

[MS-CEPM]: Microsoft Complex Event Processing Engine Manageability 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.aspx>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplq@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

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.

Preliminary Documentation. This Open Specification is preliminary documentation for this technology. Since the documentation may change between this preliminary version and the final version, there are risks in relying on preliminary documentation. To the extent that you incur

additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
08/01/2009	0.1	Major	First release

Preliminary

Contents

1 Introduction	8
1.1 Glossary	8
1.2 References	9
1.2.1 Normative References	9
1.2.1.1 Normative API References	10
1.2.2 Informative References	11
1.3 Protocol Overview (Synopsis)	11
1.4 Relationship to Other Protocols	12
1.5 Prerequisites/Preconditions	12
1.6 Applicability Statement	13
1.7 Versioning and Capability Negotiation	13
1.8 Vendor-Extensible Fields	13
1.9 Standards Assignments	13
2 Messages	14
2.1 Transport	14
2.2 Messages	14
2.2.1 Namespaces	14
2.2.2 Methods	15
2.2.2.1 Metadata Methods	16
2.2.2.1.1 Create Message	16
2.2.2.1.1.1 CreateRequest Message	16
2.2.2.1.1.1.1 CreateRequest SOAP Header	16
2.2.2.1.1.1.2 CreateRequest SOAP Body	16
2.2.2.1.1.2 CreateResponse Message	17
2.2.2.1.1.2.1 CreateResponse SOAP Header	17
2.2.2.1.1.2.2 CreateResponse SOAP Body	17
2.2.2.1.1.2.3 Faults	17
2.2.2.1.1.3 Create Example	18
2.2.2.1.1.3.1 CreateRequest	18
2.2.2.1.1.3.2 CreateResponse	18
2.2.2.1.2 Get Message	19
2.2.2.1.2.1 GetRequest Message	19
2.2.2.1.2.1.1 GetRequest SOAP Header	19
2.2.2.1.2.1.2 GetRequest SOAP Body	19
2.2.2.1.2.2 GetResponse Message	19
2.2.2.1.2.2.1 GetResponse SOAP Header	19
2.2.2.1.2.2.2 GetResponse SOAP Body	20
2.2.2.1.2.2.3 Faults	20
2.2.2.1.2.3 Get Examples	20
2.2.2.1.2.3.1 GetRequest	20
2.2.2.1.2.3.2 GetResponse	21
2.2.2.1.3 Delete Message	21
2.2.2.1.3.1 DeleteRequest Message	21
2.2.2.1.3.1.1 DeleteRequest SOAP Header	21
2.2.2.1.3.1.2 DeleteRequest SOAP Body	21
2.2.2.1.3.2 DeleteResponse Message	22
2.2.2.1.3.2.1 DeleteResponse SOAP Header	22
2.2.2.1.3.2.2 DeleteResponse SOAP Body	22
2.2.2.1.3.2.3 Faults	22

