[MS-WFIM]:
Workflow Instance Management Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
09/25/2009	0.1	Major	First Release.
11/06/2009	0.1.1	Editorial	Revised and edited the technical content.
12/18/2009	0.1.2	Editorial	Revised and edited the technical content.
01/29/2010	0.2	Minor	Updated the technical content.
03/12/2010	0.2.1	Editorial	Revised and edited the technical content.
04/23/2010	0.3	Minor	Updated the technical content.
06/04/2010	0.3.1	Editorial	Revised and edited the technical content.
07/16/2010	1.0	Major	Significantly changed the technical content.
08/27/2010	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	2.0	Major	Significantly changed the technical content.
02/11/2011	2.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

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.

1.1 Glossary

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

globally unique identifier (GUID)
.NET Framework
SOAP fault
SOAP message
Web Services Description Language (WSDL)
WSDL message
WSDL operation
WSDL port type
XML namespace
XML schema (XSD)

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.

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

SOAP: Either the Simple Object Access Protocol (SOAP) 1.1 [SOAP1.1] or SOAP 1.2 [SOAP1.2-1/2007]. This term is used in cases where the difference between the two SOAP version specifications has no impact on the specification of the Workflow Instance Management Protocol.

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

1.2 References

1.2.1 Normative References

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

[MS-DTCO] Microsoft Corporation, "MSDTC Connection Manager: OleTx Transaction Protocol Specification", July 2007.

[MS-DTYP] Microsoft Corporation, "Windows Data Types", January 2007.

[MS-WSPOL] Microsoft Corporation, "Web Services: Policy Assertions and WSDL Extensions", September 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.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.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

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

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-WSRVCAT] Microsoft Corporation, "WS-AtomicTransaction (WS-AT) Version 1.0 Protocol Extensions", April 2008.

[PROCESS] Microsoft Corporation, "About Processes and Threads", http://msdn.microsoft.com/en-us/library/ms681917.aspx

[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 internals 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:

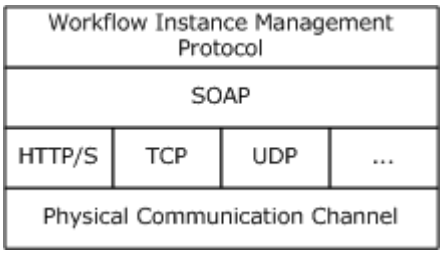

Figure 1: Protocol stack for the Workflow Instance Management Protocol

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

Prefix	Namespace URI	Reference
Wsam	http://www.w3.org/2007/05/addressing/metadata	
Wsdl	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. 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 2.3.2, 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.

	Workflow instance state when operation invoked			
Operation	Active	Suspended	Completed	
Abandon	<end>	<end>	<end>	Workflow instance state after operation completes
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	

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.

