

## [MS-WFIM-Diff]:

# Workflow Instance Management 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 as well as overviews of the interaction among each of these technologies 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 maycan make copies of it in order to develop implementations of the technologies that are described in the Open Specifications-~~this documentation~~ and maycan distribute portions of it in your implementations usingthat use these technologies or in your documentation as necessary to properly document the implementation. You maycan also distribute in your implementation, with or without modification, any schema, IDL'sschemas, 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 maymight cover your implementations of the technologies described in the Open Specifications-documentation. Neither this notice nor Microsoft's delivery of thethis documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specification maySpecifications 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 the Open Specifications>this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation maymight be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mailemail 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 standardstandards specifications and network programming art; and assumes, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

Date	Revision History	Revision Class	Comments
9/25/2009	0.1	Major	First Release.
11/6/2009	0.1.1	Editorial	Changed language and formatting in the technical content.
12/18/2009	0.1.2	Editorial	Changed language and formatting in the technical content.
1/29/2010	0.2	Minor	Clarified the meaning of the technical content.
3/12/2010	0.2.1	Editorial	Changed language and formatting in the technical content.
4/23/2010	0.3	Minor	Clarified the meaning of the technical content.
6/4/2010	0.3.1	Editorial	Changed language and formatting in the technical content.
7/16/2010	1.0	Major	Updated and revised the technical content.
8/27/2010	1.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	1.0	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	1.0	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	2.0	Major	Updated and revised the technical content.
2/11/2011	2.0	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	2.0	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	2.0	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	2.1	Minor	Clarified the meaning of the technical content.
9/23/2011	2.1	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	3.0	Major	Updated and revised the technical content.
3/30/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	3.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	3.0	None	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	3.0	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
2/13/2014	3.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	3.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	4.0	Major	Significantly changed the technical content.
10/16/2015	5.0	Major	Significantly changed the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>7</b>
1.1	Glossary .....	7
1.2	References .....	8
1.2.1	Normative References .....	8
1.2.2	Informative References .....	9
1.3	Overview .....	9
1.4	Relationship to Other Protocols .....	10
1.5	Prerequisites/Preconditions .....	10
1.6	Applicability Statement .....	10
1.7	Versioning and Capability Negotiation .....	10
1.8	Vendor-Extensible Fields .....	10
1.9	Standards Assignments.....	11
<b>2</b>	<b>Messages.....</b>	<b>12</b>
2.1	Transport .....	12
2.2	Common Message Syntax .....	12
2.2.1	Namespaces .....	12
2.2.2	Messages.....	13
2.2.3	Elements .....	13
2.2.4	Complex Types.....	13
2.2.5	Simple Types .....	13
2.2.6	Attributes .....	13
2.2.7	Groups .....	13
2.2.8	Attribute Groups.....	13
<b>3</b>	<b>Protocol Details .....</b>	<b>14</b>
3.1	IWorkflowInstanceManagement Server Details.....	14
3.1.1	Abstract Data Model.....	14
3.1.1.1	Active State .....	14
3.1.1.2	Suspended State.....	15
3.1.1.3	Completed State .....	15
3.1.2	Timers .....	15
3.1.3	Initialization .....	15
3.1.4	Message Processing Events and Sequencing Rules .....	16
3.1.4.1	Run.....	17
3.1.4.1.1	Messages .....	17
3.1.4.1.1.1	IWorkflowInstanceManagement_Run_InputMessage.....	17
3.1.4.1.1.2	IWorkflowInstanceManagement_Run_OutputMessage .....	18
3.1.4.1.2	Elements.....	18
3.1.4.1.2.1	Run.....	18
3.1.4.1.2.2	RunResponse .....	18
3.1.4.2	TransactedRun .....	19
3.1.4.2.1	Messages .....	19
3.1.4.2.1.1	IWorkflowInstanceManagement_TransactedRun_InputMessage .....	19
3.1.4.2.1.2	IWorkflowInstanceManagement_TransactedRun_OutputMessage .....	20
3.1.4.2.2	Elements.....	20
3.1.4.2.2.1	TransactedRun .....	20
3.1.4.2.2.2	TransactedRunResponse .....	20
3.1.4.3	Abandon .....	21
3.1.4.3.1	Messages .....	21
3.1.4.3.1.1	IWorkflowInstanceManagement_Abandon_InputMessage.....	22
3.1.4.3.1.2	IWorkflowInstanceManagement_Abandon_OutputMessage.....	22
3.1.4.3.2	Elements.....	22
3.1.4.3.2.1	Abandon .....	22
3.1.4.3.2.2	AbandonResponse.....	23

3.1.4.4	Cancel .....	23
3.1.4.4.1	Messages .....	23
3.1.4.4.1.1	IWorkflowInstanceManagement_Cancel_InputMessage .....	24
3.1.4.4.1.2	IWorkflowInstanceManagement_Cancel_OutputMessage .....	24
3.1.4.4.2	Elements .....	24
3.1.4.4.2.1	Cancel .....	24
3.1.4.4.2.2	CancelResponse .....	24
3.1.4.5	TransactedCancel .....	25
3.1.4.5.1	Messages .....	25
3.1.4.5.1.1	IWorkflowInstanceManagement_TransactedCancel_InputMessage ...	26
3.1.4.5.1.2	IWorkflowInstanceManagement_TransactedCancel_OutputMessage .	26
3.1.4.5.2	Elements .....	26
3.1.4.5.2.1	TransactedCancel .....	26
3.1.4.5.2.2	TransactedCancelResponse .....	27
3.1.4.6	Terminate .....	27
3.1.4.6.1	Messages .....	27
3.1.4.6.1.1	IWorkflowInstanceManagement_Terminate_InputMessage .....	28
3.1.4.6.1.2	IWorkflowInstanceManagement_Terminate_OutputMessage .....	28
3.1.4.6.2	Elements .....	28
3.1.4.6.2.1	Terminate .....	28
3.1.4.6.2.2	TerminateResponse .....	29
3.1.4.7	TransactedTerminate .....	29
3.1.4.7.1	Messages .....	30
3.1.4.7.1.1	IWorkflowInstanceManagement_TransactedTerminate_InputMessage	30
3.1.4.7.1.2	IWorkflowInstanceManagement_TransactedTerminate_OutputMessage	30
3.1.4.7.2	Elements .....	30
3.1.4.7.2.1	TransactedTerminate .....	31
3.1.4.7.2.2	TransactedTerminateResponse .....	31
3.1.4.8	Suspend .....	31
3.1.4.8.1	Messages .....	32
3.1.4.8.1.1	IWorkflowInstanceManagement_Suspend_InputMessage .....	32
3.1.4.8.1.2	IWorkflowInstanceManagement_Suspend_OutputMessage .....	32
3.1.4.8.2	Elements .....	32
3.1.4.8.2.1	Suspend .....	33
3.1.4.8.2.2	SuspendResponse .....	33
3.1.4.9	TransactedSuspend .....	33
3.1.4.9.1	Messages .....	34
3.1.4.9.1.1	IWorkflowInstanceManagement_TransactedSuspend_InputMessage .	34
3.1.4.9.1.2	IWorkflowInstanceManagement_TransactedSuspend_OutputMessage	34
3.1.4.9.2	Elements .....	35
3.1.4.9.2.1	TransactedSuspend .....	35
3.1.4.9.2.2	TransactedSuspendResponse .....	35
3.1.4.10	Unsuspend .....	36
3.1.4.10.1	Messages .....	36
3.1.4.10.1.1	IWorkflowInstanceManagement_Unsuspend_InputMessage .....	36
3.1.4.10.1.2	IWorkflowInstanceManagement_Unsuspend_OutputMessage .....	37
3.1.4.10.2	Elements .....	37
3.1.4.10.2.1	Unsuspend .....	37
3.1.4.10.2.2	UnsuspendResponse .....	37
3.1.4.11	TransactedUnsuspend .....	38
3.1.4.11.1	Messages .....	38
3.1.4.11.1.1	IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage	38
3.1.4.11.1.2	IWorkflowInstanceManagement_TransactedUnsuspend_OutputMessage	39
3.1.4.11.2	Elements .....	39
3.1.4.11.2.1	TransactedUnsuspend .....	39
3.1.4.11.2.2	TransactedUnsuspendResponse .....	39

3.1.4.12	Update .....	40
3.1.4.12.1	Messages .....	40
3.1.4.12.1.1	IWorkflowInstanceManagement_Update_InputMessage .....	40
3.1.4.12.1.2	IWorkflowInstanceManagement_Update_OutputMessage .....	40
3.1.4.12.2	Elements .....	41
3.1.4.12.2.1	Update .....	41
3.1.4.12.2.2	UpdateResponse .....	41
3.1.4.13	TransactedUpdate .....	42
3.1.4.13.1	Messages .....	42
3.1.4.13.1.1	IWorkflowInstanceManagement_TransactedUpdate_InputMessage ...	42
3.1.4.13.1.2	IWorkflowInstanceManagement_TransactedUpdate_OutputMessage.	42
3.1.4.13.2	Elements .....	43
3.1.4.13.2.1	TransactedUpdate .....	43
3.1.4.13.2.2	TransactedUpdateResponse.....	43
3.1.5	Timer Events.....	44
3.1.6	Other Local Events.....	44
3.2	IWorkflowInstanceManagement Client Details.....	44
<b>4</b>	<b>Protocol Examples .....</b>	<b>45</b>
<b>5</b>	<b>Security .....</b>	<b>46</b>
5.1	Security Considerations for Implementers .....	46
5.2	Index of Security Parameters .....	46
<b>6</b>	<b>Appendix A: Full WSDL .....</b>	<b>47</b>
6.1	Workflow Instance Management Protocol WSDL .....	47
6.2	Workflow Instance Management Schema for the WSDL.....	52
6.3	Workflow Identity Management Schema for the WSDL.....	56
<b>7</b>	<b>Appendix B: Product Behavior .....</b>	<b>57</b>
<b>8</b>	<b>Change Tracking.....</b>	<b>59</b>
<b>9</b>	<b>Index.....</b>	<b>61</b>

# 1 Introduction

This document specifies the Workflow Instance Management Protocol, which defines a set of SOAP messages for the management of **durable program instances**, such as suspending, resuming, or canceling an instance.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative ~~and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms.~~ All other sections and examples in this specification are informative.

## 1.1 Glossary

~~The~~This document uses the following terms ~~are specific to this document:~~

**durable program:** A program whose lifetime is not bound to a single operating system process. For more information about these processes, see [PROCESS]. The execution of the **durable program** starts in one process with a durable state, survives process termination, and can continue to execute in another process at a later point in time.

**durable program instance:** An identifiable occurrence of the execution of a **durable program**. The **durable program instance** captures the complete state of the execution. The execution of a **durable program instance** is limited to a single process at a time.

**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).

**management operation:** An operation on a **durable program instance** that is not related to the business logic of the **durable program**.

**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 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.

**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.

**WSDL message:** An abstract, typed definition of the data that is communicated during a **WSDL operation** [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

**WSDL operation:** A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

**WSDL port type:** A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

**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 Schema (XSD):** A language that defines the elements, attributes, namespaces, and data types for XML documents as defined by [XMLSCHEMA1/2] and [W3C-XSD] standards. An XML schema uses XML syntax for its language.

**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 refers 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-DTCO] Microsoft Corporation, "MSDTC Connection Manager: OleTx Transaction Protocol".

