[MS-CDEPLOY]:

Content Deployment Remote Import Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

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

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
4/25/2008	0.2	Editorial	Revised and edited the technical content
6/27/2008	1.0	Major	Revised and edited the technical content
10/6/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
7/13/2009	1.03	Major	Revised and edited the technical content
8/28/2009	1.04	Editorial	Revised and edited the technical content
11/6/2009	1.05	Editorial	Revised and edited the technical content
2/19/2010	2.0	Minor	Updated the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	3.0	Major	Significantly changed the technical content.
6/10/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.1	Minor	Clarified the meaning of the technical content.
4/11/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	3.1.1	Editorial	Changed language and formatting in the technical content.
2/11/2013	3.1.1	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	3.2	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
11/18/2013	3.2	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	3.2	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	3.2	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	3.2	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	3.2	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	4.0	Major	Significantly changed the technical content.
7/15/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/24/2018	5.0	Major	Significantly changed the technical content.
10/1/2018	6.0	Major	Significantly changed the technical content.
6/18/2019	6.0	None	No changes to the meaning, language, or formatting of the technical content.
7/20/2021	7.0	Major	Significantly changed the technical content.
10/5/2021	8.0	Major	Significantly changed the technical content.
2/15/2022	8.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1   Introduction

The Content Deployment Remote Import Web Service Protocol enables a protocol client to remotely manage the import of a set of content, previously exported from an export server, to a protocol server.

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

1.1   Glossary

This document uses the following terms:

Central Administration site: A SharePoint site that an administrator can use to manage all of the sites and servers in a server farm that is running SharePoint Products and Technologies.

content deployment: The act of exporting content from a source system and importing it to a destination system.

deployment package: A collection of files that represent a serialized snapshot of data. A deployment package is stored as XML files that describe the deployment objects and their relationships, and a binary file for each object. Optionally, the resulting set of files can be compressed into one or more files in the compressed PRIME data format (CMP).

editor: The user who last modified an item or document in a SharePoint list.

event receiver: A structured modular component that enables built-in or user-defined managed code classes to act upon objects, such as list items, lists, or content types, when specific triggering actions occur.

export server: A server that serves as the source of a data export operation.

farm: A group of computers that work together as a single system to help ensure that applications and resources are available. Also referred to as server farm.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

HTTP POST: An HTTP method, as described in [RFC2616].

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

import job: A timer job that is used to import data from a content migration package to a remote server.

operator account: The account of the user who is managing the import process for a deployment package.

site collection: A set of websites that are in the same content database, have the same owner, and share administration settings. A site collection can be identified by a **GUID** or the **URL** of the top-level site for the site collection. Each site collection contains a top-level site, can contain one or more subsites, and can have a shared navigational structure.

site-relative URL: A URL that is relative to the site that contains a resource and does not begin with a leading slash (/).

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

SOAP action: The HTTP request header field used to indicate the intent of the **SOAP** request, using a URI value. See [SOAP1.1] section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a **SOAP message** to its recipient. See [SOAP1.2-1/2007] section 5.3 for more information.

SOAP fault: A container for error and status information within a **SOAP message**. See [SOAP1.2-1/2007] section 5.4 for more information.

SOAP message: An XML document consisting of a mandatory SOAP envelope, an optional SOAP header, and a mandatory **SOAP body**. See [SOAP1.2-1/2007] section 5 for more information.

timer job: A built-in SharePoint object that can perform various tasks within the environment on a scheduled or one-time event basis.

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

web application: A container in a configuration database that stores administrative settings and entry-point **URLs** for **site collections**.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [XML].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition

refs to the fully specified and currently recommended standard for use in authoring **XML schemas**.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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.

[MS-NRTP] Microsoft Corporation, ".NET Remoting: Core Protocol".

[MS-PRIMEPF] Microsoft Corporation, "Deployment Package Format".

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

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

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[SOAP1.2-2/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 2: Adjuncts (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part2-20070427

[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

[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/

[XMLSCHEMA1/2] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

1.2.2 Informative References

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

1.3 Overview

This protocol enables a protocol client to remotely manage the import of a **deployment package** to a protocol server. This protocol consists of nine stateless operations. These operations enable a protocol client to create an **import job**, run the import job, monitor the status of the import job, cancel the import job, and delete the import job or the log files created by the import job on the remote import server. The operations also enable the protocol client to gather information about the remote import server, such as the **URL** of the **Central Administration site**, the names and the URLs of all **web applications**, and the names of all **site collections** within a specified web application.

1.4 Relationship to Other Protocols

This protocol uses the **SOAP** message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2-1/2007] and [SOAP1.2-2/2007]. It transmits those messages by using **HTTP**, as described in [RFC2616], or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818].

The following diagram shows the underlying messaging and transport stack used by the protocol:

Content Deployment Remote Import Web Service	This Protocol
SOAP	Industry Standard
HTTP | HTTPS	
TCP	
IP	
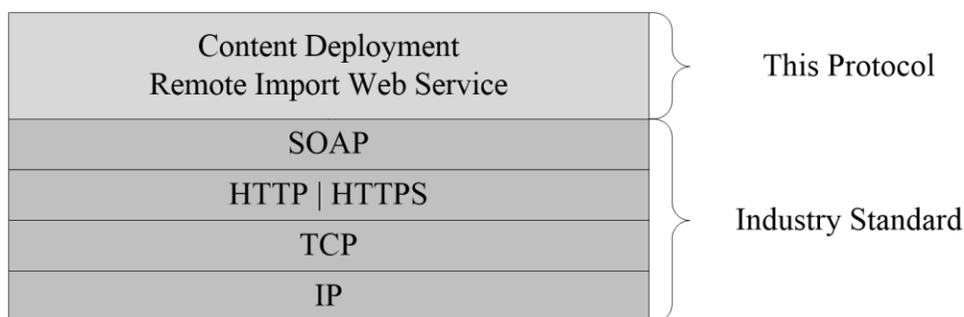
Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against the central administration site of a remote import server with a URL that is known to the protocol client. The protocol server endpoint is formed by appending "_vti_adm/ContentDeploymentRemoteImport.asmx" to the URL of the central administration site of the remote import server. For example, if the URL of the central administration site of the remote import server were http://www.contoso.com:8080, the protocol server endpoint would be http://www.contoso.com:8080/_vti_adm/ContentDeploymentRemoteImport.asmx.

This protocol assumes that authentication has been performed by the underlying protocols.

The remote import server has been configured to accept incoming **content deployment** import jobs.

1.6 Applicability Statement

This protocol is not applicable to the scenario of importing content to a site collection while users are editing content in that site collection. The protocol is also not applicable to the scenario of concurrent imports of multiple **deployment packages** to the same site collection. When used to query for the status of an import job, this protocol is intended for querying no more frequently than once every 10 seconds.

1.7 Versioning and Capability Negotiation

This protocol uses multiple transports with SOAP as specified in section 2.1.

1.8   Vendor-Extensible Fields

None.

1.9   Standards Assignments

None.

2   Messages

2.1   Transport

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

Protocol messages MUST be formatted as specified either in [SOAP1.1] (Section 4, SOAP Envelope) or in [SOAP1.2-1/2007] SOAP Message Construct section 5. Protocol server faults MUST be returned either using HTTP status codes as specified in [RFC2616] Status Code Definitions section 10 or using **SOAP faults** as specified either in [SOAP1.1] SOAP Fault section 4.4 or in [SOAP1.2-1/2007] SOAP Fault section 5.4.

2.2   Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as specified in [XMLSCHEMA1/2] and [XMLSCHEMA2/2], and **WSDL**, as specified in [WSDL].

