[MS-INFODCF]:
InfoPath Data Connection File Download Protocol Specification

---

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
07/13/2009	1.02	Major	Revised and edited the technical content
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Minor	Updated the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the InfoPath Data Connection File Download Protocol. This protocol enables a protocol client to download information defining the connection parameters for a specific remote data store.

1.1 Glossary

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

ASCII
Augmented Backus-Naur Form (ABNF)
authentication
authorization
HTTP OK
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
UTF-8

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

data source
failure response
form definition (.xsf) file
form template (.xsn) file
HTTP method
message body
path component
path segment
query component
Request-URI
site
Status-Code
Status-Line
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
Universal Data Connection (.udc, .udcx) file

The following terms are specific to this document:

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

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-IPFF] Microsoft Corporation, "InfoPath Form Template Format"

[MS-IPFF2] Microsoft Corporation, "InfoPath Form Template Format Version 2"

[MS-UDCX] Microsoft Corporation, "Universal Data Connection 2.0 XML File Format"

[RFC1945] Berners-Lee, T., Fielding, R., and Frystyk, H., "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996, http://www.ietf.org/rfc/rfc1945.txt

[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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, http://www.ietf.org/rfc/rfc3986.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

[MSDN-UDCV2] Microsoft Corporation, "Universal Data Connection v2.0 Reference and Schema", http://msdn.microsoft.com/en-us/library/ms772017.aspx

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

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.ietf.org/rfc/rfc2818.txt

1.3 Protocol Overview (Synopsis)

The InfoPath Data Connection File Download protocol provides a method for protocol clients to request a **Universal Data Connection (.udc, .udcx) file** in the format specified by [MS-UDCX]. This method requires the protocol client has a **form template (.xsn) file** that uses information in the requested UDC file to connect to a **data source**. The protocol client provides query parameters in the **query component (1)** submitted to the protocol server to identify the requested UDC file and the form template (.xsn) file that contains a reference to this UDC file.

This protocol uses **HTTP** for transport. The protocol client can use the GET **HTTP method** to download the file, or the HEAD HTTP method to determine whether the file exists without downloading it.

A typical scenario for using this protocol would be to access a UDC file that is used by several different form template (.xsn) file**s** which are located on different **sites (2)**. In this scenario, the protocol server can restrict access to the UDC file by verifying the protocol client has **authorization** to use a form template (.xsn) file that contains a reference to this UDC file.

For more information about using the UDC file format, see [MSDN-UDCV2].

1.4 Relationship to Other Protocols

For message transport the InfoPath Data Connection File Download protocol uses the HTTP/1.0 protocol as described in [RFC1945], the HTTP/1.1 protocol as described in [RFC2616], or the HTTPS protocol as described [RFC2818].

The following diagram shows the transport stack that this protocol uses:

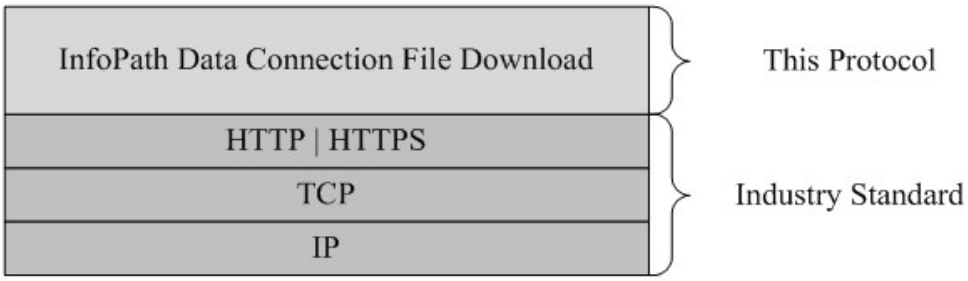

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a site(2) that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending "/_layouts/GetDataConnectionFile.aspx " to the URL of the site, for example http://www.contoso.com/Repository/_layouts/GetDataConnectionFile.aspx.

This protocol assumes that **authentication (2)** has been performed by the underlying protocols.

This protocol assumes that both the protocol client and protocol server have copies of a form template (.xsn) file resource. This protocol does not specify how the protocol client and protocol server obtain their respective copies of this resource.