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

[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>

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", 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 27, 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-2ED] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, <http://www.w3.org/TR/2006/REC-xml-names-20060816/>

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

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

## 1.2.2 Informative References

[MS-WSPOL] Microsoft Corporation, "Web Services: Policy Assertions and WSDL Extensions".

[MS-WSRVCAT] Microsoft Corporation, "WS-AtomicTransaction (WS-AT) Version 1.0 Protocol Extensions".

[WSS1] Nadalin, A., Kaler, C., Hallam-Baker, P., et al., "Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)", March 2004, <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

## 1.3 Overview

The familiar control operations of starting, pausing, and terminating processes are sufficient for managing programs where execution is contained within a single process; however, these operations are insufficient when the program is durable because a **durable program** spans multiple processes over time. A similar control mechanism that is not scoped to a single process is required for managing durable programs. The Workflow Instance Management Protocol specifies such a control mechanism.

Durable program instances can be hosted on a variety of execution environments or hosts, for example on a desktop computer, a server farm, and so on. The Workflow Instance Management Protocol is provided on durable program hosts that support messaging (that is, messaging hosts) for the external control of various lifetime and execution aspects of the durable program instances running on that host. External control consists of operations for terminating, suspending, and resuming the execution of durable program instances where the client for these operations is typically system administration tooling.

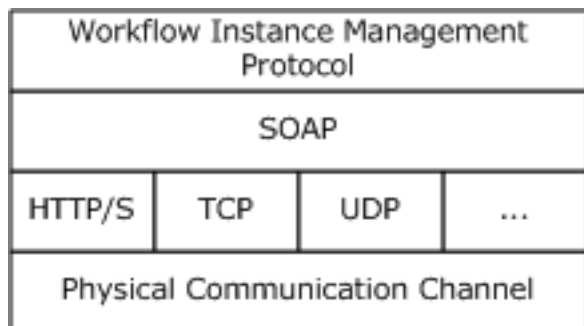
The Workflow Instance Management Protocol defines a set of request and reply **SOAP messages** that specify these external control operations. This specification also describes the interdependencies of these operations and how they relate to an abstract model of the durable program instance state.

For example, consider an expense approval durable program that is running in a messaging host. The host for the expense approval durable program exposes an expense approval messaging endpoint. The expense approval endpoint and its protocol are part of the definition of the expense approval application. The host can also expose a messaging endpoint that supports the Workflow Instance Management Protocol. This is a generic, infrastructural endpoint provided by the host for the administration of instances of the expense approval durable program. Using this infrastructural endpoint, an administrator of the application can have available tooling that uses the Workflow Instance Management Protocol to control the execution of instances of the expense approval workflows. Using the **Abandon**, **Cancel**, **Terminate**, **Suspend**, and **Unsuspend** operations defined in this protocol, the tooling enables the administrator to perform tasks, such as terminating a particular Instance or temporarily suspending its execution.

In some scenarios, operations in the Workflow Instance Management Protocol are used by the system internally itself. For example, the **Run** operation can be utilized internally by the system for recovery after system failure.

## 1.4 Relationship to Other Protocols

The Workflow Instance Management Protocol can be used with **SOAP**-formatted messages. The following figure shows a protocol stack:



**Figure 1: Protocol stack for the Workflow Instance Management Protocol**

## 1.5 Prerequisites/Preconditions

The Workflow Instance Management Protocol requires that:

1. The client role can communicate with the server role so that messages can be exchanged between client and server.
2. The server role can create and host durable program instances and associate a unique identifier to each durable program instance.
3. The client role can determine the unique identifier associated by the server role to the durable program instance on which **management operation(s)** need to be performed. This unique identifier is used by the client to identify the target instance of the management operation on the server.

## 1.6 Applicability Statement

The Workflow Instance Management Protocol is applicable to scenarios where management of durable program instances is required. The client and server use this protocol to perform management operations on durable program instances.

## 1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP as specified in section 2.1.
- **Protocol Versions:** This protocol has only one **WSDL port type** version with a single set of operations. The use of these operations is specified in section 3.2.
- **Capability Negotiation:** The Workflow Instance Management Protocol does not support negotiation of the version to use. Instead, an implementation has to be configured to process messages only as described in section 2.1.

## 1.8 Vendor-Extensible Fields

There are no vendor-extensible fields in this protocol.

## 1.9 Standards Assignments

None.

## 2 Messages

### 2.1 Transport

The Workflow Instance Management Protocol can be used over any transport protocol that supports transmitting messages that are specified by the following protocols:

- SOAP 1.1 [SOAP1.1]
- SOAP 1.2 [SOAP1.2-1/2007]

This specification uses the term SOAP to mean either SOAP 1.1 or SOAP 1.2. An implementation of the Workflow Instance Management Protocol **MUST** support the processing of messages that are specified by either of these SOAP versions.

### 2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses **XML schema (XSD)** as defined in [XMLSCHEMA1] and [XMLSCHEMA2], and Web Services Description Language as defined in [WSDL].

#### 2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [XMLNS-2ED]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and is not significant for interoperability.

Prefix	Namespace URI	Reference
soap	<a href="http://schemas.xmlsoap.org/wsdl/soap/">http://schemas.xmlsoap.org/wsdl/soap/</a>	[SOAP1.1]
Soapenc	<a href="http://schemas.xmlsoap.org/soap/encoding/">http://schemas.xmlsoap.org/soap/encoding/</a>	[SOAP1.1]
Wsu	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a>	
Xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	
soap12	<a href="http://schemas.xmlsoap.org/wsdl/soap12/">http://schemas.xmlsoap.org/wsdl/soap12/</a>	[SOAP1.2-1/2007], [SOAP1.2-2/2007]
Tns	<a href="http://schemas.datacontract.org/2008/10/WorkflowServices">http://schemas.datacontract.org/2008/10/WorkflowServices</a>	
Wsa	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>	
Wsp	<a href="http://schemas.xmlsoap.org/ws/2004/09/policy">http://schemas.xmlsoap.org/ws/2004/09/policy</a>	
Wsap	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy">http://schemas.xmlsoap.org/ws/2004/08/addressing/policy</a>	
WsaW	<a href="http://www.w3.org/2006/05/addressing/wsdl">http://www.w3.org/2006/05/addressing/wsdl</a>	
Msc	<a href="http://schemas.microsoft.com/ws/2005/12/wsdl/contract">http://schemas.microsoft.com/ws/2005/12/wsdl/contract</a>	[MS-WSPOL]
wsa10	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>	
Wsx	<a href="http://schemas.xmlsoap.org/ws/2004/09/mex">http://schemas.xmlsoap.org/ws/2004/09/mex</a>	
Wsam	<a href="http://www.w3.org/2007/05/addressing/metadata">http://www.w3.org/2007/05/addressing/metadata</a>	

Prefix	Namespace URI	Reference
Wsdli	http://schemas.xmlsoap.org/wsdl/	[WSDL]
Xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1], [XMLSCHEMA2]
q4	http://schemas.microsoft.com/2003/10/Serialization/	

## 2.2.2 Messages

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

## 2.2.3 Elements

This specification does not define any common XSD element definitions.

## 2.2.4 Complex Types

This specification does not define any common XSD complex-type definitions.

## 2.2.5 Simple Types

This specification does not define any common XSD simple-type definitions.

## 2.2.6 Attributes

This specification does not define any common XSD attribute definitions.

## 2.2.7 Groups

This specification does not define any common XSD group definitions.

## 2.2.8 Attribute Groups

This specification does not define any common XSD attribute group definitions.

### 3 Protocol Details

The client side of this protocol is simply a pass-through mechanism. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

#### 3.1 IWorkflowInstanceManagement Server Details

##### 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 server MUST maintain the following data element:

- **Durable Program Instance Table:** A table that associates a **globally unique identifier (GUID)**, as specified in [MS-DTYP] section 2.3.4, to a durable program instance and durable program instance state. The durable program instance state is an enumeration that identifies the current state of the durable program instance:
  - Active
  - Suspended
  - Completed

The following table shows the relationship between durable program instance states and Workflow Instance Management Protocol operations. The table identifies the durable program instance state when the operation completes, based on the durable program instance state when the operation was invoked.

Operation	Workflow instance state when operation invoked		
	Active	Suspended	Completed
Abandon	<end>	<end>	<end>
Run	Active	Suspended	Completed
Cancel	Completed	Completed	Completed
Terminate	Completed	Completed	Completed
Suspend	Suspended	Suspended	Completed
Unsuspend	Active	Active	Completed
TransactedRun	Active	Suspended	Completed
TransactedSuspend	Suspended	Suspended	Completed
TrnasactedUnsuspend	Active	Active	Completed
TransactedCancel	Completed	Completed	Completed
TransactedTerminate	Completed	Completed	Completed

} Workflow instance state after operation completes

**Figure 2: Durable program instance states when operation is invoked and completed**

##### 3.1.1.1 Active State

The durable program instance is in the active state before it reaches the completed state and when it is not in the suspended state. In the active state, the durable program instance SHOULD execute and process application messages.

### 3.1.1.2 Suspended State

In the suspended state, the durable program instance MUST NOT execute.

### 3.1.1.3 Completed State

The completed state is a final state of the durable program instance. The durable program instance MUST NOT execute in this state.

In a typical implementation, other parts of the system will interact with the durable program instance and can cause the state to be changed. The current state of the durable program instance can also be a snapshot into a durable store, where durability affects the system in the sense that a durable program instance can be reloaded from the durable store, or can be reset to the last durable state. As a result, the Workflow Instance Management Protocol does not prescribe a durable program instance state machine. In the absence of any other interactions, an implementation MAY<1> implement the following durable program instance state machine.

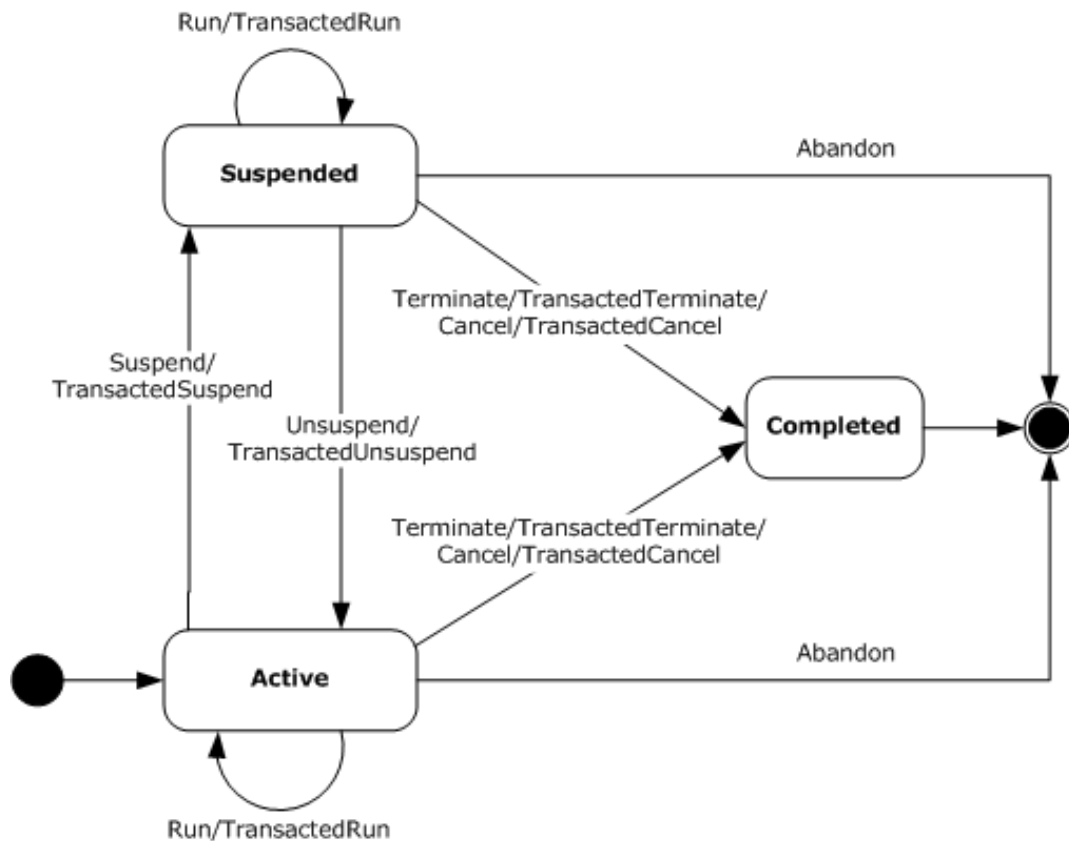


Figure 3: Durable program instance state machine

### 3.1.2 Timers

None.