2.2.1   Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [XMLNS]. Although this specification associates a specific **XML namespace prefix** to each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1/2] [XMLSCHEMA2/2]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2-1/2007] [SOAP1.2-2/2007]
tns	http://schemas.microsoft.com/sharepoint/soap/	
s1	http://microsoft.com/wsdl/types/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
(none)	http://schemas.microsoft.com/sharepoint/soap/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]

2.2.2   Messages

This specification does not define any common **WSDL** message definitions.

2.2.3   Elements

This specification does not define any common **XML schema** element definitions.

2.2.4 Complex Types

This specification does not define any common **XML schema** complex type definitions.

2.2.5 Simple Types

The following table summarizes the set of common **XML schema** simple type definitions defined by this specification. XML schema simple type definitions that are specific to a particular operation are described with the operation.

Simple type	Description
guid	A globally unique identifier (GUID).

2.2.5.1 guid

A **guid** simple type specifies a GUID. Its structure is specified by the following **XML schema definition (XSD)**:

```
<s:simpleType name="guid">
  <s:restriction base="s:string">
    <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}" />
  </s:restriction>
</s:simpleType>
```

2.2.6 Attributes

This specification does not define any common **XML schema** attribute definitions.

2.2.7 Groups

This specification does not define any common **XML schema** group definitions.

2.2.8 Attribute Groups

This specification does not define any common **XML schema** attribute group definitions.

3   Protocol Details

Except where specified, the protocol client SHOULD interpret HTTP status codes returned by the protocol server as specified in [RFC2616] Status Code Definitions section 10.

This protocol allows protocol servers to notify the protocol client of application-level faults using SOAP faults. Except where specified, these SOAP faults are not significant for interoperability, and the protocol client can interpret them in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and notify the protocol client of authorization faults either using HTTP status codes or using SOAP faults as specified previously in this section.

3.1   ContentDeploymentRemoteImportSoap Server Details

All operations consist of a basic **SOAP** request-response pair and the protocol server treats each request as an independent transaction, unrelated to any previous request.

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.

The protocol server MUST maintain a mapping of **GUIDs** to import jobs such that, for any GUID, at most one import job is found.

3.1.2   Timers

None.

3.1.3   Initialization

None.

3.1.4   Message Processing Events and Sequencing Rules

The following table summarizes the list of **WSDL** operations as defined by this specification:

Operations	Description
CancelJob	Cancels the specified import job.
CreateJob	Creates a new import job.
DeleteImportLogs	Deletes log files created by import job(s) associated with the specified **content deployment** job.
DeleteJob	Deletes the specified import job.
GetJobStatus	Returns the status of the specified import job.

Operations	Description
GetRemoteAdminServerUrl	Returns the URL of the central administration site on the remote import server.
GetSiteCollectionNames	Returns the names of all site collections within the specified **web application**.
GetVirtualServersInformation	Returns the names and URLs of all web applications within the current **farm**.
RunJob	Runs the specified import job.

3.1.4.1 CancelJob

This operation cancels the specified import job, as follows.

```
<wsdl:operation name="CancelJob">
    <wsdl:input message="tns:CancelJobSoapIn" />
    <wsdl:output message="tns:CancelJobSoapOut" />
</wsdl:operation>
```

A protocol client initiates the operation by sending a **CancelJobSoapIn** message to the protocol server, formatted as specified in section 3.1.4.1.1.1.

If the status of the specified import job is ImportPreparing, as specified in section 3.1.4.5.4.1, the protocol server MUST cancel the import job. Otherwise, the protocol server MUST take no action.

The protocol server MUST respond with a **CancelJobSoapOut** response message, formatted as specified in section 3.1.4.1.1.2.

3.1.4.1.1 Messages

The following table summarizes the set of **WSDL** message definitions that are specific to this operation.

Message	Description
CancelJobSoapIn	A request to initiate a **CancelJob** operation on the protocol server.
CancelJobSoapOut	A response from the protocol server at completion of the **CancelJob** operation.

3.1.4.1.1.1   CancelJobSoapIn

This message is sent from the protocol client to the protocol server to initiate a **CancelJob** operation.

The **SOAP action** value of the message is specified as follows:

```
http://schemas.microsoft.com/sharepoint/soap/CancelJob
```

The **SOAP body** MUST contain a **CancelJob** element.

3.1.4.1.1.2   CancelJobSoapOut

This message is sent from the protocol server to the protocol client when the **CancelJob** operation is finished.

The SOAP body MUST contain a **CancelJobResponse** element.

3.1.4.1.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
CancelJob	Specifies the body of the **CancelJobSoapIn** message.
CancelJobResponse	Specifies the body of the **CancelJobSoapOut** message.

3.1.4.1.2.1   CancelJob

This element forms the body of the **CancelJobSoapIn** message. It MUST adhere to the following XSD:

```
<s:element name="CancelJob">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="jobId" type="s1:guid" />
    </s:sequence>
  </s:complexType>
</s:element>
```

jobId: The identifier of the import job.

3.1.4.1.2.2   CancelJobResponse

This element forms the body of the **CancelJobSoapOut** message. It MUST adhere to the following XSD:

```
<s:element name="CancelJobResponse">
  <s:complexType />
</s:element>
```

3.1.4.1.3 Complex Types

None.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 CreateJob

This operation creates a new import job, as follows.

```
<wsdl:operation name="CreateJob">
    <wsdl:input message="tns:CreateJobSoapIn" />
    <wsdl:output message="tns:CreateJobSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **CreateJobSoapIn** request message to the server, formatted as specified in section 3.1.4.2.1.1.

If an import job already exists with the specified **sourceId**, the protocol server MUST delete the existing import job. The protocol server SHOULD<1> create a new import job with the specified parameter values.

When the operation is finished, the protocol server MUST respond with a **CreateJobSoapOut** response message, formatted as specified in section 3.1.4.2.1.2.

3.1.4.2.1 Messages

The following table summarizes the set of **WSDL** message definitions that are specific to this operation.

Message	Description
CreateJobSoapIn	A request to initiate a **CreateJob** operation on the protocol server.
CreateJobSoapOut	A response from the protocol server at completion of the **CreateJob** operation.

3.1.4.2.1.1   CreateJobSoapIn

This message is sent from the protocol client to the protocol server to initiate a **CreateJob** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/CreateJob
```

The SOAP body MUST contain a **CreateJob** element.

3.1.4.2.1.2   CreateJobSoapOut

This message is sent from the protocol server to the protocol client when the **CreateJob** operation is finished.

The SOAP body MUST contain a **CreateJobResponse** element.

3.1.4.2.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
CreateJob	Specifies the body of the **CreateJobSoapIn** message.
CreateJobResponse	Specifies the body of the **CreateJobSoapOut** message.

3.1.4.2.2.1 CreateJob

This element forms the body of the **CreateJobSoapIn** message. It MUST adhere to the following XSD<2>:

```
<s:element name="CreateJob">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="description" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="sourceId" type="s1:guid" />
      <s:element minOccurs="1" maxOccurs="1" name="destinationServerUrl" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="destinationSiteCollection" type="s:string"
/>
      <s:element minOccurs="1" maxOccurs="1" name="includeSecurity"
type="tns:SPIncludeSecurity" />
      <s:element minOccurs="1" maxOccurs="1" name="includeUserInfoDateTime"
type="tns:SPImportUserInfoDateTimeOption" />
      <s:element minOccurs="1" maxOccurs="1" name="enableEventReceivers" type="s:boolean" />
      <s:element minOccurs="1" maxOccurs="1" name="fileCompression" type="s:boolean" />
    </s:sequence>
  </s:complexType>
</s:element>
```

description: The description of the new import job**.** It MUST be less than 256 characters in length.

sourceId: The identifier of the content deployment job that associates it with the import job that is to be created.

destinationServerUrl: The URL of the remote import server for the new import job. It MUST be less than 255 characters in length and MUST be the URL of an existing **web application**.

destinationSiteCollection: The **site-relative URL** of the destination site collection for the new import job. It MUST be less than 256 characters in length and MUST be the URL of an existing site collection within the web application specified by **destinationServerUrl**.

includeSecurity: The user and group information to import. This MUST be set to the same value as was specified for **IncludeSecurity** in the ExportSettings.xml file (specified in [MS-PRIMEPF] section 2.2) when the **deployment package** was created.

includeUserInfoDateTime: The level of user and creation date information to retain during the import. The default value of this element is None.

enableEventReceivers: Whether **event receivers** are notified while content is being imported.

fileCompression: Whether the deployment package is in a compressed format.