1.6 Applicability Statement

This protocol is appropriate for providing protocol clients access over HTTP or HTTPS to a UDC file when both the following conditions apply:

- The protocol client is requesting the UDC file because it is used by a form template (.xsn) file that the protocol server also has a copy of.
- The UDC file is referenced by a **connectoid** element in the form template (.xsn) file as specified in [MS-IPFF] section 2.2.147.30 and [MS-IPFF2] section 2.2.2.2.23 with a **connectionLinkType** attribute equal to "store".
- The UDC file is in the format specified by [MS-UDCX].

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with HTTP as specified in Transport, section 2.1.
- This protocol refers to different file format specifications, [MS-IPFF] and [MS-IPFF2], both of which define the structure of a valid form template (.xsn) file. In cases where both specifications are cited as references, the **SolutionFormatVersion** attribute of the **xDocumentClass** element, as described in [MS-IPFF2] section 2.2.1.2.1, specifies whether to use the InfoPath Form Template Format Structure, as described in [MS-IPFF], or the InfoPath Form Template Format Version 2 Structure, as described in [MS-IPFF2].

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support HTTP. Protocol servers SHOULD additionally support **HTTPS** for securing communication with clients.

2.2 Message Syntax

All messages in this protocol MUST be valid HTTP requests and responses as specified in [RFC1945] or [RFC2616]. The HTTP version as specified in [RFC1945] or [RFC2616], section 3.1 MUST be either "HTTP/1.0" or "HTTP/1.1". The following subsections detail the relevant portions of HTTP request and response messages.

2.2.1 Request Syntax

2.2.1.1 Request HTTP Method

The protocol client MUST use the GET or HEAD HTTP methods specified in [RFC1945], section 8 or [RFC2616], section 9. The protocol server will determine whether to return a **message body** in the response based on the HTTP method, as specified in Response Syntax, section 2.2.2.

2.2.1.2 Request-URI Syntax

The **Request-URI** sent in the HTTP request MUST be a valid **URI** as specified in [RFC3986]. The **path component** of the Request-URI MUST end with "/_layouts/GetDataConnectionFile.aspx". The following **ABNF** specifies the syntax that the path component MUST adhere using the notation specified in [RFC5234]. The ABNF rules **path-absolute** and **path-rootless** are defined in [RFC3986], section 3.3.

```
path    = [base]"/_layouts/GetDataConnectionFile.aspx"
base    = path-absolute / path-rootless
```

The value of the **base** ABNF rule identifies the site the request operates on, and MUST be negotiated prior to initiating this protocol as described in Prerequisites/Preconditions, section 1.5.

The query component of the Request-URI MUST be present, and MUST contain 3 query parameters with the following case-insensitive **ASCII** names:

- "Udc"
- "Urn"
- "Version"

The following ABNF specifies the syntax for the query component. The ABNF for the **pct-encoded** rule is specified in [RFC3986].

```
query-component       = <query-parameter>"&"<query-parameter>"&"
   <query-parameter>
query-parameter       = name"="value
name                  = "Udc" / "Urn" / "Version"
value                 = 1*(allowed-char | optional-allowed-char)
allowed-char          = ALPHA / DIGIT / pct-encoded / "-" / "_" / "." /
```

```
    "!" / "@" / "$" / "," / "="
optional-allowed-char = "+" / "'" ; plus and apostrophe symbols
```

The **value** for all query parameters MUST be a non-empty ASCII string, and MUST NOT contain "&". The protocol server MUST support values that use only characters matching the **allowed-char** rule. The protocol server SHOULD<1> also support characters matching the **optional-allowed-char** rule. The escaped encoding specified in [RFC3986], section 2 SHOULD<2> be used for other punctuation, space, and non- ASCII characters in the query parameters.

The query component MUST NOT contain extra query parameters beyond the 3 required parameters. The protocol server MUST ignore the order of these parameters. For example, the following two query components are identical for purposes of this protocol:

```
Udc=sample.udcx&Urn=urn:sample&Version=1.0.0.0
```

and

```
Version=1.0.0.0&Udc=sample.udcx&Urn=urn:sample
```

Complete example Request-URIs are shown in Protocol Examples, section 4.

2.2.1.3 Request Headers Syntax

The following request header is relevant to this protocol:

- Accept - [RFC1945], section D.2 or [RFC2616], section 14.1. The protocol client SHOULD specify this header with the value "*/*". The protocol server MAY<3> ignore the value of this header.

2.2.2 Response Syntax

2.2.2.1 Response Status-Line

The response **Status-Line** MUST be valid according to [RFC1945] or [RFC2616], section 6.1. The protocol server MUST return a **HTTP OK** for successful requests.

