

[MS-DOCTRANS]: Document Transforms Service Protocol Specification

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's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.msp>). 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 iplq@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.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail 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
04/04/2008	0.1		Initial Availability
04/25/2008	0.2	Editorial	Revised and edited the technical content
06/27/2008	1.0	Major	Revised and edited the technical content
08/15/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
07/13/2009	1.03	Major	Revised and edited the technical content
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	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	Editorial	Revised and edited 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	2.5	Minor	Clarified the meaning of the technical content.

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References.....	5
1.2.1	Normative References.....	5
1.2.2	Informative References	6
1.3	Protocol Overview (Synopsis)	6
1.4	Relationship to Other Protocols.....	7
1.5	Prerequisites/Preconditions	8
1.6	Applicability Statement.....	8
1.7	Versioning and Capability Negotiation.....	8
1.8	Vendor-Extensible Fields.....	8
1.9	Standards Assignments	8
2	Messages.....	9
2.1	Transport.....	9
2.2	Common Data Types	9
2.2.1	CreateDCInfo.....	10
2.2.2	CreateHtmlInfo	11
2.2.3	BrowserType.....	12
2.2.4	CreationErrorType	12
3	Protocol Details.....	14
3.1	IHtmlTrLoadBalancer Server Details	14
3.1.1	Abstract Data Model	14
3.1.2	Timers	14
3.1.3	Initialization	14
3.1.4	Message Processing Events and Sequencing Rules.....	14
3.1.4.1	RegisterLauncher.....	14
3.1.4.2	UnRegisterLauncher	15
3.1.4.3	StrGetLauncher	15
3.1.4.4	LauncherTaskCompleted.....	16
3.2	IHtmlTrLoadBalancer Client Details	16
3.2.1	Abstract Data Model	16
3.2.2	Timers	16
3.2.3	Initialization	16
3.3	IDocumentConversionsLauncher Server Details	16
3.3.1	Abstract Data Model	16
3.3.2	Timers	16
3.3.3	Initialization	16
3.3.4	Message Processing Events and Sequencing Rules.....	17
3.3.4.1	ConvertFile	17
3.4	IDocumentConversionsLauncher Client Details	18
3.4.1	Abstract Data Model	18
3.4.2	Timers	18
3.4.3	Initialization	18
3.5	IHtmlTrLauncher Server Details.....	18
3.5.1	Abstract Data Model	18
3.5.2	Timers	18
3.5.3	Initialization	18
3.5.4	Message Processing Events and Sequencing Rules.....	18

3.5.4.1	CHICreateHtml	18
3.6	IHtmlTrLauncher Client Details	19
3.6.1	Abstract Data Model	19
3.6.2	Timers	19
3.6.3	Initialization	19
3.6.4	Timer Events	19
3.6.5	Other Local Events	20
4	Protocol Examples	21
4.1	RegisterLauncher	21
4.1.1	RegisterLauncher Request Message	21
4.1.2	RegisterLauncher Response Message	21
4.2	StrGetLauncher	21
4.2.1	StrGetLauncher Request Message.....	21
4.2.2	StrGetLauncher Response Message.....	22
4.3	LauncherTaskCompleted	22
4.3.1	LauncherTaskCompleted Request Message	22
4.3.2	LauncherTaskCompleted Response Message	22
4.4	UnRegisterLauncher	23
4.4.1	UnRegisterLauncher Request Message	23
4.4.2	UnRegisterLauncher Response Message	23
4.5	ConvertFile	24
4.5.1	ConvertFile Request Message.....	24
4.5.2	ConvertFile Response Message.....	24
5	Security	26
5.1	Security Considerations for Implementers.....	26
5.2	Index of Security Parameters	26
6	Appendix A: Full Definition of Interfaces and Types	27
7	Appendix B: The ConfigInfo Parameter for Specific Conversion Applications	29
7.1	Common configInfo parameter structure.....	29
7.1.1	RcaTransformation	29
7.1.2	ConverterSettings	30
7.2	The configInfo parameter for the docxpageconverter.exe.....	30
7.3	The configInfo parameter for the XslApplicatorConverter.exe	30
7.4	The configInfo parameter for the InfoPathPageConverter.exe	31
7.5	Example Data.....	32
7.5.1	Example for the configInfo for the DocxPageConverter.exe.....	32
7.5.2	Example for the configInfo for the XslApplicatorConverter.exe	33
7.5.3	Example for the configInfo parameter for the InfoPathPageConverter.exe	33
8	Appendix C: Product Behavior	34
9	Change Tracking.....	35
10	Index	37

1 Introduction

This document specifies the Document Transforms Service Protocol, which allows a protocol client to call a file conversion service which converts a file from one file format to another.

1.1 Glossary

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

GUIDString
Unicode
XML

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

base64 encoding
document library
farm
file extension
form
relative path
root element
server-relative URL
Uniform Resource Locator (URL)
user-agent string
XML schema
XSL Transformation (XSLT)

The following terms are specific to this document:

conversion application: An application that can be used to convert a file from one format to another format.

conversion client: An entity that issues a request to convert a file from one format to another format.

conversion task: An operation that converts a file from one format to another format.

result file: The primary file that is generated by a conversion application when the conversion task is finished.

supporting file: Any additional file, other than the result file, that is created during document transformation.

supporting folder: A folder in which supporting files are stored.

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.

[ECMA-376] ECMA International, "Office Open XML File Formats", 1st Edition, ECMA-376, December 2006, <http://www.ecma-international.org/publications/standards/Ecma-376.htm>

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

[MS-NRTP] Microsoft Corporation, "[.NET Remoting: Core Protocol Specification](#)".

[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>

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[XML10] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Third Edition)", February 2004, <http://www.w3.org/TR/REC-xml>

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA1] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

1.2.2 Informative References

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

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

[RFC1738] Berners-Lee, T., Masinter, L., and McCahill, M., "Uniform Resource Locators (URL)", RFC 1738, December 1994, <http://www.ietf.org/rfc/rfc1738.txt>

1.3 Protocol Overview (Synopsis)

This protocol grants a protocol client the ability to call a launcher service which converts a file from one format to another. In typical operation, the protocol client requests the load balancer service for an available launcher service, and then passes a file to this launcher service for conversion. The launcher service responds by converting the file from the original file format to the target file format and then sends the converted file to the protocol client. A typical scenario for using this protocol is a content management application that grants a user the ability to convert into web pages documents which are created by a word processor.

The following diagram shows how this protocol is used.