3.1.4.2.2.2   CreateJobResponse

This element forms the body of the **CreateJobSoapOut** message. It MUST adhere to the following XSD:

```
<s:element name="CreateJobResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="CreateJobResult" type="s1:guid" />
    </s:sequence>
  </s:complexType>
</s:element>
```

CreateJobResult: The identifier of the newly created import job.

3.1.4.2.3 Complex Types

None.

3.1.4.2.4 Simple Types

The following table summarizes the **XML schema** simple type definitions that are specific to this operation.

Simple type	Description
SPIncludeSecurity	Specifies user and group information included for imported content.
SPImportUserInfoDateTimeOption	Specifies level of user and creation date retained for imported content.

3.1.4.2.4.1   SPIncludeSecurity

The **SPIncludeSecurity** simple type specifies the user and group information to include when the content is imported. It is specified in [MS-PRIMEPF] section 2.2.4.4.

3.1.4.2.4.2   SPImportUserInfoDateTimeOption

The **SPImportUserInfoDateTimeOption** simple type specifies the level of user and creation date information to retain when the content is imported. It is specified by the following XSD:

```
<s:simpleType name="SPImportUserInfoDateTimeOption">
  <s:restriction base="s:string">
    <s:enumeration value="None"/>
    <s:enumeration value="ReplaceUserWithSystemAccount"/>
    <s:enumeration value="ImportAll"/>
  </s:restriction>
</s:simpleType>
```

The following table defines the valid values.

Value	Meaning
None	Sets the creation date/time values for all content to the date/time of the import, replaces all authors and **editors** with the **operator account**, and

Value	Meaning
	replaces all references to users with the empty string.
ReplaceUserWithSystemAccount	Retains the original creation date/time values for all content, but replaces all authors and editors with the system account, and replaces all references to users with the empty string.
ImportAll	Retains all original creation date/time values, all original author and editor information, and all references to users.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 DeleteImportLogs

This operation deletes log files created by import job(s) associated with the specified content deployment job, as follows.<3>

```
<wsdl:operation name="DeleteImportLogs">
  <wsdl:input message="tns:DeleteImportLogsSoapIn" />
  <wsdl:output message="tns:DeleteImportLogsSoapOut" />
</wsdl:operation>
```

A protocol client initiates the operation by sending a **DeleteImportLogsSoapIn** message to the protocol server, formatted as specified in section 3.1.4.3.1.1.

The protocol server MUST delete any existing log files created by import job(s) associated with the specified content deployment job. If no such log files exist, the protocol server MUST take no action.

When the operation is finished, the protocol server MUST respond with a **DeleteImportLogsSoapOut** response message, formatted as specified in section 3.1.4.3.1.2.

3.1.4.3.1 Messages

The following table summarizes the set of **WSDL** message definitions that are specific to this operation.

Message	Description
DeleteImportLogsSoapIn	A request to initiate a **DeleteImportLogs** operation on the protocol server.

Message	Description
DeleteImportLogsSoapOut	A response from the protocol server at completion of the **DeleteImportLogs** operation.

3.1.4.3.1.1   DeleteImportLogsSoapIn

This message is sent from the protocol client to the protocol server to initiate a **DeleteImportLogs** operation.

The **SOAP action** value of the message is specified as follows:

```
http://schemas.microsoft.com/sharepoint/soap/DeleteImportLogs
```

The **SOAP body** MUST contain a **DeleteImportLogs** element.

3.1.4.3.1.2   DeleteImportLogsSoapOut

This message is sent from the protocol server to the protocol client when the **DeleteImportLogs** operation is finished.

The SOAP body MUST contain a **DeleteImportLogsResponse** element.

3.1.4.3.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
DeleteImportLogs	Specifies the body of the **DeleteImportLogsSoapIn** message.
DeleteImportLogsReponse	Specifies the body of the **DeleteImportLogsSoapOut** message.

3.1.4.3.2.1   DeleteImportLogs

This element forms the body of the **DeleteImportLogsSoapIn** message. It MUST adhere to the following XSD:

```
<s:element name="DeleteImportLogs">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="sourceJobId" type="s1:guid" />
    </s:sequence>
  </s:complexType>
</s:element>
```

sourceJobId: The identifier of the content deployment job.

3.1.4.3.2.2   DeleteImportLogsResponse

This element forms the body of the **DeleteImportLogsResponse** message. It MUST adhere to the following XSD:

```
<s:element name="DeleteImportLogsResponse">
  <s:complexType />
</s:element>
```

3.1.4.3.3 Complex Types

None.

3.1.4.3.4 Simple Types

None.

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.4.4 DeleteJob

This operation deletes the specified import job, as follows.

```
<wsdl:operation name="DeleteJob">
    <wsdl:input message="tns:DeleteJobSoapIn" />
    <wsdl:output message="tns:DeleteJobSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **DeleteJobSoapIn** request message to the protocol server, formatted as specified in section 3.1.4.4.1.1.

If the specified import job exists the protocol server MUST delete the import job. Otherwise, the protocol server MUST take no action.

When the operation is finished, the protocol server MUST respond with a **DeleteJobSoapOut** response message, formatted as specified in section 3.1.4.4.1.2.

3.1.4.4.1 Messages

The following table summarizes the set of **WSDL** message definitions that are specific to this operation.

Message	Description
DeleteJobSoapIn	A request to initiate a **DeleteJob** operation on the protocol server.
DeleteJobSoapOut	A response from the protocol server at completion of the **DeleteJob** operation.

3.1.4.4.1.1 DeleteJobSoapIn

This message is sent from the protocol client to the protocol server to initiate a **DeleteJob** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/DeleteJob
```

The SOAP body MUST contain a **DeleteJob** element.

3.1.4.4.1.2 DeleteJobSoapOut

This message is sent from the protocol server to the protocol client when the **DeleteJob** operation is finished.

The SOAP body MUST contain a **DeleteJobResponse** element.

3.1.4.4.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
DeleteJob	Specifies the body of the **DeleteJobSoapIn** message.
DeleteJobResponse	Specifies the body of the **DeleteJobSoapOut** message.

3.1.4.4.2.1 DeleteJob

This element forms the body of the **DeleteJobSoapIn** message. It MUST adhere to the following XSD:

```
<s:element name="DeleteJob">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="jobId" type="s1:guid" />
    </s:sequence>
  </s:complexType>
</s:element>
```

jobId: The identifier of the import job.

3.1.4.4.2.2 DeleteJobResponse

This element forms the body of the **DeleteJobSoapOut** message. It MUST adhere to the following XSD:

```
<s:element name="DeleteJobResponse">
  <s:complexType />
</s:element>
```

3.1.4.4.3 Complex Types

None.