2.2.2.1.3.3	Delete Examples	22
2.2.2.1.3.3.1	DeleteRequest.....	22
2.2.2.1.3.3.2	Delete Response	23
2.2.2.1.4	Enumerate Message	23
2.2.2.1.4.1	EnumerateRequest Message.....	23
2.2.2.1.4.1.1	EnumerateRequest SOAP Header.....	23
2.2.2.1.4.1.2	EnumerateRequest SOAP Body	24
2.2.2.1.4.2	EnumerateResponse Message.....	24
2.2.2.1.4.2.1	EnumerateResponse SOAP Header.....	24
2.2.2.1.4.2.2	EnumerateResponse SOAP Body.....	24
2.2.2.1.4.2.3	Faults.....	24
2.2.2.1.4.3	Enumerate Examples.....	25
2.2.2.1.4.3.1	EnumerateRequest.....	25
2.2.2.1.4.3.2	EnumerateResponse.....	25
2.2.2.1.5	ChangeQueryState Message	26
2.2.2.1.5.1	ChangeQueryStateRequest Message	26
2.2.2.1.5.1.1	ChangeQueryStateRequest SOAP Header.....	26
2.2.2.1.5.1.2	ChangeQueryStateRequest SOAP Body	26
2.2.2.1.5.2	ChangeQueryStateResponse Message.....	26
2.2.2.1.5.2.1	ChangeQueryStateResponse SOAP Header.....	26
2.2.2.1.5.2.2	ChangeQueryStateResponse SOAP Body.....	26
2.2.2.1.5.2.3	Faults.....	27
2.2.2.1.5.3	ChangeQueryState Examples.....	27
2.2.2.1.5.3.1	ChangeQueryStateRequest.....	27
2.2.2.1.5.3.2	ChangeQueryStateResponse.....	28
2.2.2.2	Diagnostic Methods	28
2.2.2.2.1	GetDiagnosticSettings Message	28
2.2.2.2.1.1	GetDiagnosticSettingsRequest	28
2.2.2.2.1.1.1	GetDiagnosticSettingsRequest SOAP Header.....	28
2.2.2.2.1.1.2	GetDiagnosticSettingsRequest SOAP Body	29
2.2.2.2.1.2	GetDiagnosticSettingsResponse	29
2.2.2.2.1.2.1	GetDiagnosticSettingsResponse SOAP Header.....	29
2.2.2.2.1.2.2	GetDiagnosticSettingsResponse SOAP Body.....	29
2.2.2.2.1.2.3	Faults.....	29
2.2.2.2.1.3	GetDiagnosticSettings Examples.....	29
2.2.2.2.1.3.1	GetDiagnosticSettingsRequest	29
2.2.2.2.1.3.2	GetDiagnosticSettingsResponse	30
2.2.2.2.2	SetDiagnosticSettings.....	30
2.2.2.2.2.1	SetDiagnosticSettingsRequest	30
2.2.2.2.2.1.1	SetDiagnosticSettingsRequest SOAP Header	31
2.2.2.2.2.1.2	SetDiagnosticSettingsRequest SOAP Body	31
2.2.2.2.2.2	SetDiagnosticSettingsResponse	31
2.2.2.2.2.2.1	SetDiagnosticSettingsResponse SOAP Header.....	31
2.2.2.2.2.2.2	SetDiagnosticSettingsResponse SOAP Body	31
2.2.2.2.2.2.3	Faults.....	31
2.2.2.2.2.3	SetDiagnosticSettings Examples	32
2.2.2.2.2.3.1	SetDiagnosticSettingsRequest	32
2.2.2.2.2.3.2	SetDiagnosticSettingsResponse	32
2.2.2.2.3	ClearDiagnosticSettings.....	33
2.2.2.2.3.1	ClearDiagnosticSettingsRequest.....	33
2.2.2.2.3.1.1	ClearDiagnosticSettingsRequest SOAP Header	33
2.2.2.2.3.1.2	ClearDiagnosticSettingsRequest SOAP Body.....	33
2.2.2.2.3.2	ClearDiagnosticSettingsResponse	33