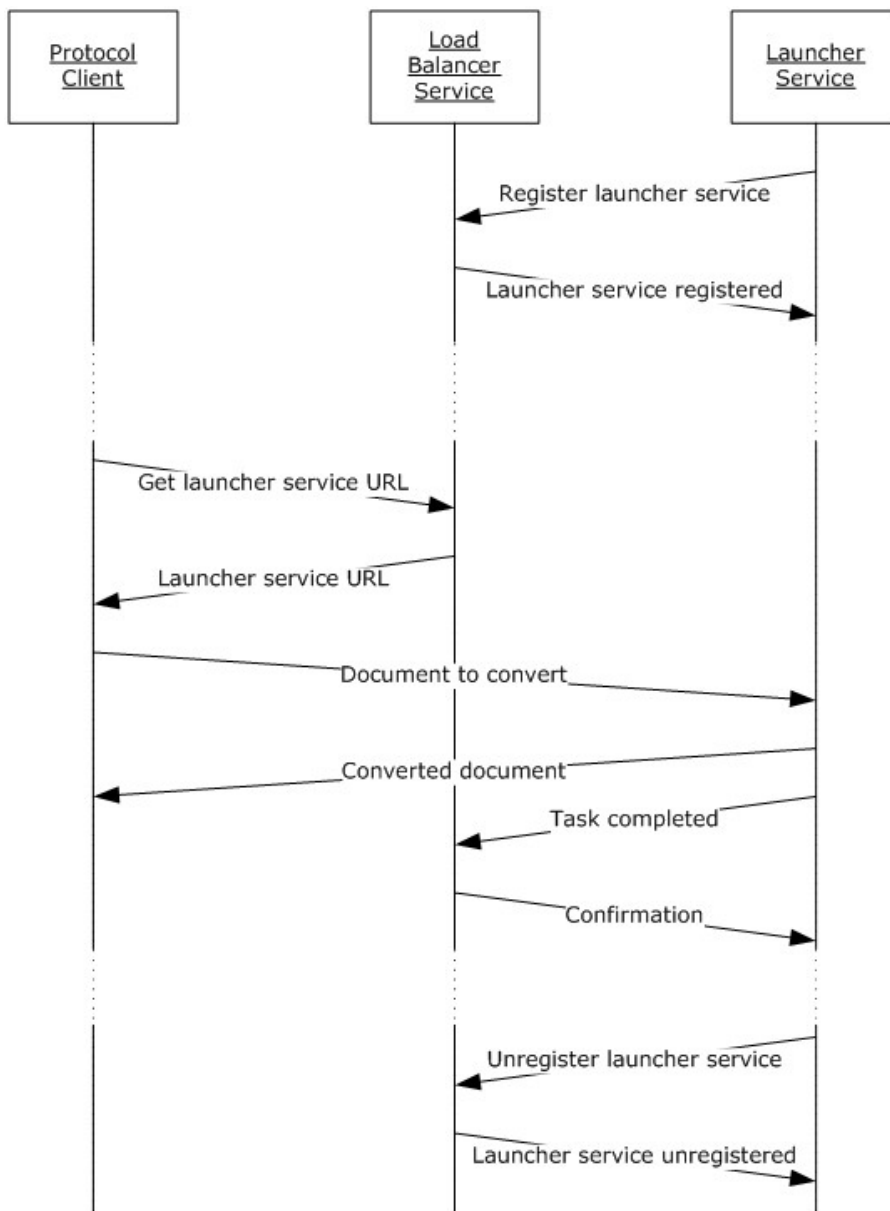


Figure 1: Overview of how this protocol is used to convert a document

1.4 Relationship to Other Protocols

The Document Transforms Service Protocol depends on the .NET Remoting Core Protocol described in [\[MS-NRTP\]](#), shown in the following layering diagram:

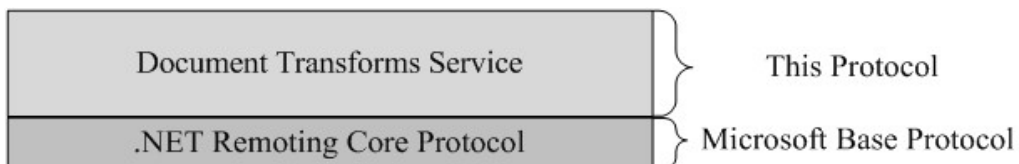


Figure 2: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

The protocol client and launcher service is configured so that the **URL** as described in [\[RFC1738\]](#) of the load balancer service is known or can be retrieved. The protocol client is configured so that the names of the **conversion applications** and the schemas of the **configInfo** parameters (see section [3.3.4.1](#)) match the conversion applications. How to configure the load balancer service to know or retrieve the URL, the names of the conversion application, and the **XML schemas** of the **configInfo** parameters, are all part of the administration of the **farm** and outside the scope of this specification.

1.6 Applicability Statement

This protocol applies only to the full conversion of files which don't require outside data sources (such as external images) from one format to another for which a designated conversion application has been supplied. This protocol does not apply to extracting only parts of files, such as a first page of a document. It is also not designed for low latency, as the conversion process typically take a long time. The protocol is designed to facilitate the distribution of **conversion tasks** to multiple servers.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

Vendors can install their own conversion applications in implementation-specific ways and use them through this protocol.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Document Transforms Service Protocol is composed of three interfaces, which are based on the protocol specified in [\[MS-NRTP\]](#).

- **IHtmlTrLoadBalancer**
- **IHtmlTrLauncher**
- **IDocumentConversionsLauncher**

The **IHtmlTrLoadBalancer** interface MUST be exposed by the implementing load balancer service at the following URL:

- [baseURL]\HtmlTrLoadBalancer

The **IHtmlTrLauncher** interface MUST be exposed by the implementing launcher service at the following URL:

- [baseURL]\HtmlTrLauncher

The **IDocumentConversionsLauncher** interface MUST be exposed by the implementing launcher service at the following URL:

- [baseURL]\HtmlTrLauncher

2.2 Common Data Types

The following table specifies the **Unicode** character codes that are not valid for use in some string fields as specified in this section. Characters that are not valid for a string are explicitly specified in the section associated with the string.

Character Codes that are Commonly Not Valid
0x0000 - 0x001f
0x0021
0x0022
0x002f
0x003a
0x003c
0x003e

Character Codes that are Commonly Not Valid
0x003f
0x005c
0x007c

2.2.1 CreateDCInfo

The **CreateDCInfo** common data type stores the result of a conversion task performed by a server that implements the **IDocumentConversionsLauncher** interface. The **CreateDCInfo** common data type is specified by the **Microsoft.HtmlTrans.Interface** library as follows.

```
namespace Microsoft.HtmlTrans
{
    class CreateDCInfo
    {
        CreationErrorType m_ce;
        String m_strMainFileName;
        Byte[] m_rgbMainFile;
        String[] m_rgstrSupportingFileNames;
        Byte[][] m_rgrgbSupportingFiles;
        String m_strSupportingFolderName;
        String m_strErrorString;
    }
}
```

m_ce: The error type of the conversion task that finished execution.

m_strMainFileName: The file name of the **result file**. If the conversion task finishes successfully, this member MUST be set to a non-empty value, MUST be 128 or fewer characters in length, and MUST NOT contain any of the characters specified in Section [2.2](#). If the conversion task was not successful, this member MUST be set to an empty string.

m_rgbMainFile: The **result file** of the conversion task. This member MUST be non-null if the conversion task finishes successfully and the **fReturnFileBits** parameter of **IDocumentConversionsLauncher.ConvertFile** is set to true. Otherwise, this member MUST be set to null.

m_rgstrSupportingFileNames: The filenames of the **supporting files**. MUST be set to a non-null value if any supporting files are created. Otherwise, this member MUST be set to null. Each string within the array MUST be 128 or fewer characters in length and MUST NOT contain any of the characters specified in Section [2.2](#).

m_rgrgbSupportingFiles: The supporting files created by the conversion task. The size of the array MUST be identical to that of the **m_rgstrSupportingFileNames** parameter. The index of a file in this array MUST be the same as the index of the corresponding file name in the **m_rgstrSupportingFileNames** member. The array MUST be set to null if no supporting files are created. Otherwise, this member MUST be set to a non-null value.

m_strSupportingFolderName: The name of the **supporting folder**. If any supporting files are created, this member MUST be set to a non-empty value, MUST be 128 or fewer characters in

length, and MUST NOT contain any of the characters specified in Section 2.2. Otherwise, this member MUST be set to an empty string value.

m_strErrorString: An implementation-specific description of the actions performed by the conversion application.