### 3.1.3 Initialization

When a server role is initialized:

- The **Durable Program Instance Table** MUST be set to a value that is obtained from an implementation-specific source.
- A listening infrastructural endpoint is created.

When a durable program instance is initialized:

- An entry for the durable program instance MUST be made in the **Durable Program Instance Table**.
- A GUID to identify the durable program instance MUST be set to a value that is obtained from an implementation-specific source.
- The durable program instance state MUST be set to one of the enumerated values: active, suspended, or completed.

### 3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
<b>Abandon</b>	SHOULD forcefully stop the execution of the durable program instance and indicate to the system that the durable program instance SHOULD be disposed.
<b>Cancel</b>	Transitions a durable program instance from the active or suspended state to the completed state. The operation SHOULD gracefully cancel any remaining work and clean up resources being used by the durable program instance.
<b>Run</b>	SHOULD provide the durable program instance an opportunity to execute.
<b>Suspend</b>	Transitions a durable program instance from the active state to the suspended state.
<b>Terminate</b>	Transitions a durable program instance from the active or suspended state to the completed state. It SHOULD perform the minimum possible work needed to transition the durable program instance to the completed state.
<b>TransactedCancel</b>	Performs the <b>Cancel</b> operation under a transaction (flowed in from the client or locally created). If the system maintains the durable state of the durable program instance, the durable state MUST be updated during execution of this operation.
<b>TransactedRun</b>	Performs the <b>Run</b> operation under a transaction (flowed in from the client or locally created). If the system maintains the durable state of the durable program instance, the durable state MUST be updated during execution of this operation.
<b>TransactedSuspend</b>	Performs the <b>Suspend</b> operation under a transaction (flowed in from the client or locally created). If the system maintains the durable state of the durable program instance, the durable state MUST be updated during execution of this operation.
<b>TransactedTerminate</b>	Performs the <b>Terminate</b> operation under a transaction (flowed in from the client or locally created). If the system maintains the durable state of the durable program instance, the durable state MUST be updated during execution of this operation.
<b>TransactedUnsuspend</b>	Performs the <b>Unsuspend</b> operation under a transaction (flowed in from the client or locally created). If the system maintains the durable state of the durable program instance, the durable state MUST be updated during execution of this operation.
<b>TransactedUpdate</b>	Performs the <b>Update</b> operation under a transaction (flowed in from the client or locally created).
<b>Unsuspend</b>	Transitions a durable program instance from the suspended state to the active state.
<b>Update</b>	Transitions the identity of a durable program instance from its current identity to an



Operation	Description
	updated identity.

### 3.1.4.1 Run

The **WSDL** definition of the **Run** operation is as follows.

```
<wsdl:operation name="Run">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/Run"
    message="tns:IWorkflowInstanceManagement_Run_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/RunResponse"
    message="tns:IWorkflowInstanceManagement_Run_OutputMessage" />
</wsdl:operation>
```

The **Run** operation SHOULD provide the durable program instance with an opportunity to execute in an implementation-specific manner. A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. The operation SHOULD return a **SOAP fault** message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
- The durable program instance associated with the value of the <instanceId> element is in the suspended state.
- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The server encounters an internal error while executing the **Run** operation.

#### 3.1.4.1.1 Messages

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

Message	Description
IWorkflowInstanceManagement_Run_InputMessage	Sent from the client to invoke the <b>Run</b> operation.
IWorkflowInstanceManagement_Run_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_Run_InputMessage.

##### 3.1.4.1.1.1 IWorkflowInstanceManagement\_Run\_InputMessage

The IWorkflowInstanceManagement\_Run\_InputMessage message is the request message for the **Run** operation. The client SHOULD send this message to invoke the **Run** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_Run_InputMessage">
  <wsdl:part name="parameters" element="tns:Run" />
</wsdl:message>
```

```
</wsdl:message>
```

**Run:** The <Run> element, as specified in section 3.1.4.1.2.1.

### 3.1.4.1.1.2 IWorkflowInstanceManagement\_Run\_OutputMessage

The IWorkflowInstanceManagement\_Run\_OutputMessage message is the reply message for the **Run** operation. The message indicates that the **Run** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_Run_OutputMessage">  
  <wsdl:part name="parameters" element="tns:RunResponse" />  
</wsdl:message>
```

**RunResponse:** The <RunResponse> element, as specified in section 3.1.4.1.2.2.

### 3.1.4.1.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<Run>	Contains the body of the IWorkflowInstanceManagement_Run_InputMessage message.
<RunResponse>	Contains the body of the IWorkflowInstanceManagement_Run_OutputMessage message.

#### 3.1.4.1.2.1 Run

<Run> is an XSD element that has a child element <instanceId>. The XSD definition of the <Run> element is as follows:

```
<xs:element name="Run">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element minOccurs="0" name="instanceId"  
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

#### 3.1.4.1.2.2 RunResponse

<RunResponse> is an XSD element that has no child elements. The XSD definition of the <RunResponse> element is as follows:

```
<xs:element name="RunResponse">  
  <xs:complexType>  
    <xs:sequence />  
  </xs:complexType>  
</xs:element>
```

### 3.1.4.2 TransactedRun

The WSDL definition of the **TransactedRun** operation is as follows:

```
<wsdl:operation name="TransactedRun">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedRun"
    message="tns:IWorkflowInstanceManagement_TransactedRun_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedRunResponse"
    message="tns:IWorkflowInstanceManagement_TransactedRun_OutputMessage" />
</wsdl:operation>
```

**TransactedRun** is an atomic operation that SHOULD provide the durable program instance with an opportunity to execute in an implementation-specific manner. The operation SHOULD be performed under the scope of a transaction flowed in from the client, if one is flowed in, using a protocol that is recognized by the client and server roles, such as [MS-WSRVCAT].<2>

If the system maintains the durable state of the durable program instance, then the durable state MUST be updated during execution of this operation. If the durable store is a transactional resource manager, the same transaction SHOULD be used for the durable state change. Failure to make the durable state change MUST result in failure of the operation.

The durable program instance SHOULD start executing when in the active state. A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
- The durable program instance associated with the value of the <instanceId> element is in the suspended state.
- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The server encounters an internal error while executing the **TransactedRun** operation.

#### 3.1.4.2.1 Messages

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

Message	Description
IWorkflowInstanceManagement_TransactedRun_InputMessage	Sent from the client to invoke the <b>TransactedRun</b> operation.
IWorkflowInstanceManagement_TransactedRun_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_TransactedRun_InputMessage.

##### 3.1.4.2.1.1 IWorkflowInstanceManagement\_TransactedRun\_InputMessage

The `IWorkflowInstanceManagement_TransactedRun_InputMessage` message is the request message for the **TransactedRun** operation. The client SHOULD send this message to invoke the **TransactedRun** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedRun_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedRun" />
</wsdl:message>
```

**TransactedRun:** The `<TransactedRun>` element, as specified in section 3.1.4.2.2.1.

### 3.1.4.2.1.2 IWorkflowInstanceManagement\_TransactedRun\_OutputMessage

The `IWorkflowInstanceManagement_TransactedRun_OutputMessage` message is the reply message for the **TransactedRun** operation. The message indicates that the **TransactedRun** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedRun_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedRunResponse" />
</wsdl:message>
```

**TransactedRunResponse:** The `<TransactedRunResponse>` element, as specified in section 3.1.4.2.2.2.

### 3.1.4.2.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<code>&lt;TransactedRun&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_TransactedRun_InputMessage</code> message.
<code>&lt;TransactedRunResponse&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_TransactedRun_OutputMessage</code> message.

#### 3.1.4.2.2.1 TransactedRun

`<TransactedRun>` is an XSD element that has a child element `<instanceId>`. The XSD definition of the `<TransactedRun>` element is as follows:

```
<xs:element name="TransactedRun">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

#### 3.1.4.2.2.2 TransactedRunResponse

<TransactedRunResponse> is an XSD element that has no child elements. The XSD definition of the <TransactedRunResponse> element is as follows:

```
<xs:element name="TransactedRunResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.1.4.3 Abandon

The WSDL definition of the **Abandon** operation is as follows:

```
<wsdl:operation name="Abandon">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/Abandon"
    message="tns:IWorkflowInstanceManagement_Abandon_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/AbandonResponse"
    message="tns:IWorkflowInstanceManagement_Abandon_OutputMessage" />
</wsdl:operation>
```

The **Abandon** operation SHOULD forcefully stop the execution of the durable program instance and indicate to the system that the current durable program instance execution image SHOULD be disposed. If the system maintains the durable state of the durable program instances, then the durable state SHOULD NOT be updated during execution of this operation.

For example, in an expense report processing system, an administrator might decide to **Abandon** all active reports and ask for them to be resubmitted. A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The server encounters an internal error while executing the **Abandon** operation.

#### 3.1.4.3.1 Messages

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

Message	Description
IWorkflowInstanceManagement_Abandon_InputMessage	Sent from the client to invoke the <b>Abandon</b> operation.
IWorkflowInstanceManagement_Abandon_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_Abandon_InputMessage

Message	Description
	e.

### 3.1.4.3.1.1 IWorkflowInstanceManagement\_Abandon\_InputMessage

The IWorkflowInstanceManagement\_Abandon\_InputMessage message is the request message for the **Abandon** operation. The client role SHOULD send this message to invoke the **Abandon** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_Abandon_InputMessage">
  <wsdl:part name="parameters" element="tns:Abandon" />
</wsdl:message>
```

**Abandon:** The <Abandon> element, as specified in section 3.1.4.1.2.2.

### 3.1.4.3.1.2 IWorkflowInstanceManagement\_Abandon\_OutputMessage

The IWorkflowInstanceManagement\_Abandon\_OutputMessage message is the reply message for the **Abandon** operation. The message indicates that the **Abandon** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_Abandon_OutputMessage">
  <wsdl:part name="parameters" element="tns:AbandonResponse" />
</wsdl:message>
```

**AbandonResponse:** The <AbandonResponse> element, as specified in section 3.1.4.3.2.2.

### 3.1.4.3.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<Abandon>	Contains the body of the IWorkflowInstanceManagement_Abandon_InputMessage message.
<AbandonResponse>	Contains the body of the IWorkflowInstanceManagement_Abandon_OutputMessage message.

### 3.1.4.3.2.1 Abandon

<Abandon> is an XSD element that has two child elements: <instanceId> and <reason>. The XSD definition of the <Abandon> element is as follows:

```
<xs:element name="Abandon">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
      <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

**reason:** The value of this element is a description of the reason for performing the **Abandon** operation.

### 3.1.4.3.2.2 AbandonResponse

<AbandonResponse> is an XSD element that has no child elements. The XSD definition of the <AbandonResponse> element is as follows:

```
<xs:element name="AbandonResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.1.4.4 Cancel

The WSDL definition of the **Cancel** operation is as follows:

```
<wsdl:operation name="Cancel">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/Cancel"
    message="tns:IWorkflowInstanceManagement_Cancel_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/CancelResponse"
    message="tns:IWorkflowInstanceManagement_Cancel_OutputMessage" />
</wsdl:operation>
```

The **Cancel** operation transitions a durable program instance from the active or suspended state to the completed state. The operation SHOULD gracefully cancel any remaining work and clean up resources being used by the durable program instance, such as open network connections. Completed is a final state and the durable program instance MUST NOT execute in the completed state. A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The server encounters an internal error while executing the **Cancel** operation.

#### 3.1.4.4.1 Messages

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

Message	Description
IWorkflowInstanceManagement_Cancel_InputMessage	Sent from the client to invoke the <b>Cancel</b> operation.
IWorkflowInstanceManagement_Cancel_OutputMessage	Sent from the server as a reply to

Message	Description
	IWorkflowInstanceManagement_Cancel_InputMessage.

### 3.1.4.4.1.1 IWorkflowInstanceManagement\_Cancel\_InputMessage

The IWorkflowInstanceManagement\_Cancel\_InputMessage message is the request message for the **Cancel** operation. The client role SHOULD send this message to invoke the **Cancel** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_Cancel_InputMessage">
  <wsdl:part name="parameters" element="tns:Cancel" />
</wsdl:message>
```

**Cancel:** The <Cancel> element, as specified in section 3.1.4.4.2.1.

### 3.1.4.4.1.2 IWorkflowInstanceManagement\_Cancel\_OutputMessage

The IWorkflowInstanceManagement\_Cancel\_OutputMessage message is the reply message for the **Cancel** operation. The message indicates that the **Cancel** operation has successfully completed. The SOAP:body of this message consists of the <CancelResponse> element.

```
<wsdl:message name="IWorkflowInstanceManagement_Cancel_OutputMessage">
  <wsdl:part name="parameters" element="tns:CancelResponse" />
</wsdl:message>
```

**CancelResponse:** The <CancelResponse> element, as specified in section 3.1.4.4.2.2.

### 3.1.4.4.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<Cancel>	Contains the body of the IWorkflowInstanceManagement_Cancel_InputMessage message.
<CancelResponse>	Contains the body of the IWorkflowInstanceManagement_Cancel_OutputMessage message.

#### 3.1.4.4.2.1 Cancel

<Cancel> is an XSD element that has a child element <instanceId>. The XSD definition of the <Cancel> element is as follows:

```
<xs:element name="Cancel">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

#### 3.1.4.4.2.2 CancelResponse



<CancelResponse> is an XSD element that has no child elements. The XSD definition of the <CancelResponse> element is as follows:

```
<xs:element name="CancelResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.1.4.5 TransactedCancel

Following is the WSDL definition of the **TransactedCancel** operation:

```
<wsdl:operation name="TransactedCancel">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedCancel"
    message="tns:IWorkflowInstanceManagement_TransactedCancel_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedCancelResponse"
    message="tns:IWorkflowInstanceManagement_TransactedCancel_OutputMessage" />
</wsdl:operation>
```

**TransactedCancel** is an atomic operation that transitions the durable program instance from the active or suspended state to the completed state. The operation SHOULD gracefully cancel any remaining work and clean up resources being used by the durable program instance. This operation SHOULD be performed under the scope of a transaction flowed in from the client, if one is flowed in, using a protocol that is recognized by the client and server roles, such as [MS-WSRVCAT].

If the system maintains the durable state of the durable program instance, then the durable state MUST be updated during execution of this operation. If the durable store is a transactional resource manager, then the same transaction SHOULD be used for the durable state change. Failure to make the durable state change MUST result in failure of the operation.

A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed.

The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The server encounters an internal error while executing the **TransactedCancel** operation.

#### 3.1.4.5.1 Messages

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

Message	Description
IWorkflowInstanceManagement_TransactedCancel_InputMessage	Sent from the client to invoke the <b>TransactedCancel</b> operation.
IWorkflowInstanceManagement_TransactedCancel_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_TransactedCancel_InputMessage.

### 3.1.4.5.1.1 IWorkflowInstanceManagement\_TransactedCancel\_InputMessage

The IWorkflowInstanceManagement\_TransactedCancel\_InputMessage message is the request message for the **TransactedCancel** operation. The client role SHOULD send this message to invoke the **TransactedCancel** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedCancel_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedCancel" />
</wsdl:message>
```

**TransactedCancel:** The <TransactedCancel> element, as specified in section 3.1.4.5.2.1.

### 3.1.4.5.1.2 IWorkflowInstanceManagement\_TransactedCancel\_OutputMessage

The IWorkflowInstanceManagement\_TransactedCancel\_OutputMessage message is the reply message for the **TransactedCancel** operation. The message indicates that the **TransactedCancel** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedCancel_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedCancelResponse" />
</wsdl:message>
```

**TransactedCancelResponse:** The <TransactedCancelResponse> element, as specified in section 3.1.4.5.2.2.

### 3.1.4.5.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<TransactedCancel>	Contains the body of the IWorkflowInstanceManagement_TransactedCancel_InputMessage message.
<TransactedCancelResponse>	Contains the body of the IWorkflowInstanceManagement_TransactedCancel_OutputMessage message.

#### 3.1.4.5.2.1 TransactedCancel

<TransactedCancel> is an XSD element that has a child element <instanceId>. The XSD definition of the <TransactedCancel> element is as follows:

```
<xs:element name="TransactedCancel">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
</xs:complexType>
</xs:element>
```

**instanceId**: The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

### 3.1.4.5.2.2 TransactedCancelResponse

<TransactedCancelResponse> is an XSD element that has no child elements. The XSD definition of the <TransactedCancelResponse> element is as follows:

```
<xs:element name="TransactedCancelResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.1.4.6 Terminate

Following is the WSDL definition of the **Terminate** operation:

```
<wsdl:operation name="Terminate">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/Terminate"
    message="tns:IWorkflowInstanceManagement_Terminate_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TerminateResponse"
    message="tns:IWorkflowInstanceManagement_Terminate_OutputMessage" />
</wsdl:operation>
```

The **Terminate** operation transitions a durable program instance from the active or suspended state to the completed state. It SHOULD perform the minimal possible work needed to transition the durable program instance to the completed state. A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The server encounters an internal error while executing the Terminate operation.

#### 3.1.4.6.1 Messages

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

Message	Description
IWorkflowInstanceManagement_Terminate_InputMessage	Sent from the client to invoke the <b>Terminate</b> operation.
IWorkflowInstanceManagement_Terminate_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_Terminate_InputMessage.

### 3.1.4.6.1.1 IWorkflowInstanceManagement\_Terminate\_InputMessage

The IWorkflowInstanceManagement\_Terminate\_InputMessage message is the request message for the **Terminate** operation. The client SHOULD send this message to invoke the **Terminate** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_Terminate_InputMessage">
  <wsdl:part name="parameters" element="tns:Terminate" />
</wsdl:message>
```

**Terminate:** The <Terminate> element, as specified in section 3.1.4.6.2.1.

### 3.1.4.6.1.2 IWorkflowInstanceManagement\_Terminate\_OutputMessage

The IWorkflowInstanceManagement\_Terminate\_OutputMessage message is the reply message for the **Terminate** operation. The message indicates that the **Terminate** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_Terminate_OutputMessage">
  <wsdl:part name="parameters" element="tns:TerminateResponse" />
</wsdl:message>
```

**TerminateResponse:** The <TerminateResponse> element, as specified in section 3.1.4.6.2.2.

### 3.1.4.6.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<Terminate>	Contains the body of the IWorkflowInstanceManagement_Terminate_InputMessage message.
<TerminateResponse>	Contains the body of the IWorkflowInstanceManagement_Terminate_OutputMessage message.

### 3.1.4.6.2.1 Terminate

<Terminate> is an XSD element that has two child elements: <instanceId> and <reason>. The XSD definition of the <Terminate> element is as follows:

```
<xs:element name="Terminate">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
      <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
```

```
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

**reason:** The value of this element is a description of the reason for performing the **Terminate** operation.

### 3.1.4.6.2.2 TerminateResponse

<TerminateResponse> is an XSD element that has no child elements. The XSD definition of the <TerminateResponse> element is as follows:

```
<xs:element name="TerminateResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.1.4.7 TransactedTerminate

The WSDL definition of the **TransactedTerminate** operation is as follows:

```
<wsdl:operation name="TransactedTerminate">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedTerminate"
    message="tns:IWorkflowInstanceManagement_TransactedTerminate_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedTerminateResponse"
    message="tns:IWorkflowInstanceManagement_TransactedTerminate_OutputMessage" />
</wsdl:operation>
```

**TransactedTerminate** is an atomic operation that transitions a durable program instance from the active or suspended state to the completed state. It SHOULD perform the minimal possible work needed to transition the durable program instance to the completed state. This operation SHOULD be performed under the scope of a transaction flowed in from the client, if one is flowed in, using a protocol that is recognized by the client and server roles, such as [MS-WSRVCAT].

If the system maintains the durable state of the durable program instance, then the durable state MUST be updated during execution of this operation. If the durable store is a transactional resource manager, then the same transaction SHOULD be used for the durable state change. Failure to make the durable state change MUST result in failure of the operation.

A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed.

The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.

- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The server encounters an internal error while executing the **TransactedTerminate** operation.

### 3.1.4.7.1 Messages

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

Message	Description
IWorkflowInstanceManagement_TransactedTerminate_InputMessage	Sent from the client to invoke the <b>TransactedTerminate</b> operation.
IWorkflowInstanceManagement_TransactedTerminate_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_TransactedTerminate_InputMessage.

#### 3.1.4.7.1.1 IWorkflowInstanceManagement\_TransactedTerminate\_InputMessage

The IWorkflowInstanceManagement\_TransactedTerminate\_InputMessage message is the request message for the **TransactedTerminate** operation. The client SHOULD send this message to invoke the **TransactedTerminate** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedTerminate_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedTerminate" />
</wsdl:message>
```

**TransactedTerminate:** The <TransactedTerminate> element, as specified in section 3.1.4.7.2.1.

#### 3.1.4.7.1.2 IWorkflowInstanceManagement\_TransactedTerminate\_OutputMessage

The IWorkflowInstanceManagement\_TransactedTerminate\_OutputMessage message is the reply message for the **TransactedTerminate** operation. The message indicates that the **TransactedTerminate** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedTerminate_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedTerminateResponse" />
</wsdl:message>
```

**TransactedTerminateResponse:** The <TransactedTerminateResponse> element, as specified in section 3.1.4.7.2.2.

### 3.1.4.7.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<TransactedTerminate>	Contains the body of the IWorkflowInstanceManagement_TransactedTerminate_InputMessage message.
<TransactedTerminateResponse>	Contains the body of the IWorkflowInstanceManagement_TransactedTerminate_OutputMessage message.

### 3.1.4.7.2.1 TransactedTerminate

<TransactedTerminate> is an XSD element that has a child element <instanceId>. The XSD definition of the <TransactedTerminate> element is as follows:

```
<xs:element name="TransactedTerminate">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
      <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

**reason:** The value of this element is a description of the reason for performing the **TransactedTerminate** operation.