2.2.2.2.3.2.1	ClearDiagnosticSettingsResponse SOAP Header	33
2.2.2.2.3.2.2	ClearDiagnosticSettingsResponse SOAP Body	33
2.2.2.2.3.2.3	Faults.....	34
2.2.2.2.3.3	ClearDiagnosticSettings Examples	34
2.2.2.2.3.3.1	ClearDiagnosticSettingsRequest.....	34
2.2.2.2.3.3.2	ClearDiagnosticSettingsResponse	34
2.2.2.2.4	GetDiagnosticView	35
2.2.2.2.4.1	GetDiagnosticView Request	35
2.2.2.2.4.1.1	GetDiagnosticView Request SOAP Header	35
2.2.2.2.4.1.2	GetDiagnosticView Request SOAP Body	35
2.2.2.2.4.2	GetDiagnosticView Response	35
2.2.2.2.4.2.1	GetDiagnosticView Response SOAP Header.....	35
2.2.2.2.4.2.2	GetDiagnosticView Response SOAP Body	36
2.2.2.2.4.2.3	Faults.....	36
2.2.2.2.4.3	GetDiagnosticView Examples.....	36
2.2.2.2.4.3.1	GetDiagnosticView Request	36
2.2.2.2.4.3.2	GetDiagnosticView Response	37
2.2.2.3	Faults.....	38
2.2.2.3.1	InvalidNameFault.....	38
2.2.2.3.1.1	InvalidNameFault SOAP Header.....	38
2.2.2.3.1.2	InvalidNameFault SOAP Body.....	38
2.2.2.3.1.3	InvalidNameFault Example	39
2.2.2.3.2	InvalidDefinitionFault	39
2.2.2.3.2.1	InvalidDefinitionFault SOAP Header	40
2.2.2.3.2.2	InvalidDefinitionFault SOAP Body.....	40
2.2.2.3.2.3	InvalidDefinitionFault Example	40
2.2.2.3.3	ManagementFault.....	41
2.2.2.3.3.1	ManagementFault SOAP Header	41
2.2.2.3.3.2	ManagementFault SOAP Body.....	41
2.2.2.3.3.3	ManagementFault Example.....	42
2.2.2.3.4	RuntimeFault.....	42
2.2.2.3.4.1	RuntimeFault SOAP Header	42
2.2.2.3.4.2	RuntimeFault SOAP Body.....	43
2.2.2.3.4.3	RuntimeFault Example.....	43
2.2.2.3.5	GetDiagnosticSettingsNotSupportedFault.....	44
2.2.2.3.5.1	GetDiagnosticSettingsNotSupportedFault SOAP Header.....	44
2.2.2.3.5.2	GetDiagnosticSettingsNotSupportedFault SOAP Body	44
2.2.2.3.5.3	GetDiagnosticSettingsNotSupportedFault Example	44
2.2.2.3.6	SetDiagnosticSettingsNotSupportedFault	45
2.2.2.3.6.1	SetDiagnosticSettingsNotSupportedFault SOAP Header	45
2.2.2.3.6.2	SetDiagnosticSettingsNotSupportedFault SOAP Body	45
2.2.2.3.6.3	SetDiagnosticSettingsNotSupportedFault Example	46
2.2.2.3.7	ClearDiagnosticSettingsNotSupportedFault	46
2.2.2.3.7.1	ClearDiagnosticSettingsNotSupportedFault SOAP Header	47
2.2.2.3.7.2	ClearDiagnosticSettingsNotSupportedFault SOAP Body.....	47
2.2.2.3.7.3	ClearDiagnosticSettingsNotSupportedFault Example	47
2.2.2.3.8	GetDiagnosticView NotSupportedFault	48
2.2.2.3.8.1	GetDiagnosticView NotSupportedFault SOAP Header	48
2.2.2.3.8.2	GetDiagnosticView NotSupportedFault SOAP Body	48
2.2.2.3.8.3	GetDiagnosticView NotSupportedFault Example	48
2.2.3	Types.....	49
2.2.3.1	Metadata Method Types.....	50
2.2.3.1.1	CreateRequest	50