3.1.4.4.4 Simple Types

None.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.4.5 GetJobStatus

This operation returns the status and properties of the specified import job, as follows.

```
<wsdl:operation name="GetJobStatus">
    <wsdl:input message="tns:GetJobStatusSoapIn" />
    <wsdl:output message="tns:GetJobStatusSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetJobStatusSoapIn** request message to the protocol server, formatted as specified in section 3.1.4.5.1.1.

If the **GetJobStatus** operation is called for a specified import job before the **RunJob** operation is called for that import job, the protocol server MUST respond with a SOAP fault. Otherwise, the protocol server MUST respond with a **GetJobStatusSoapOut** response message, formatted as specified in section 3.1.4.5.1.2.

3.1.4.5.1 Messages

The following table summarizes the set of **WSDL** message definitions that are specific to this operation.

Message	Description
GetJobStatusSoapIn	A request to initiate a **GetJobStatus** operation on the protocol server.
GetJobStatusSoapOut	A response from the protocol server at completion of the **GetJobStatus** operation.

3.1.4.5.1.1   GetJobStatusSoapIn

This message is sent from the protocol client to the protocol server to initiate a **GetJobStatus** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/GetJobStatus
```

The SOAP body MUST contain a **GetJobStatus** element.

3.1.4.5.1.2   GetJobStatusSoapOut

This message is sent from the protocol server to the protocol client when the **GetJobStatus** operation is finished.

The SOAP body MUST contain a **GetJobStatusResponse** element.

3.1.4.5.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
GetJobStatus	Specifies the body of the **GetJobStatusSoapIn** message.
GetJobStatusResponse	Specifies the body of the **GetJobStatusSoapOut** message.

3.1.4.5.2.1   GetJobStatus

This element forms the body of the **GetJobStatusSoapIn** message. It MUST adhere to the following XSD:

```
<s:element name="GetJobStatus">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="jobId" type="s1:guid" />
    </s:sequence>
  </s:complexType>
</s:element>
```

jobId: The identifier of the import job.

3.1.4.5.2.2   GetJobStatusResponse

This element forms the body of the **GetJobStatusSoapOut** message. It MUST adhere to the following XSD:

```
<s:element name="GetJobStatusResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetJobStatusResult"
type="tns:ContentDeploymentRemoteJobStatus" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetJobStatusResult: The status and properties of the specified import job.

3.1.4.5.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type	Description
ContentDeploymentRemoteJobStatus	Specifies the status and properties of an import job.
ArrayOfReportMessage	Specifies a sequence of **ReportMessages.**
ReportMessage	Specifies a single error, warning, or informational message

3.1.4.5.3.1   ContentDeploymentRemoteJobStatus

A **ContentDeploymentRemoteJobStatus** object specifies the status and properties of an import job. Its structure is specified by the following XSD:<u><4></u>

```
<s:complexType name="ContentDeploymentRemoteJobStatus">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="Status"
type="tns:ContentDeploymentJobStatus" />
    <s:element minOccurs="1" maxOccurs="1" name="StatusMessage" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="LastModifiedTime" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="StartTime" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="EndTime" type="s:dateTime" />
    <s:element minOccurs="1" maxOccurs="1" name="ImportedObjectCount" type="s:int" />
    <s:element minOccurs="1" maxOccurs="1" name="ContentSize" type="s:long" />
    <s:element minOccurs="1" maxOccurs="1" name="ImportLogFileMachineName" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ImportLogFilePath" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="ReportMessages"
type="tns:ArrayOfReportMessage" />
  </s:sequence>
</s:complexType>
```

Status: The status of the import job.

StatusMessage: The status message of the import job.

LastModifiedTime: The date and time at which the import job was last modified.

StartTime: The date and time at which the import job started.

EndTime: The date and time at which the import job ended.

ImportedObjectCount: The current count of imported objects. This parameter MUST be a non-negative value.

ContentSize: This parameter MUST be 0.

ImportLogFileMachineName: If not an empty string, the name of the computer containing the import job log file. If empty, the protocol server did not have this information available at the time the message was sent.

ImportLogFilePath: If not an empty string, the absolute path to the import job log file. If empty, the protocol server did not have this information available at the time the message was sent.

ReportMessages: The sequence of error, warning, and informational messages generated by the import job during its run.

3.1.4.5.3.2   ArrayOfReportMessage

An **ArrayOfReportMessage** is a sequence of **ReportMessages** and is specified by the following XSD:

```
<s:complexType name="ArrayOfReportMessage">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="ReportMessage" nillable="true"
type="tns:ReportMessage" />
  </s:sequence>
</s:complexType>
```

ReportMessage: A single error, warning, or informational message.

3.1.4.5.3.3   ReportMessage

The **ReportMessage** type specifies a single error, warning, or informational message. Its structure is specified by the following XSD:

```
<s:complexType name="ReportMessage">
  <s:attribute name="Title" type="s:string"/>
  <s:attribute name="Time" type="s:dateTime" use="required"/>
  <s:attribute name="Severity" type="tns:MessageSeverity" use="required"/>
  <s:attribute name="Description" type="s:string"/>
  <s:attribute name="Recommendation" type="s:string"/>
  <s:attribute name="ObjectName" type="s:string"/>
  <s:attribute name="Phase" type="s:string"/>
</s:complexType>
```

Title: The title of the message.

Time: The date and time at which the message was generated.

Severity: The severity of the message as specified in section 3.1.4.5.4.2.

Description: The text of the message.

Recommendation: The recommended action in response to the message.

ObjectName: The URL or name of the object to which the message applies.

Phase: The phase of the import job when the message was generated.

3.1.4.5.4 Simple Types

The following table summarizes the **XML schema** simple type definitions that are specific to this operation.

Simple type	Description
ContentDeploymentJobStatus	Specifies the status of an import job.
MessageSeverity	Specifies the severity level of a message.

3.1.4.5.4.1 ContentDeploymentJobStatus

A **ContentDeploymentJobStatus** object specifies the status of an import job. Its structure is specified by the following XSD:

```
<s:simpleType name="ContentDeploymentJobStatus">
  <s:restriction base="s:string">
    <s:enumeration value="Success"/>
    <s:enumeration value="Failure"/>
    <s:enumeration value="Cancelled"/>
    <s:enumeration value="ImportInProgress"/>
    <s:enumeration value="CancelInProgress"/>
    <s:enumeration value="ImportPreparing"/>
    <s:enumeration value="ImportTimedOut"/>
  </s:restriction>
</s:simpleType>
```

The following table specifies possible values.

Value	Meaning
Success	Indicates that the import job succeeded.
Failure	Indicates that the import job failed.
Cancelled	Indicates that the import job has been cancelled.
ImportInProgress	Indicates that the import job is currently in progress.
CancelInProgress	Indicates that a request to cancel the import job is currently being processed.
ImportPreparing	Indicates that the import job is preparing to run.
ImportTimedOut	Indicates that the import job has timed out.

3.1.4.5.4.2 MessageSeverity

A **MessageSeverity** object specifies the severity level of a message. Its structure is specified by the following XSD:

```
<s:simpleType name="MessageSeverity">
  <s:restriction base="s:string">
    <s:enumeration value="Informational"/>
    <s:enumeration value="Warning"/>
    <s:enumeration value="Error"/>
  </s:restriction>
</s:simpleType>
```

The following table specifies possible values.

Value	Meaning
Informational	The message is informational.
Warning	The message is a warning. A Warning is more severe than an Informational message.

Value	Meaning
Error	The message describes an error. An Error is more severe than a Warning.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.4.6 GetRemoteAdminServerUrl

This operation returns the URL of the central administration site on the remote import server as follows.

```
<wsdl:operation name="GetRemoteAdminServerUrl">
    <wsdl:input message="tns:GetRemoteAdminServerUrlSoapIn" />
    <wsdl:output message="tns:GetRemoteAdminServerUrlSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetRemoteAdminServerUrlSoapIn** request message to the protocol server, formatted as specified in section 3.1.4.6.1.1.