### 3.1.4.7.2.2 TransactedTerminateResponse

<TransactedTerminateResponse> is an XSD element that has no child elements. The XSD definition of the <TransactedTerminateResponse> element is as follows:

```
<xs:element name="TransactedTerminateResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.1.4.8 Suspend

The WSDL definition of the **Suspend** operation is as follows:

```
<wsdl:operation name="Suspend">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/Suspend"
    message="tns:IWorkflowInstanceManagement_Suspend_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/SuspendResponse"
    message="tns:IWorkflowInstanceManagement_Suspend_OutputMessage" />
</wsdl:operation>
```

The **Suspend** operation transitions a durable program instance from the active state to the suspended state. The durable program instance MUST NOT execute when in the suspended state.

The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.

- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The <reason> element is missing, empty, or has the xsi:nil attribute set to a value of true.
- The server encounters an internal error while executing the **Suspend** operation.

A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. If the durable program instance associated with the identifier passed to the **Suspend** operation is already in the suspended state, the state is not modified.

### 3.1.4.8.1 Messages

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

Message	Description
IWorkflowInstanceManagement_Suspend_InputMessage	Sent from the client to invoke the <b>Suspend</b> operation.
IWorkflowInstanceManagement_Suspend_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_Suspend_InputMessage.

#### 3.1.4.8.1.1 IWorkflowInstanceManagement\_Suspend\_InputMessage

The IWorkflowInstanceManagement\_Suspend\_InputMessage message is the request message for the **Suspend** operation. The client SHOULD send this message to invoke the **Suspend** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_Suspend_InputMessage">
  <wsdl:part name="parameters" element="tns:Suspend" />
</wsdl:message>
```

**Suspend:** The <Suspend> element, as specified in section 3.1.4.8.2.1.

#### 3.1.4.8.1.2 IWorkflowInstanceManagement\_Suspend\_OutputMessage

The IWorkflowInstanceManagement\_Suspend\_OutputMessage message is the reply message for the **Suspend** operation. The message indicates that the **Suspend** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_Suspend_OutputMessage">
  <wsdl:part name="parameters" element="tns:SuspendResponse" />
</wsdl:message>
```

**SuspendResponse:** The <SuspendResponse> element, as specified in section 3.1.4.8.2.2.

### 3.1.4.8.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<Suspend>	Contains the body of the IWorkflowInstanceManagement_Suspend_InputMessage message.



Element	Description
<SuspendResponse>	Contains the body of the IWorkflowInstanceManagement_Suspend_OutputMessage message.

### 3.1.4.8.2.1 Suspend

<Suspend> is an XSD element that has two child elements: <instanceId> and <reason>. The XSD definition of the <Suspend> element is as follows:

```
<xs:element name="Suspend">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
      <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

**reason:** The value of this element is a description of the reason for performing the **Suspend** operation.

### 3.1.4.8.2.2 SuspendResponse

<SuspendResponse> is an XSD element that has no child elements. The XSD definition of the <SuspendResponse> element is as follows:

```
<xs:element name="SuspendResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.1.4.9 TransactedSuspend

The WSDL definition of the **TransactedSuspend** operation is as follows:

```
<wsdl:operation name="TransactedSuspend">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedSuspend"
    message="tns:IWorkflowInstanceManagement_TransactedSuspend_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedSuspendResponse"
    message="tns:IWorkflowInstanceManagement_TransactedSuspend_OutputMessage" />
</wsdl:operation>
```

**TransactedSuspend** is an atomic operation that SHOULD perform the following tasks under the scope of a transaction flowed in from the client, if one is flowed in, using a protocol that is recognized by the client and server roles, such as [MS-WSRVCAT]:

- Transitions a durable program instance from the active state to the suspended state. If the durable program instance is already in the suspended state, then this task is not performed. The durable program instance **MUST NOT** execute when in the suspended state.
- The operation **SHOULD** return a SOAP fault message if one or more of the following conditions exist:
  - The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
  - The <instanceId> element is absent.
  - The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
  - The durable program instance associated with the value of the <instanceId> element is in the completed state.
  - The <reason> element is missing, empty, or has the xsi:nil attribute set to a value of true.
  - The server encounters an internal error while executing the **TransactedSuspend** operation.
- If the system maintains the durable state of the durable program instance, then the durable state **MUST** be updated during execution of this operation. If the durable store is a transactional resource manager, then the same transaction **SHOULD** be used for the durable state change. Failure to make the durable state change **MUST** result in failure of the operation.

A GUID **MUST** be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. If the durable program instance associated with the identifier passed to the **Suspend** operation is already in the suspended state, then the state is not modified.

### 3.1.4.9.1 Messages

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

Message	Description
IWorkflowInstanceManagement_TransactedSuspend_InputMessage	Sent from the client to invoke the <b>TransactedSuspend</b> operation.
IWorkflowInstanceManagement_TransactedSuspend_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_TransactedSuspend_InputMessage.

#### 3.1.4.9.1.1 IWorkflowInstanceManagement\_TransactedSuspend\_InputMessage

The IWorkflowInstanceManagement\_TransactedSuspend\_InputMessage message is the request message for the **TransactedSuspend** operation. The client **SHOULD** send this message to invoke the **TransactedSuspend** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedSuspend_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedSuspend" />
</wsdl:message>
```

**TransactedSuspend**: The <TransactedSuspend> element, as specified in section 3.1.4.9.2.1.

#### 3.1.4.9.1.2 IWorkflowInstanceManagement\_TransactedSuspend\_OutputMessage

The `IWorkflowInstanceManagement_TransactedSuspend_OutputMessage` message is the reply message for the **TransactedSuspend** operation. The message indicates that the **TransactedSuspend** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedSuspend_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedSuspendResponse" />
</wsdl:message>
```

**TransactedSuspendResponse:** The `<TransactedSuspendResponse>` element, as specified in section 3.1.4.9.2.2.

### 3.1.4.9.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<code>&lt;TransactedSuspend&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_TransactedSuspend_InputMessage</code> message.
<code>&lt;TransactedSuspendResponse&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_TransactedSuspend_OutputMessage</code> message.

#### 3.1.4.9.2.1 TransactedSuspend

`<TransactedSuspend>` is an XSD element that has two child elements: `<instanceId>` and `<reason>`. The XSD definition of the `<TransactedSuspend>` element is as follows:

```
<xs:element name="TransactedSuspend">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
      <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

**reason:** The value of this element is a description of the reason for performing the **TransactedSuspend** operation.

#### 3.1.4.9.2.2 TransactedSuspendResponse

`<TransactedSuspendResponse>` is an XSD element that has no child elements. The XSD definition of the `<TransactedSuspendResponse>` element is as follows:

```
<xs:element name="TransactedSuspendResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.1.4.10 Unsuspend

The WSDL definition of the **Unsuspend** operation is as follows:

```
<wsdl:operation name="Unsuspend">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/Unsuspend"
    message="tns:IWorkflowInstanceManagement_Unsuspend_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/UnsuspendResponse"
    message="tns:IWorkflowInstanceManagement_Unsuspend_OutputMessage" />
</wsdl:operation>
```

The **Unsuspend** operation transitions a durable program instance from the suspended state to the active state. A GUID **MUST** be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. The operation has no effect if the durable program instance associated with the provided identifier is already in the active state.

A GUID **MUST** be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. The operation **SHOULD** return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The server encounters an internal error while executing the **Unsuspend** operation.

#### 3.1.4.10.1 Messages

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

Message	Description
IWorkflowInstanceManagement_Unsuspend_InputMessage	Sent from the client to invoke the <b>Unsuspend</b> operation.
IWorkflowInstanceManagement_Unsuspend_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_Unsuspend_InputMessage.

##### 3.1.4.10.1.1 IWorkflowInstanceManagement\_Unsuspend\_InputMessage

The IWorkflowInstanceManagement\_Unsuspend\_InputMessage message is the request message for the **Unsuspend** operation. The client **SHOULD** send this message to invoke the **Unsuspend** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_Unsuspend_InputMessage">
  <wsdl:part name="parameters" element="tns:Unsuspend" />
</wsdl:message>
```

```
</wsdl:message>
```

**Unsuspend:** The <Unsuspend> element, as specified in section 3.1.4.10.2.1.

### 3.1.4.10.1.2 IWorkflowInstanceManagement\_Unsuspend\_OutputMessage

The IWorkflowInstanceManagement\_Unsuspend\_OutputMessage message is the reply message for the **Unsuspend** operation. The message indicates that the **Unsuspend** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_Unsuspend_OutputMessage">  
  <wsdl:part name="parameters" element="tns:UnsuspendResponse" />  
</wsdl:message>
```

**UnsuspendResponse:** The <UnsuspendResponse> element, as specified in section 3.1.4.10.2.2.

### 3.1.4.10.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<Unsuspend>	Contains the body of the IWorkflowInstanceManagement_Unsuspend_InputMessage message.
<UnsuspendResponse>	Contains the body of the IWorkflowInstanceManagement_Unsuspend_OutputMessage message.

#### 3.1.4.10.2.1 Unsuspend

<Unsuspend> is an XSD element that has a child element <instanceId>. The XSD definition of the <Unsuspend> element is as follows:

```
<xs:element name="Unsuspend">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element minOccurs="0" name="instanceId"  
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

#### 3.1.4.10.2.2 UnsuspendResponse

<UnsuspendResponse> is an XSD element that has no child elements. The XSD definition of the <UnsuspendResponse> element is as follows:

```
<xs:element name="UnsuspendResponse">  
  <xs:complexType>  
    <xs:sequence />  
  </xs:complexType>  
</xs:element>
```

### 3.1.4.11 TransactedUnsuspend

The WSDL definition of the **TransactedUnsuspend** operation is as follows:

```
<wsdl:operation name="TransactedUnsuspend">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedUnsuspend"
    message="tns:IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedUnsuspendResponse"
    message="tns:IWorkflowInstanceManagement_TransactedUnsuspend_OutputMessage" />
</wsdl:operation>
```

**TransactedUnsuspend** is an atomic operation that transitions a durable program instance from the suspended state to the active state. The operation SHOULD be performed under the scope of a transaction flowed in from the client, if one is flowed in, using a protocol that is recognized by the client and server roles, such as [MS-WSRVCAT].

If the system maintains the durable state of the durable program instance, then the durable state MUST be updated during execution of this operation. If the durable store is a transactional resource manager, then the same transaction SHOULD be used for the durable state change. Failure to make the durable state change MUST result in failure of the operation.

The durable program instance SHOULD start executing when in the active state. A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed.

The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
- The durable program instance associated with the value of the <instanceId> element is in the completed state.
- The server encounters an internal error while executing the **TransactedUnsuspend** operation.