2.2.3.1.2	GetResponse	51
2.2.3.1.3	QueryState	51
2.2.3.2	Metadata Definition Types	52
2.2.3.2.1	Metadata Object Types	52
2.2.3.2.1.1	QueryType	52
2.2.3.2.1.1.1	OutputStreamBindingType	53
2.2.3.2.1.1.2	InputStreamBindingType	54
2.2.3.2.1.2	QueryTemplateType	55
2.2.3.2.1.2.1	ImportOperatorType	56
2.2.3.2.1.2.2	ExportOperatorType	57
2.2.3.2.1.3	ApplicationType	57
2.2.3.2.1.4	Adapter Types	58
2.2.3.2.1.4.1	AdapterBaseType	58
2.2.3.2.1.4.2	InputAdapterType	58
2.2.3.2.1.4.3	OutputAdapterType	59
2.2.3.2.1.5	EventType	59
2.2.3.2.1.5.1	FieldType	60
2.2.3.2.2	AnyOperator Group	61
2.2.3.2.2.1	QueryTemplateReferenceOperatorType	62
2.2.3.2.2.1.1	QTrefInputStreamType	63
2.2.3.2.2.1.2	QTrefOutputStreamType	64
2.2.3.2.2.1.3	Example	64
2.2.3.2.2.2	MultiCastOperatorType	65
2.2.3.2.2.2.1	Example	65
2.2.3.2.2.3	ProjectOperatorType	65
2.2.3.2.2.3.1	ProjectExpressionContainerType	66
2.2.3.2.2.3.2	Example	67
2.2.3.2.2.4	SelectOperatorType	67
2.2.3.2.2.4.1	Example	68
2.2.3.2.2.5	JoinOperatorType	68
2.2.3.2.2.5.1	Example	70
2.2.3.2.2.6	UnionOperatorType	71
2.2.3.2.2.6.1	Example	71
2.2.3.2.2.7	AggregationOperatorType	72
2.2.3.2.2.7.1	AggregateExpressionContainerType	72
2.2.3.2.2.7.1.1	AggregateFunctionType	73
2.2.3.2.2.7.2	Example	74
2.2.3.2.2.8	AlterLifeTimeOperatorType	74
2.2.3.2.2.8.1	Example	75
2.2.3.2.2.9	GroupAndApplyOperatorType	75
2.2.3.2.2.9.1	ApplyBranchType	76
2.2.3.2.2.9.1.1	ApplyInputType	77
2.2.3.2.2.9.1.2	ApplyOutputType	78
2.2.3.2.2.9.2	Example	78
2.2.3.2.2.10	TopKOperatorType	79
2.2.3.2.2.10.1	RankExpressionContainerType	80
2.2.3.2.2.10.1.1	RankOrderType	81
2.2.3.2.2.10.2	Example	81
2.2.3.2.3	Additional Types, Groups, and AttributeGroups	82
2.2.3.2.3.1	BuiltinType	82
2.2.3.2.3.2	OperatorBaseType	82
2.2.3.2.3.3	StreamReferenceType	83
2.2.3.2.3.4	StreamDefinitionType	83

2.2.3.2.3.5	ExpressionContainerType	84
2.2.3.2.3.6	TerminatorBaseType	84
2.2.3.2.3.7	AnyExpression Group	84
2.2.3.2.3.7.1	UnaryArithmeticExpression	87
2.2.3.2.3.7.2	BinaryArithmeticExpression	87
2.2.3.2.3.7.3	ComparisonExpression	88
2.2.3.2.3.7.4	ConstantExpression	89
2.2.3.2.3.7.5	ConvertExpression	89
2.2.3.2.3.7.6	HashExpression	90
2.2.3.2.3.7.7	InputFieldExpression	91
2.2.3.2.3.7.8	NaryArithmeticExpression	91
2.2.3.2.3.7.9	MethodCallExpression	92
2.2.3.2.3.7.10	UnaryExpression	93
2.2.3.2.3.7.11	BinaryExpression	93
2.2.3.2.3.7.12	SystemFieldExpression	94
2.2.3.2.3.8	NullaryExpression	95
2.2.3.2.3.9	TypeIdentifier AttributeGroup	95
2.2.3.2.3.10	Date TimeType	96
2.2.3.2.3.11	ExpressionReturn TypeFacets AttributeGroup	96
2.2.3.2.3.12	StreamIdentifier AttributeGroup	97
2.2.3.2.3.13	ExpressionBase	97
2.2.3.2.3.14	FieldIdentifier	98
2.2.3.3	Diagnostic Method Types	98
2.2.3.3.1	SetDiagnosticSettings	98
2.2.3.3.1.1	DiagnosticAspects	99
2.2.3.3.1.2	DiagnosticLevel	100
2.2.3.3.2	GetDiagnosticSettingsResponse	101
2.2.3.3.3	GetDiagnosticView Response	102
2.2.3.3.3.1	DiagnosticView	102
2.2.3.3.3.1.1	Properties	103
2.2.3.4	Fault Types	103
2.2.3.4.1	InvalidNameFault	103
2.2.3.4.2	InvalidDefinitionFault	104
2.2.3.4.3	ManagementFault	104
2.2.3.4.4	RuntimeFault	105
2.2.3.4.5	GetDiagnosticSettingsNotSupportedFault	105
2.2.3.4.6	ClearDiagnosticSettingsNotSupportedFault	105
2.2.3.4.7	GetDiagnosticView NotSupportedFault	106
2.2.4	SOAP Headers	106
3	Appendix A: Full WSDL	108
4	Change Tracking	118
5	Index	119

