

# [MS-SHLLINK]: Shell Link (.LNK) Binary File Format

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

| Date       | Revision History | Revision Class | Comments   |
|------------|------------------|----------------|--|
| 07/16/2010 | 1.0              | New            | First Release.   |
| 08/27/2010 | 1.1              | Minor          | Clarified the meaning of the technical content.                              |
| 10/08/2010 | 1.1              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 11/19/2010 | 1.1              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 01/07/2011 | 1.1              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 02/11/2011 | 1.1              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 03/25/2011 | 1.1              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 05/06/2011 | 1.1              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 06/17/2011 | 1.2              | Minor          | Clarified the meaning of the technical content.                              |
| 09/23/2011 | 1.2              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 12/16/2011 | 1.2              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 03/30/2012 | 1.2              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 07/12/2012 | 1.2              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 10/25/2012 | 1.2              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 01/31/2013 | 1.2              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 08/08/2013 | 2.0              | Major          | Significantly changed the technical content.                                 |
| 11/14/2013 | 2.0              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 02/13/2014 | 2.0              | No change      | No changes to the meaning, language, or formatting of the technical content. |
| 05/15/2014 | 2.0              | No change      | No changes to the meaning, language, or formatting of the technical content. |



# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction .....</b>                            | <b>5</b>  |
| 1.1      | Glossary .....                                       | 5         |
| 1.2      | References.....                                      | 6         |
| 1.2.1    | Normative References.....                            | 6         |
| 1.2.2    | Informative References .....                         | 7         |
| 1.3      | Overview .....                                       | 7         |
| 1.4      | Relationship to Protocols and Other Structures ..... | 8         |
| 1.5      | Applicability Statement.....                         | 8         |
| 1.6      | Versioning and Localization .....                    | 8         |
| 1.7      | Vendor-Extensible Fields.....                        | 8         |
| <b>2</b> | <b>Structures .....</b>                              | <b>9</b>  |
| 2.1      | ShellLinkHeader.....                                 | 9         |
| 2.1.1    | LinkFlags.....                                       | 11        |
| 2.1.2    | FileAttributesFlags .....                            | 13        |
| 2.1.3    | HotKeyFlags .....                                    | 15        |
| 2.2      | LinkTargetIDList .....                               | 18        |
| 2.2.1    | IDList .....   | 18        |
| 2.2.2    | ItemID.....  | 19        |
| 2.3      | LinkInfo.....  | 19        |
| 2.3.1    | VolumeID.....  | 22        |
| 2.3.2    | CommonNetworkRelativeLink.....                       | 24        |
| 2.4      | StringData .....                                     | 27        |
| 2.5      | ExtraData .....                                      | 28        |
| 2.5.1    | ConsoleDataBlock.....                                | 29        |
| 2.5.2    | ConsoleFEDataBlock .....                             | 34        |
| 2.5.3    | DarwinDataBlock .....                                | 35        |
| 2.5.4    | EnvironmentVariableDataBlock .....                   | 36        |
| 2.5.5    | IconEnvironmentDataBlock.....                        | 37        |
| 2.5.6    | KnownFolderDataBlock.....                            | 38        |
| 2.5.7    | PropertyStoreDataBlock .....                         | 39        |
| 2.5.8    | ShimDataBlock .....                                  | 39        |
| 2.5.9    | SpecialFolderDataBlock .....                         | 40        |
| 2.5.10   | TrackerDataBlock .....                               | 41        |
| 2.5.11   | VistaAndAboveIDListDataBlock.....                    | 42        |
| <b>3</b> | <b>Structure Examples .....</b>                      | <b>43</b> |
| 3.1      | Shortcut to a File .....                             | 43        |
| <b>4</b> | <b>Security.....</b>                                 | <b>47</b> |
| <b>5</b> | <b>Appendix A: Product Behavior .....</b>            | <b>48</b> |
| <b>6</b> | <b>Change Tracking.....</b>                          | <b>50</b> |
| <b>7</b> | <b>Index .....</b>                                   | <b>51</b> |

# 1 Introduction

This is a specification of the Shell Link Binary File Format. In this format a structure is called a **shell link**, or **shortcut**, and is a data **object** that contains information that can be used to access another data object. The Shell Link Binary File Format is the format of Windows files with the extension ".LNK".

Shell links are commonly used to support application launching and linking scenarios, such as **Object Linking and Embedding (OLE)**, but they also can be used by applications that need the ability to store a reference to a target file.

Sections 1.7 and 2 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. All other sections and examples in this specification are informative.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**American National Standards Institute (ANSI) character set**  
**Augmented Backus-Naur Form (ABNF)**  
**class identifier (CLSID)**  
**code page**  
**Component Object Model (COM)**  
**Coordinated Universal Time (UTC)**  
**GUID**  
**little-endian**  
**NetBIOS name**  
**object (3)**  
**reparse point**  
**sparse file**  
**Unicode**  
**Universal Naming Convention (UNC)**

The following terms are specific to this document:

**extra data section:** A data structure appended to the basic Shell Link Binary File Format data that contains additional information about the **link target**.

**folder integer ID:** An integer value that identifies a known folder.

**folder GUID ID:** A **GUID** value that identifies a known folder. Some folder GUID ID values correspond to **folder integer ID** values.

**item ID (ItemID):** A structure that represents an item in the context of a **shell data source**.

**item ID list (IDList):** A data structure that refers to a location. An item ID list is a multi-segment data structure where each segment's content is defined by a data source that is responsible for the location in the **namespace** referred to by the preceding segments.

**link:** An **object** that refers to another item.

**link target:** The item that a **link** references. In the case of a **shell link**, the referenced item is identified by its location in the **link target namespace** using an **item ID list (IDList)**.

**link target namespace:** A hierarchical **namespace**. In Windows, the link target namespace is the **Windows Explorer namespace**, as described in [\[MSDN-ShellNamespace\]](#).

**namespace:** An abstract container used to hold a set of unique identifiers.

**Object Linking and Embedding (OLE):** A technology for transferring and sharing information between applications by inserting a file or part of a file into a compound document. The inserted file can be either linked or embedded. An embedded item is stored as part of the compound document that contains it; a linked item stores its data in a separate file.

**relative path:** A path that is implied by the current working directory or is calculated based on a specified directory. When a user enters a command that refers to a file, and the full path is not entered, the current working directory becomes the relative path of the referenced file.

**resolve a link:** The act of finding a specific **link target**, confirming that it exists, and finding whether it has moved.

**Red-Green-Blue (RGB):** A mapping of color components in which red, green, and blue and an intensity value are combined in various ways to reproduce a range of colors.

**shell data source:** An **object** that is responsible for a specific location in the **namespace** and for enumerating and binding **IDLists** to handlers.

**shell link:** A structure in Shell Link Binary File Format.

**shim:** A mechanism used to provide custom behavior to applications that do not work on newer versions of the operating system.

**shortcut:** A term that is used synonymously with **shell link**.

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

## 1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 1.2.1 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[MS-DFSNM] Microsoft Corporation, "[Distributed File System \(DFS\): Namespace Management Protocol](#)".

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](#)".

[MS-LCID] Microsoft Corporation, "[Windows Language Code Identifier \(LCID\) Reference](#)".

[MS-PROPSTORE] Microsoft Corporation, "[Property Store Binary File Format](#)".

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

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <http://www.rfc-editor.org/rfc/rfc5234.txt>

## 1.2.2 Informative References

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, <https://www2.opengroup.org/ogsys/catalog/c706>

[MS-CFB] Microsoft Corporation, "[Compound File Binary File Format](#)".

[MS-DLTW] Microsoft Corporation, "[Distributed Link Tracking: Workstation Protocol](#)".

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[MSCHARSET] Microsoft Corporation, "INFO: Windows, Code Pages, and Character Sets", February 2005, <http://support.microsoft.com/kb/75435>

[MSDN-CODEPAGE] Microsoft Corporation, "Common Pages", <http://msdn.microsoft.com/en-us/goglobal/bb964653.aspx>

[MSDN-ISHELLINK] Microsoft Corporation, "IShellLink Interface", <http://msdn.microsoft.com/en-us/library/bb774950.aspx>

[MSDN-MSISHORTCUTS] Microsoft Corporation, "How Windows Installer Shortcuts Work", <http://support.microsoft.com/kb/243630>

## 1.3 Overview

The Shell Link Binary File Format specifies a structure called a shell link. That structure is used to store a reference to a location in a **link target namespace**, which is referred to as a **link target**. The most important component of a link target namespace is a link target in the form of an **item ID list (IDList)**.

The shell link structure stores various information that is useful to end users, including:

- A keyboard shortcut that can be used to launch an application.
- A descriptive comment.
- Settings that control application behavior.
- Optional data stored in **extra data sections**.

Optional data can include a property store that contains an extensible set of properties in the format that is described in [\[MS-PROPSTORE\]](#).

The Shell Link Binary File Format can be managed using a COM object, programmed using the **IShellLink** interface, and saved into its persistence format using the **IPersistStream** or **IPersistFile** interface. It is most common for shell links to be stored in a file with the .LNK file extension. By using the **IPersistStream** interface, a shell link can be saved into another storage system, for example a database or the registry, or embedded in another file format. For more information, see [\[MSDN-ISHELLINK\]](#).

Multi-byte data values in the Shell Link Binary File Format are stored in **little-endian** format.

## 1.4 Relationship to Protocols and Other Structures

The Shell Link Binary File Format is used by the Compound File Binary File Format [\[MS-CFB\]](#).

The Shell Link Binary File Format uses the Property Store Binary File Format [\[MS-PROPSTORE\]](#).

## 1.5 Applicability Statement

This document specifies a persistence format for **links** to files in a file system or to applications that are available for installation. This persistence format is applicable for use as a stand-alone file and for containment within other structures.

## 1.6 Versioning and Localization

This specification covers versioning issues in the following areas:

**Localization:** The Shell Link Binary File Format defines the [ConsoleFEDataBlock](#) structure (section [2.5.2](#)), which specifies a **code page** for displaying text. That value can be used to specify a set of characters for a particular language or locale.

## 1.7 Vendor-Extensible Fields

A **shell data source** can extend the persistence format by storing custom data inside **ItemID** structure.

The ItemIDs embedded in an IDList are in a format specified by the shell data sources that manage the ItemIDs. The ItemIDs are free to store whatever data is needed in this structure to uniquely identify the items in their **namespace**.