If the **CreateDCInfo** method returns supporting files, the result file MUST reference these files using **relative paths** under the assumption that the **supporting folder** appears at the same level in the directory hierarchy as the result file. For example, if the result file needs to reference a supporting file named "a.txt" and the name of the supporting folder is "B" a correct reference is in the form, "B\a.txt".

2.2.2 CreateHtmlInfo

The **CreateHtmlInfo** common data type stores the result of a conversion task performed by a server which implements the **IHtmlTrLauncher** interface. This common data type is specified by the **Microsoft.HtmlTrans.Interface** library as follows.

```
namespace Microsoft.HtmlTrans
{
    class CreateHtmlInfo
    {
        CreationErrorType m_ce;
        String m_strMainFileName;
        String m_strMainFilePath;
        Byte [][]m_rgrgbThicketFiles;
        String []m_rgstrThicketFileNames;
        Byte []m_rgbMainFile;
        String m_strThicketFolderName;
    }
}
```

m_ce: The error level of the conversion task that finished execution.

m_strMainFileName: The file name of the result file. If the conversion task finishes successfully, this member MUST be set to a non-empty value, MUST be 128 or fewer characters in length, and MUST NOT contain any of the characters specified in Section 2.2. If the conversion task is unsuccessful, this member MUST be set to an empty string.

m_strMainFilePath: A relative path for the result file and supporting files. This member MUST be set to a lower-case **GUIDString** followed by a "/" (forward slash) character.

m_rgbMainFile: The result file of the conversion task. This member MUST be set to a non-null value if the conversion task finishes successfully and the **fReturnFileBits** parameter of **IHtmlTrLauncher.CHICreateHtml** is set to true. Otherwise, this member MUST be set to null.

m_rgstrThicketFileNames: The filenames of the supporting files. This member MUST be set to a non-null value if any supporting files are created. Otherwise, this member MUST be set to null. Each string within the array MUST be 128 or fewer characters in length and MUST NOT contain any of the characters specified in Section 2.2.

m_rgrgbThicketFiles: The supporting files created by the conversion. The size of the array MUST be identical to that of **m_rgstrThicketFileNames**. The index of a file in this array MUST be the same as the index of the corresponding file name in **m_rgstrThicketFileNames**. The array MUST be set to null if no supporting files are created. Otherwise, this member MUST be set to a non-null value.

m_strThicketFolderName: The name of the supporting folder. If any supporting files are created, this member MUST be set to a non-empty value, MUST be 128 or fewer characters in length, and MUST NOT contain any of the characters specified in Section [2.2](#). Otherwise, this member MUST be set to an empty string.

If supporting files are returned in the **CreateHtmlInfo** common data type, the result file MUST reference the supporting files using relative paths under the assumption that the supporting folder appears at the same level in the directory hierarchy as the result file. For example, if the result file needs to reference a supporting file named "a.txt" and the name of the supporting folder is "B", a correct reference would be of the form, "B\a.txt".

2.2.3 BrowserType

The **BrowserType** enumeration enables the protocol client to specify the capabilities of its browser.

```
namespace Microsoft.HtmlTrans
{
    enum BrowerType : Int32
    {
        BT_IE3          = 0,
        BT_IE4          = 1,
        BT_UNKNOWN      = 2
    }
}
```

BT_IE3: The application name and version of the browser as specified by the **user-agent string** is set to one of the following values in application name/version format. For example: "Mozilla/1", "Mozilla/2", or "Mozilla/3".

BT_IE4: The application name and version of the browser as specified by the user-agent string is set to the following value in application name/version format: For example: "Mozilla/4".

BT_UNKNOWN: The application name and version of the browser as specified by the user-agent string is a value other than those specified by the **BT_IE3** or **BT_IE4** values.

2.2.4 CreationErrorType

The **CreationErrorType** enumeration specifies information for errors encountered during a conversion task.

```
namespace Microsoft.HtmlTrans
{
    enum CreationErrorType : Int32
    {
        CE_NONE          = 0,
        CE_CRASH         = 1,
        CE_HANG          = 2,
        CE_OTHER_BLOCKLIST = 3,
        CE_SERVERFILENOTFOUND = 4,
        CE_RESOURCESUNAVAILABLE = 5,
        CE_SPGETCONTENTFAILED = 6,
        CE_DISKWRITEERROR = 7,
        CE_BADINPUT      = 8,
        CE_BACKENDUNAVAILABLE = 9,
        CE_ALREADYRUNNING = 10,
        CE_STARTTIMEOUT  = 11,
    }
}
```

```

        CE_OTHER
    }
}

```

The following table specifies the values of the **CreationErrorType** enumeration. The enumerations **CE_SPGETCONTENTFAILED**, **CE_BACKENDUNAVAILABLE**, and **CE_ALREADYRUNNING** MUST NOT be used by this protocol because they are undefined.

Value	Meaning
CE_NONE	Conversion finished successfully and a result file was created.
CE_CRASH	The conversion application exited unexpectedly.
CE_HANG	The conversion application took longer than the specified timeout limit to complete the conversion task.
CE_OTHER_BLOCKLIST	The conversion application finished the conversion task, but did not generate a result file.
CE_SERVERFILENOTFOUND	The launcher service encountered an error.
CE_RESOURCESUNAVAILABLE	The launcher service was unable to create an event or allocate shared memory for the conversion task.
CE_SPGETCONTENTFAILED	This value is implementation-specific and MUST NOT be used.
CE_DISKWRITEERROR	The conversion application was unable to write the result file or one of the supporting files to the disk.
CE_BADINPUT	One of the parameter values passed to either the IDocumentConversionsLauncher.ConvertFile interface or the IHtmlTrLauncher.CHICreateHtml interface is not valid for one of the following reasons: <ul style="list-style-type: none"> ▪ The appExe parameter contains path information before the file name. ▪ The timeout parameter is less than 0 or greater than 1800. ▪ The file extension in the strReqFile parameter does not represent a supported file format for conversion.
CE_BACKENDUNAVAILABLE	This value is implementation-specific and MUST NOT be used.
CE_ALREADYRUNNING	This value is implementation-specific and MUST NOT be used.
CE_STARTTIMEOUT	The launcher service timed out waiting for the currently running conversion task to finish.
CE_OTHER	The conversion task did not finish successfully for a reason other than errors specified in this table. This includes failure to begin the conversion application process.

