Section 2.2.2.2](#Appendix_A_Target_3): Exchange 2010 does not include the *Size* parameter.

[<4> Section 3.1.4.1](#Appendix_A_Target_4): Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 ignore the **TI\_HELPFILE\_NAME** and **TI\_HELPFILE\_CONTENTS** flags. Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 pass these flags, but they have no effect on the results of the **NspiGetTemplateInfo** function.

[<5> Section 3.1.4.2](#Appendix_A_Target_5): Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 ignore the **TI\_HELPFILE\_NAME** and **TI\_HELPFILE\_CONTENTS** flags. Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 pass these flags but they have no effect on the results of the **NspiGetTemplateInfo** function.

[<6> Section 3.1.4.3](#Appendix_A_Target_6): Exchange 2003, Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 ignore the **TI\_HELPFILE\_NAME** and **TI\_HELPFILE\_CONTENTS** flags. Office Outlook 2003, Office Outlook 2007, Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 pass these flags, but they have no effect on the results of the **NspiGetTemplateInfo** function.

# Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

* A document revision that incorporates changes to interoperability requirements.
* A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

| Section | Description | Revision class |
| --- | --- | --- |
| [6](#Section_3724a3609fa14271aa84ae2389f8520e) Appendix A: Product Behavior | Updated list of supported products. | Major |

# Index

A

Abstract data model

 [client](#section_4946fe1e77384f89b55bcee7505a26d7) 20

 [server](#section_08ff6e8655354c388ef95f9541038abd) 24

[Applicability](#section_d6f0bc688dfb4fbcacf81dbc8447122a) 9

C

[Capability negotiation](#section_bd5370d16ec34ccdacc1624d266dbca4) 9

[Change tracking](#section_0ec5b573ee25400fb74a923f1edab1bb) 51

Client

 [abstract data model](#section_4946fe1e77384f89b55bcee7505a26d7) 20

 [higher-layer triggered events](#section_de37324cabf247b78309eb809a874628) 21

 [initialization](#section_b68717e2f74e45e2a3a2e27b26cac03c) 21

 [message processing](#section_069592c0164b4e168bd07d26694e2fa0) 22

 [other local events](#section_a53fa2b3962c47ee986763b99e7bd5bf) 24

 [sequencing rules](#section_069592c0164b4e168bd07d26694e2fa0) 22

 [timer events](#section_6acc3288f0b94adb9676e03d6643ca0f) 24

 [timers](#section_6da777cf32814bf69d23e6cba1965581) 21

D

Data model - abstract

 [client](#section_4946fe1e77384f89b55bcee7505a26d7) 20

 [server](#section_08ff6e8655354c388ef95f9541038abd) 24

F

[Fields - vendor-extensible](#section_663cc493ec9f4485a076588d51417c1a) 9

G

[Glossary](#section_d6dced03dea244029684802b51c34947) 6

H

Higher-layer triggered events

 [client](#section_de37324cabf247b78309eb809a874628) 21

 [server](#section_5a12a95f7c214db4b5c2c7697d2ab5a2) 24

I

[Implementer - security considerations](#section_2e283667f11c47769558baeb35dd646c) 48

[Index of security parameters](#section_19c84b355b9944c687bfee51e4fd9645) 48

[Informative references](#section_dba9b67cbae2466b8d2be611d8edb434) 8

Initialization

 [client](#section_b68717e2f74e45e2a3a2e27b26cac03c) 21

 [server](#section_9f7999a0764b449499a7490f6bcdebab) 24

[Introduction](#section_1884d1cde08e4da0a518fb7892957879) 6

M

Message processing

 [client](#section_069592c0164b4e168bd07d26694e2fa0) 22

 [server](#section_9538d9ca2d9e475bac31a3d7834d2360) 24

Messages

 [NspiGetSpecialTable PropertyRowSet\_r Format](#section_4ac5435ad3b448fe9c69ce85e00cf39a) 10

 [NspiGetTemplateInfo PropertyRow\_r Format](#section_68d4dd27ac444fbc9a0e20fbbf906389) 11

 [transport](#section_dbd1e55173014887a79118ffb428c8f1) 10

N

[Normative references](#section_2a1ce8ed9ae041fc8eef55eec9916b22) 7

[NspiGetSpecialTable PropertyRowSet\_r Format message](#section_4ac5435ad3b448fe9c69ce85e00cf39a) 10

[NspiGetTemplateInfo PropertyRow\_r Format message](#section_68d4dd27ac444fbc9a0e20fbbf906389) 11

O

Other local events

 [client](#section_a53fa2b3962c47ee986763b99e7bd5bf) 24

 [server](#section_345a8ad1692b4d94b3387da029639ba2) 25

[Overview (synopsis)](#section_3c06c5ed20f14ac4a1d7747ca37d7392) 8

P

[Parameters - security index](#section_19c84b355b9944c687bfee51e4fd9645) 48

[Preconditions](#section_880c0b0d26254530836cdd09f39e9aab) 9

[Prerequisites](#section_880c0b0d26254530836cdd09f39e9aab) 9

[Product behavior](#section_3724a3609fa14271aa84ae2389f8520e) 49

R

[References](#section_1e061e0a2c804e8a91491fe172fd5c40) 7

 [informative](#section_dba9b67cbae2466b8d2be611d8edb434) 8

 [normative](#section_2a1ce8ed9ae041fc8eef55eec9916b22) 7

[Relationship to other protocols](#section_ad0f8f6a6e6c4fa295fd25e7052fc443) 9

S

Security

 [implementer considerations](#section_2e283667f11c47769558baeb35dd646c) 48

 [parameter index](#section_19c84b355b9944c687bfee51e4fd9645) 48

Sequencing rules

 [client](#section_069592c0164b4e168bd07d26694e2fa0) 22

 [server](#section_9538d9ca2d9e475bac31a3d7834d2360) 24

Server

 [abstract data model](#section_08ff6e8655354c388ef95f9541038abd) 24

 [higher-layer triggered events](#section_5a12a95f7c214db4b5c2c7697d2ab5a2) 24

 [initialization](#section_9f7999a0764b449499a7490f6bcdebab) 24

 [message processing](#section_9538d9ca2d9e475bac31a3d7834d2360) 24

 [other local events](#section_345a8ad1692b4d94b3387da029639ba2) 25

 [sequencing rules](#section_9538d9ca2d9e475bac31a3d7834d2360) 24

 [timer events](#section_5d1940d687d14595b04e4584f608ad75) 25

 [timers](#section_5f7e1969141d48f999e7e91eea4945d8) 24

[Standards assignments](#section_ca74a39b85de4c0187d2e345a1bb7f04) 9

T

Timer events

 [client](#section_6acc3288f0b94adb9676e03d6643ca0f) 24

 [server](#section_5d1940d687d14595b04e4584f608ad75) 25

Timers

 [client](#section_6da777cf32814bf69d23e6cba1965581) 21

 [server](#section_5f7e1969141d48f999e7e91eea4945d8) 24

[Tracking changes](#section_0ec5b573ee25400fb74a923f1edab1bb) 51

[Transport](#section_dbd1e55173014887a79118ffb428c8f1) 10

Triggered events - higher-layer

 [client](#section_de37324cabf247b78309eb809a874628) 21

 [server](#section_5a12a95f7c214db4b5c2c7697d2ab5a2) 24

V

[Vendor-extensible fields](#section_663cc493ec9f4485a076588d51417c1a) 9

[Versioning](#section_bd5370d16ec34ccdacc1624d266dbca4) 9