The protocol server MUST respond with a **GetRemoteAdminServerUrlSoapOut** response message, formatted as specified in section 3.1.4.6.1.2.

3.1.4.6.1 Messages

The following table summarizes the set of **WSDL** message definitions that are specific to this operation.

Message	Description
GetRemoteAdminServerUrlSoapIn	A request to initiate a **GetRemoteAdminServerUrl** operation on the protocol server.
GetRemoteAdminServerUrlSoapOut	A response from the protocol server at completion of the **GetRemoteAdminServerUrl** operation.

3.1.4.6.1.1 GetRemoteAdminServerUrlSoapIn

This message is sent from the protocol client to the protocol server to initiate a **GetRemoteAdminServerUrl** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/GetRemoteAdminServerUrl
```

The SOAP body MUST contain a **GetRemoteAdminServerUrl** element.

3.1.4.6.1.2   GetRemoteAdminServerUrlSoapOut

This message is sent from the protocol server to the protocol client when the
GetRemoteAdminServerUrl operation is finished.

The SOAP body MUST contain a **GetRemoteAdminServerUrlResponse** element.

3.1.4.6.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this
operation.

Element	Description
GetRemoteAdminServerUrl	A request to initiate a **GetRemoteAdminServerUrlSoapIn** operation on the protocol server.
GetRemoteAdminServerUrlResponse	A response from the protocol server at completion of the **GetRemoteAdminServerUrlSoapOut** operation.

3.1.4.6.2.1   GetRemoteAdminServerUrl

This element forms the body of the **GetRemoteAdminServerUrlSoapIn** message. It MUST adhere to
the following XSD:

```
<s:element name="GetRemoteAdminServerUrl">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="clientVersion" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

clientVersion: Version of the library which implements the protocol client. The value MUST be
formatted as specified by **VersionValue** in [MS-NRTP] section 2.2.1.3. This element MUST be ignored
by the protocol server.

3.1.4.6.2.2   GetRemoteAdminServerUrlResponse

This element forms the body of the **GetRemoteAdminServerUrlSoapOut** message. It MUST adhere
to the following XSD:

```
<s:element name="GetRemoteAdminServerUrlResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetRemoteAdminServerUrlResult"
type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetRemoteAdminServerUrlResult: The URL of the central administration site on the remote import server.

3.1.4.6.3 Complex Types

None.

3.1.4.6.4 Simple Types

None.

3.1.4.6.5 Attributes

None.

3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.4.7 GetSiteCollectionNames

This operation returns the names of all site collections within the specified **web application** as follows.

```
<wsdl:operation name="GetSiteCollectionNames">
    <wsdl:input message="tns:GetSiteCollectionNamesSoapIn" />
    <wsdl:output message="tns:GetSiteCollectionNamesSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetSiteCollectionNamesSoapIn** request message to the protocol server, formatted as specified in section 3.1.4.7.1.1.

The protocol server MUST respond with a **GetSiteCollectionNamesSoapOut** response message, formatted as specified in section 3.1.4.7.1.2.

3.1.4.7.1 Messages

The following table summarizes the set of **WSDL** message definitions that are specific to this operation.

Message	Description
GetSiteCollectionNamesSoapIn	A request to initiate a **GetSiteCollectionNames** operation on the protocol server.
GetSiteCollectionNamesSoapOut	A response from the protocol server at completion of the **GetSiteCollectionNames** operation.

3.1.4.7.1.1 GetSiteCollectionNamesSoapIn

This message is sent from the protocol client to the protocol server to initiate a **GetSiteCollectionNames** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/GetSiteCollectionNames
```

The SOAP body MUST contain a **GetSiteCollectionNames** element.

3.1.4.7.1.2 GetSiteCollectionNamesSoapOut

This message is sent from the protocol server to the protocol client when the **GetSiteCollectionNames** operation is finished.

The SOAP body MUST contain a **GetSiteCollectionNamesResponse** element.

3.1.4.7.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
GetSiteCollectionNames	Specifies the body of the **GetSiteCollectionNamesSoapIn** message.
GetSiteCollectionNamesResponse	Specifies the body of the **GetSiteCollectionNamesSoapOut** message.

3.1.4.7.2.1 GetSiteCollectionNames

This element forms the body of the **GetSiteCollectionNamesSoapIn** message. It MUST adhere to the following XSD:

```
<s:element name="GetSiteCollectionNames">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="virtualServerUrl" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

virtualServerUrl: The URL of an existing **web application** on the farm.

3.1.4.7.2.2 GetSiteCollectionNamesResponse

This element forms the body of the **GetSiteCollectionNamesSoapOut** message. It MUST adhere to the following XSD:

```
<s:element name="GetSiteCollectionNamesResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetSiteCollectionNamesResult"
type="tns:ArrayOfString" />
    </s:sequence>
  </s:complexType>
```

```
    </s:element>
```

GetSiteCollectionNamesResult: The names of all site collections within the specified **web application**.

3.1.4.7.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type	Description
ArrayOfString	Specifies a sequence of strings.

3.1.4.7.3.1   ArrayOfString

A sequence of strings as specified by the following XSD:

```
<s:complexType name="ArrayOfString">
  <s:sequence>
    <s:element name="string" type="s:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

string: A single string within the **ArrayOfString**.

3.1.4.7.4 Simple Types

None.

3.1.4.7.5 Attributes

None.

3.1.4.7.6 Groups

None.

3.1.4.7.7 Attribute Groups

None.

3.1.4.8 GetVirtualServersInformation

This operation returns the names and URLs of all **web applications** within the farm, as follows.

```
<wsdl:operation name="GetVirtualServersInformation">
    <wsdl:input message="tns:GetVirtualServersInformationSoapIn" />
    <wsdl:output message="tns:GetVirtualServersInformationSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetVirtualServersInformationSoapIn** request message to the protocol server, formatted as specified in section 3.1.4.8.1.1.

The protocol server MUST respond with a **GetVirtualServersInformationSoapOut** response message, formatted as specified in section 3.1.4.8.1.2.

3.1.4.8.1 Messages

The following table summarizes the set of **WSDL** message definitions that are specific to this operation.

Message	Description
GetVirtualServersInformationSoapIn	A request to initiate a **GetVirtualServersInformation** operation on the protocol server.
GetVirtualServersInformationSoapOut	A response from the protocol server at completion of the **GetVirtualServersInformation** operation.

3.1.4.8.1.1 GetVirtualServersInformationSoapIn

This message is sent from the protocol client to the protocol server to initiate a **GetVirtualServersInformation** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/GetVirtualServersInformation
```

The SOAP body MUST contain a **GetVirtualServersInformation** element.

3.1.4.8.1.2 GetVirtualServersInformationSoapOut

This message is sent from the protocol server to the protocol client when the **GetVirtualServersInformation** operation is finished.

The SOAP body MUST contain a **GetVirtualServersInformationResponse** element.

3.1.4.8.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
GetVirtualServersInformation	Specifies the body of the **GetVirtualServersInformationSoapIn** message.
GetVirtualServersInformationResponse	Specifies the body of the **GetVirtualServersInformationSoapOut** message.

3.1.4.8.2.1 GetVirtualServersInformation

This element forms the body of the **GetVirtualServersInformationSoapIn** message. It MUST adhere to the following XSD:

```
<s:element name="GetVirtualServersInformation">
  <s:complexType/>
</s:element>
```

3.1.4.8.2.2   GetVirtualServersInformationResponse

This element forms the body of the **GetVirtualServersInformationSoapOut** message. It MUST
adhere to the following XSD:

```
<s:element name="GetVirtualServersInformationResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetVirtualServersInformationResult"
type="tns:ArrayOfVirtualServerInformation" />
    </s:sequence>
  </s:complexType>
</s:element>
```

GetVirtualServersInformationResult: The names and URLs of all **web applications** within the
farm.

3.1.4.8.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this
operation.

Complex type	Description
ArrayOfVirtualServerInformation	Specifies a sequence of **VirtualServerInformation** elements.
VirtualServerInformation	Specifies the name and URL of a **web application**.

3.1.4.8.3.1   ArrayOfVirtualServerInformation

A sequence of **VirtualServerInformation** elements as specified by the following XSD:

```
<s:complexType name="ArrayOfVirtualServerInformation">
  <s:sequence>
    <s:element name="VirtualServerInformation" type="tns:VirtualServerInformation"
nillable="true" minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

VirtualServerInformation: The name and URL of a single **web application** within the farm.