3 Protocol Details

The following sections specify details of the Document Transforms Service Protocol, including interface method syntax and message processing rules.

3.1 IHtmlTrLoadBalancer Server Details

3.1.1 Abstract Data Model

The load balancer service **MUST** maintain a table of registered launcher services and **MUST** maintain a table of in-progress conversion tasks and their associated start times. The following figure describes the tables that are maintained by the load balancer service.

Launcher Service Table	In-Progress Task Table
Launcher URL	Task Name Task Start Time

Figure 3: Tables that the IHtmlTrLoadBalancer interface maintains

3.1.2 Timers

None.

3.1.3 Initialization

The interface uses initialization as specified in [\[MS-NRTP\]](#).

3.1.4 Message Processing Events and Sequencing Rules

This interface comprises the following methods:

Method	Description	Used By
RegisterLauncher	Registers a launcher service with the load balancer service.	launcher service
UnRegisterLauncher	Unregisters a launcher service with the load balancer service.	launcher service
StrGetLauncher	Returns the URL of a launcher service that is available to perform a conversion task.	conversion client
LauncherTaskCompleted	Notifies the load balancer service that a conversion task is complete.	conversion client and launcher service

3.1.4.1 RegisterLauncher

The **RegisterLauncher** method is called to register a launcher service with the load balancer service. If the specified launcher service is not already registered with the load balancer service, the load balancer service **MUST** act on this call by adding an entry for the specified launcher service to its table of registered launcher services.

```
bool RegisterLauncher(String strLauncherUri);
```

strLauncherUri: The URL of the launcher service. This string MUST be a URL and MUST be 260 or fewer characters in length.

Return Value: This method MUST return a value of true if the launcher service is successfully registered or if the launcher service was already registered. Otherwise, this method MUST return false.

Exceptions Thrown: No exceptions are thrown beyond those thrown by the .NET Remoting Core Protocol as specified in [\[MS-NRTP\]](#).

3.1.4.2 UnRegisterLauncher

The **UnRegisterLauncher** method is called to unregister a launcher service from the load balancer service. The load balancer service MUST respond to this call by removing the launcher service from its table of registered launcher services.

```
bool UnRegisterLauncher(String strLauncherUri);
```

strLauncherUri: The URL of the launcher service. This string MUST be a URL and MUST be 260 or fewer characters in length.

Return Value: This method MUST return a value of true if the launcher service is successfully unregistered. It MUST return a value of false if unregistration fails or if the launcher service is unknown to the load balancer service.

Exceptions Thrown: No exceptions are thrown beyond those thrown by the .NET Remoting Core Protocol as specified in [\[MS-NRTP\]](#).

3.1.4.3 StrGetLauncher

The **StrGetLauncher** method passes a task name to the load balancer service and returns the URL of a launcher service that is available to perform a conversion task. If the load balancer service is able to assign a launcher service to the conversion task, it MUST add the task name of the conversion task and current time to its table of in-progress conversion tasks.

```
String StrGetLauncher(String strTaskName);
```

strTaskname: The name of the conversion task that runs on the returned launcher service. This parameter MUST be set to a non-empty value and MUST be 270 or fewer characters in length.

Return Value: If a conversion task with the specified name exists in the load balancer service's table of in-progress conversion tasks, and if the current time is less than 1200 seconds after the start time for the conversion task, the return value MUST be equal to "TransformCurrentlyRunning". Otherwise, if a launcher service is available to process the conversion task, the return value MUST be set to the URL of the launcher service assigned to the conversion task. Otherwise, the return value MUST be set to an empty string value.

Exceptions Thrown: No exceptions are thrown beyond those thrown by the .NET Remoting Core Protocol as specified in [\[MS-NRTP\]](#).

3.1.4.4 LauncherTaskCompleted

This protocol calls the **LauncherTaskCompleted** method to notify the load balancer service that a conversion task finished execution. The load balancer service MUST respond to this call by removing the task from its table of in-progress conversion tasks.

```
bool LauncherTaskCompleted(String strLauncherUri, String strTaskName);
```

strLauncherUri: The URL of the launcher service that was assigned to the conversion task. This parameter MUST be a URL and MUST be 260 or fewer characters in length.

strTaskName: The name of the conversion task that finished execution. This parameter MUST be set to a non-empty value and MUST be 270 or fewer characters in length.

Return Value: This method MUST return a value of true if the conversion task is successfully removed from the load balancer service's table of in-progress tasks. Otherwise, this method MUST return false.

Exceptions Thrown: No exceptions are thrown beyond those thrown by the .NET Remoting Core Protocol as specified in [\[MS-NRTP\]](#).

3.2 IHtmlTrLoadBalancer Client Details

3.2.1 Abstract Data Model

For each conversion task, the protocol client MUST maintain a record of the conversion task name passed to the **StrGetLauncher** method. The protocol client MUST use this same conversion task name when calling the **LauncherTaskCompleted** method for the conversion task. The protocol client also MUST pass this conversion task name in the **strTaskName** parameter when calling either the **IDocumentConversionsLauncher.ConvertFile** interface or the **IHtmlTrLoadBalancer.CHICreateHtml** interface.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.3 IDocumentConversionsLauncher Server Details

3.3.1 Abstract Data Model

None.

3.3.2 Timers

The **ConvertFile** method maintains a timer to convert the file as specified in Section [3.3.4.1](#).

3.3.3 Initialization

The interface uses initialization as specified in [\[MS-NRTP\]](#).

3.3.4 Message Processing Events and Sequencing Rules

This interface is comprised of the **ConvertFile** method, which takes a file passed as an array of bytes and uses the specified conversion application to convert it from one specified file format to another specified file format. When the conversion is done, the protocol server MUST call the **LauncherTaskCompleted** method of the load balancer service.

3.3.4.1 ConvertFile

The **ConvertFile** method takes a file passed as an array of bytes and uses the specified conversion application to convert it from one specified file format to another specified file format.

```
CreateDCInfo ConvertFile(String launcherUri,  
                          String appExe,  
                          String convertFrom,  
                          String convertTo,  
                          Byte[] fileBits,  
                          String taskName,  
                          String configInfo,  
                          Int32 timeout,  
                          bool fReturnFileBits);
```

launcherUri: The URL of the launcher service that performs the conversion. It MUST be 260 or fewer characters in length, and MUST be the URL of a launcher service that has implemented the **IDocumentConversionsLauncher** interface.

appExe: The name of the executable file of the conversion application that converts the document. This parameter MUST be set to a non-empty value.

convertFrom: The **file extension** of the file's original file format. This parameter MUST be set to a non-null value.

convertTo: The file extension of the file format to which the file MUST be converted. This parameter MUST be set to a non-null value.

fileBits: The file to be converted. This parameter MUST be set to a non-null value.

taskName: The name of the conversion task. This parameter MUST be set to a non-empty value and MUST be 270 or fewer characters in length.

configInfo: The configuration information that is specific to the conversion application. The value of this parameter is written to a file which is passed as a parameter to the conversion application. Specific uses of this parameter for specific types of conversions are discussed in [Section 7](#).

timeout: The number of seconds that the launcher service MUST wait for the conversion application to complete before canceling the conversion. The launcher service MUST cancel the conversion if the conversion application does not complete in the specified amount of time. This parameter MUST be set to a value greater than 0 and less than or equal to 1800.

fReturnFileBits: Specifies whether to return the result file in the **m_rgbMainFile** member of the **CreateDCInfo** common data type returned by the function. If this parameter is set to true, the result file MUST be returned in the **m_rgbMainFile** member. If set to false, the result file MUST NOT be returned.