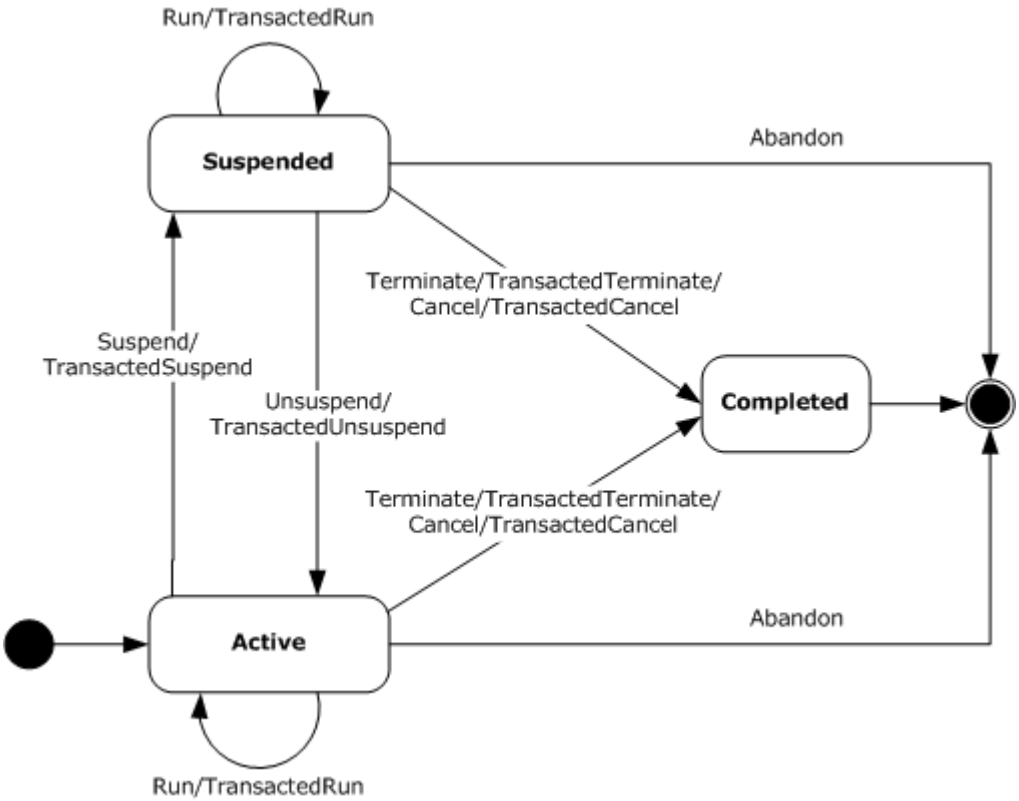

Figure 3: Durable program instance state machine

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
Abandon	SHOULD forcefully stop the execution of the durable program instance and indicate to the system that the durable program instance SHOULD be disposed.
Cancel	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.
Run	SHOULD provide the durable program instance an opportunity to execute.
Suspend	Transitions a durable program instance from the active state to the suspended state.
Terminate	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.
TransactedCancel	Performs the **Cancel** 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.
TransactedRun	Performs the **Run** 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.
TransactedSuspend	Performs the **Suspend** operation under a transaction (flowed in from the client or locally created). If the system maintains the durable state of the durable

Operation	Description
	program instance, the durable state MUST be updated during execution of this operation.
TransactedTerminate	Performs the **Terminate** 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.
TransactedUnsuspend	Performs the **Unsuspend** 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.
Unsuspend	Transitions a durable program instance from the suspended state to the active state.

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 2.3.2.
- 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 **Run** 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>
```

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_ResponseMessage** 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 2.3.2.
- 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 **TransactedRun** 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
<TransactedRun>	Contains the body of the IWorkflowInstanceManagement_TransactedRun_InputMessage message.
<TransactedRunResponse>	Contains the body of the **IWorkflowInstanceManagement_TransactedRun_ResponseMessage** 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 2.3.2.
- 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 **Abandon** operation.
IWorkflowInstanceManagement_Abandon_OutputMessage	Sent from the server as a reply to IWorkflowInstanceManagement_Abandon_InputMessage.

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_ResponseMessage** 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 2.3.2.
- 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 **Cancel** operation.
IWorkflowInstanceManagement_Cancel_OutputMessage	Sent from the server as a reply to 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_ResponseMessage** 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 2.3.2.
- 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 **TransactedCancel** 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_ResponseMessage** 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>
```

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 2.3.2.
- 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 **Terminate** 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_ResponseMessage** 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 2.3.2.
- 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 **TransactedTerminate** 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_ResponseMessage** 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 2.3.2.
- 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 **Suspend** 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.
<SuspendResponse>	Contains the body of the **IWorkflowInstanceManagement_Suspend_ResponseMessage** 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 2.3.2.
  - 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 **TransactedSuspend** 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
<TransactedSuspend>	Contains the body of the IWorkflowInstanceManagement_TransactedSuspend_InputMessage message.
<TransactedSuspendResponse>	Contains the body of the **IWorkflowInstanceManagement_TransactedSuspend_ResponseMessage** 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 2.3.2.
- 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 **Unsuspend** 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>
```

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_ResponseMessage** 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 2.3.2.
- 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 **TransactedUnsuspend** 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
<TransactedUnsuspend>	Contains the body of the IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage message.
<TransactedUnsuspendResponse>	Contains the body of the **IWorkflowInstanceManagement_TransactedUnsuspend_ResponseMessage** 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>
```

3.1.5 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. 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.

Request SOAP Message sent from the Client role to the Server role:

```
<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/IWorkflowInstan
ceManagement/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>
```

Response SOAP message sent from Server role to the Client role 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/IWorkflowInstan
ceManagement/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 Workflow Instance Management Protocol should be secured 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 Protocol Schema	xs:	Section 6.2

For ease of implementation the full WSDLs 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:portType name="IWorkflowInstanceManagement">
    <wsdl:operation name="TransactedUnsuspend">
      <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedUnsuspend"
message="tns:IWorkflowInstanceManagement_TransactedUnsuspend_InputMessage" />
      <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/TransactedUnsuspendResponse"
message="tns:IWorkflowInstanceManagement_TransactedUnsuspend_OutputMessage" />
    </wsdl:operation>
    <wsdl:operation name="Abandon">
      <wsdl:input
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/Abandon" message="tns:IWorkflowInstanceManagement_Abandon_InputMessage" />
```

```
      <wsdl:output
wsaw:Action="http://schemas.datacontract.org/2008/10/WorkflowServices/IWorkflowInstanceManage
ment/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: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: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:binding>
</wsdl:definitions>
```

6.2 Workflow Instance Management Protocol Schema

```
<?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 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: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>
```

7 Appendix B: Product Behavior

This document specifies version-specific details in the Microsoft® .NET Framework. For information about which versions of .NET Framework are available in each released Microsoft Windows® product or as supplemental software, see **.NET Framework**.

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

- Microsoft® .NET Framework 4.0

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 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 and Windows Server 2008 R2, .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 and Windows Server 2008 R2, .NET Framework 4.0 is not supported in the Server Core Role.

8 Change Tracking

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

9 Index

P

R

S

T

V

W