The property store embedded in a link can be used to store property values in the shell link.



## 2 Structures

The Shell Link Binary File Format consists of a sequence of structures that conform to the following **ABNF** rules [\[RFC5234\]](#).

```
SHELL_LINK = SHELL_LINK_HEADER [LINKTARGET_IDLIST] [LINKINFO]
              [STRING_DATA] *EXTRA_DATA
```

**SHELL\_LINK\_HEADER:** A [ShellLinkHeader](#) structure (section [2.1](#)), which contains identification information, timestamps, and flags that specify the presence of optional structures.

**LINKTARGET\_IDLIST:** An optional [LinkTargetIDList](#) structure (section [2.2](#)), which specifies the target of the link. The presence of this structure is specified by the **HasLinkTargetIDList** bit ([LinkFlags](#) section 2.1.1) in the ShellLinkHeader.

**LINKINFO:** An optional [LinkInfo](#) structure (section [2.3](#)), which specifies information necessary to resolve the link target. The presence of this structure is specified by the **HasLinkInfo** bit ([LinkFlags](#) section 2.1.1) in the ShellLinkHeader.

**STRING\_DATA:** Zero or more optional [StringData](#) structures (section [2.4](#)), which are used to convey user interface and path identification information. The presence of these structures is specified by bits ([LinkFlags](#) section 2.1.1) in the ShellLinkHeader.

**EXTRA\_DATA:** Zero or more [ExtraData](#) structures (section [2.5](#)).

**Note** Structures of the Shell Link Binary File Format can define strings in fixed-length fields. In fixed-length fields, strings **MUST** be null-terminated. If a string is smaller than the size of the field that contains it, the bytes in the field following the terminating null character are undefined and can have any value. The undefined bytes **MUST NOT** be used.

### 2.1 ShellLinkHeader

The ShellLinkHeader structure contains identification information, timestamps, and flags that specify the presence of optional structures, including [LinkTargetIDList](#) (section [2.2](#)), [LinkInfo](#) (section [2.3](#)), and [StringData](#) (section [2.4](#)).

|            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0          | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| HeaderSize |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LinkCLSID  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LinkFlags  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

|                |           |
|----------------|-----------|
| FileAttributes |           |
| CreationTime   |           |
| ...            |           |
| AccessTime     |           |
| ...            |           |
| WriteTime      |           |
| ...            |           |
| FileSize       |           |
| IconIndex      |           |
| ShowCommand    |           |
| HotKey         | Reserved1 |
| Reserved2      |           |
| Reserved3      |           |

**HeaderSize (4 bytes):** The size, in bytes, of this structure. This value MUST be 0x0000004C.

**LinkCLSID (16 bytes):** A **class identifier (CLSID)**. This value MUST be 00021401-0000-0000-C000-000000000046.

**LinkFlags (4 bytes):** A [LinkFlags](#) structure (section [2.1.1](#)) that specifies information about the shell link and the presence of optional portions of the structure.

**FileAttributes (4 bytes):** A [FileAttributesFlags](#) structure (section [2.1.2](#)) that specifies information about the link target.

**CreationTime (8 bytes):** A **FILETIME** structure ([\[MS-DTYP\]](#) section 2.3.3) that specifies the creation time of the link target in **UTC (Coordinated Universal Time)**. If the value is zero, there is no creation time set on the link target.

**AccessTime (8 bytes):** A **FILETIME** structure ([\[MS-DTYP\]](#) section 2.3.3) that specifies the access time of the link target in **UTC (Coordinated Universal Time)**. If the value is zero, there is no access time set on the link target.

**WriteTime (8 bytes):** A **FILETIME** structure ([\[MS-DTYP\]](#) section 2.3.3) that specifies the write time of the link target in **UTC (Coordinated Universal Time)**. If the value is zero, there is no write time set on the link target.

**FileSize (4 bytes):** A 32-bit unsigned integer that specifies the size, in bytes, of the link target. If the link target file is larger than 0xFFFFFFFF, this value specifies the least significant 32 bits of the link target file size.

**IconIndex (4 bytes):** A 32-bit signed integer that specifies the index of an icon within a given icon location.

**ShowCommand (4 bytes):** A 32-bit unsigned integer that specifies the expected window state of an application launched by the link. This value SHOULD be one of the following.

| Value                            | Meaning   |
|----------------------------------|---|
| SW_SHOWNORMAL<br>0x00000001      | The application is open and its window is open in a normal fashion.                                   |
| SW_SHOWMAXIMIZED<br>0x00000003   | The application is open, and keyboard focus is given to the application, but its window is not shown. |
| SW_SHOWMINNOACTIVE<br>0x00000007 | The application is open, but its window is not shown. It is not given the keyboard focus.             |

All other values MUST be treated as **SW\_SHOWNORMAL**.

**HotKey (2 bytes):** A [HotKeyFlags](#) structure (section [2.1.3](#)) that specifies the keystrokes used to launch the application referenced by the shortcut key. This value is assigned to the application after it is launched, so that pressing the key activates that application.

**Reserved1 (2 bytes):** A value that MUST be zero.

**Reserved2 (4 bytes):** A value that MUST be zero.

**Reserved3 (4 bytes):** A value that MUST be zero.

### 2.1.1 LinkFlags

The LinkFlags structure defines bits that specify which shell link structures are present in the file format after the [ShellLinkHeader](#) structure (section [2.1](#)).

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |   |   |   | 1 |   |   |   |   |   |   |   |   |   | 2 |   |   |   |   |   |   |   |   |   |   | 3 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | 0 | 0 | 0 | 0 | 0 |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | A |   |   |   |   |   |   |

Where the bits are defined as:

| Value                    | Description   |
|--------------------------|---|
| A<br>HasLinkTargetIDList | The shell link is saved with an item ID list (IDList). If this bit is set, a <a href="#">LinkTargetIDList</a> structure (section <a href="#">2.2</a> ) MUST follow the ShellLinkHeader. If this bit is not set, this structure MUST NOT be present. |

| Value                     | Description   |
|---------------------------|---|
| B<br>HasLinkInfo          | The shell link is saved with link information. If this bit is set, a <a href="#">LinkInfo</a> structure (section 2.3) MUST be present. If this bit is not set, this structure MUST NOT be present.  |
| C<br>HasName              | The shell link is saved with a name string. If this bit is set, a <b>NAME_STRING</b> <a href="#">StringData</a> structure (section 2.4) MUST be present. If this bit is not set, this structure MUST NOT be present.  |
| D<br>HasRelativePath      | The shell link is saved with a <b>relative path</b> string. If this bit is set, a <b>RELATIVE_PATH</b> <a href="#">StringData</a> structure (section 2.4) MUST be present. If this bit is not set, this structure MUST NOT be present.                                |
| E<br>HasWorkingDir        | The shell link is saved with a working directory string. If this bit is set, a <b>WORKING_DIR</b> <a href="#">StringData</a> structure (section 2.4) MUST be present. If this bit is not set, this structure MUST NOT be present.                                     |
| F<br>HasArguments         | The shell link is saved with command line arguments. If this bit is set, a <b>COMMAND_LINE_ARGUMENTS</b> <a href="#">StringData</a> structure (section 2.4) MUST be present. If this bit is not set, this structure MUST NOT be present.                              |
| G<br>HasIconLocation      | The shell link is saved with an icon location string. If this bit is set, an <b>ICON_LOCATION</b> <a href="#">StringData</a> structure (section 2.4) MUST be present. If this bit is not set, this structure MUST NOT be present.                                     |
| H<br>IsUnicode            | The shell link contains Unicode encoded strings. This bit SHOULD be set. If this bit is set, the <a href="#">StringData</a> section should contain Unicode-encoded strings; otherwise, it should contain strings that are encoded using the system default code page. |
| I<br>ForceNoLinkInfo      | The <a href="#">LinkInfo</a> structure (section 2.3) is ignored.  |
| J<br>HasExpString         | The shell link is saved with an <a href="#">EnvironmentVariableDataBlock</a> (section 2.5.4).   |
| K<br>RunInSeparateProcess | The target is run in a separate virtual machine when launching a link target that is a 16-bit application.  |
| L<br>Unused1              | A bit that is undefined and MUST be ignored.  |
| M<br>HasDarwinID          | The shell link is saved with a <a href="#">DarwinDataBlock</a> (section 2.5.3).   |
| N<br>RunAsUser            | The application is run as a different user when the target of the shell link is activated.  |
| O<br>HasExpIcon           | The shell link is saved with an <a href="#">IconEnvironmentDataBlock</a> (section 2.5.5).   |
| P<br>NoPidAlias           | The file system location is represented in the shell namespace when the path to an item is parsed into an IDList.   |

| Value                             | Description  |
|-----------------------------------|--|
| Q<br>Unused2                      | A bit that is undefined and MUST be ignored.   |
| R<br>RunWithShimLayer             | The shell link is saved with a <a href="#">ShimDataBlock (section 2.5.8)</a> .   |
| S<br>ForceNoLinkTrack             | The <a href="#">TrackerDataBlock (section 2.5.10)</a> is ignored.  |
| T<br>EnableTargetMetadata         | The shell link attempts to collect target properties and store them in the <a href="#">PropertyStoreDataBlock (section 2.5.7)</a> when the link target is set.   |
| U<br>DisableLinkPathTracking      | The EnvironmentVariableDataBlock is ignored.   |
| V<br>DisableKnownFolderTracking   | The <a href="#">SpecialFolderDataBlock (section 2.5.9)</a> and the <a href="#">KnownFolderDataBlock (section 2.5.6)</a> are ignored when loading the shell link. If this bit is set, these extra data blocks SHOULD NOT be saved when saving the shell link. |
| W<br>DisableKnownFolderAlias      | If the link has a KnownFolderDataBlock (section 2.5.6), the unaliased form of the known folder IDList SHOULD be used when translating the target IDList at the time that the link is loaded.   |
| X<br>AllowLinkToLink              | Creating a link that references another link is enabled. Otherwise, specifying a link as the target IDList SHOULD NOT be allowed.  |
| Y<br>UnaliasOnSave                | When saving a link for which the target IDList is under a known folder, either the unaliased form of that known folder or the target IDList SHOULD be used.  |
| Z<br>PreferEnvironmentPath        | The target IDList SHOULD NOT be stored; instead, the path specified in the EnvironmentVariableDataBlock (section 2.5.4) SHOULD be used to refer to the target.   |
| AA<br>KeepLocalIDListForUNCtarget | When the target is a <b>UNC</b> name that refers to a location on a local machine, the local path IDList in the PropertyStoreDataBlock (section 2.5.7) SHOULD be stored, so it can be used when the link is loaded on the local machine.                     |