1 Introduction

This document specifies the CEP protocol, a Web service protocol that defines the communication protocol between a client application and a complex event processing (CEP) server. Using this protocol, a client application can create metadata objects on a CEP server, start and stop queries, and query about the CEP system state.

1.1 Glossary

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

SOAP body
SOAP fault
SOAP header
Web Services Description Language (WSDL)
XML namespace
XML schema

The following terms are specific to this document:

Application object: A CEP metadata object that defines a containing namespace for all child objects, which can be any the following: **Query**, **QueryTemplate**, **InputAdapter**, **OutputAdapter**, **EventType**

CEP metadata object: An object that the CEP server allows an implementer to name and define. A metadata object can be any of the following types:

- **Application**
- **EventType**
- **InputAdapter**
- **OutputAdapter**
- **Query**
- **QueryTemplate**

complex event processing (CEP): The continuous and incremental processing of event streams from multiple sources, based on declarative query and pattern specifications with near-zero latency.

CTI (current time increment): A "heartbeat" event type that does not carry any payload, only a single timestamp. CTIs advance application time in the CEP engine.

event sink: A destination for an event stream within the CEP platform. In the current version, only output adapters can be event sinks.

EventType object: A CEP metadata object that is used to define the structure of the payload of an event, including the associated fields.

expand: An event type that extends the lifetime of an event. To be associated with an event, the expand operation **MUST** match its start time, end time, and entire set of payload field values.

InputAdapter object: A CEP metadata object that is the registration of the binary file compiled from user-written code, which makes the input adapter available to the CEP system so that it

can be used in query definitions. This object represents an input stream source and converts proprietary event data into CEP event format.

insert: An event type that declares that a payload is valid for the actual observed duration of the specified event's lifetime (start time – end time).

OutputAdapter object: A CEP metadata object that is the registration of the binary file compiled from user-written code, which makes the output adapter available to the CEP system so that it can be used in query definitions. This object represents an output stream source and receives events that are produced by the CEP engine for further processing.

Query object: A CEP metadata object that represents the binding of input and output adapters and a **QueryTemplate** object within an application

QueryTemplate object: A CEP metadata object that defines how to compute an output stream from one or more input streams.

retract: An event type that shortens the lifetime of an event. To be associated with an event, the retract MUST match the specified event's start time, end time, and entire set of payload field values.

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.

[ISO-3166] International Organization for Standardization, "Codes for the Representation of Names of Countries and Their Subdivisions", ISO 3166, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24591

Note There is a charge to download the specification.

[ISO-639] International Organization for Standardization, "Codes for the Representation of Names of Languages", ISO 639, <http://www.loc.gov/standards/iso639-2/>

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

[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>

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

[SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>.