The protocol server MUST return a 4*xx* or 5*xx* **Status-Code** as specified in [RFC1945] or [RFC2616], section 6.1.1 to indicate that a request failed. The protocol server MUST return the Status-Code "403" if the query component of the Request-URI does not match the syntax specified in section 2.2.1.2, Request-URI Syntax. The protocol server SHOULD<4> use the Status-Code "401" to indicate that the protocol client can retry the request using a different authentication (2) protocol or different properties.

Response Message Body Syntax , section 2.2.2.3 specifies the appropriate message body to return for different Status-Codes.

2.2.2.2 Response Headers Syntax

The following response headers are relevant to this protocol:

Content-Type: MUST be present and set to "text/xml; charset=utf-8" on HTTP OK responses.

2.2.2.3 Response Message Body Syntax

The following table specifies the required message body content based on the HTTP method and response Status-Code.

HTTP Method	Response Status-Code	Response Message-Body
GET	200	MUST be a valid UDC file in the format specified by [MS-UDCX] and MUST use **UTF-8** encoding.
GET	4*xx* or 5*xx*	MUST NOT be a UDC file. SHOULD<5> be a message describing the failure reason, but MAY<6> be omitted by the protocol server.
HEAD	Any valid Status-Code	MUST be omitted by the protocol server.

3 Protocol Details

The following sections specify details of the InfoPath Data Connection File Download protocol, including message processing rules.

3.1 Common Details

This section specifies details common to both protocol server and protocol client behavior.

Except where specified, protocol clients SHOULD interpret HTTP Status-Codes returned by the protocol server as specified in [RFC1945], section 9 or [RFC2616], section 10, Status Code Definitions.

This protocol allows protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults using HTTP Status-Codes.

3.1.1 Abstract Data Model

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

Requested UDC file: The UDC file that the protocol client is attempting to download using this protocol. The requested UDC file is identified by the *Udc* query parameter in the query component submitted to the protocol server as specified in the following table.

Request Site: The site this request is running on. This is determined from the **path segment** of the Request-URI that match the **base** ABNF rule as specified in Request-URI Syntax, section 2.2.1.2.

Referring Form Template: A form template (.xsn) file that references the requested UDC file. The protocol client and protocol server are both expected to have a copy of this file as discussed in Prerequisites/Preconditions, section 1.5. The referring form template is uniquely identified by the *Urn* and *Version* query parameters as specified in the following table.

Referring Element: An element in the **form definition (.xsf) file** of the referring form template that refers to the requested UDC File. The reason this protocol requires *Urn* and *Version* query parameters is so that the protocol server can verify a referring element exists. A process for finding a referring element is specified in Message Processing Events and Sequencing Rules, section 3.2.5.

Query Parameters: The **Udc**, **Urn**, and **Version** query parameters specified in Request-URI Syntax, section 2.2.1.2. The values of these parameters are strings matching the **value** rule in the ABNF specified in Request-URI Syntax, section 2.2.1.2. The protocol client and protocol server interpret values of these parameters as specified in the following table:

Parameter	Description
Udc	This parameter identifies the file name of the UDC file to be returned.
Urn	This parameter partially identifies the referring form template. The value of this parameter MUST be the value of the **name** attribute on the **xDocumentClass** element in the referring form template, as specified in [MS-IPFF] section 2.2.20 or [MS-IPFF2] section 2.2.1.2.1.

Parameter	Description
Version	This parameter partially identifies the referring form template. This parameter MUST be a string valid according to the **xdSolutionVersion** type specified in [MS-IPFF] section 2.2.10 or [MS-IPFF2] section 2.2.1.1.10. This value MUST be the value of the **solutionVersion** attribute on the **xDocumentClass** element specified in [MS-IPFF] section 2.2.20 or [MS-IPFF2] section 2.2.1.2.1 in the referring form template.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Protocol Server Details

3.2.1 Abstract Data Model

Requested UDC file: described in Common Details, Abstract Data Model, section 3.1.1 **.**

Request Site: described in Common Details, Abstract Data Model, section 3.1.1 .

Referring Form Template: described in the Common Details, Abstract Data Model, section 3.1.1 .

Referring Element: described in Common Details, Abstract Data Model, section 3.1.1 .

Query Parameters: described in Common Details, Abstract Data Model, section 3.1.1 .