## 2.1.2 FileAttributesFlags

The FileAttributesFlags structure defines bits that specify the file attributes of the link target, if the target is a file system item. File attributes can be used if the link target is not available, or if accessing the target would be inefficient. It is possible for the target items attributes to be out of sync with this value.

|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| A | B | C | D | E | F | G | H | I | J | K  | L  | M  | N  | O  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

Where the bits are defined as:

| Value                                   | Description  |
|---|--|
| A<br>FILE_ATTRIBUTE_READONLY            | The file or directory is read-only. For a file, if this bit is set, applications can read the file but cannot write to it or delete it. For a directory, if this bit is set, applications cannot delete the directory. |
| B<br>FILE_ATTRIBUTE_HIDDEN              | The file or directory is hidden. If this bit is set, the file or folder is not included in an ordinary directory listing.  |
| C<br>FILE_ATTRIBUTE_SYSTEM              | The file or directory is part of the operating system or is used exclusively by the operating system.  |
| D<br>Reserved1                          | A bit that MUST be zero.   |
| E<br>FILE_ATTRIBUTE_DIRECTORY           | The link target is a directory instead of a file.  |
| F<br>FILE_ATTRIBUTE_ARCHIVE             | The file or directory is an archive file. Applications use this flag to mark files for backup or removal.  |
| G<br>Reserved2                          | A bit that MUST be zero.   |
| H<br>FILE_ATTRIBUTE_NORMAL              | The file or directory has no other flags set. If this bit is 1, all other bits in this structure MUST be clear.  |
| I<br>FILE_ATTRIBUTE_TEMPORARY           | The file is being used for temporary storage.  |
| J<br>FILE_ATTRIBUTE_SPARSE_FILE         | The file is a <b>sparse file</b> .   |
| K<br>FILE_ATTRIBUTE_REPARSE_POINT       | The file or directory has an associated <b>reparse point</b> .   |
| L<br>FILE_ATTRIBUTE_COMPRESSED          | The file or directory is compressed. For a file, this means that all data in the file is compressed. For a directory, this means that compression is the default for newly created files and subdirectories.           |
| M<br>FILE_ATTRIBUTE_OFFLINE             | The data of the file is not immediately available.   |
| N<br>FILE_ATTRIBUTE_NOT_CONTENT_INDEXED | The contents of the file need to be indexed.   |
| O<br>FILE_ATTRIBUTE_ENCRYPTED           | The file or directory is encrypted. For a file, this means that all data in the file is encrypted. For a directory, this means that encryption is the default for newly created files and subdirectories.              |

### 2.1.3 HotKeyFlags

The HotKeyFlags structure specifies input generated by a combination of keyboard keys being pressed.

|         |   |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---------|---|---|---|---|---|---|---|---|---|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10       | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| LowByte |   |   |   |   |   |   |   |   |   | HighByte |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**LowByte (1 byte):** An 8-bit unsigned integer that specifies a virtual key code that corresponds to a key on the keyboard. This value **MUST** be one of the following:

| Value | Meaning |
|-------|---------|
| 0x30  | "0" key |
| 0x31  | "1" key |
| 0x32  | "2" key |
| 0x33  | "3" key |
| 0x34  | "4" key |
| 0x35  | "5" key |
| 0x36  | "6" key |
| 0x37  | "7" key |
| 0x38  | "8" key |
| 0x39  | "9" key |
| 0x41  | "A" key |
| 0x42  | "B" key |
| 0x43  | "C" key |
| 0x44  | "D" key |
| 0x45  | "E" key |
| 0x46  | "F" key |
| 0x47  | "G" key |
| 0x48  | "H" key |
| 0x49  | "I" key |
| 0x4A  | "J" key |
| 0x4B  | "K" key |

| Value         | Meaning  |
|---------------|----------|
| 0x4C          | "L" key  |
| 0x4D          | "M" key  |
| 0x4E          | "N" key  |
| 0x4F          | "O" key  |
| 0x50          | "P" key  |
| 0x51          | "Q" key  |
| 0x52          | "R" key  |
| 0x53          | "S" key  |
| 0x54          | "T" key  |
| 0x55          | "U" key  |
| 0x56          | "V" key  |
| 0x57          | "W" key  |
| 0x58          | "X" key  |
| 0x59          | "Y" key  |
| 0x5A          | "Z" key  |
| VK_F1<br>0x70 | "F1" key |
| VK_F2<br>0x71 | "F2" key |
| VK_F3<br>0x72 | "F3" key |
| VK_F4<br>0x73 | "F4" key |
| VK_F5<br>0x74 | "F5" key |
| VK_F6<br>0x75 | "F6" key |
| VK_F7<br>0x76 | "F7" key |
| VK_F8<br>0x77 | "F8" key |
| VK_F9<br>0x78 | "F9" key |



| Value              | Meaning           |
|--------------------|-------------------|
| VK_F10<br>0x79     | "F10" key         |
| VK_F11<br>0x7A     | "F11" key         |
| VK_F12<br>0x7B     | "F12" key         |
| VK_F13<br>0x7C     | "F13" key         |
| VK_F14<br>0x7D     | "F14" key         |
| VK_F15<br>0x7E     | "F15" key         |
| VK_F16<br>0x7F     | "F16" key         |
| VK_F17<br>0x80     | "F17" key         |
| VK_F18<br>0x81     | "F18" key         |
| VK_F19<br>0x82     | "F19" key         |
| VK_F20<br>0x83     | "F20" key         |
| VK_F21<br>0x84     | "F21" key         |
| VK_F22<br>0x85     | "F22" key         |
| VK_F23<br>0x86     | "F23" key         |
| VK_F24<br>0x87     | "F24" key         |
| VK_NUMLOCK<br>0x90 | "NUM LOCK" key    |
| VK_SCROLL<br>0x91  | "SCROLL LOCK" key |

**HighByte (1 byte):** An 8-bit unsigned integer that specifies bits that correspond to modifier keys on the keyboard. This value **MUST** be one or a combination of the following:

| Value                   | Meaning                          |
|-------------------------|----------------------------------|
| HOTKEYF_SHIFT<br>0x01   | The "SHIFT" key on the keyboard. |
| HOTKEYF_CONTROL<br>0x02 | The "CTRL" key on the keyboard.  |
| HOTKEYF_ALT<br>0x04     | The "ALT" key on the keyboard.   |

## 2.2 LinkTargetIDList

The LinkTargetIDList structure specifies the target of the link. The presence of this optional structure is specified by the **HasLinkTargetIDList** bit ([LinkFlags](#) section 2.1.1) in the [ShellLinkHeader](#) ([section 2.1](#)).

|            |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |                   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
|------------|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|-------------------|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|
| 0          | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6                 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| IDListSize |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   | IDList (variable) |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
| ...        |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |                   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |

**IDListSize (2 bytes):** The size, in bytes, of the **IDList** field.

**IDList (variable):** A stored [IDList](#) structure ([section 2.2.1](#)), which contains the item ID list. An IDList structure conforms to the following ABNF [\[RFC5234\]](#):

```
IDLIST = *ITEMID TERMINALID
```

### 2.2.1 IDList

The stored IDList structure specifies the format of a persisted item ID list.

|                       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0                     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| ItemIDList (variable) |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| TerminalID            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**ItemIDList (variable):** An array of zero or more [ItemID](#) structures ([section 2.2.2](#)).

**TerminalID (2 bytes):** A 16-bit, unsigned integer that indicates the end of the item IDs. This value **MUST** be zero.

## 2.2.2 ItemID

An ItemID is an element in an [IDList](#) structure (section [2.2.1](#)). The data stored in a given ItemID is defined by the source that corresponds to the location in the target namespace of the preceding ItemIDs. This data uniquely identifies the items in that part of the namespace.

|            |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |                 |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
|------------|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|-----------------|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|
| 0          | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6               | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| ItemIDSize |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   | Data (variable) |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
| ...        |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |                 |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |

**ItemIDSize (2 bytes):** A 16-bit, unsigned integer that specifies the size, in bytes, of the ItemID structure, including the **ItemIDSize** field.

**Data (variable):** The shell data source-defined data that specifies an item.

## 2.3 LinkInfo

The LinkInfo structure specifies information necessary to **resolve a link** target if it is not found in its original location. This includes information about the volume that the target was stored on, the mapped drive letter, and a Universal Naming Convention (UNC) form of the path if one existed when the link was created. For more details about UNC paths, see [\[MS-DFSNM\]](#) section 2.2.1.4.

|  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|--|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| LinkInfoSize                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LinkInfoHeaderSize                       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LinkInfoFlags                            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| VolumeIDOffset                           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LocalBasePathOffset                      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CommonNetworkRelativeLinkOffset          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CommonPathSuffixOffset                   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LocalBasePathOffsetUnicode (optional)    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CommonPathSuffixOffsetUnicode (optional) |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| VolumeID (variable)                      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                                      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |



| Value                                       | Description  |
|---|--|
| A<br>VolumeIDAndLocalBasePath               | <p>If set, the <b>VolumeID</b> and <b>LocalBasePath</b> fields are present, and their locations are specified by the values of the <b>VolumeIDOffset</b> and <b>LocalBasePathOffset</b> fields, respectively. If the value of the <b>LinkInfoHeaderSize</b> field is greater than or equal to 0x00000024, the <b>LocalBasePathUnicode</b> field is present, and its location is specified by the value of the <b>LocalBasePathOffsetUnicode</b> field.</p> <p>If not set, the <b>VolumeID</b>, <b>LocalBasePath</b>, and <b>LocalBasePathUnicode</b> fields are not present, and the values of the <b>VolumeIDOffset</b> and <b>LocalBasePathOffset</b> fields are zero. If the value of the <b>LinkInfoHeaderSize</b> field is greater than or equal to 0x00000024, the value of the <b>LocalBasePathOffsetUnicode</b> field is zero.</p> |
| B<br>CommonNetworkRelativeLinkAndPathSuffix | <p>If set, the <b>CommonNetworkRelativeLink</b> field is present, and its location is specified by the value of the <b>CommonNetworkRelativeLinkOffset</b> field.</p> <p>If not set, the <b>CommonNetworkRelativeLink</b> field is not present, and the value of the <b>CommonNetworkRelativeLinkOffset</b> field is zero.</p>   |

**VolumeIDOffset (4 bytes):** A 32-bit, unsigned integer that specifies the location of the **VolumeID** field. If the **VolumeIDAndLocalBasePath** flag is set, this value is an offset, in bytes, from the start of the LinkInfo structure; otherwise, this value MUST be zero.