#### 3.1.4.11.1 Messages

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

Message	Description
IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage	Sent from the client to invoke the <b>TransactedUnsuspend</b> operation.
IWorkflowInstanceManagement_TransactedUnsuspend_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage.

#### 3.1.4.11.1.1 IWorkflowInstanceManagement\_TransactedUnsuspend\_InputMessage

The `IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage` message is the request message for the **TransactedUnsuspend** operation. The client SHOULD send this message to invoke the **TransactedUnsuspend** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedUnsuspend" />
</wsdl:message>
```

**TransactedUnsuspend:** The `<TransactedUnsuspend>` element, as specified in section 3.1.4.11.2.1.

### 3.1.4.11.1.2 IWorkflowInstanceManagement\_TransactedUnsuspend\_OutputMessage

The `IWorkflowInstanceManagement_TransactedUnsuspend_OutputMessage` message is the reply message for the **TransactedUnsuspend** operation. The message indicates that the **TransactedUnsuspend** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedUnsuspend_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedUnsuspendResponse" />
</wsdl:message>
```

#### 3.1.4.11.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<code>&lt;TransactedUnsuspend&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage</code> message.
<code>&lt;TransactedUnsuspendResponse&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_TransactedUnsuspend_OutputMessage</code> message.

#### 3.1.4.11.2.1 TransactedUnsuspend

`<TransactedUnsuspend>` is an XSD element that has a child element `<instanceId>`. The XSD definition of the `<TransactedUnsuspend>` element is as follows:

```
<xs:element name="TransactedUnsuspend">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
        xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

#### 3.1.4.11.2.2 TransactedUnsuspendResponse

`<TransactedUnsuspendResponse>` is an XSD element that has no child elements. The XSD definition of the `<TransactedUnsuspendResponse>` element is as follows:

```

<xs:element name="TransactedUnsuspendResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>

```

### 3.1.4.12 Update

The WSDL definition of the **Update** operation is as follows:

```

<wsdl:operation name="Update">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/Update"
    message="tns:IWorkflowInstanceManagement_Update_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/UpdateResponse"
    message="tns:IWorkflowInstanceManagement_Update_OutputMessage" />
</wsdl:operation>

```

The **Update** operation SHOULD provide the durable program instance with the opportunity to update its identity. A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.

#### 3.1.4.12.1 Messages

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

Message	Description
IWorkflowInstanceManagement_Update_InputMessage	Sent from the client to invoke the <b>Update</b> operation.
IWorkflowInstanceManagement_Update_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_Update_InputMessage.

##### 3.1.4.12.1.1 IWorkflowInstanceManagement\_Update\_InputMessage

The IWorkflowInstanceManagement\_Update\_InputMessage message is the request message for the **Update** operation. The client SHOULD send this message to invoke the **Update** operation.

```

<wsdl:message name="IWorkflowInstanceManagement_Update_InputMessage">
  <wsdl:part name="parameters" element="tns:Update" />
</wsdl:message>

```

**Update:** The <Update> element, as specified in section 3.1.4.12.2.1.

##### 3.1.4.12.1.2 IWorkflowInstanceManagement\_Update\_OutputMessage



The `IWorkflowInstanceManagement_Update_OutputMessage` message is the reply message for the **Update** operation. This message indicates that the **Update** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_Update_OutputMessage">
  <wsdl:part name="parameters" element="tns:UpdateResponse" />
</wsdl:message>
```

**UpdateResponse:** The `<UpdateResponse>` element, as specified in section 3.1.4.12.2.2.

### 3.1.4.12.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<code>&lt;Update&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_Update_InputMessage</code> message.
<code>&lt;UpdateResponse&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_Update_OutputMessage</code> message.

#### 3.1.4.12.2.1 Update

`<Update>` is an XSD element that has two child elements, `<instanceId>` and `<updateDefinitionIdentity>`. The XSD definition of the `<Update>` element is as follows:

```
<xs:element name="Update">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId" type="q3:guid"
        xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xs:element minOccurs="0" name="updatedDefinitionIdentity" nillable="true"
        type="q4:WorkflowIdentity"
        xmlns:q4="http://schemas.datacontract.org/2004/07/System.Activities" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

**updateDefinitionIdentity:** The value of this element is of type `WorkflowIdentity` and SHOULD match the identity of the durable program instance on which the `<Update>` operation SHOULD be performed.

#### 3.1.4.12.2.2 UpdateResponse

`<UpdateResponse>` is an XSD element that has no child elements. The XSD definition of the `<UpdateResponse>` element is as follows:

```
<xs:element name="UpdateResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
```

### 3.1.4.13 TransactedUpdate

The WSDL definition of the **TransactedUpdate** operation is as follows:

```
<wsdl:operation name="TransactedUpdate">
  <wsdl:input wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedUpdate"
    message="tns:IWorkflowInstanceManagement_TransactedUpdate_InputMessage" />
  <wsdl:output wsaw:Action="http://schemas.datacontract.org/2008/10/
    WorkflowServices/IWorkflowInstanceManagement/TransactedUpdateResponse"
    message="tns:IWorkflowInstanceManagement_TransactedUpdate_OutputMessage" />
</wsdl:operation>
```

**TransactedUpdate** is an atomic operation that SHOULD update the identity of the durable program instance. The operation SHOULD be performed under the scope of a transaction flowed in from the client, if one is flowed in, using a protocol that is recognized by the client and server, such as the one specified in [MS-WSRVCAT]. A GUID MUST be passed to the operation as the value of the <instanceId> element to identify the durable program instance on which the operation is to be performed. The operation SHOULD return a SOAP fault message if one or more of the following conditions exist:

- The value of the <instanceId> element is not in the correct format, as specified in [MS-DTYP] section 2.3.4.
- The <instanceId> element is absent.
- The value of the <instanceId> element does not exist in the **Durable Program Instance Table** on the server.
- The server encounters an internal error while executing the **TransactedUpdate** operation.

#### 3.1.4.13.1 Messages

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

Message	Description
IWorkflowInstanceManagement_TransactedUpdate_InputMessage	Sent from the client to invoke the <b>TransactedUpdate</b> operation.
IWorkflowInstanceManagement_TransactedUpdate_OutputMessage	Sent from the server as a reply to the IWorkflowInstanceManagement_TransactedUpdate_InputMessage message.

##### 3.1.4.13.1.1 IWorkflowInstanceManagement\_TransactedUpdate\_InputMessage

The IWorkflowInstanceManagement\_TransactedUpdate\_InputMessage message is the request message for the **TransactedUpdate** operation. The client SHOULD send this message to invoke the **TransactedUpdate** operation.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedUpdate_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedUpdate" />
</wsdl:message>
```

**TransactedUpdate:** The <TransactedUpdate> element, as specified in section 3.1.4.13.2.1.

##### 3.1.4.13.1.2 IWorkflowInstanceManagement\_TransactedUpdate\_OutputMessage

The `IWorkflowInstanceManagement_TransactedUpdate_OutputMessage` message is the reply message for the **TransactedUpdate** operation. This message indicates that the **TransactedUpdate** operation has successfully completed.

```
<wsdl:message name="IWorkflowInstanceManagement_TransactedUpdate_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedUpdateResponse" />
</wsdl:message>
```

**TransactedUpdateResponse:** The `<TransactedUpdateResponse>` element, as specified in section 3.1.4.13.2.2.

### 3.1.4.13.2 Elements

The following table summarizes the XSD element definitions that are specific to this operation.

Element	Description
<code>&lt;TransactedUpdate&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_TransactedUpdate_InputMessage</code> message.
<code>&lt;TransactedUpdateResponse&gt;</code>	Contains the body of the <code>IWorkflowInstanceManagement_TransactedUpdate_OutputMessage</code> message.

#### 3.1.4.13.2.1 TransactedUpdate

`<TransactedUpdate>` is an XSD element that has two child elements `<instanceId>` and `<updateDefinitionIdentity>`. The XSD definition of the `<TransactedUpdate>` element is as follows:

```
<xs:element name="TransactedUpdate">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId" type="q1:guid"
        xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xs:element minOccurs="0" name="updatedDefinitionIdentity" nillable="true"
        type="q2:WorkflowIdentity"
        xmlns:q2="http://schemas.datacontract.org/2004/07/System.Activities" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**instanceId:** The value of this element is of type GUID and SHOULD match the identifier that is associated with the durable program instance in the **Durable Program Instance Table** on which this operation SHOULD be performed.

**updateDefinitionIdentity:** The value of this element is of type `WorkflowIdentity` and SHOULD match the identity of the durable program instance on which the `<TransactedUpdate>` operation SHOULD be performed.

#### 3.1.4.13.2.2 TransactedUpdateResponse

`<TransactedUpdateResponse>` is an XSD element that has no child elements. The XSD definition of the `<TransactedUpdateResponse>` element is as follows:

```
<xs:element name="TransactedUpdateResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
```

</xs:element>

### **3.1.5 Timer Events**

None.

### **3.1.6 Other Local Events**

None.

## **3.2 IWorkflowInstanceManagement Client Details**

The client side of this protocol is simply a pass-through mechanism. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

## 4 Protocol Examples

The following is an example message exchange using the Workflow Instance Management Protocol to suspend a durable program instance.

A SOAP request message is sent from the client to the server:

```
<s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagement/Suspend</a:Action>
    <a:MessageID>urn:uuid:8afb36d3-9a6e-47df-9313-f005242ea3ed</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To
s:mustUnderstand="1">net.pipe://localhost/workflowControlServiceEndpoint/2308/c50fb3bb-6c52-43b3-af57-8acb43a487b7</a:To>
  </s:Header>
  <s:Body>
    <Suspend xmlns="http://schemas.datacontract.org/2008/10/WorkflowServices">
      <instanceId>349be129-fb36-49e5-abb8-76b9831fc7b6</instanceId>
      <reason>
        Suspend the instance
      </reason>
    </Suspend>
  </s:Body>
</s:Envelope>
```

A SOAP response message is sent from the server to the client after successfully processing the request:

```
<s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagement/SuspendResponse</a:Action>
    <a:RelatesTo>urn:uuid:89a7d122-208f-443b-8f16-44bfe7fb684e</a:RelatesTo>
  </s:Header>
  <s:Body>
    <SuspendResponse xmlns="http://schemas.datacontract.org/2008/10/WorkflowServices" />
  </s:Body>
</s:Envelope>
```

## 5 Security

### 5.1 Security Considerations for Implementers

The ~~Secure the~~ Workflow Instance Management Protocol ~~should be secured by~~ using the security mechanisms provided by the underlying layers including WS-\* security mechanisms, such as [WSS1] and those provided by the transport, such as HTTPS.

### 5.2 Index of Security Parameters

None.

## 6 Appendix A: Full WSDL

WSDL or schema name	Prefix	Section
Workflow Instance Management Protocol WSDL	wsdl:	Section 6.1
Workflow Instance Management Schema for the WSDL	xs:	Section 6.2
Workflow Identity Management Schema for the WSDL	xs:	Section 6.3

For ease of implementation the full WSDL with schemas are provided in the following sections.

### 6.1 Workflow Instance Management Protocol WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tns="http://schemas.datacontract.org/2008/10/WorkflowServices"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
targetNamespace="http://schemas.datacontract.org/2008/10/WorkflowServices"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xsd:schema
targetNamespace="http://schemas.datacontract.org/2008/10/WorkflowServices/Imports">
      <xsd:import namespace="http://schemas.datacontract.org/2008/10/WorkflowServices" />
      <xsd:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage">
    <wsdl:part name="parameters" element="tns:TransactedUnsuspend" />
  </wsdl:message>
  <wsdl:message name="IWorkflowInstanceManagement_TransactedUnsuspend_OutputMessage">
    <wsdl:part name="parameters" element="tns:TransactedUnsuspendResponse" />
  </wsdl:message>
  <wsdl:message name="IWorkflowInstanceManagement_Abandon_InputMessage">
    <wsdl:part name="parameters" element="tns:Abandon" />
  </wsdl:message>
  <wsdl:message name="IWorkflowInstanceManagement_Abandon_OutputMessage">
    <wsdl:part name="parameters" element="tns:AbandonResponse" />
  </wsdl:message>
  <wsdl:message name="IWorkflowInstanceManagement_Cancel_InputMessage">
    <wsdl:part name="parameters" element="tns:Cancel" />
  </wsdl:message>
  <wsdl:message name="IWorkflowInstanceManagement_Cancel_OutputMessage">
    <wsdl:part name="parameters" element="tns:CancelResponse" />
  </wsdl:message>
  <wsdl:message name="IWorkflowInstanceManagement_Run_InputMessage">
    <wsdl:part name="parameters" element="tns:Run" />
  </wsdl:message>
  <wsdl:message name="IWorkflowInstanceManagement_Run_OutputMessage">
    <wsdl:part name="parameters" element="tns:RunResponse" />
  </wsdl:message>
  <wsdl:message name="IWorkflowInstanceManagement_Suspend_InputMessage">
    <wsdl:part name="parameters" element="tns:Suspend" />
  </wsdl:message>
  <wsdl:message name="IWorkflowInstanceManagement_Suspend_OutputMessage">
```

```

    <wsdl:part name="parameters" element="tns:SuspendResponse" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_Terminate_InputMessage">
  <wsdl:part name="parameters" element="tns:Terminate" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_Terminate_OutputMessage">
  <wsdl:part name="parameters" element="tns:TerminateResponse" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_Unsuspend_InputMessage">
  <wsdl:part name="parameters" element="tns:Unsuspend" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_Unsuspend_OutputMessage">
  <wsdl:part name="parameters" element="tns:UnsuspendResponse" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedCancel_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedCancel" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedCancel_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedCancelResponse" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedRun_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedRun" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedRun_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedRunResponse" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedSuspend_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedSuspend" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedSuspend_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedSuspendResponse" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedTerminate_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedTerminate" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedTerminate_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedTerminateResponse" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedUpdate_InputMessage">
  <wsdl:part name="parameters" element="tns:TransactedUpdate" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_TransactedUpdate_OutputMessage">
  <wsdl:part name="parameters" element="tns:TransactedUpdateResponse" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_Update_InputMessage">
  <wsdl:part name="parameters" element="tns:Update" />
</wsdl:message>
<wsdl:message name="IWorkflowInstanceManagement_Update_OutputMessage">
  <wsdl:part name="parameters" element="tns:UpdateResponse" />
</wsdl:message>
<wsdl:portType name="IWorkflowInstanceManagement">
  <wsdl:operation name="TransactedUnsuspend">
    <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagement/TransactedUnsuspend"
message="tns:IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage" />
    <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagement/TransactedUnsuspendResponse"
message="tns:IWorkflowInstanceManagement_TransactedUnsuspend_OutputMessage" />
    </wsdl:operation>
  <wsdl:operation name="Abandon">
    <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagement/Abandon" message="tns:IWorkflowInstanceManagement_Abandon_InputMessage" />
    <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagement/AbandonResponse" message="tns:IWorkflowInstanceManagement_Abandon_OutputMessage" />
    </wsdl:operation>
  <wsdl:operation name="Cancel">

```



```

        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/Cancel" message="tns:IWorkflowInstanceManagement_Cancel_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/CancelResponse" message="tns:IWorkflowInstanceManagement_Cancel_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="Run">
        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/Run" message="tns:IWorkflowInstanceManagement_Run_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/RunResponse" message="tns:IWorkflowInstanceManagement_Run_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="Suspend">
        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/Suspend" message="tns:IWorkflowInstanceManagement_Suspend_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/SuspendResponse" message="tns:IWorkflowInstanceManagement_Suspend_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="Terminate">
        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/Terminate" message="tns:IWorkflowInstanceManagement_Terminate_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TerminateResponse" message="tns:IWorkflowInstanceManagement_Terminate_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="Unsuspend">
        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/Unsuspend" message="tns:IWorkflowInstanceManagement_Unsuspend_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/UnsuspendResponse" message="tns:IWorkflowInstanceManagement_Unsuspend_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="TransactedCancel">
        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedCancel"
message="tns:IWorkflowInstanceManagement_TransactedCancel_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedCancelResponse"
message="tns:IWorkflowInstanceManagement_TransactedCancel_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="TransactedRun">
        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedRun" message="tns:IWorkflowInstanceManagement_TransactedRun_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedRunResponse"
message="tns:IWorkflowInstanceManagement_TransactedRun_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="TransactedSuspend">
        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedSuspend"
message="tns:IWorkflowInstanceManagement_TransactedSuspend_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedSuspendResponse"
message="tns:IWorkflowInstanceManagement_TransactedSuspend_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="TransactedTerminate">

```

```

        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedTerminate"
message="tns:IWorkflowInstanceManagement_TransactedTerminate_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedTerminateResponse"
message="tns:IWorkflowInstanceManagement_TransactedTerminate_OutputMessage" />
    </wsdl:operation>
    <wsdl:operation name="Update">
        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/Update" message="tns:IWorkflowInstanceManagement_Update_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/UpdateResponse" message="tns:IWorkflowInstanceManagement_Update_OutputMessage" />
    </wsdl:operation>
    <wsdl:operation name="TransactedUpdate">
        <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedUpdate"
message="tns:IWorkflowInstanceManagement_TransactedUpdate_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedUpdateResponse"
message="tns:IWorkflowInstanceManagement_TransactedUpdate_OutputMessage" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="DefaultBinding_IWorkflowInstanceManagement"
type="tns:IWorkflowInstanceManagement">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="TransactedUnsuspend">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/TransactedUnsuspend" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Abandon">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/Abandon" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Cancel">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/Cancel" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Run">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/Run" style="document" />
        <wsdl:input>
            <soap:body use="literal" />

```

```

        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Suspend">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/Suspend" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
    <wsdl:operation name="Terminate">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/Terminate" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
    <wsdl:operation name="Unsuspend">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/Unsuspend" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
    <wsdl:operation name="TransactedCancel">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/TransactedCancel" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
    <wsdl:operation name="TransactedRun">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/TransactedRun" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
    <wsdl:operation name="TransactedSuspend">
        <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/TransactedSuspend" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:output>

```

```

    </wsdl:operation>
    <wsdl:operation name="TransactedTerminate">
      <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/TransactedTerminate" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="TransactedUpdate">
      <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/TransactedUpdate" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Update">
      <soap:operation
soapAction="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManagem
ent/Update" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>

```

## 6.2 Workflow Instance Management Schema for the WSDL

```

<?xml version="1.0" encoding="utf-8"?>
  <xs:schema xmlns:tns="http://schemas.datacontract.org/2008/10/WorkflowServices"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2008/10/WorkflowServices"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
    <xs:element name="TransactedUnsuspend">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="instanceId"
xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/" type="q1:guid" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="TransactedUnsuspendResponse">
      <xs:complexType>
        <xs:sequence />
      </xs:complexType>
    </xs:element>
    <xs:element name="Abandon">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="instanceId"
xmlns:q2="http://schemas.microsoft.com/2003/10/Serialization/" type="q2:guid" />
          <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="AbandonResponse">

```

```

    <xs:complexType>
      <xs:sequence />
    </xs:complexType>
  </xs:element>
  <xs:element name="Cancel">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="instanceId"
xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/" type="q3:guid" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CancelResponse">
    <xs:complexType>
      <xs:sequence />
    </xs:complexType>
  </xs:element>
  <xs:element name="Run">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="instanceId"
xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/" type="q4:guid" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="RunResponse">
    <xs:complexType>
      <xs:sequence />
    </xs:complexType>
  </xs:element>
  <xs:element name="Suspend">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="instanceId"
xmlns:q5="http://schemas.microsoft.com/2003/10/Serialization/" type="q5:guid" />
        <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SuspendResponse">
    <xs:complexType>
      <xs:sequence />
    </xs:complexType>
  </xs:element>
  <xs:element name="Terminate">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="instanceId"
xmlns:q6="http://schemas.microsoft.com/2003/10/Serialization/" type="q6:guid" />
        <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="TerminateResponse">
    <xs:complexType>
      <xs:sequence />
    </xs:complexType>
  </xs:element>
  <xs:element name="Unsuspend">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="instanceId"
xmlns:q7="http://schemas.microsoft.com/2003/10/Serialization/" type="q7:guid" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="UnsuspendResponse">
    <xs:complexType>
      <xs:sequence />
    </xs:complexType>
  </xs:element>

```

```

</xs:element>
<xs:element name="TransactedCancel">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
xmlns:q8="http://schemas.microsoft.com/2003/10/Serialization/" type="q8:guid" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TransactedCancelResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="TransactedRun">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
xmlns:q9="http://schemas.microsoft.com/2003/10/Serialization/" type="q9:guid" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TransactedRunResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="TransactedSuspend">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
xmlns:q10="http://schemas.microsoft.com/2003/10/Serialization/" type="q10:guid" />
      <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TransactedSuspendResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="TransactedTerminate">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId"
xmlns:q11="http://schemas.microsoft.com/2003/10/Serialization/" type="q11:guid" />
      <xs:element minOccurs="0" name="reason" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TransactedTerminateResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
<xs:element name="TransactedUpdate">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId" type="q1:guid"
xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xs:element minOccurs="0" name="updatedDefinitionIdentity" nillable="true"
type="q2:WorkflowIdentity"
xmlns:q2="http://schemas.datacontract.org/2004/07/System.Activities" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TransactedUpdateResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>

```

```

</xs:element>
<xs:element name="Update">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="instanceId" type="q3:guid"
xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xs:element minOccurs="0" name="updatedDefinitionIdentity" nillable="true"
type="q4:WorkflowIdentity"
xmlns:q4="http://schemas.datacontract.org/2004/07/System.Activities" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="UpdateResponse">
  <xs:complexType>
    <xs:sequence />
  </xs:complexType>
</xs:element>
</xs:schema>

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="anyType" nillable="true" type="xs:anyType" />
  <xs:element name="anyURI" nillable="true" type="xs:anyURI" />
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary" />
  <xs:element name="boolean" nillable="true" type="xs:boolean" />
  <xs:element name="byte" nillable="true" type="xs:byte" />
  <xs:element name="dateTime" nillable="true" type="xs:dateTime" />
  <xs:element name="decimal" nillable="true" type="xs:decimal" />
  <xs:element name="double" nillable="true" type="xs:double" />
  <xs:element name="float" nillable="true" type="xs:float" />
  <xs:element name="int" nillable="true" type="xs:int" />
  <xs:element name="long" nillable="true" type="xs:long" />
  <xs:element name="QName" nillable="true" type="xs:QName" />
  <xs:element name="short" nillable="true" type="xs:short" />
  <xs:element name="string" nillable="true" type="xs:string" />
  <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte" />
  <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt" />
  <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong" />
  <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort" />
  <xs:element name="char" nillable="true" type="tns:char" />
  <xs:simpleType name="char">
    <xs:restriction base="xs:int" />
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns:duration" />
  <xs:simpleType name="duration">
    <xs:restriction base="xs:duration">
      <xs:pattern value="-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?" />
      <xs:minInclusive value="-P10675199DT2H48M5.4775808S" />
      <xs:maxInclusive value="P10675199DT2H48M5.4775807S" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="guid" nillable="true" type="tns:guid" />
  <xs:simpleType name="guid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}" />
    </xs:restriction>
  </xs:simpleType>
  <xs:attribute name="FactoryType" type="xs:QName" />
  <xs:attribute name="Id" type="xs:ID" />
  <xs:attribute name="Ref" type="xs:IDREF" />
</xs:schema>