Return Value: An object with information about the result of the conversion. This method MUST return a non-null value.

Exceptions Thrown: No exceptions are thrown beyond those thrown by the .NET Remoting Core Protocol as specified in [\[MS-NRTP\]](#).

3.4 IDocumentConversionsLauncher Client Details

3.4.1 Abstract Data Model

The URL that the protocol client passes as the **launcherUri** parameter to the **IDocumentConversionsLauncher** interface. The **taskName** parameter of the **ConvertFile** method MUST be set to a URL which was returned to the protocol client from an earlier call to the **IHtmlTrLoadBalancer.StrGetLauncher** method.

3.4.2 Timers

None.

3.4.3 Initialization

None.

3.5 IHtmlTrLauncher Server Details

Deprecated in favor of the **IDocumentConversionsLauncher** interface.

3.5.1 Abstract Data Model

None.

3.5.2 Timers

The **CHICreateHtml** method maintains a timer to create the HTML as specified in Section [3.5.4.1](#).

3.5.3 Initialization

The interface uses initialization as specified in [\[MS-NRTP\]](#).

3.5.4 Message Processing Events and Sequencing Rules

This interface is comprised of the **CHICreateHtml** method, which takes a file passed as an array of bytes and converts it to HTML that can be displayed by the specified type of browser.

3.5.4.1 CHICreateHtml

The **CHICreateHtml** method takes a file passed as an array of bytes and converts the file into HTML, which is displayed by the specified type of browser.

```
CreateHtmlInfo CHICreateHtml(String strLauncherUri,
                               Byte[] rgbFile,
                               BrowserType bt,
                               String strReqFile,
                               String strTaskName,
                               Int32 timeout,
                               bool fReturnFileBits);
```

strLauncherUri: The URL of the launcher service that performs the conversion. This parameter MUST be set to a value that is 260 or fewer characters in length, and MUST be the URL of a launcher service that implements the **IHtmlTrLauncher** interface.

rgbFile: The file to be converted. This parameter MUST be set to a non-null value.

bt: The type of browser in which the result file is displayed.

strReqFile: The URL of the file being converted. This parameter MUST be set to a non-null value, MUST be 260 or fewer characters in length, MUST end with a file extension, and MUST NOT contain any of the characters specified in Section [2.2](#).

strTaskName: The name of the conversion task. This parameter MUST be set to a non-empty value and MUST be 270 or fewer characters in length.

timeout: The number of seconds that the launcher service MUST wait for the conversion application to complete before canceling the conversion. The launcher service MUST cancel the conversion if the conversion application does not complete in the specified amount of time. This parameter MUST be set to a value greater than 0 and less than or equal to 1800.

fReturnFileBits: Specifies whether to return the result file in the **m_rgbMainFile** member of the **CreateHtmlInfo** common data type returned by the method. If this parameter is set to true, the result file MUST be returned in the **m_rgbMainFile** member. If this parameter is set to false, the result file MUST NOT be returned.

Return Value: An object with information about the result of the conversion. This method MUST return a non-null value. The **m_ce** property MUST be set to CE_OTHER, which indicates that the conversion failed. As mentioned in Section [3.5](#), this interface is deprecated.

Exceptions Thrown: No exceptions are thrown beyond those thrown by the .NET Remoting Core Protocol as specified in [\[MS-NRTP\]](#).

3.6 IHtmlTrLauncher Client Details

3.6.1 Abstract Data Model

The URL that the protocol client passes as the **launcherUri** parameter to the **IHtmlTrLauncher** interface. The **strTaskName** parameter of the **CHICreateHtml** method MUST be set to a non-empty value and MUST be 270 or fewer or fewer characters in length. The protocol client MUST use the user-agent string of its browser to determine the appropriate value of the **bt** parameter to the **IHtmlTrLauncher.CHICreateHtml** method.

3.6.2 Timers

None.

3.6.3 Initialization

None.

3.6.4 Timer Events

None.

3.6.5 Other Local Events

None.

4 Protocol Examples

The following sections provide samples of **XML** messages sent over the wire by the Document Transforms Service Protocol.

4.1 RegisterLauncher

In the following sample, a launcher service at `http://contoso:12345/HtmlTrLauncher` is registering itself with a load balancer service. The response contains the return value "true" to indicate that registration was successful.

4.1.1 RegisterLauncher Request Message

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:clr="http://schemas.microsoft.com/soap/encoding/clr/1.0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <i2:RegisterLauncher id="ref-1"
xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IHtmlTrLoadBalancer/Mi
crosoft.HtmlTrans.Interface">
      <strLauncherUri id="ref-3">http://contoso:12345/HtmlTrLauncher</strLauncherUri>
    </i2:RegisterLauncher>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.1.2 RegisterLauncher Response Message

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:clr="http://schemas.microsoft.com/soap/encoding/clr/1.0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <i2:RegisterLauncherResponse id="ref-1"
xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IHtmlTrLoadBalancer/Mi
crosoft.HtmlTrans.Interface">
      <return>true</return>
    </i2:RegisterLauncherResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.2 StrGetLauncher

In the following sample, the protocol client is requesting the URL of a launcher service to send the "brochure_to_html" conversion task to. The response contains the URL `http://contoso:12345/HtmlTrLauncher`.

4.2.1 StrGetLauncher Request Message

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
```

```

ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:clr="http://schemas.microsoft.com/soap/encoding/clr/1.0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <i2:StrGetLauncher id="ref-1"
xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IHtmlTrLoadBalancer/Mi
crosoft.HtmlTrans.Interface">
      <strTaskName id="ref-3">brochure_to_html</strTaskName>
    </i2:StrGetLauncher>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.2.2 StrGetLauncher Response Message

```

<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:clr="http://schemas.microsoft.com/soap/encoding/clr/1.0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <i2:StrGetLauncherResponse id="ref-1"
xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IHtmlTrLoadBalancer/Mi
crosoft.HtmlTrans.Interface">
      <return id="ref-3">http://contoso:12345/HtmlTrLauncher</return>
    </i2:StrGetLauncherResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.3 LauncherTaskCompleted

In the following sample, a launcher service at `http://contoso:12345/HtmlTrLauncher` notifies the load balancer service that the **brochure_to_html** task is complete. The response message is set to "true" to indicate that the task has been removed from the table of in-progress tasks.

4.3.1 LauncherTaskCompleted Request Message