**LocalBasePathOffset (4 bytes):** A 32-bit, unsigned integer that specifies the location of the **LocalBasePath** field. If the **VolumeIDAndLocalBasePath** flag is set, this value is an offset, in bytes, from the start of the LinkInfo structure; otherwise, this value MUST be zero.

**CommonNetworkRelativeLinkOffset (4 bytes):** A 32-bit, unsigned integer that specifies the location of the **CommonNetworkRelativeLink** field. If the **CommonNetworkRelativeLinkAndPathSuffix** flag is set, this value is an offset, in bytes, from the start of the LinkInfo structure; otherwise, this value MUST be zero.

**CommonPathSuffixOffset (4 bytes):** A 32-bit, unsigned integer that specifies the location of the **CommonPathSuffix** field. This value is an offset, in bytes, from the start of the LinkInfo structure.

**LocalBasePathOffsetUnicode (4 bytes):** An optional, 32-bit, unsigned integer that specifies the location of the **LocalBasePathUnicode** field. If the **VolumeIDAndLocalBasePath** flag is set, this value is an offset, in bytes, from the start of the LinkInfo structure; otherwise, this value MUST be zero. This field can be present only if the value of the **LinkInfoHeaderSize** field is greater than or equal to 0x00000024.

**CommonPathSuffixOffsetUnicode (4 bytes):** An optional, 32-bit, unsigned integer that specifies the location of the **CommonPathSuffixUnicode** field. This value is an offset, in bytes, from the start of the LinkInfo structure. This field can be present only if the value of the **LinkInfoHeaderSize** field is greater than or equal to 0x00000024.

**VolumeID (variable):** An optional [VolumeID](#) structure (section [2.3.1](#)) that specifies information about the volume that the link target was on when the link was created. This field is present if the **VolumeIDAndLocalBasePath** flag is set.

**LocalBasePath (variable):** An optional, NULL-terminated string, defined by the system default code page, which is used to construct the full path to the link item or link target by appending the string in the **CommonPathSuffix** field. This field is present if the **VolumeIDAndLocalBasePath** flag is set.

**CommonNetworkRelativeLink (variable):** An optional [CommonNetworkRelativeLink](#) structure (section [2.3.2](#)) that specifies information about the network location where the link target is stored.

**CommonPathSuffix (variable):** A NULL-terminated string, defined by the system default code page, which is used to construct the full path to the link item or link target by being appended to the string in the **LocalBasePath** field.

**LocalBasePathUnicode (variable):** An optional, NULL-terminated, **Unicode** string that is used to construct the full path to the link item or link target by appending the string in the **CommonPathSuffixUnicode** field. This field can be present only if the **VolumeIDAndLocalBasePath** flag is set and the value of the **LinkInfoHeaderSize** field is greater than or equal to 0x00000024.

**CommonPathSuffixUnicode (variable):** An optional, NULL-terminated, Unicode string that is used to construct the full path to the link item or link target by being appended to the string in the **LocalBasePathUnicode** field. This field can be present only if the value of the **LinkInfoHeaderSize** field is greater than or equal to 0x00000024.

### 2.3.1 VolumeID

The VolumeID structure specifies information about the volume that a link target was on when the link was created. This information is useful for resolving the link if the file is not found in its original location.

|                                     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0                                   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| VolumeIDSize                        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| DriveType                           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| DriveSerialNumber                   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| VolumeLabelOffset                   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| VolumeLabelOffsetUnicode (optional) |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Data (variable)                     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                                 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**VolumeIDSize (4 bytes):** A 32-bit, unsigned integer that specifies the size, in bytes, of this structure. This value **MUST** be greater than 0x00000010. All offsets specified in this structure **MUST** be less than this value, and all strings contained in this structure **MUST** fit within the extent defined by this size.

**DriveType (4 bytes):** A 32-bit, unsigned integer that specifies the type of drive the link target is stored on. This value **MUST** be one of the following:

| Value                           | Meaning   |
|---------------------------------|---|
| DRIVE_UNKNOWN<br>0x00000000     | The drive type cannot be determined.  |
| DRIVE_NO_ROOT_DIR<br>0x00000001 | The root path is invalid; for example, there is no volume mounted at the path.            |
| DRIVE_REMOVABLE<br>0x00000002   | The drive has removable media, such as a floppy drive, thumb drive, or flash card reader. |
| DRIVE_FIXED<br>0x00000003       | The drive has fixed media, such as a hard drive or flash drive.                           |
| DRIVE_REMOTE<br>0x00000004      | The drive is a remote (network) drive.  |
| DRIVE_CDROM<br>0x00000005       | The drive is a CD-ROM drive.  |
| DRIVE_RAMDISK<br>0x00000006     | The drive is a RAM disk.  |

**DriveSerialNumber (4 bytes):** A 32-bit, unsigned integer that specifies the drive serial number of the volume the link target is stored on.

**VolumeLabelOffset (4 bytes):** A 32-bit, unsigned integer that specifies the location of a string that contains the volume label of the drive that the link target is stored on. This value is an offset, in bytes, from the start of the VolumeID structure to a NULL-terminated string of characters, defined by the system default code page. The volume label string is located in the **Data** field of this structure.

If the value of this field is 0x00000014, it **MUST** be ignored, and the value of the **VolumeLabelOffsetUnicode** field **MUST** be used to locate the volume label string.

**VolumeLabelOffsetUnicode (4 bytes):** An optional, 32-bit, unsigned integer that specifies the location of a string that contains the volume label of the drive that the link target is stored on. This value is an offset, in bytes, from the start of the VolumeID structure to a NULL-terminated string of Unicode characters. The volume label string is located in the **Data** field of this structure.

If the value of the **VolumeLabelOffset** field is not 0x00000014, this field **MUST** be ignored, and the value of the **VolumeLabelOffset** field **MUST** be used to locate the volume label string.

**Data (variable):** A buffer of data that contains the volume label of the drive as a string defined by the system default code page or Unicode characters, as specified by preceding fields.

### 2.3.2 CommonNetworkRelativeLink

The CommonNetworkRelativeLink structure specifies information about the network location where a link target is stored, including the mapped drive letter and the UNC path prefix. For details on UNC paths, see [\[MS-DFSNM\]](#) section 2.2.1.4.

| 0                                  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|------------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| CommonNetworkRelativeLinkSize      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CommonNetworkRelativeLinkFlags     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| NetNameOffset                      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| DeviceNameOffset                   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| NetworkProviderType                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| NetNameOffsetUnicode (optional)    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| DeviceNameOffsetUnicode (optional) |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| NetName (variable)                 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| DeviceName (variable)              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| NetNameUnicode (variable)          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| DeviceNameUnicode (variable)       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**CommonNetworkRelativeLinkSize (4 bytes):** A 32-bit, unsigned integer that specifies the size, in bytes, of the CommonNetworkRelativeLink structure. This value **MUST** be greater than or equal to 0x00000014. All offsets specified in this structure **MUST** be less than this value, and all strings contained in this structure **MUST** fit within the extent defined by this size.

**CommonNetworkRelativeLinkFlags (4 bytes):** Flags that specify the contents of the **DeviceNameOffset** and **NetProviderType** fields.



|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| A | B | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

Where the bits are defined as:

| Value             | Description  |
|-------------------|--|
| A<br>ValidDevice  | If set, the <b>DeviceNameOffset</b> field contains an offset to the device name.<br>If not set, the <b>DeviceNameOffset</b> field does not contain an offset to the device name, and its value MUST be zero. |
| B<br>ValidNetType | If set, the <b>NetProviderType</b> field contains the network provider type.<br>If not set, the <b>NetProviderType</b> field does not contain the network provider type, and its value MUST be zero.         |

**NetNameOffset (4 bytes):** A 32-bit, unsigned integer that specifies the location of the **NetName** field. This value is an offset, in bytes, from the start of the CommonNetworkRelativeLink structure.

**DeviceNameOffset (4 bytes):** A 32-bit, unsigned integer that specifies the location of the **DeviceName** field. If the **ValidDevice** flag is set, this value is an offset, in bytes, from the start of the CommonNetworkRelativeLink structure; otherwise, this value MUST be zero.

**NetworkProviderType (4 bytes):** A 32-bit, unsigned integer that specifies the type of network provider. If the **ValidNetType** flag is set, this value MUST be one of the following; otherwise, this value MUST be ignored.