3.1.4.8.3.2   VirtualServerInformation

The name and URL of a **web application** as specified by the following XSD:

```
<s:complexType name="VirtualServerInformation">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="VirtualServerName" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="VirtualServerUrl" type="s:string" />
  </s:sequence>
```

```
        </s:complexType>
```

VirtualServerName: The name of the web application.

VirtualServerUrl: The URL of the web application.

3.1.4.8.4 Simple Types

None.

3.1.4.8.5 Attributes

None.

3.1.4.8.6 Groups

None.

3.1.4.8.7 Attribute Groups

None.

3.1.4.9 RunJob

This operation runs the specified import job, as follows.

```
<wsdl:operation name="RunJob">
    <wsdl:input message="tns:RunJobSoapIn" />
    <wsdl:output message="tns:RunJobSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **RunJobSoapIn** request message to the protocol server, formatted as specified in section 3.1.4.9.1.1.

If the **jobId** parameter of the **RunJobSoapIn** message identifies an existing import job that is not in any of the following states—**ImportPreparing**, **ImportInProgress**, **ImportTimedOut**, or **CancelInProgress**—the protocol server MUST execute the import job. Otherwise, the protocol server MUST take no action.

When the operation is finished, the protocol server MUST respond with a **RunJobSoapOut** response message, formatted as specified in section 3.1.4.9.1.2.

3.1.4.9.1 Messages

The following table summarizes the set of **WSDL** message definitions that are specific to this operation.

Message	Description
RunJobSoapIn	A request to initiate a **RunJob** operation on the protocol server.
RunJobSoapOut	A response from the protocol server at completion of the **RunJob** operation.

3.1.4.9.1.1 RunJobSoapIn

This message is sent from the protocol client to the protocol server to initiate a **RunJob** operation.

The SOAP action value of the message is defined as follows:

```
http://schemas.microsoft.com/sharepoint/soap/RunJob
```

The SOAP body MUST contain a **RunJob** element.

3.1.4.9.1.2 RunJobSoapOut

This message is sent from the protocol server to the protocol client when the **RunJob** operation is finished.

The SOAP body MUST contain a **RunJobResponse** element.

3.1.4.9.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.

Element	Description
RunJob	Specifies the body of the **RunJobSoapIn** message.
RunJobResponse	Specifies the body of the **RunJobSoapOut** message.

3.1.4.9.2.1 RunJob

This element forms the body of the **RunJobSoapIn** message. It MUST adhere to the following XSD:

```
<s:element name="RunJob">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="jobId" type="s1:guid" />
    </s:sequence>
  </s:complexType>
</s:element>
```

jobId: The identifier of the import job.

3.1.4.9.2.2 RunJobResponse

This element forms the body of the **RunJobSoapOut** message. It MUST adhere to the following XSD:

```
<s:element name="RunJobResponse">
  <s:complexType/>
</s:element>
```

3.1.4.9.3 Complex Types

None.

3.1.4.9.4 Simple Types

None.

3.1.4.9.5 Attributes

None.

3.1.4.9.6 Groups

None.

3.1.4.9.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 ContentDeploymentRemoteImportSoap Client Details

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing Rules

To perform a successful remote import operation, the protocol client MUST have farm administrator rights on the remote import server and MUST perform the following operations in order:

1. Export content from the **export server** and store it in a local **deployment package**.

    The deployment package<5> MUST adhere to the format specified in [MS-PRIMEPF].

    The names of the deployment package (1) files MUST be as specified in [MS-PRIMEPF] section 2. For a compressed deployment package (1) (consisting of one or more cabinet files) names of the cabinet files MUST be **BaseFileName**.cab for the first file and **BaseFileName**n.cab for subsequent files, where **BaseFileName** equals ExportedFiles<6>, and n is an integer starting with 1 and incrementing monotonically.

    The deployment package MUST contain the file ExportSettings.xml specified in [MS-PRIMEPF] section 2.2 with the following settings:

IncludeSecurity: MUST be set to one of the values specified in [MS-PRIMEPF] section 2.2.4.4.

IncludeVersions: MUST be set to **SPIncludeVersions.LastMajorAndMinor** as specified in [MS-PRIMEPF] section 2.2.4.5.

ExcludeDependencies: MUST be set to false.

The mechanism by which the protocol client performs this export operation is outside the scope of this protocol.

1. Call the **CreateJob** operation to create an import job.

   The protocol client MUST save the identifier of the newly created import job that is returned by the protocol server. This identifier will be needed in step 4.

2. Call the **GetRemoteAdminServerUrl** operation to retrieve the base URL for uploading the deployment package.

3. Use the **HTTP POST** operation to upload the deployment package to the following URL:

   AdminServerUrl/_admin/Content
   Deployment/DeploymentUpload.aspx?filename=*FileName*&remoteJobId=*JobId*

   where *AdminServerUrl* MUST be the URL retrieved in step 3, *FileName* MUST be the name of the file being uploaded, and *JobId* MUST be the identifier of the import job returned by the call to **CreateJob** in step 2.

   If there are multiple files in the deployment package, the protocol client MUST use the HTTP POST operation repeatedly to upload each file.

4. Call the **RunJob** operation to import the deployment package.

5. Optionally, call the **GetJobStatus** operation to retrieve the status of the import job.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

4 Protocol Examples

4.1 GetRemoteAdminServerUrl

If a protocol client wants to retrieve the URL of the central administration site of the remote import server, the protocol client sends a **SOAP message** such as the following:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetRemoteAdminServerUrl
      xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <clientVersion>
        12.0.6211.1000
      </clientVersion>
    </GetRemoteAdminServerUrl>
  </soap:Body>
</soap:Envelope>
```

The protocol server responds with a SOAP message such as the following:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetRemoteAdminServerUrlResponse
     xmlns="http://schemas.microsoft.com/sharepoint/soap/">
     <GetRemoteAdminServerUrlResult>
        http://www.contoso.com:8000
      </GetRemoteAdminServerUrlResult>
    </GetRemoteAdminServerUrlResponse>
  </soap:Body>
</soap:Envelope>
```

4.2 CreateJob

If a protocol client wants to create a new import job, the protocol client sends a SOAP message such as the following:

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateJob
      xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <description>
        Update Product Sites
      </description>
      <sourceId>
        6faeca33-f332-488b-93b8-2ecfb858ca4e
      </sourceId>
      <destinationServerUrl>
        http://contoso.com:90
      </destinationServerUrl>
```

```
<destinationSiteCollection>
  /Product Sites
</destinationSiteCollection>
<includeSecurity>
  All
</includeSecurity>
<includeUserInfoDateTime>
  ImportAll
</includeUserInfoDateTime>
<enableEventReceivers>
  true
</enableEventReceivers>
<fileCompression>
  true
</fileCompression>
    </CreateJob>
  </soap:Body>
</soap:Envelope>
```