```

<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:clr="http://schemas.microsoft.com/soap/encoding/clr/1.0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <i2:LauncherTaskCompleted id="ref-1"
xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IHtmlTrLoadBalancer/Mi
crosoft.HtmlTrans.Interface">
      <strLauncherUri id="ref-3">http://contoso:12345/HtmlTrLauncher</strLauncherUri>
      <strTaskName id="ref-4">brochure_to_html</strTaskName>
    </i2:LauncherTaskCompleted>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.3.2 LauncherTaskCompleted Response Message

```

<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-

```

```

ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:clr="http://schemas.microsoft.com/soap/encoding/clr/1.0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  <SOAP-ENV:Body>
    <i2:LauncherTaskCompletedResponse id="ref-1"
xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IHtmlTrLoadBalancer/Mi
crosoft.HtmlTrans.Interface">
      <return>true</return>
    </i2:LauncherTaskCompletedResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.4 UnRegisterLauncher

In the following sample, a launcher service at `http://contoso:12345/HtmlTrLauncher` notifies a load balancer service that it is no longer accepting conversion tasks. The return value of `true` indicates that the load balancer service successfully removed the launcher service from its table of registered launcher services.

4.4.1 UnRegisterLauncher Request Message

```

<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:clr="http://schemas.microsoft.com/soap/encoding/clr/1.0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <i2:UnRegisterLauncher id="ref-1"
xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IHtmlTrLoadBalancer/Mi
crosoft.HtmlTrans.Interface">
      <strLauncherUri id="ref-3">http://contoso:12345/HtmlTrLauncher</strLauncherUri>
    </i2:UnRegisterLauncher>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.4.2 UnRegisterLauncher Response Message

```

<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:clr="http://schemas.microsoft.com/soap/encoding/clr/1.0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <i2:UnRegisterLauncherResponse id="ref-1"
xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IHtmlTrLoadBalancer/Mi
crosoft.HtmlTrans.Interface">
      <return>true</return>
    </i2:UnRegisterLauncherResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4.5 ConvertFile

In the following sample, the conversion client requests that the launcher service at <http://contoso:12345/HtmlTrLauncher> use the **docxpageconverter.exe** conversion application to convert a document from the **.docx** file format to the **html** file format. The conversion client passes in the file in the **SOAP-ENC:Array** node with the **identifier** set to ref-7. The return value is a **CreateDCInfo** object with the **error level** set to CE_NONE, the **file name** of the converted file set to output.html, and the **converted file** in the SOAP-ENC:Array node. For brevity, the value A1B2C3D4 represents the content, encoded with **base64 encoding**, of the file to be converted and the converted file.

4.5.1 ConvertFile Request Message

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
  ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:clr="http://schemas.microsoft.com/soap/encoding clr/1.0" SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <i2:ConvertFile id="ref-1"
  xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IDocumentConversionsLa
  uncher/Microsoft.HtmlTrans.Interface">
      <launcherUri id="ref-3">http://contoso:12345/HtmlTrLauncher</launcherUri>
      <appExe id="ref-4">docxpageconverter.exe</appExe>
      <convertFrom id="ref-5">docx</convertFrom>
      <convertTo id="ref-6">html</convertTo>
      <fileBits href="#ref-7"/>
      <taskName id="ref-8">brochure_to_html</taskName>
      <configInfo id="ref-9"></configInfo>
      <timeout>20</timeout>
      <fReturnFileBits>true</fReturnFileBits>
    </i2:ConvertFile>
    <SOAP-ENC:Array id="ref-7" xsi:type="SOAP-ENC:base64">A1B2C3D4</SOAP-ENC:Array>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.5.2 ConvertFile Response Message

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
  ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:clr="http://schemas.microsoft.com/soap/encoding clr/1.0" SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <i2:ConvertFileResponse id="ref-1"
  xmlns:i2="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans.IDocumentConversionsLa
  uncher/Microsoft.HtmlTrans.Interface">
      <return href="#ref-4"/>
    </i2:ConvertFileResponse>
    <a1>CreateDCInfo id="ref-4"
  xmlns:a1="http://schemas.microsoft.com/clr/nsassem/Microsoft.HtmlTrans/Microsoft.HtmlTrans.In
  terface%2C%20Version%3D12.0.0.0%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3D71e9bce111e9429c
  ">
      <m_ce>CE_NONE</m_ce>
      <m_strMainFileName id="ref-5">output.html</m_strMainFileName>
      <m_rgbMainFile href="#ref-6"/>
    </a1>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```
<m_rgstrSupportingFileNames xsi:null="1"/>
<m_rgrgbSupportingFiles xsi:null="1"/>
<m_strSupportingFolderName id="ref-7"></m_strSupportingFolderName>
<m_strErrorString id="ref-8">1 embedded image(s) from the document were not
preserved in the HTML...</m_strErrorString>
</a1:CreateDCInfo>
<SOAP-ENC:Array id="ref-6" xsi:type="SOAP-ENC:base64">A1B2C3D4</SOAP-ENC:Array>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

This protocol specifies no authentication or authorization mechanisms. A protocol server can perform implementation-specific authorization based only on evidence from underlying transport mechanisms. In operation, transport layer access to the protocol servers can be restricted to known protocol clients, such as other machines in the farm.

5.2 Index of Security Parameters

None.

6 Appendix A: Full Definition of Interfaces and Types

For ease of implementation, the complete definitions of the types and interfaces used by this protocol are as follows, and are specified in the Microsoft.HtmlTrans.Interface assembly.

```
namespace Microsoft.HtmlTrans
{
    interface IHttpTrLauncher
    {
        CreateHtmlInfo CHICreateHtml(String strLauncherUri, Byte[] rgbFile, BrowserType bt,
            String strReqFile, String strTaskName, Int32 timeout, bool fReturnFileBits);
    }
    interface IDocumentConversionsLauncher
    {
        CreateDCInfo ConvertFile(String launcherUri, String appExe, String convertFrom,
            String convertTo,
            Byte[] fileBits, String taskName, String configInfo, Int32 timeout, bool
            fReturnFileBits);
    }
    interface IHttpTrLoadBalancer
    {
        String StrGetLauncher(String strTaskName);
        bool RegisterLauncher(String strLauncherUri);
        bool UnRegisterLauncher(String strLauncherUri);
        bool LauncherTaskCompleted(String strLauncherUri, String strTaskName);
    }

    class CreateHtmlInfo
    {
        CreationErrorType m_ce;
        String m_strMainFileName;
        String m_strMainFilePath;
        Byte [][]m_rgrgbThicketFiles;
        String []m_rgstrThicketFileNames;
        Byte []m_rgbMainFile;
        String m_strThicketFolderName;
    }

    class CreateDCInfo
    {
        CreationErrorType m_ce;
        String m_strMainFileName;
        Byte[] m_rgbMainFile;
        String[] m_rgstrSupportingFileNames;
        Byte[][] m_rgrgbSupportingFiles;
        String m_strSupportingFolderName;
        String m_strErrorString;
    }

    enum BrowserType : Int32 {
        BT_IE3 = 0,
        BT_IE4 = 1,
        BT_UNKNOWN = 2
    }

    enum CreationErrorType : Int32 {
        CE_NONE = 0,
        CE_CRASH = 1,
    }
}
```