| Vendor name          | Value      |
|----------------------|------------|
| WNNC_NET_AVID        | 0x001A0000 |
| WNNC_NET_DOCUSPACE   | 0x001B0000 |
| WNNC_NET_MANGOSOFT   | 0x001C0000 |
| WNNC_NET_SERNET      | 0x001D0000 |
| WNNC_NET_RIVERFRONT1 | 0x001E0000 |
| WNNC_NET_RIVERFRONT2 | 0x001F0000 |
| WNNC_NET_DECORB      | 0x00200000 |
| WNNC_NET_PROTSTOR    | 0x00210000 |
| WNNC_NET_FJ_REDIR    | 0x00220000 |
| WNNC_NET_DISTINCT    | 0x00230000 |
| WNNC_NET_TWINS       | 0x00240000 |
| WNNC_NET_RDR2SAMPLE  | 0x00250000 |
| WNNC_NET_CSC         | 0x00260000 |

| Vendor name          | Value      |
|----------------------|------------|
| WNNC_NET_3IN1        | 0x00270000 |
| WNNC_NET_EXTENDNET   | 0x00290000 |
| WNNC_NET_STAC        | 0x002A0000 |
| WNNC_NET_FOXBAT      | 0x002B0000 |
| WNNC_NET_YAHOO       | 0x002C0000 |
| WNNC_NET_EXIFS       | 0x002D0000 |
| WNNC_NET_DAV         | 0x002E0000 |
| WNNC_NET_KNOWARE     | 0x002F0000 |
| WNNC_NET_OBJECT_DIRE | 0x00300000 |
| WNNC_NET_MASFAX      | 0x00310000 |
| WNNC_NET_HOB_NFS     | 0x00320000 |
| WNNC_NET_SHIVA       | 0x00330000 |
| WNNC_NET_IBMAL       | 0x00340000 |
| WNNC_NET_LOCK        | 0x00350000 |
| WNNC_NET_TERMSRV     | 0x00360000 |
| WNNC_NET_SRT         | 0x00370000 |
| WNNC_NET_QUINCY      | 0x00380000 |
| WNNC_NET_OPENAFS     | 0x00390000 |
| WNNC_NET_AVID1       | 0X003A0000 |
| WNNC_NET_DFS         | 0x003B0000 |
| WNNC_NET_KWNP        | 0x003C0000 |
| WNNC_NET_ZENWORKS    | 0x003D0000 |
| WNNC_NET_DRIVEONWEB  | 0x003E0000 |
| WNNC_NET_VMWARE      | 0x003F0000 |
| WNNC_NET_RSFX        | 0x00400000 |
| WNNC_NET_MFILES      | 0x00410000 |
| WNNC_NET_MS_NFS      | 0x00420000 |
| WNNC_NET_GOOGLE      | 0x00430000 |

**NetNameOffsetUnicode (4 bytes):** An optional, 32-bit, unsigned integer that specifies the location of the **NetNameUnicode** field. This value is an offset, in bytes, from the start of the

CommonNetworkRelativeLink structure. This field MUST be present if the value of the **NetNameOffset** field is greater than 0x00000014; otherwise, this field MUST NOT be present.

**DeviceNameOffsetUnicode (4 bytes):** An optional, 32-bit, unsigned integer that specifies the location of the **DeviceNameUnicode** field. This value is an offset, in bytes, from the start of the CommonNetworkRelativeLink structure. This field MUST be present if the value of the **NetNameOffset** field is greater than 0x00000014; otherwise, this field MUST NOT be present.

**NetName (variable):** A NULL-terminated string, as defined by the system default code page, which specifies a server share path; for example, "\\server\share".

**DeviceName (variable):** A NULL-terminated string, as defined by the system default code page, which specifies a device; for example, the drive letter "D:".

**NetNameUnicode (variable):** An optional, NULL-terminated, Unicode string that is the Unicode version of the **NetName** string. This field MUST be present if the value of the **NetNameOffset** field is greater than 0x00000014; otherwise, this field MUST NOT be present.

**DeviceNameUnicode (variable):** An optional, NULL-terminated, Unicode string that is the Unicode version of the **DeviceName** string. This field MUST be present if the value of the **NetNameOffset** field is greater than 0x00000014; otherwise, this field MUST NOT be present.

## 2.4 StringData

StringData refers to a set of structures that convey user interface and path identification information. The presence of these optional structures is controlled by [LinkFlags \(section 2.1.1\)](#) in the [ShellLinkHeader \(section 2.1\)](#).

The StringData structures conform to the following ABNF rules [\[RFC5234\]](#).

```
STRING_DATA = [NAME_STRING] [RELATIVE_PATH] [WORKING_DIR]
               [COMMAND_LINE_ARGUMENTS] [ICON_LOCATION]
```

**NAME\_STRING:** An optional structure that specifies a description of the shortcut that is displayed to end users to identify the purpose of the shell link. This structure MUST be present if the **HasName** flag is set.

**RELATIVE\_PATH:** An optional structure that specifies the location of the link target relative to the file that contains the shell link. When specified, this string SHOULD be used when resolving the link. This structure MUST be present if the **HasRelativePath** flag is set.

**WORKING\_DIR:** An optional structure that specifies the file system path of the working directory to be used when activating the link target. This structure MUST be present if the **HasWorkingDir** flag is set.

**COMMAND\_LINE\_ARGUMENTS:** An optional structure that stores the command-line arguments that should be specified when activating the link target. This structure MUST be present if the **HasArguments** flag is set.

**ICON\_LOCATION:** An optional structure that specifies the location of the icon to be used when displaying a shell link item in an icon view. This structure **MUST** be present if the **HasIconLocation** flag is set.

All StringData structures have the following structure.

|                 |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |                   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
|-----------------|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|-------------------|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|
| 0               | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6                 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| CountCharacters |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   | String (variable) |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
| ...             |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |                   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |

- CountCharacters (2 bytes):** A 16-bit, unsigned integer that specifies either the number of characters, defined by the system default code page, or the number of Unicode characters found in the **String** field. A value of zero specifies an empty string.
- String (variable):** An optional set of characters, defined by the system default code page, or a Unicode string with a length specified by the **CountCharacters** field. This string **MUST NOT** be NULL-terminated.

2.5 ExtraData

ExtraData refers to a set of structures that convey additional information about a link target. These optional structures can be present in an extra data section that is appended to the basic Shell Link Binary File Format.

The ExtraData structures conform to the following ABNF rules [\[RFC5234\]](#):

```
EXTRA_DATA      = *EXTRA_DATA_BLOCK TERMINAL_BLOCK

EXTRA_DATA_BLOCK = CONSOLE_PROPS / CONSOLE_FE_PROPS / DARWIN_PROPS /
ENVIRONMENT_PROPS / ICON_ENVIRONMENT_PROPS /
KNOWN_FOLDER_PROPS / PROPERTY_STORE_PROPS /
SHIM_PROPS / SPECIAL_FOLDER_PROPS /
TRACKER_PROPS / VISTA_AND_ABOVE_IDLIST_PROPS
```

**EXTRA\_DATA:** A structure consisting of zero or more property data blocks followed by a terminal block.

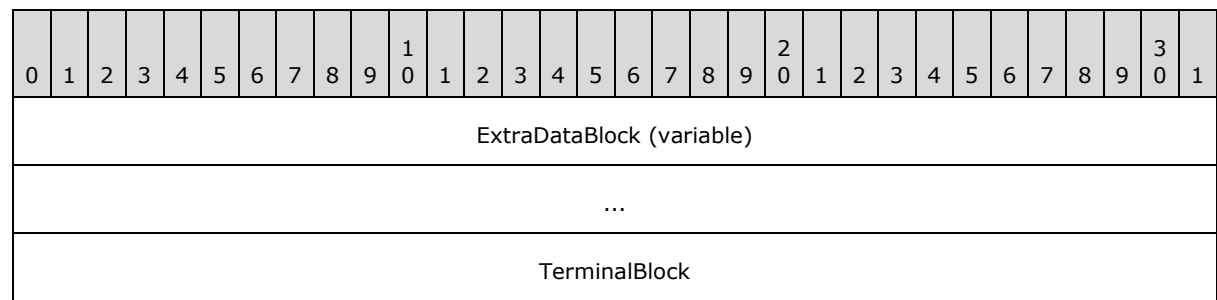
**EXTRA\_DATA\_BLOCK:** A structure consisting of any one of the following property data blocks.

- **CONSOLE\_PROPS:** A [ConsoleDataBlock](#) structure (section [2.5.1](#)).
- **CONSOLE\_FE\_PROPS:** A [ConsoleFEDataBlock](#) structure (section [2.5.2](#)).
- **DARWIN\_PROPS:** A [DarwinDataBlock](#) structure (section [2.5.3](#)).
- **ENVIRONMENT\_PROPS:** An [EnvironmentVariableDataBlock](#) structure (section [2.5.4](#)).
- **ICON\_ENVIRONMENT\_PROPS:** An [IconEnvironmentDataBlock](#) structure (section [2.5.5](#)).
- **KNOWN\_FOLDER\_PROPS:** A [KnownFolderDataBlock](#) structure (section [2.5.6](#)).

- **PROPERTY\_STORE\_PROPS:** A [PropertyStoreDataBlock](#) structure (section [2.5.7](#)).
- **SHIM\_PROPS:** A [ShimDataBlock](#) structure (section [2.5.8](#)).
- **SPECIAL\_FOLDER\_PROPS:** A [SpecialFolderDataBlock](#) structure (section [2.5.9](#)).
- **TRACKER\_PROPS:** A [TrackerDataBlock](#) structure (section [2.5.10](#)).
- **VISTA\_AND\_ABOVE\_IDLIST\_PROPS:** A [VistaAndAboveIDListDataBlock](#) structure (section [2.5.11](#)).

**TERMINAL\_BLOCK** A structure that indicates the end of the extra data section.

The general structure of an extra data section is shown in the following diagram.

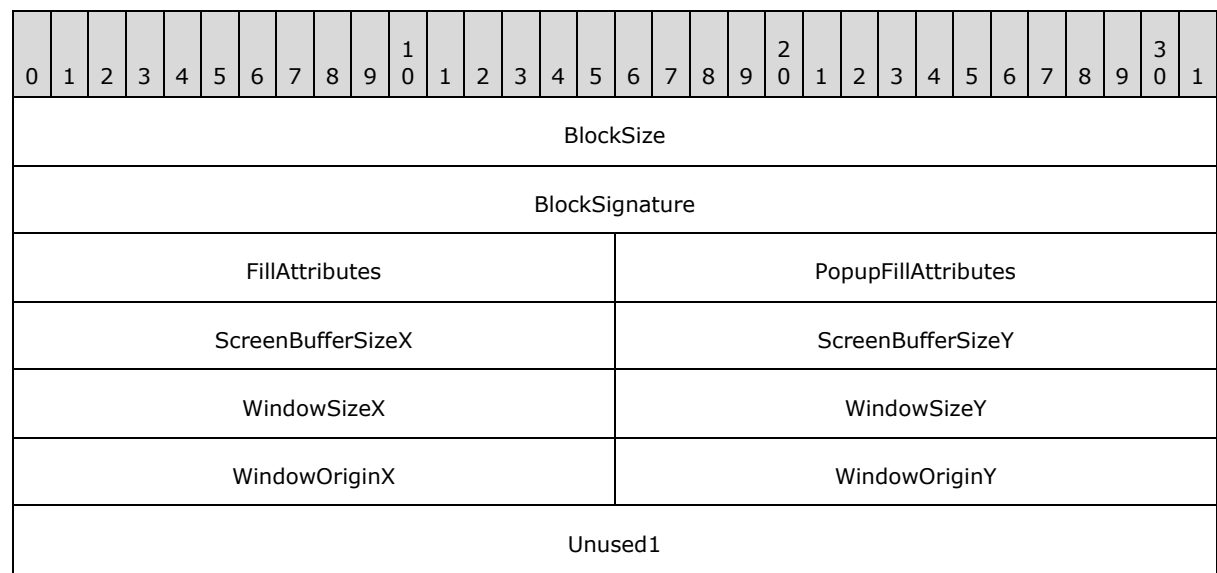


**ExtraDataBlock (variable):** An optional array of bytes that contains zero or more property data blocks listed in the **EXTRA\_DATA\_BLOCK** syntax rule.