The protocol server responds with a SOAP message such as the following:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateJobResponse
      xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <CreateJobResult>
        0fdac8a8-87ca-45d7-8eb6-ba5c976f5b57
      </CreateJobResult>
    </CreateJobResponse>
  </soap:Body>
</soap:Envelope>
```

4.3 GetJobStatus

If a protocol client wants to obtain the status of an existing import job, the protocol client sends a SOAP message, such as the following:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetJobStatus
      xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <jobId>
        0fdac8a8-87ca-45d7-8eb6-ba5c976f5b57
      </jobId>
    </GetJobStatus>
  </soap:Body>
</soap:Envelope>
```

The protocol server responds with a SOAP message, such as the following:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Body>
     <GetJobStatusResponse
        xmlns="http://schemas.microsoft.com/sharepoint/soap/">
       <GetJobStatusResult>
         <Status>
            ImportInProgress
         </Status>
         <StatusMessage>
            Content deployment job 'Remote import job with sourceID = CDJob001': Import in
progress
         </StatusMessage>
         <LastModifiedTime>
            2008-02-15T19:09:09
         </LastModifiedTime>
         <StartTime>
            2008-02-15T19:08:31
         </StartTime>
         <EndTime>
            2008-02-15T19:09:09
         </EndTime>
         <ImportedObjectCount>
            324
         </ImportedObjectCount>
         <ContentSize>
            0
         </ContentSize>
         <ImportLogFileMachineName>cox64-112</ImportLogFileMachineName>
         <ImportLogFilePath>
D:\ProgramData\ContentDeployment\ImportExportLogs\398052fb-86ca-4d3d-914b-
2e952cebcff2\ExportLog 20091210 082455.zip
         </ImportLogFilePath>
         <ReportMessages/>
       </GetJobStatusResult>
     </GetJobStatusResponse>
   </soap:Body>
</soap:Envelope>
```

5   Security

5.1   Security Considerations for Implementers

None.

5.2   Index of Security Parameters

None.

6   Appendix A: Full WSDL

For ease of implementation, the full WSDL and schema are provided in this appendix.

```xml
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/"
xmlns:s1="http://microsoft.com/wsdl/types/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/">
      <s:import namespace="http://microsoft.com/wsdl/types/" />
      <s:element name="GetRemoteAdminServerUrl">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="clientVersion" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetRemoteAdminServerUrlResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetRemoteAdminServerUrlResult"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetVirtualServersInformation">
        <s:complexType />
      </s:element>
      <s:element name="GetVirtualServersInformationResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetVirtualServersInformationResult"
type="tns:ArrayOfVirtualServerInformation" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfVirtualServerInformation">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="VirtualServerInformation"
nillable="true" type="tns:VirtualServerInformation" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="VirtualServerInformation">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="VirtualServerName" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="VirtualServerUrl" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="GetSiteCollectionNames">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="virtualServerUrl" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetSiteCollectionNamesResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetSiteCollectionNamesResult"
type="tns:ArrayOfString" />
          </s:sequence>
```

```
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfString">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="CreateJob">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="description" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="sourceId" type="s1:guid" />
            <s:element minOccurs="1" maxOccurs="1" name="destinationServerUrl"
type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="destinationSiteCollection"
type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="includeSecurity"
type="tns:SPIncludeSecurity" />
            <s:element minOccurs="1" maxOccurs="1" name="includeUserInfoDateTime"
type="tns:SPImportUserInfoDateTimeOption" />
            <s:element minOccurs="1" maxOccurs="1" name="enableEventReceivers"
type="s:boolean" />
            <s:element minOccurs="1" maxOccurs="1" name="fileCompression" type="s:boolean" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:simpleType name="SPIncludeSecurity">
        <s:restriction base="s:string">
          <s:enumeration value="None" />
          <s:enumeration value="WssOnly" />
          <s:enumeration value="All" />
        </s:restriction>
      </s:simpleType>
      <s:simpleType name="SPImportUserInfoDateTimeOption">
        <s:restriction base="s:string">
          <s:enumeration value="None" />
          <s:enumeration value="ReplaceUserWithSystemAccount" />
          <s:enumeration value="ImportAll" />
        </s:restriction>
      </s:simpleType>
      <s:element name="CreateJobResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="CreateJobResult" type="s1:guid" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetJobStatus">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="jobId" type="s1:guid" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetJobStatusResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetJobStatusResult"
type="tns:ContentDeploymentRemoteJobStatus" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ContentDeploymentRemoteJobStatus">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="Status"
type="tns:ContentDeploymentJobStatus" />
          <s:element minOccurs="1" maxOccurs="1" name="StatusMessage" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="LastModifiedTime" type="s:dateTime" />
```

```
              <s:element minOccurs="1" maxOccurs="1" name="StartTime" type="s:dateTime" />
              <s:element minOccurs="1" maxOccurs="1" name="EndTime" type="s:dateTime" />
              <s:element minOccurs="1" maxOccurs="1" name="ImportedObjectCount" type="s:int" />
              <s:element minOccurs="1" maxOccurs="1" name="ContentSize" type="s:long" />
              <s:element minOccurs="1" maxOccurs="1" name="ImportLogFileMachineName"
  type="s:string" />
              <s:element minOccurs="1" maxOccurs="1" name="ImportLogFilePath" type="s:string" />
              <s:element minOccurs="1" maxOccurs="1" name="ReportMessages"
  type="tns:ArrayOfReportMessage" />
          </s:sequence>
        </s:complexType>
        <s:simpleType name="ContentDeploymentJobStatus">
          <s:restriction base="s:string">
            <s:enumeration value="Success" />
            <s:enumeration value="Failure" />
            <s:enumeration value="Cancelled" />
            <s:enumeration value="ImportInProgress" />
            <s:enumeration value="CancelInProgress" />
            <s:enumeration value="ImportPreparing" />
            <s:enumeration value="ImportTimedOut" />
          </s:restriction>
        </s:simpleType>
        <s:complexType name="ArrayOfReportMessage">
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="unbounded" name="ReportMessage" nillable="true"
  type="tns:ReportMessage" />
          </s:sequence>
        </s:complexType>
        <s:complexType name="ReportMessage">
          <s:attribute name="Title" type="s:string" />
          <s:attribute name="Time" type="s:dateTime" use="required" />
          <s:attribute name="Severity" type="tns:MessageSeverity" use="required" />
          <s:attribute name="Description" type="s:string" />
          <s:attribute name="Recommendation" type="s:string" />
          <s:attribute name="ObjectName" type="s:string" />
          <s:attribute name="Phase" type="s:string" />
        </s:complexType>
        <s:simpleType name="MessageSeverity">
          <s:restriction base="s:string">
            <s:enumeration value="Informational" />
            <s:enumeration value="Warning" />
            <s:enumeration value="Error" />
          </s:restriction>
        </s:simpleType>
        <s:element name="RunJob">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="1" maxOccurs="1" name="jobId" type="s1:guid" />
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="RunJobResponse">
          <s:complexType />
        </s:element>
        <s:element name="DeleteJob">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="1" maxOccurs="1" name="jobId" type="s1:guid" />
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="DeleteJobResponse">
          <s:complexType />
        </s:element>
        <s:element name="CancelJob">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="1" maxOccurs="1" name="jobId" type="s1:guid" />
            </s:sequence>
```