```
    CE_HANG = 2,  
    CE_OTHER_BLOCKLIST = 3,  
    CE_SERVERFILENOTFOUND = 4,  
    CE_RESOURCESUNAVAILABLE = 5,  
    CE_SPGETCONTENTFAILED = 6,  
    CE_DISKWRITEERROR = 7,  
    CE_BADINPUT = 8,  
    CE_BACKENDUNAVAILABLE = 9,  
    CE_ALREADYRUNNING = 10,  
    CE_STARTTIMEOUT = 11,  
    CE_OTHER = 12  
}  
}
```

7 Appendix B: The ConfigInfo Parameter for Specific Conversion Applications

If the name of the conversion application executable that is being requested is in the following table, additional context information is passed to the conversion application through the **configInfo** parameter.

Name	Description
Docxpageconverter.exe	The conversion application that converts [ECMA-376] documents to HTML
InfoPathPageConverter.exe	The conversion application that converts [MS-IPFF] documents to HTML
XslApplicatorConverter.exe	The conversion application that converts XML files to HTML by applying a user-provided XSL transformation (see XSLT)

The following sections describe the data which the conversion applications are expected to read.

Vendors can extend the range of conversion applications to which this applies by implementation-specific means.

7.1 Common configInfo parameter structure

The conversion applications in Section [Z](#) expect an XML string that follows this XML schema:

```
<xs:schema
  elementFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
>
  <xs:element name="RcaTransformation">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ConverterSettings">
          <xs:complexType>
            <xs:sequence minOccurs="0" maxOccurs="unbounded">
              <xs:any />
            </xs:sequence>
            <xs:attribute name="SourceDocLibUrl" type="xs:string" use="required" />
            <xs:attribute name="AllocationLimitHint" type="xs:positiveInteger" use="optional" />
          </xs:complexType>
        </xs:element>
        <xs:any />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

7.1.1 RcaTransformation

This is the **root element** of the XML that is described by [\[XML10\]](#). It includes the **ConverterSettings** child element, and can include additional nodes that are implementation-specific to the protocol client. Conversion applications cannot use any nodes in the **RcaTransformation** node other than the **ConverterSettings** node.

7.1.2 ConverterSettings

This node contains settings for the conversion that conversion applications are expected to honor. It can include child elements of type that depend on the conversion applications. It has the following attributes:

SourceDocLibUrl: The **server-relative URL** of the **document library** that contains the file being converted.

AllocationLimitHint: The maximum file size in bytes to which compressed files are allowed to expand when decompressed. This attribute is only present if the document conversion services are configured to grant permission to convert files only up to a certain size. The uncompressed file size depends on the compression, which is specific to the file format. Only the conversion applications enforce the size limit, which they do to guard against attacks through files that expand to many times their original size.

7.2 The configInfo parameter for the docxpageconverter.exe

The docxpageconverter.exe conversion application does not expect child elements to be present in the **ConverterSettings** node. The full XML schema for the **configInfo** parameter is as follows.

```
<xs:schema
  elementFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
>
  <xs:element name="RcaTransformation">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ConverterSettings">
          <xs:complexType>
            <xs:attribute name="SourceDocLibUrl" type="xs:string" use="required" />
            <xs:attribute name="AllocationLimitHint" type="xs:positiveInteger" use="optional"
          />
        </xs:complexType>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

All elements are described in Section [7.1](#).

The docxpageconverter.exe works with an empty **configInfo** parameter.

7.3 The configInfo parameter for the XslApplicatorConverter.exe

The full XML schema for the **configInfo** parameter of the XslApplicatorConverter.exe is as follows.

```
<xs:schema
  elementFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
>
  <xs:element name="RcaTransformation">
    <xs:complexType>
      <xs:sequence>
```

```

<xs:element name="ConverterSettings">
  <xs:complexType>
    <xs:element name="XslApplicatorConverterSettings">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="FilePlaceholder">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:base64Binary">
                  <xs:attribute name="Url" type="xs:anyURI"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name='Version' fixed='1' />
      </xs:complexType>
    </xs:element>
    <xs:attribute name="SourceDocLibUrl" type="xs:string" use="required" />
    <xs:attribute name="AllocationLimitHint" type="xs:positiveInteger"
      use="optional" />
  </xs:complexType>
</xs:element>
<xs:any />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

The **RcaTransformation** and **ConverterSettings** elements are described in Section [7.1](#) and the **ConverterSettings** node contains a child element named **XslApplicatorConverterSettings**, with the following child element and attribute:

FilePlaceholder: The content, encoded with base64 encoding, of the file specified by the value of the attribute **Url**, which is a fully qualified URL of an XSLT file which is capable of converting the XML schema of the file being converted to HTML.

Version: Reserved. Always set to 1.

7.4 The configInfo parameter for the InfoPathPageConverter.exe

The full XML schema of the **configInfo** parameter for the InfoPathPageConverter is as follows.

```

<xs:schema
  elementFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
>
  <xs:element name="RcaTransformation">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ConverterSettings">
          <xs:complexType>
            <xs:element name="InfoPathPageConverterSettings">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="FilePlaceholder">
                    <xs:complexType>

```

```

        <xs:simpleContent>
          <xs:extension base="xs:base64Binary">
            <xs:attribute name="Url" type="xs:anyURI"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="SelectedView">
      <xs:complexType>
        <xs:attribute name="Name" type="xs:string"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name='Version' fixed='1' />
</xs:complexType>
</xs:element>
<xs:attribute name="SourceDocLibUrl" type="xs:string" use="required" />
<xs:attribute name="AllocationLimitHint" type="xs:positiveInteger"
  use="optional" />
</xs:complexType>
</xs:element>
<xs:any />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

The **RcaTransformation** and **ConverterSettings** elements are specified in Section 7.1. The **ConverterSettings** node contains a child element named **InfoPathPageConverterSettings**, which has the following child elements and attributes:

FilePlaceholder: The content, encoded with base64 encoding, of the file specified by the value of the attribute **Url**, which is be a fully qualified URL of a file that follows the format described by [MS-IPFF]. The XML file being converted MUST have been created by filling out the **form (2)** that this file in the [MS-IPFF] format specifies.

SelectedView: The name of the view as specified in [MS-IPFF] which conversion applications are expected to apply.

Version: Reserved. Always set to 1.

7.5 Example Data

7.5.1 Example for the configInfo for the DocxPageConverter.exe

In the following sample, a maximum file size of 1,000,000 bytes is set:

```

<RcaTransformation>
  <ConverterSettings SourceDocLibUrl="/Documents" AllocationLimitHint="1000000">
  </ConverterSettings>
  ***
</RcaTransformation>

```

*** Represents additional nodes that are specific to the implementation of the protocol client, and will be ignored by the conversion applications.

7.5.2 Example for the configInfo for the XslApplicatorConverter.exe

In the following sample, no maximum file size is set.