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The protocol server MUST process request messages received from the protocol client as follows:

- First the protocol server MUST identify the request site from the Request-URI. The protocol server MUST return a **failure response** if there is no site associated with the Request-URI or if the protocol client does not have authorization to access the path in the request URI.
- Then the protocol server MUST find the referring form template identified by the **Urn** and **Version** query parameters using the interpretation of those parameters specified in Abstract Data Model, section 3.1.1 . The protocol server MUST return a failure response if no referring form template can be found on the Request Site, or if any implementation specific authorization for this file fails.
- Then the protocol server MUST find the requested UDC file identified by the **Udc** query parameter. The protocol server MUST return a failure response if it cannot find the requested UDC file or if any implementation specific authorization fails.
- The protocol server MUST find the referring element in the referring form template as follows, or return a failure response if no referring element can be found.
  - The referring element MUST be a **connectoid** element as specified in [MS-IPFF] section 2.2.147.30 or [MS-IPFF2] section 2.2.2.2.23.
  - This **connectoid** element MUST have a **source** attribute where the rightmost path **segment** equals the value of the **Udc** query parameter. The protocol server MUST use a case-insensitive ASCII string for this comparison.
  - This **connectoid** element MUST have a **connectionLinkType** attribute with the value "store".

If a **connectoid** element is found that passes all of these checks, then it is a valid referring element.

- After performing these validation steps and any other implementation specific validation<7>, the protocol server MUST return an appropriate HTTP response message as specified in Response Syntax, section 2.2.2.
  - If all validation steps succeed and a UDC file and referring element are found, then the protocol server MUST return a HTTP OK response. If the request HTTP method was GET, the protocol server MUST return a UTF-8 encoded message body containing the UDC file that is identified by the **Udc** query parameter. If the HTTP method was HEAD, the protocol server MUST NOT return a message body.
  - Otherwise a failure response MUST be returned. Response Status-Line Syntax, section 2.2.2.1 and Response Message Body Syntax, section 2.2.2.3 specify the response syntax for failure responses.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

3.3 Protocol Client Details

3.3.1 Abstract Data Model

Requested UDC file**:** described in Common Details, Abstract Data Model, section 3.1.1 **.**

Request Site: described in Common Details, Abstract Data Model, section 3.1.1 **.**

Referring Form Template: described in Common Details, Abstract Data Model, section 3.1.1 **.**

Referring Element: described in Common Details, Abstract Data Model, section 3.1.1 **.**

Query Parameters: described in Common Details, Abstract Data Model, section 3.1.1 **.**

3.3.2 Timers

None.

3.3.3 Initialization

None.

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

Request messages sent by the protocol client MUST be in the syntax specified in Message Syntax, section 2.2.1.

The protocol client MUST interpret the protocol server response based on the Status-Code as follows:

- 200 – The request was successful. If the HTTP method was GET, then the protocol client can assume the message body MUST be a valid UTF-8 encoded UDC file in the format specified by [MS-UDCX].
- 401 – The request failed because of an authentication (2) or authorization failure.
- 4*xx*/5*xx* – The request failed and if present the message body MUST NOT be interpreted as a UDC file.

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

4 Protocol Examples

The following examples show sample interactions between the protocol client and the protocol server.

4.1 Making a Successful GET Request

This example illustrates the messages exchanged when the protocol client makes a successful GET request to a protocol server using this protocol.

Protocol Client Request:

```
GET /_layouts/GetDataConnectionFile.aspx?Udc=sample.udcx&Urn=urn:sample&Version=1.0.0.0
HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0
Host: www.contoso.com
```

Protocol Server Response:

```
HTTP/1.1 200 OK
Connection: Keep-Alive
Cache-Control: Private
Date: Tue, 22 Jan 2008 01:13:30 GMT
Server: Microsoft-IIS/6.0
Content-Type: text/xml; charset=utf-8
Content-Length: 1234

<?xml version="1.0" encoding="UTF-8"?>
<?MicrosoftWindowsSharePointServices ContentTypeID="0x102030FF"?>
<udc:DataSource MajorVersion="2" MinorVersion="0"
xmlns:udc=http://schemas.microsoft.com/office/infopath/2006/udc>
...
</udc:DataSource>
```

For complete examples and specification of the **DataSource** element in a UDC file, see [MS-UDCX].