```

### 6.3 Workflow Identity Management Schema for the WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="http://schemas.datacontract.org/2004/07/System.Activities"
targetNamespace="http://schemas.datacontract.org/2004/07/System.Activities"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:complexType name="WorkflowIdentity">
    <xs:sequence>
      <xs:element name="name" type="xs:string" minOccurs="0" nillable="true">
        <xs:annotation>
          <xs:appinfo>
            <DefaultValue xmlns="http://schemas.microsoft.com/2003/10/Serialization/"
EmitDefaultValue="false"/>
          </xs:appinfo>
        </xs:annotation>
      </xs:element>
      <xs:element name="package" type="xs:string" minOccurs="0" nillable="true">
        <xs:annotation>
          <xs:appinfo>
            <DefaultValue xmlns="http://schemas.microsoft.com/2003/10/Serialization/"
EmitDefaultValue="false"/>
          </xs:appinfo>
        </xs:annotation>
      </xs:element>
      <xs:element name="version" type="xs:string" minOccurs="0" nillable="true">
        <xs:annotation>
          <xs:appinfo>
            <DefaultValue xmlns="http://schemas.microsoft.com/2003/10/Serialization/"
EmitDefaultValue="false"/>
          </xs:appinfo>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="WorkflowIdentity" type="tns:WorkflowIdentity" nillable="true"/>
</xs:schema>
```



## 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 released service packs.

This document specifies version-specific details in the Microsoft .NET Framework. For information about which versions of .NET Framework are available in each released Windows product or as supplemental software, see [MS-NETOD] section 4.

- Microsoft .NET Framework 4.0
- Microsoft .NET Framework 4.5
- Microsoft .NET Framework 4.6

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

<1> Section 3.1.1.3: The .NET Framework 4.0 implementation of the Workflow Instance Management Protocol includes features that interact with durable program instances in the system and cause the following changes to their state:

- **Persistence:** The persistence of the complete state of a durable program instance to a persistence store, thus causing the creation of a "durable instance" which can later be restored in memory.
- **Unhandled Exception behavior:** In the case of an unhandled exception from a durable program instance, a preconfigured set of actions can be performed on the in-memory, or durable, durable program instance. For example, the user can configure the system to cause the errant durable program instance to transition to the suspended state.
- **Idle behavior:** The persistence of durable program instances that are blocked on some stimuli after a user-configured duration of time, and eventually causing the unloading of these durable program instances from memory after a user-configured duration of time.

These features result in the following consequences for the .NET Framework 4.0 implementation of the Workflow Instance Management Protocol:

- The **Abandon** operation disposes the in-memory durable program instance. If the Persistence feature is enabled and a persistence record exists for the durable program instance, then the durable program instance can be reloaded from the persistence store and execution can be continued from that point. The state of the reloaded durable program instance will be the state that was stored in the persisted record for the Instance. If no persistence record exists for the durable program instance, then the durable program instance is effectively transitioned to the final state.
- The **Run** and **TransactedRun** operations load the durable program instance from the persistence store if not already in memory, the Persistence feature is enabled, and a persistence record for the durable program instance exists in the store. These two operations have no effect if the durable program instance is already in memory.
- The **TransactedSuspend**, **TransactedCancel**, **TransactedTerminate**, and **TransactedUnsuspend** operations persist the durable program instance if the Persistence feature is enabled. The **Suspend**, **Cancel**, **Terminate**, and **Unsuspend** operations do not persist the

durable program instance, and therefore, the durable state will not be up-to-date after these non-transacted operations. As a result, a sequence of commands, such as **Suspend, Abandon, Run**, might result in the in-memory durable program instance being in a different state as compared with a sequence of commands, such as **TransactedSuspend, Abandon, Run**, since the **Abandon** operation will remove the in-memory instance and the **Run** operation will reload the durable instance from the last persisted record.

**Note** In Windows Server 2008 operating system, Windows Server 2008 R2 operating system, Windows Server 2012 operating system, Windows Server 2012 R2 operating system, and Windows Server 2016 ~~Technical Preview~~ operating system, .NET Framework 4.0 is not supported in the Server Core Role.

<2> Section 3.1.4.2: The .NET Framework 4.0 implementation supports the WS-AtomicTransaction (WS-AT) Version 1.0 Protocol Extensions [MS-WSRVCAT] and the MSDTC Connection Manager: OleTx Transaction Protocol [MS-DTCO] for flowing transactions using the **TransactedRun, TransactedSuspend, TransactedUnsuspend, TransactedCancel, and TransactedTerminate** operations. If no transaction is flowed in, a local transaction is created to provide atomic semantics.

**Note:** In Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016 ~~Technical Preview~~, .NET Framework 4.0 is not supported in the Server Core Role.

## 8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

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

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

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

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

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

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

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

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
6 Appendix A: Full WSDL	72775 : Added a new sub-section 6.3 to the table, and updated title for section 6.2.	Y	Content update.
6.1 Workflow Instance Management Protocol WSDL	72775 : Updated the WSDL.	Y	Content update.
6.2 Workflow Instance Management Schema for the WSDL	72775 : Updated the section title.	N	Content update.
6.3 Workflow Identity Management Schema for the WSDL	72775 : Added new section.	Y	New content added.

## 9 Index

### A

- Abstract data model
  - server 14
  - server - IWorkflowInstanceManagement
    - active state 14
    - completed state 15
    - overview 14
    - suspended state 15
- Applicability 10
- Attribute groups 13
- Attributes 13

### C

- Capability negotiation 10
- Change tracking 59
- Client - IWorkflowInstanceManagement (section 3 14, section 3.2 44)
- Complex types 13

### D

- Data model - abstract
  - server 14
  - server - IWorkflowInstanceManagement
    - active state 14
    - completed state 15
    - overview 14
    - suspended state 15

### E

- Events
  - local - server 44
  - local - server - IWorkflowInstanceManagement 44
  - timer - server 44
  - timer - server - IWorkflowInstanceManagement 44
- Examples - overview 45

### F

- Fields - vendor-extensible 10
- Full WSDL 47
  - overview 47
  - Workflow Identity Management Schema for the WSDL 56
  - Workflow Instance Management Protocol WSDL 47
  - Workflow Instance Management Schema for the WSDL 52

### G

- Glossary 7
- Groups 13

### I

- Implementer - security considerations 46
- Index of security parameters 46
- Informative references 9
- Initialization
  - server 15
- Initialization - server - IWorkflowInstanceManagement 15

- Introduction 7
- IWorkflowInstanceManagement
  - client 44
  - server
    - Abandon operation 21
    - abstract data model
      - active state 14
      - completed state 15
      - overview 14
      - suspended state 15
    - Cancel operation 23
    - initialization 15
    - local events 44
    - message processing 16
    - Run operation 17
    - sequencing rules 16
    - Suspend operation 31
    - Terminate operation 27
    - timer events 44
    - timers 15
    - TransactedCancel operation 25
    - TransactedRun operation 19
    - TransactedSuspend operation 33
    - TransactedTerminate operation 29
    - TransactedUnsuspend operation 38
    - TransactedUpdate operation 42
    - Unsuspend operation 36
    - Update operation 40

## L

- Local events
  - server 44
- Local events - server - IWorkflowInstanceManagement 44

## M

- Message processing
  - server 16
- Message processing - server - IWorkflowInstanceManagement 16
- Messages
  - attribute groups 13
  - attributes 13
  - complex types 13
  - elements 13
  - enumerated 13
  - groups 13
  - namespaces 12
  - simple types 13
  - syntax 12
  - transport 12

## N

- Namespaces 12
- Normative references 8

## O

- Operations
  - Abandon 21
  - Cancel 23
  - Run 17
  - Suspend 31
  - Terminate 27

- TransactedCancel 25
- TransactedRun 19
- TransactedSuspend 33
- TransactedTerminate 29
- TransactedUnsuspend 38
- TransactedUpdate 42
- Unsuspend 36
- Update 40
- Overview (synopsis) 9

## **P**

- Parameter index - security 46
- Parameters - security index 46
- Preconditions 10
- Prerequisites 10
- Product behavior 57
- Protocol Details
  - overview 14

## **R**

- References 8
  - informative 9
  - normative 8
- Relationship to other protocols 10

## **S**

- Security
  - implementer considerations 46
  - parameter index 46
- Sequencing rules
  - server 16
- Sequencing rules - server - IWorkflowInstanceManagement 16
- Server
  - Abandon operation 21
  - abstract data model 14
  - Cancel operation 23
  - initialization 15
  - local events 44
  - message processing 16
  - Run operation 17
  - sequencing rules 16
  - Suspend operation 31
  - Terminate operation 27
  - timer events 44
  - timers 15
  - TransactedCancel operation 25
  - TransactedRun operation 19
  - TransactedSuspend operation 33
  - TransactedTerminate operation 29
  - TransactedUnsuspend operation 38
  - TransactedUpdate operation 42
  - Unsuspend operation 36
  - Update operation 40
- Server - IWorkflowInstanceManagement
  - Abandon operation 21
  - abstract data model
    - active state 14
    - completed state 15
    - overview 14
    - suspended state 15
  - Cancel operation 23
  - initialization 15

- local events 44
- message processing 16
- Run operation 17
- sequencing rules 16
- Suspend operation 31
- Terminate operation 27
- timer events 44
- timers 15
- TransactedCancel operation 25
- TransactedRun operation 19
- TransactedSuspend operation 33
- TransactedTerminate operation 29
- TransactedUnsuspend operation 38
- TransactedUpdate operation 42
- Unsuspend operation 36
- Update operation 40
- Simple types 13
- Standards assignments 11
- Syntax
  - attribute groups 13
  - attributes 13
  - complex types 13
  - elements 13
  - groups 13
  - message definitions 13
  - messages - overview 12
  - namespaces 12
  - overview 12
  - simple types 13

## **T**

- Timer events
  - server 44
- Timer events - server - IWorkflowInstanceManagement 44
- Timers
  - server 15
- Timers - server - IWorkflowInstanceManagement 15
- Tracking changes 59
- Transport 12
- Types
  - complex 13
  - simple 13

## **V**

- Vendor-extensible fields 10
- Versioning 10

## **W**

- WSDL 47
  - overview 47
  - Workflow Identity Management Schema for the WSDL 56
  - Workflow Instance Management Protocol WSDL 47
  - Workflow Instance Management Schema for the WSDL 52