```
<RcaTransformation>
  <ConverterSettings SourceDocLibUrl="/Documents">
    <XslApplicatorConverterSettings Version="1">
      <FilePlaceholder Url="http://contoso/Documents/Transformation.xsl">A1B2C3D4
    </FilePlaceholder>
    </XslApplicatorConverterSettings>
  </ConverterSettings>
  ***
</RcaTransformation>
```

*** Represents additional nodes that are specific to the implementation of the protocol client, which will be ignored by the conversion applications, and A1B2C3D4 represents the content, encoded with base64 encoding, of the file found at the URL <http://contoso/Documents/Transformation.xsl>.

7.5.3 Example for the configInfo parameter for the InfoPathPageConverter.exe

In the following sample, no maximum file size is set.

```
<RcaTransformation>
  <ConverterSettings SourceDocLibUrl="/Documents">
    <InfoPathPageConverterSettings Version="1">
      <FilePlaceholder Url="http://contoso/Documents/ms-ipff.xsn"> A1B2C3D4
    </FilePlaceholder>
    <SelectedView Name="view1.xsl" />
    </InfoPathPageConverterSettings>
  </ConverterSettings>
  ***
</RcaTransformation>
```

*** Represents additional nodes that are specific to the implementation of the protocol client, which will be ignored by the conversion applications, and A1B2C3D4 stands for the content, encoded with base64 encoding, of the file at the URL <http://contoso/Documents/ms-ipff.xsn>.

8 Appendix C: 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 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.

9 Change Tracking

This section identifies changes that were made to the [MS-DOCTRANS] protocol document between the March 2011 and June 2011 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
3.3.4 Message Processing Events and Sequencing Rules	Clarified behavior of the protocol server when the conversion is complete.	N	Protocol syntax updated.
8 Appendix C: Product Behavior	Updated the list of applicable product versions.	N	Content updated.

10 Index

A

Abstract data model
client ([section 3.2.1](#) 16, [section 3.4.1](#) 18, [section 3.6.1](#) 19)
server ([section 3.1.1](#) 14, [section 3.3.1](#) 16, [section 3.5.1](#) 18)
[Applicability](#) 8

B

[BrowserType data type](#) 12

C

[Capability negotiation](#) 8
[Change tracking](#) 35
Client
abstract data model ([section 3.2.1](#) 16, [section 3.4.1](#) 18, [section 3.6.1](#) 19)
[IDocumentConversionsLauncher](#) 18
[IHtmlTrLauncher](#) 19
[IHtmlTrLoadBalancer](#) 16
initialization ([section 3.2.3](#) 16, [section 3.4.3](#) 18, [section 3.6.3](#) 19)
[local events](#) 20
[overview](#) 14
[timer events](#) 19
timers ([section 3.2.2](#) 16, [section 3.4.2](#) 18, [section 3.6.2](#) 19)
Common data types
[overview](#) 9
[configInfo parameter for specific conversion applications](#) 29
[ConvertFile example](#) 24
[CreateDCInfo data type](#) 10
[CreateHtmlInfo data type](#) 11
[CreationErrorType data type](#) 12

D

Data model - abstract
client ([section 3.2.1](#) 16, [section 3.4.1](#) 18, [section 3.6.1](#) 19)
server ([section 3.1.1](#) 14, [section 3.3.1](#) 16, [section 3.5.1](#) 18)
Data types
[BrowserType](#) 12
[common](#) 9
[CreateDCInfo](#) 10
[CreateHtmlInfo](#) 11
[CreationErrorType](#) 12

E

Events
[local - client](#) 20
[timer - client](#) 19

Examples

[ConvertFile](#) 24
[LauncherTaskCompleted](#) 22
[overview](#) 21
[RegisterLauncher](#) 21
[StrGetLauncher](#) 21
[UnRegisterLauncher](#) 23

F

[Fields - vendor-extensible](#) 8

G

[Glossary](#) 5

I

IDocumentConversionsLauncher ([section 3.3](#) 16, [section 3.4](#) 18)
IHtmlTrLauncher ([section 3.5](#) 18, [section 3.6](#) 19)
IHtmlTrLoadBalancer ([section 3.1](#) 14, [section 3.2](#) 16)
[Implementer - security considerations](#) 26
[Index of security parameters](#) 26
[Informative references](#) 6
Initialization
client ([section 3.2.3](#) 16, [section 3.4.3](#) 18, [section 3.6.3](#) 19)
server ([section 3.1.3](#) 14, [section 3.3.3](#) 16, [section 3.5.3](#) 18)
[Interfaces and types - full definition](#) 27
[Introduction](#) 5

L

[LauncherTaskCompleted example](#) 22
Local events
[client](#) 20

M

Message processing
server ([section 3.1.4](#) 14, [section 3.3.4](#) 17, [section 3.5.4](#) 18)
Messages
[BrowserType data type](#) 12
[common data types](#) 9
[CreateDCInfo data type](#) 10
[CreateHtmlInfo data type](#) 11
[CreationErrorType data type](#) 12
[transport](#) 9

N

[Normative references](#) 5

O

Operations

[CHICreateHtml](#) 18
[ConvertFile](#) 17
[LauncherTaskCompleted](#) 16
[RegisterLauncher](#) 14
[StrGetLauncher](#) 15
[UnRegisterLauncher](#) 15
[Overview \(synopsis\)](#) 6

P

[Parameters - security index](#) 26
[Preconditions](#) 8
[Prerequisites](#) 8
[Product behavior](#) 34

R

References

[informative](#) 6
[normative](#) 5
[RegisterLauncher example](#) 21
[Relationship to other protocols](#) 7

S

Security

[implementer considerations](#) 26
[parameter index](#) 26

Sequencing rules

server ([section 3.1.4](#) 14, [section 3.3.4](#) 17,
[section 3.5.4](#) 18)

Server

abstract data model ([section 3.1.1](#) 14, [section 3.3.1](#) 16, [section 3.5.1](#) 18)
[CHICreateHtml operation](#) 18
[ConvertFile operation](#) 17
[IDocumentConversionsLauncher](#) 16
[IHtmlTrLauncher](#) 18
[IHtmlTrLoadBalancer](#) 14
initialization ([section 3.1.3](#) 14, [section 3.3.3](#) 16,
[section 3.5.3](#) 18)
[LauncherTaskCompleted operation](#) 16
message processing ([section 3.1.4](#) 14, [section 3.3.4](#) 17, [section 3.5.4](#) 18)
[overview](#) 14
[RegisterLauncher operation](#) 14
sequencing rules ([section 3.1.4](#) 14, [section 3.3.4](#) 17, [section 3.5.4](#) 18)
[StrGetLauncher operation](#) 15
timers ([section 3.1.2](#) 14, [section 3.3.2](#) 16,
[section 3.5.2](#) 18)
[UnRegisterLauncher operation](#) 15
[Standards assignments](#) 8
[StrGetLauncher example](#) 21

T

Timer events

[client](#) 19

Timers

client ([section 3.2.2](#) 16, [section 3.4.2](#) 18, [section 3.6.2](#) 19)
server ([section 3.1.2](#) 14, [section 3.3.2](#) 16,
[section 3.5.2](#) 18)
[Tracking changes](#) 35
[Transport](#) 9

U

[UnRegisterLauncher example](#) 23

V

[Vendor-extensible fields](#) 8
[Versioning](#) 8