**TerminalBlock (4 bytes):** A 32-bit, unsigned integer that indicates the end of the extra data section. This value **MUST** be less than 0x00000004.

### 2.5.1 ConsoleDataBlock

The ConsoleDataBlock structure specifies the display settings to use when a link target specifies an application that is run in a console window. [<2>](#)



|                               |
|-------------------------------|
| Unused2                       |
| FontSize                      |
| FontFamily                    |
| FontWeight                    |
| Face Name                     |
| ...                           |
| ...                           |
| ...                           |
| ...                           |
| ...                           |
| ...                           |
| ...                           |
| (Face Name cont'd for 8 rows) |
| CursorSize                    |
| FullScreen                    |
| QuickEdit                     |
| InsertMode                    |
| AutoPosition                  |
| HistoryBufferSize             |
| NumberOfHistoryBuffers        |
| HistoryNoDup                  |
| ColorTable                    |
| ...                           |

|                                |
|--------------------------------|
| ...                            |
| ...                            |
| ...                            |
| ...                            |
| ...                            |
| ...                            |
| (ColorTable cont'd for 8 rows) |

**BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the ConsoleDataBlock structure. This value MUST be 0x000000CC.

**BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the ConsoleDataBlock extra data section. This value MUST be 0xA0000002.

**FillAttributes (2 bytes):** A 16-bit, unsigned integer that specifies the fill attributes that control the foreground and background text colors in the console window. The following bit definitions can be combined to specify 16 different values each for the foreground and background colors:

| Value                          | Meaning                                   |
|--------------------------------|---|
| FOREGROUND_BLUE<br>0x0001      | The foreground text color contains blue.  |
| FOREGROUND_GREEN<br>0x0002     | The foreground text color contains green. |
| FOREGROUND_RED<br>0x0004       | The foreground text color contains red.   |
| FOREGROUND_INTENSITY<br>0x0008 | The foreground text color is intensified. |
| BACKGROUND_BLUE<br>0x0010      | The background text color contains blue.  |
| BACKGROUND_GREEN<br>0x0020     | The background text color contains green. |
| BACKGROUND_RED<br>0x0040       | The background text color contains red.   |
| BACKGROUND_INTENSITY<br>0x0080 | The background text color is intensified. |

**PopupFillAttributes (2 bytes):** A 16-bit, unsigned integer that specifies the fill attributes that control the foreground and background text color in the console window popup. The values are the same as for the **FillAttributes** field.

**ScreenBufferSizeX (2 bytes):** A 16-bit, signed integer that specifies the horizontal size (X axis), in characters, of the console window buffer.

**ScreenBufferSizeY (2 bytes):** A 16-bit, signed integer that specifies the vertical size (Y axis), in characters, of the console window buffer.

**WindowSizeX (2 bytes):** A 16-bit, signed integer that specifies the horizontal size (X axis), in characters, of the console window.

**WindowSizeY (2 bytes):** A 16-bit, signed integer that specifies the vertical size (Y axis), in characters, of the console window.

**WindowOriginX (2 bytes):** A 16-bit, signed integer that specifies the horizontal coordinate (X axis), in pixels, of the console window origin.

**WindowOriginY (2 bytes):** A 16-bit, signed integer that specifies the vertical coordinate (Y axis), in pixels, of the console window origin.

**Unused1 (4 bytes):** A value that is undefined and MUST be ignored.

**Unused2 (4 bytes):** A value that is undefined and MUST be ignored.

**FontSize (4 bytes):** A 32-bit, unsigned integer that specifies the size, in pixels, of the font used in the console window.

**FontFamily (4 bytes):** A 32-bit, unsigned integer that specifies the family of the font used in the console window. This value MUST be one of the following:

| Value                   | Meaning  |
|-------------------------|--|
| FF_DONTCARE<br>0x0000   | The font family is unknown.  |
| FF_ROMAN<br>0x0010      | The font is variable-width with serifs; for example, "Times New Roman".      |
| FF_SWISS<br>0x0020      | The font is variable-width without serifs; for example, "Arial".             |
| FF_MODERN<br>0x0030     | The font is fixed-width, with or without serifs; for example, "Courier New". |
| FF_SCRIPT<br>0x0040     | The font is designed to look like handwriting; for example, "Cursive".       |
| FF_DECORATIVE<br>0x0050 | The font is a novelty font; for example, "Old English".                      |

**FontWeight (4 bytes):** A 16-bit, unsigned integer that specifies the stroke weight of the font used in the console window.



| Value                   | Meaning                |
|-------------------------|------------------------|
| $700 \leq \text{value}$ | A bold font.           |
| $\text{value} < 700$    | A regular-weight font. |

**Face Name (64 bytes):** A 32-character Unicode string that specifies the face name of the font used in the console window.

**CursorSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the cursor, in pixels, used in the console window.

| Value                  | Meaning          |
|------------------------|------------------|
| $\text{value} \leq 25$ | A small cursor.  |
| 26 — 50                | A medium cursor. |
| 51 — 100               | A large cursor.  |

**FullScreen (4 bytes):** A 32-bit, unsigned integer that specifies whether to open the console window in full-screen mode.

| Value                       | Meaning                  |
|-----------------------------|--------------------------|
| 0x00000000                  | Full-screen mode is off. |
| $0x00000000 < \text{value}$ | Full-screen mode is on.  |

**QuickEdit (4 bytes):** A 32-bit, unsigned integer that specifies whether to open the console window in QuickEdit mode. In QuickEdit mode, the mouse can be used to cut, copy, and paste text in the console window.

| Value                       | Meaning                |
|-----------------------------|------------------------|
| 0x00000000                  | QuickEdit mode is off. |
| $0x00000000 < \text{value}$ | QuickEdit mode is on.  |

**InsertMode (4 bytes):** A 32-bit, unsigned integer that specifies insert mode in the console window.

| Value                       | Meaning                  |
|-----------------------------|--------------------------|
| 0x00000000                  | Insert mode is disabled. |
| $0x00000000 < \text{value}$ | Insert mode is enabled.  |

**AutoPosition (4 bytes):** A 32-bit, unsigned integer that specifies auto-position mode of the console window.

| Value                     | Meaning   |
|---------------------------|---|
| 0x00000000                | The values of the <b>WindowOriginX</b> and <b>WindowOriginY</b> fields are used to position the console window. |
| 0x00000000 < <i>value</i> | The console window is positioned automatically.   |

**HistoryBufferSize (4 bytes):** A 32-bit, unsigned integer that specifies the size, in characters, of the buffer that is used to store a history of user input into the console window.

**NumberOfHistoryBuffers (4 bytes):** A 32-bit, unsigned integer that specifies the number of history buffers to use.

**HistoryNoDup (4 bytes):** A 32-bit, unsigned integer that specifies whether to remove duplicates in the history buffer.

| Value                     | Meaning                     |
|---------------------------|-----------------------------|
| 0x00000000                | Duplicates are not allowed. |
| 0x00000000 < <i>value</i> | Duplicates are allowed.     |

**ColorTable (64 bytes):** A table of 16 32-bit, unsigned integers specifying the **RGB** colors that are used for text in the console window. The values of the fill attribute fields **FillAttributes** and **PopupFillAttributes** are used as indexes into this table to specify the final foreground and background color for a character.

## 2.5.2 ConsoleFEDataBlock

The ConsoleFEDataBlock structure specifies the code page to use for displaying text when a link target specifies an application that is run in a console window. [<3>](#)

|                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0              | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| BlockSize      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BlockSignature |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CodePage       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the ConsoleFEDataBlock structure. This value MUST be 0x0000000C.

**BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the ConsoleFEDataBlock extra data section. This value MUST be 0xA0000004.

**CodePage (4 bytes):** A 32-bit, unsigned integer that specifies a code page language code identifier. For details concerning the structure and meaning of language code identifiers, see [\[MS-LCID\]](#). For additional background information, see [\[MSCHARSET\]](#) and [\[MSDN-CODEPAGE\]](#).

### 2.5.3 DarwinDataBlock

The DarwinDataBlock structure specifies an application identifier that can be used instead of a link target IDList to install an application when a shell link is activated.

|  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|--|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| BlockSize  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BlockSignature                                     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| DarwinDataAnsi                                     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| (DarwinDataAnsi cont'd for 57 rows)                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| DarwinDataUnicode (optional)                       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| (DarwinDataUnicode (optional) cont'd for 122 rows) |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the DarwinDataBlock structure. This value **MUST** be 0x00000314.

**BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the DarwinDataBlock extra data section. This value MUST be 0xA0000006.

**DarwinDataAnsi (260 bytes):** A NULL-terminated string, defined by the system default code page, which specifies an application identifier. This field SHOULD be ignored.

**DarwinDataUnicode (520 bytes):** An optional, NULL-terminated, Unicode string that specifies an application identifier. [<4>](#)

#### 2.5.4 EnvironmentVariableDataBlock

The `EnvironmentVariableDataBlock` structure specifies a path to environment variable information when the link target refers to a location that has a corresponding environment variable.

|                                 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
|---------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|---|
| 0                               | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| BlockSize                       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| BlockSignature                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| TargetAnsi                      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| ...                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| ...                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| ...                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| ...                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| ...                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| ...                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| (TargetAnsi cont'd for 57 rows) |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| TargetUnicode                   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| ...                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| ...                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| ...                             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |

|                                     |
|-------------------------------------|
| ...                                 |
| ...                                 |
| ...                                 |
| ...                                 |
| (TargetUnicode cont'd for 122 rows) |

- BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the EnvironmentVariableDataBlock structure. This value **MUST** be 0x00000314.
- BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the EnvironmentVariableDataBlock extra data section. This value **MUST** be 0xA0000001.
- TargetAnsi (260 bytes):** A NULL-terminated string, defined by the system default code page, which specifies a path to environment variable information.
- TargetUnicode (520 bytes):** An optional, NULL-terminated, Unicode string that specifies a path to environment variable information.