```xml
          </s:complexType>
        </s:element>
        <s:element name="CancelJobResponse">
          <s:complexType />
        </s:element>
        <s:element name="DeleteImportLogs">
          <s:complexType>
            <s:sequence>
              <s:element minOccurs="1" maxOccurs="1" name="sourceJobId" type="s1:guid" />
            </s:sequence>
          </s:complexType>
        </s:element>
        <s:element name="DeleteImportLogsResponse">
          <s:complexType />
        </s:element>
      </s:schema>
      <s:schema elementFormDefault="qualified"
    targetNamespace="http://microsoft.com/wsdl/types/">
        <s:simpleType name="guid">
          <s:restriction base="s:string">
            <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-
    9a-fA-F]{12}" />
          </s:restriction>
        </s:simpleType>
      </s:schema>
    </wsdl:types>
    <wsdl:message name="GetRemoteAdminServerUrlSoapIn">
      <wsdl:part name="parameters" element="tns:GetRemoteAdminServerUrl" />
    </wsdl:message>
    <wsdl:message name="GetRemoteAdminServerUrlSoapOut">
      <wsdl:part name="parameters" element="tns:GetRemoteAdminServerUrlResponse" />
    </wsdl:message>
    <wsdl:message name="GetVirtualServersInformationSoapIn">
      <wsdl:part name="parameters" element="tns:GetVirtualServersInformation" />
    </wsdl:message>
    <wsdl:message name="GetVirtualServersInformationSoapOut">
      <wsdl:part name="parameters" element="tns:GetVirtualServersInformationResponse" />
    </wsdl:message>
    <wsdl:message name="GetSiteCollectionNamesSoapIn">
      <wsdl:part name="parameters" element="tns:GetSiteCollectionNames" />
    </wsdl:message>
    <wsdl:message name="GetSiteCollectionNamesSoapOut">
      <wsdl:part name="parameters" element="tns:GetSiteCollectionNamesResponse" />
    </wsdl:message>
    <wsdl:message name="CreateJobSoapIn">
      <wsdl:part name="parameters" element="tns:CreateJob" />
    </wsdl:message>
    <wsdl:message name="CreateJobSoapOut">
      <wsdl:part name="parameters" element="tns:CreateJobResponse" />
    </wsdl:message>
    <wsdl:message name="GetJobStatusSoapIn">
      <wsdl:part name="parameters" element="tns:GetJobStatus" />
    </wsdl:message>
    <wsdl:message name="GetJobStatusSoapOut">
      <wsdl:part name="parameters" element="tns:GetJobStatusResponse" />
    </wsdl:message>
    <wsdl:message name="RunJobSoapIn">
      <wsdl:part name="parameters" element="tns:RunJob" />
    </wsdl:message>
    <wsdl:message name="RunJobSoapOut">
      <wsdl:part name="parameters" element="tns:RunJobResponse" />
    </wsdl:message>
    <wsdl:message name="DeleteJobSoapIn">
      <wsdl:part name="parameters" element="tns:DeleteJob" />
    </wsdl:message>
    <wsdl:message name="DeleteJobSoapOut">
      <wsdl:part name="parameters" element="tns:DeleteJobResponse" />
    </wsdl:message>
    <wsdl:message name="CancelJobSoapIn">
```

```
      <wsdl:part name="parameters" element="tns:CancelJob" />
    </wsdl:message>
    <wsdl:message name="CancelJobSoapOut">
      <wsdl:part name="parameters" element="tns:CancelJobResponse" />
    </wsdl:message>
    <wsdl:message name="DeleteImportLogsSoapIn">
      <wsdl:part name="parameters" element="tns:DeleteImportLogs" />
    </wsdl:message>
    <wsdl:message name="DeleteImportLogsSoapOut">
      <wsdl:part name="parameters" element="tns:DeleteImportLogsResponse" />
    </wsdl:message>
    <wsdl:portType name="ContentDeploymentRemoteImportSoap">
      <wsdl:operation name="GetRemoteAdminServerUrl">
        <wsdl:input message="tns:GetRemoteAdminServerUrlSoapIn" />
        <wsdl:output message="tns:GetRemoteAdminServerUrlSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="GetVirtualServersInformation">
        <wsdl:input message="tns:GetVirtualServersInformationSoapIn" />
        <wsdl:output message="tns:GetVirtualServersInformationSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="GetSiteCollectionNames">
        <wsdl:input message="tns:GetSiteCollectionNamesSoapIn" />
        <wsdl:output message="tns:GetSiteCollectionNamesSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="CreateJob">
        <wsdl:input message="tns:CreateJobSoapIn" />
        <wsdl:output message="tns:CreateJobSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="GetJobStatus">
        <wsdl:input message="tns:GetJobStatusSoapIn" />
        <wsdl:output message="tns:GetJobStatusSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="RunJob">
        <wsdl:input message="tns:RunJobSoapIn" />
        <wsdl:output message="tns:RunJobSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="DeleteJob">
        <wsdl:input message="tns:DeleteJobSoapIn" />
        <wsdl:output message="tns:DeleteJobSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="CancelJob">
        <wsdl:input message="tns:CancelJobSoapIn" />
        <wsdl:output message="tns:CancelJobSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="DeleteImportLogs">
        <wsdl:input message="tns:DeleteImportLogsSoapIn" />
        <wsdl:output message="tns:DeleteImportLogsSoapOut" />
      </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="ContentDeploymentRemoteImportSoap"
type="tns:ContentDeploymentRemoteImportSoap">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="GetRemoteAdminServerUrl">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetRemoteAdminServerUrl"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetVirtualServersInformation">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetVirtualServersInformation"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
```

```
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSiteCollectionNames">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSiteCollectionNames"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateJob">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/CreateJob"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetJobStatus">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetJobStatus"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RunJob">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/RunJob"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteJob">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteJob"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CancelJob">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/CancelJob"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteImportLogs">
```

```
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteImportLogs" style="document"
/>
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="ContentDeploymentRemoteImportSoap12"
type="tns:ContentDeploymentRemoteImportSoap">
      <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="GetRemoteAdminServerUrl">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetRemoteAdminServerUrl"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetVirtualServersInformation">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetVirtualServersInformation"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetSiteCollectionNames">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetSiteCollectionNames"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="CreateJob">
        <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/CreateJob"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetJobStatus">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetJobStatus" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="RunJob">
```

```
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/RunJob"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteJob">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteJob"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CancelJob">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/CancelJob"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteImportLogs">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteImportLogs" style="document"
/>
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

7   Appendix B: Product Behavior

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

- Microsoft Office SharePoint Server 2007

- Microsoft SharePoint Server 2010

- Microsoft SharePoint Server 2013

- Microsoft SharePoint Server 2016

- Microsoft SharePoint Server 2019

- Microsoft SharePoint Server Subscription Edition

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

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

<1> Section 3.1.4.2: Office SharePoint Server 2007 does not create a new import job if the protocol server finds no existing import job with the same **sourceId** parameter to delete but a **timer job** exists with the following name: "ContentDeploymentJobDefinition: Remote import job for job with sourceID = *sourceId*" where *sourceId* is the value of the **sourceId** parameter.

<2> Section 3.1.4.2.2.1: For versions previous to SharePoint Server 2010 the **sourceId** parameter is declared with the type set to "s:string". It MUST adhere to the format specified in section 2.2.5.1.

<3> Section 3.1.4.3: The **DeleteImportLogs** operation is not included in Office SharePoint Server 2007 before update KB2345212 is installed.

<4> Section 3.1.4.5.3.1: The **ImportLogFileMachineName** and **ImportLogFilePath** elements in the **ContentDeploymentRemoteJobStatus** complex type are not included in Office SharePoint Server 2007 before update KB2345212 is installed.

<5> Section 3.2.4: The **deployment package** generated by Office SharePoint Server 2007 contains the file ContentDeploymentInformation.xml. This file is reserved and MUST be ignored by import jobs.

<6> Section 3.2.4: For remote import servers running Office SharePoint Server 2007 **BaseFileName** MUST be either ExportedFiles.cab or ExportedFilesRetryNNN_.cab where NNN is a three-digit integer between 001 and 999.

8 Change Tracking

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

9 Index