4.2 Making a Successful HEAD Request

This example illustrates the messages exchanged when the protocol client makes a successful HEAD request to a protocol server using this protocol.

Protocol Client Request:

```
HEAD /_layouts/GetDataConnectionFile.aspx?Udc=sample.udcx&Urn=urn:sample&Version=1.0.0.0
HTTP/1.1
Host: www.contoso.com
Content-Length: 0
Pragma: no-cache
```

Protocol Server response:

```
HTTP/1.1 200 OK
Cache-Control: private
Date: Tue, 22 Jan 2008 01:13:30 GMT
Server: Microsoft-IIS/6.0
Content-Type: text/xml; charset=utf-8
Content-Length: 1252
```

4.3 Receiving a Failure Response From the Protocol Server

This example illustrates the messages exchanged when the protocol server returns a failure response:

Protocol Client Request:

```
GET /_layouts/GetDataConnectionFile.aspx?Udc=NotFound.udcx&Urn=urn:sample&Version=1.0.0.1
HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0
Host: www.contoso.com
```

Protocol Server Response:

```
HTTP/1.1 403 Forbidden
Connection: Keep-Alive
Cache-Control: Private
Date: Tue, 22 Jan 2008 01:13:30 GMT
Server: Microsoft-IIS/6.0
Content-Length: 0
```

5 Security

5.1 Security Considerations for Implementers

The information in a UDC file is likely to be security-sensitive; for example it could contain server names, account names and passwords. Using a method of authentication (2) is recommended to establish the protocol client's identity and validate authorization for the requested resource.

The UDC file returned by this protocol could describe a data connection that is not located on the protocol server. Sending data to or receiving data from such a data connection could pose security risks if that data connection is un-trusted. The protocol client implementation can mitigate this risk by validating data is not transferred to an un-trusted location. The protocol server implementation can mitigate this risk by only allowing highly trusted users to add or modify the UDC file files to be returned by this protocol.

5.2 Index of Security Parameters

None.

6 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:

- Microsoft® Office Forms Server 2007
- Microsoft® Office InfoPath® 2007
- Microsoft® InfoPath® 2010
- Microsoft® Office SharePoint® Server 2007
- Microsoft® SharePoint® Server 2010

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 2.2.1.2: Characters that match the **optional-allowed-char** rule are supported in query parameter values by SharePoint Server 2010 Enterprise. Office SharePoint Server 2007 fails the request with a "403" Status-Code if either of the **optional-allowed-char** characters is present in a query parameter value.

<2> Section 2.2.1.2: The Office InfoPath 2007 client will send non- ASCII characters in the *udc* parameter if the **source** attribute on the **connectoid** element specified in [MS-IPFF], 2.2.147.30 contains characters not in the ASCII character set. Office SharePoint Server 2007 and SharePoint Server 2010 Enterprise will respond to these requests with a "403" failure Status-Code.

<3> Section 2.2.1.3: Office SharePoint Server 2007 and SharePoint Server 2010 Enterprise will ignore the Accept, Accept-Charset, Accept-Encoding, and Accept-Language headers when returning a UDC file. A message body of content-type "text/xml; charset=utf-8" can be returned even if that conflicts with the request headers sent by the protocol client.

<4> Section 2.2.2.1: Office SharePoint Server 2007 and SharePoint Server 2010 Enterprise will return a "401 Unauthorized" Status-Code if the client is not authorized to access the path in the Request-URI, and will return a "403 Forbidden" Status-Code if the client is authorized to access the path in the Request-URI but is not authorized to access a resource identified by the query parameters in the Request-URI.

<5> Section 2.2.2.3: The Office InfoPath 2007 and InfoPath 2010 protocol clients ignore the message body of any failure response.

<6> Section 2.2.2.3: Office SharePoint Server 2007 and SharePoint Server 2010 Enterprise will always return an empty body when returning a"403" Status-Code, and when returning a "401" Status-Code will return either an empty body or a text/html message body depending on authentication settings configured by the administrator and the credentials provided in the request message.

<7> Section 3.2.5: The Office SharePoint Server 2007 and SharePoint Server 2010 Enterprise implementations of this protocol do the following to address security considerations:

1. This implementation requires that both the UDC file returned by this protocol and the referring form template are uploaded by an administrator.

2. This implementation will validate that the protocol client has authorization to view the form template (.xsn) file identified by the query parameters in a Web browser.

7 Change Tracking

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

8 Index

T

V