### 2.5.5 IconEnvironmentDataBlock

The IconEnvironmentDataBlock structure specifies the path to an icon. The path is encoded using environment variables, which makes it possible to find the icon across machines where the locations vary but are expressed using environment variables.

|                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0              | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| BlockSize      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BlockSignature |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| TargetAnsi     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

|                                     |
|-------------------------------------|
| ...                                 |
| (TargetAnsi cont'd for 57 rows)     |
| TargetUnicode                       |
| ...                                 |
| ...                                 |
| ...                                 |
| ...                                 |
| ...                                 |
| ...                                 |
| ...                                 |
| ...                                 |
| ...                                 |
| (TargetUnicode cont'd for 122 rows) |

- BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the IconEnvironmentDataBlock structure. This value MUST be 0x00000314.
- BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the IconEnvironmentDataBlock extra data section. This value MUST be 0xA0000007.
- TargetAnsi (260 bytes):** A NULL-terminated string, defined by the system default code page, which specifies a path that is constructed with environment variables.
- TargetUnicode (520 bytes):** An optional, NULL-terminated, Unicode string that specifies a path that is constructed with environment variables.

### 2.5.6 KnownFolderDataBlock

The KnownFolderDataBlock structure specifies the location of a known folder. This data can be used when a link target is a known folder to keep track of the folder so that the link target IDList can be translated when the link is loaded.

|                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0              | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| BlockSize      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BlockSignature |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| KnownFolderID  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

|        |
|--------|
| ...    |
| ...    |
| ...    |
| Offset |

**BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the KnownFolderDataBlock structure. This value **MUST** be 0x0000001C.

**BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the KnownFolderDataBlock extra data section. This value **MUST** be 0xA000000B.

**KnownFolderID (16 bytes):** A value in **GUID** packet representation ([\[MS-DTYP\]](#) section 2.3.2.2) that specifies the **folder GUID ID**.

**Offset (4 bytes):** A 32-bit, unsigned integer that specifies the location of the ItemID of the first child segment of the IDList specified by **KnownFolderID**. This value is the offset, in bytes, into the link target IDList.

## 2.5.7 PropertyStoreDataBlock

A PropertyStoreDataBlock structure specifies a set of properties that can be used by applications to store extra data in the shell link.

|                          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|--------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0                        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| BlockSize                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BlockSignature           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| PropertyStore (variable) |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the PropertyStoreDataBlock structure. This value **MUST** be greater than or equal to 0x0000000C.

**BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the PropertyStoreDataBlock extra data section. This value **MUST** be 0xA0000009.

**PropertyStore (variable):** A serialized property storage structure ([\[MS-PROPSTORE\]](#) section 2.2).

## 2.5.8 ShimDataBlock

The ShimDataBlock structure specifies the name of a **shim** that can be applied when activating a link target.

|                      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0                    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| BlockSize            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BlockSignature       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LayerName (variable) |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the ShimDataBlock structure. This value MUST be greater than or equal to 0x00000088.

**BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the ShimDataBlock extra data section. This value MUST be 0xA0000008.

**LayerName (variable):** A Unicode string that specifies the name of a shim layer to apply to a link target when it is being activated.

## 2.5.9 SpecialFolderDataBlock

The SpecialFolderDataBlock structure specifies the location of a special folder. This data can be used when a link target is a special folder to keep track of the folder, so that the link target IDList can be translated when the link is loaded.

|                 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0               | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| BlockSize       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BlockSignature  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| SpecialFolderID |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Offset          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the SpecialFolderDataBlock structure. This value MUST be 0x00000010.

**BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the SpecialFolderDataBlock extra data section. This value MUST be 0xA0000005.

**SpecialFolderID (4 bytes):** A 32-bit, unsigned integer that specifies the **folder integer ID**.

**Offset (4 bytes):** A 32-bit, unsigned integer that specifies the location of the ItemID of the first child segment of the IDList specified by **SpecialFolderID**. This value is the offset, in bytes, into the link target IDList.



## 2.5.10 TrackerDataBlock

The TrackerDataBlock structure specifies data that can be used to resolve a link target if it is not found in its original location when the link is resolved. This data is passed to the Link Tracking service [\[MS-DLTW\]](#) to find the link target.

|                      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0                    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| BlockSize            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| BlockSignature       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Length               |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Version              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| MachineID (variable) |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Droid                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| DroidBirth           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| ...                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

|     |
|-----|
| ... |
| ... |
| ... |

**BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the TrackerDataBlock structure. This value MUST be 0x00000060.

**BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the TrackerDataBlock extra data section. This value MUST be 0xA0000003.

**Length (4 bytes):** A 32-bit, unsigned integer. This value MUST be greater than or equal to 0x00000058.

**Version (4 bytes):** A 32-bit, unsigned integer. This value MUST be 0x00000000.

**MachineID (variable):** A character string, as defined by the system default code page, which specifies the **NetBIOS name** of the machine where the link target was last known to reside.

**Droid (32 bytes):** Two values in GUID packet representation ([\[MS-DTYP\]](#) section 2.3.2.2) that are used to find the link target with the Link Tracking service, as specified in [\[MS-DLTW\]](#).

**DroidBirth (32 bytes):** Two values in GUID packet representation that are used to find the link target with the Link Tracking service

### 2.5.11 VistaAndAboveIDListDataBlock

The VistaAndAboveIDListDataBlock structure specifies an alternate IDList that can be used instead of the [LinkTargetIDList](#) structure (section [2.2](#)) on platforms that support it.[<5>](#)

|                   |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
|-------------------|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|
| 0                 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| BlockSize         |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
| BlockSignature    |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
| IDList (variable) |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |
| ...               |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |   |   |   |   |   |   |   |   |    |   |

**BlockSize (4 bytes):** A 32-bit, unsigned integer that specifies the size of the VistaAndAboveIDListDataBlock structure. This value MUST be greater than or equal to 0x0000000A.

**BlockSignature (4 bytes):** A 32-bit, unsigned integer that specifies the signature of the VistaAndAboveIDListDataBlock extra data section. This value MUST be 0xA000000C.

**IDList (variable):** An [IDList](#) structure (section [2.2.1](#)).

## 3 Structure Examples

### 3.1 Shortcut to a File

This section presents a sample of the Shell Link Binary File Format, consisting of a shortcut to a file with the path "C:\test\a.txt".

The following is the hexadecimal representation of the contents of the shell link.

|      | x0 | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | xA | xB | xC | xD | xE | xF |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0000 | 4C | 00 | 00 | 00 | 01 | 14 | 02 | 00 | 00 | 00 | 00 | 00 | C0 | 00 | 00 | 00 |
| 0010 | 00 | 00 | 00 | 46 | 9B | 00 | 08 | 00 | 20 | 00 | 00 | 00 | D0 | E9 | EE | F2 |
| 0020 | 15 | 15 | C9 | 01 | D0 | E9 | EE | F2 | 15 | 15 | C9 | 01 | D0 | E9 | EE | F2 |
| 0030 | 15 | 15 | C9 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 |
| 0040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | BD | 00 | 14 | 00 |
| 0050 | 1F | 50 | E0 | 4F | D0 | 20 | EA | 3A | 69 | 10 | A2 | D8 | 08 | 00 | 2B | 30 |
| 0060 | 30 | 9D | 19 | 00 | 2F | 43 | 3A | 5C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 46 | 00 | 31 | 00 | 00 |
| 0080 | 00 | 00 | 00 | 2C | 39 | 69 | A3 | 10 | 00 | 74 | 65 | 73 | 74 | 00 | 00 | 32 |
| 0090 | 00 | 07 | 00 | 04 | 00 | EF | BE | 2C | 39 | 65 | A3 | 2C | 39 | 69 | A3 | 26 |
| 00A0 | 00 | 00 | 00 | 03 | 1E | 00 | 00 | 00 | 00 | F5 | 1E | 00 | 00 | 00 | 00 | 00 |
| 00B0 | 00 | 00 | 00 | 00 | 00 | 74 | 00 | 65 | 00 | 73 | 00 | 74 | 00 | 00 | 00 | 14 |
| 00C0 | 00 | 48 | 00 | 32 | 00 | 00 | 00 | 00 | 00 | 2C | 39 | 69 | A3 | 20 | 00 | 61 |
| 00D0 | 2E | 74 | 78 | 74 | 00 | 34 | 00 | 07 | 00 | 04 | 00 | EF | BE | 2C | 39 | 69 |
| 00E0 | A3 | 2C | 39 | 69 | A3 | 26 | 00 | 00 | 00 | 2D | 6E | 00 | 00 | 00 | 00 | 96 |
| 00F0 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 61 | 00 | 2E | 00 | 74 |
| 0100 | 00 | 78 | 00 | 74 | 00 | 00 | 00 | 14 | 00 | 00 | 00 | 3C | 00 | 00 | 00 | 1C |
| 0110 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 1C | 00 | 00 | 00 | 2D | 00 | 00 | 00 | 00 |
| 0120 | 00 | 00 | 00 | 3B | 00 | 00 | 00 | 11 | 00 | 00 | 00 | 03 | 00 | 00 | 00 | 81 |
| 0130 | 8A | 7A | 30 | 10 | 00 | 00 | 00 | 00 | 43 | 3A | 5C | 74 | 65 | 73 | 74 | 5C |
| 0140 | 61 | 2E | 74 | 78 | 74 | 00 | 00 | 07 | 00 | 2E | 00 | 5C | 00 | 61 | 00 | 2E |
| 0150 | 00 | 74 | 00 | 78 | 00 | 74 | 00 | 07 | 00 | 43 | 00 | 3A | 00 | 5C | 00 | 74 |
| 0160 | 00 | 65 | 00 | 73 | 00 | 74 | 00 | 60 | 00 | 00 | 00 | 03 | 00 | 00 | A0 | 58 |
| 0170 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 63 | 68 | 72 | 69 | 73 | 2D | 78 | 70 | 73 |

|             | x0 | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | xA | xB | xC | xD | xE | xF |
|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| <b>0180</b> | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 40 | 78 | C7 | 94 | 47 | FA | C7 | 46 | B3 |
| <b>0190</b> | 56 | 5C | 2D | C6 | B6 | D1 | 15 | EC | 46 | CD | 7B | 22 | 7F | DD | 11 | 94 |
| <b>01A0</b> | 99 | 00 | 13 | 72 | 16 | 87 | 4A | 40 | 78 | C7 | 94 | 47 | FA | C7 | 46 | B3 |
| <b>01B0</b> | 56 | 5C | 2D | C6 | B6 | D1 | 15 | EC | 46 | CD | 7B | 22 | 7F | DD | 11 | 94 |
| <b>01C0</b> | 99 | 00 | 13 | 72 | 16 | 87 | 4A | 00 | 00 | 00 | 00 |    |    |    |    |    |

**HeaderSize:** (4 bytes, offset 0x0000), 0x0000004C as required.

**LinkCLSID:** (16 bytes, offset 0x0004), 00021401-0000-0000-C000-0000000000046.

**LinkFlags:** (4 bytes, offset 0x0014), 0x0008009B means the following [LinkFlags \(section 2.1.1\)](#) are set:

- HasLinkTargetIDList
- HasLinkInfo
- HasRelativePath
- HasWorkingDir
- IsUnicode
- EnableTargetMetadata

**FileAttributes:** (4 bytes, offset 0x0018), 0x00000020, means the following [FileAttributesFlags \(section 2.1.2\)](#) are set:

- FILE\_ATTRIBUTE\_ARCHIVE

**CreationTime:** (8 bytes, offset 0x001C) FILETIME 9/12/08, 8:27:17PM.

**AccessTime:** (8 bytes, offset 0x0024) FILETIME 9/12/08, 8:27:17PM.

**WriteTime:** (8 bytes, offset 0x002C) FILETIME 9/12/08, 8:27:17PM.

**FileSize:** (4 bytes, offset 0x0034), 0x00000000.

**IconIndex:** (4 bytes, offset 0x0038), 0x00000000.

**ShowCommand:** (4 bytes, offset 0x003C), SW\_SHOWNORMAL(1).

**Hotkey:** (2 bytes, offset 0x0040), 0x0000.

**Reserved:** (2 bytes, offset 0x0042), 0x0000.

**Reserved2:** (4 bytes, offset 0x0044), 0 x00000000.

**Reserved3:** (4 bytes, offset 0x0048), 0 x00000000.

Because **HasLinkTargetIDList** is set, a [LinkTargetIDList](#) structure (section [2.2](#)) follows:

- **IDListSize:** (2 bytes, offset 0x004C), 0x00BD, the size of **IDList**.

- **IDList:** (189 bytes, offset 0x004E) an [IDList](#) structure (section [2.2.1](#)) follows:
  - **ItemIDList:** (187 bytes, offset 0x004E), [ItemID](#) structures (section [2.2.2](#)) follow:
    - **ItemIDSize:** (2 bytes, offset 0x004E), 0x0014
    - **Data:** (12 bytes, offset 0x0050), <18 bytes of data> [computer]
    - **ItemIDSize:** (2 bytes, offset 0x0062), 0x0019
    - **Data:** (23 bytes, offset 0x0064), <23 bytes of data> [c:]
    - **ItemIDSize:** (2 bytes, offset 0x007B), 0x0046
    - **Data:** (68 bytes, offset 0x007D), <68 bytes of data> [test]
    - **ItemIDSize:** (2 bytes, offset 0x00C1), 0x0048
    - **Data:** (68 bytes, offset 0x00C3), <70 bytes of data> [a.txt]
  - **TerminalID:** (2 bytes, offset 0x0109), 0x0000 indicates the end of the **IDList**.

Because **HasLinkInfo** is set, a [LinkInfo](#) structure (section [2.3](#)) follows:

- **LinkInfoSize:** (4 bytes, offset 0x010B), 0x0000003C
- **LinkInfoHeaderSize:** (4 bytes, offset 0x010F), 0x0000001C as specified in the LinkInfo structure definition.
- **LinkInfoFlags:** (4 bytes, offset 0x0113), 0x00000001 **VolumeIDAndLocalBasePath** is set.
- **VolumeIDOffset:** (4 bytes, offset 0x0117), 0x0000001C, references offset 0x0127.
- **LocalBasePathOffset:** (4 bytes, offset 0x011B), 0x0000002D, references the character string "C:\test\a.txt".
- **CommonNetworkRelativeLinkOffset:** (4 bytes, offset 0x011F), 0x00000000 indicates **CommonNetworkRelativeLink** is not present.
- **CommonPathSuffixOffset:** (4 bytes, offset 0x0123), 0x0000003B, references offset 0x00000146, the character string "" (empty string).
- **VolumeID:** (17 bytes, offset 0x0127), because **VolumeIDAndLocalBasePath** is set, a [VolumeID](#) structure (section [2.3.1](#)) follows:
  - **VolumeIDSize:** (4 bytes, offset 0x0127), 0x00000011 indicates the size of the **VolumeID** structure.
  - **DriveType:** (4 bytes, offset 0x012B), DRIVE\_FIXED(3).
  - **DriveSerialNumber:** (4 bytes, offset 0x012F), 0x307A8A81.
  - **VolumeLabelOffset:** (4 bytes, offset 0x0133), 0x00000010, indicates that Volume Label Offset Unicode is not specified and references offset 0x0137 where the Volume Label is stored.
  - **Data:** (1 byte, offset 0x0137), "" an empty character string.
- **LocalBasePath:** (14 bytes, offset 0x0138), because **VolumeIDAndLocalBasePath** is set, the character string "c:\test\a.txt" is present.

- **CommonPathSuffix:** (1 byte, offset 0x0146), "" an empty character string.

Because **HasRelativePath** is set, the **RELATIVE\_PATH StringData** structure (section [2.4](#)) follows:

- **CountCharacters:** (2 bytes, offset 0x0147), 0x0007 Unicode characters.
- **String** (14 bytes, offset 0x0149), the Unicode string: ".\a.txt".

Because **HasWorkingDir** is set, the **WORKING\_DIR StringData** structure (section [2.4](#)) follows:

- **CountCharacters:** (2 bytes, offset 0x0157), 0x0007 Unicode characters.
- **String** (14 bytes, offset 0x0159), the Unicode string: "c:\test".

Extra data section: (100 bytes, offset 0x0167), an **ExtraData** structure (section [2.5](#)) follows:

- **ExtraDataBlock** (96 bytes, offset 0x0167), the **TrackerDataBlock** structure (section [2.5.10](#)) follows:
  - **BlockSize:** (4 bytes, offset 0x0167), 0x00000060
  - **BlockSignature:** (4 bytes, offset 0x016B), 0xA000003, which identifies the TrackerDataBlock structure (section [2.5.10](#)).
  - **Length:** (4 bytes, offset 0x016F), 0x00000058, the required minimum size of this extra data block.
  - **Version:** (4 bytes, offset 0x0173), 0x00000000, the required version.
  - **MachineID:** (16 bytes, offset 0x0177), the character string "chris-xps", with zero fill.
  - **Droid:** (32 bytes, offset 0x0187), 2 GUID values.
  - **DroidBirth:** (32 bytes, offset 0x01A7), 2 GUID values.
- **TerminalBlock:** (4 bytes, offset 0x01C7), 0x00000000 indicates the end of the extra data section.

## 4 Security

None.

## 5 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Windows NT 3.1 operating system
- Windows NT 3.5 operating system
- Windows NT 3.51 operating system
- Windows NT 4.0 operating system
- Windows 2000 operating system
- Windows XP operating system
- Windows Server 2003 operating system
- Windows Vista operating system
- Windows Server 2008 operating system
- Windows 7 operating system
- Windows Server 2008 R2 operating system
- Windows 8 operating system
- Windows Server 2012 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product 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 2.3:](#) In Windows, Unicode characters are stored in this structure if the data cannot be represented as **ANSI** characters due to truncation of the values. In this case, the value of the **LinkInfoHeaderSize** field is greater than or equal to 36.

[<2> Section 2.5.1:](#) In Windows environments, this is commonly known as a "command prompt" window.

[<3> Section 2.5.2:](#) In Windows environments, this is commonly known as a "command prompt" window.

[<4> Section 2.5.3:](#) In Windows, this is a Windows Installer (MSI) application descriptor. For more information, see [\[MSDN-MSISHORTCUTS\]](#).



[<5> Section 2.5.11:](#) The VistaAndAboveIDListDataBlock structure is supported on Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 only.

## 6 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

## 7 Index

### A

[Applicability](#) 8

### C

[Change tracking](#) 50

[CommonNetworkRelativeLink packet](#) 24

[ConsoleDataBlock packet](#) 29

[ConsoleFEDataBlock packet](#) 34

### D

[DarwinDataBlock packet](#) 35

### E

[EnvironmentVariableDataBlock packet](#) 36

[Example - shortcut to file](#) 43

[ExtraData packet](#) 28

### F

[Fields - vendor-extensible](#) 8

[FileAttributesFlags packet](#) 13

### G

[Glossary](#) 5

### H

[HotKeyFlags packet](#) 15

### I

[IconEnvironmentDataBlock packet](#) 37

[IDList packet](#) 18

[Informative references](#) 7

[Introduction](#) 5

[ItemID packet](#) 19

### K

[KnownFolderDataBlock packet](#) 38

### L

[LinkFlags packet](#) 11

[LinkInfo packet](#) 19

[LinkTargetIDList packet](#) 18

[Localization](#) 8

### N

[Normative references](#) 6

### O

[Overview \(synopsis\)](#) 7

### P

[Product behavior](#) 48

[PropertyStoreDataBlock packet](#) 39

### R

References

[informative](#) 7

[normative](#) 6

[Relationship to protocols and other structures](#) 8

### S

[Security](#) 47

[ShellLinkHeader packet](#) 9

[ShimDataBlock packet](#) 39

[Shortcut to file example](#) 43

[SpecialFolderDataBlock packet](#) 40

[StringData packet](#) 27

[Structures](#) 9

### T

[TrackerDataBlock packet](#) 41

[Tracking changes](#) 50

### V

[Vendor-extensible fields](#) 8

[Versioning](#) 8

[VistaAndAboveIDListDataBlock packet](#) 42

[VolumeID packet](#) 22