- [WSADDR] Gudgin, M., Hadley, M., and Rogers, T., "Web Services Addressing (WS-Addressing) 1.0", W3C Recommendation, May 2006, <http://www.w3.org/2005/08/addressing>
- [WSPOLICY] Bajaj, S., et al., "Web Services Policy Framework and Attachment (WS-Policy) and (WS-PolicyAttachment)", September 2004, <http://schemas.xmlsoap.org/ws/2004/09/policy>
- [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] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, <http://www.w3.org/TR/REC-xml-names/>
- [XMLSCHEMA] World Wide Web Consortium, "XML Schema", September 2005, <http://www.w3.org/2001/XMLSchema>
- [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/>
- [XMLSCHEMA3] World Wide Web Consortium, "Schema defined in the SOAP Version 1.2 Part 1 specification", 2003, <http://www.w3.org/2003/05/soap-envelope/>
- [XMLSCHEMA4] World Wide Web Consortium, "WSDL Schema Files", 2003, <http://schemas.xmlsoap.org/wsdl/soap/>
- [XMLSCHEMA5] Organization for the Advancement of Structured Information Standards (OASIS), 2004, <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd>
- [XMLSCHEMA6] World Wide Web Consortium, "Schema for the SOAP/1.1 encoding ", 2001, <http://schemas.xmlsoap.org/soap/encoding/>
- [XMLSCHEMA7] World Wide Web Consortium, "W3C XML Schema defined in the Web Services Addressing 1.0 - Metadata specification", 2007, <http://www.w3.org/2007/05/addressing/metadata>
- [XMLSCHEMA9] World Wide Web Consortium, "WS-Addressing Specification", 2004, <http://schemas.xmlsoap.org/ws/2004/08/addressing>
- [XMLSCHEMA10] World Wide Web Consortium, "Schema for the SOAP/1.1 encoding", 2001, <http://schemas.xmlsoap.org/soap/encoding/>
- [XMLSCHEMA13] World Wide Web Consortium, "W3C XML Schema defined in the Web Services Addressing 1.0 - WSDL Binding specification", 2006, <http://www.w3.org/2006/05/addressing/wsdl>
- [XMLSCHEMA14] World Wide Web Consortium, "WSDL 1.1 Binding Extension for SOAP 1.2", March 2006, <http://schemas.xmlsoap.org/wsdl/soap12/>
- [XMLSCHEMA15] World Wide Web Consortium, "Web Services Metadata Exchange (WS-MetadataExchange)", August 2006, <http://schemas.xmlsoap.org/ws/2004/09/mex>
- [XMLSCHEMA16] World Wide Web Consortium, 2005, <http://schemas.xmlsoap.org/wsdl/>

1.2.1.1 Normative API References

There are currently no normative API references available for this protocol.

1.2.2 Informative References

[MSDN-IDPTETW] Microsoft Corporation, "Improve Debugging and Performance Tuning With ETW", April 2007, <http://msdn.microsoft.com/en-us/magazine/cc163437.aspx>

[MSDN-MPCEP] Microsoft Corporation, "Introducing Microsoft's Platform for Complex Event Processing", <http://download.microsoft.com/download/F/D/5/FD5E855C-D895-45A8-9F3E-110AFADBE51A/Microsoft%20CEP%20Overview.docx>

[MSDN-SYSNAME] Microsoft Corporation, "System Namespace", [http://msdn.microsoft.com/en-us/library/system\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/system(VS.71).aspx)

[MSDN-TAQNP] Microsoft Corporation, "Type.AssemblyQualifiedName Property," <http://msdn.microsoft.com/en-us/library/system.type.assemblyqualifiedname.aspx>.

1.3 Protocol Overview (Synopsis)

Complex event processing (CEP) is the continuous and incremental processing of event (data) streams from multiple sources based on declarative query and pattern specifications with near-zero latency. The goal is to identify meaningful patterns, relationships, and data abstractions from among seemingly unrelated events and to trigger immediate response actions.

Typical event stream sources include data from manufacturing applications, financial trading applications, Web analytics, and operational analytics.

The CEP engine provides a dedicated Web service to handle requests from client applications for managing the system. Using the protocol described in this document, applications issue instructions to the CEP engine to create, start, and stop queries, and to inquire about query status and other parameters that describe the health of a running CEP engine. The protocol also supports messages that are used to enable and disable specific performance counters and event tracing.

The CEPM protocol is used to communicate with the Web service that is provided by the CEP engine to define and manage all of the CEP system's objects. As soon as all of the objects are defined and in place in the CEP engine, a protocol message to start the query causes the CEP engine to tap into the streaming data and to calculate and send output data. Another such message will stop the engine from recording and computing data. The CEPM protocol is used to create and manage the following objects:

- **Application** object
- **Query** object
- **QueryTemplate** object
- **InputAdapter** object
- **OutputAdapter** object
- **EventType** object

The CEPM protocol is stateless. All communication is initiated by the client. The server only sends responses in response to messages received. The following figure shows the methods available in this protocol.

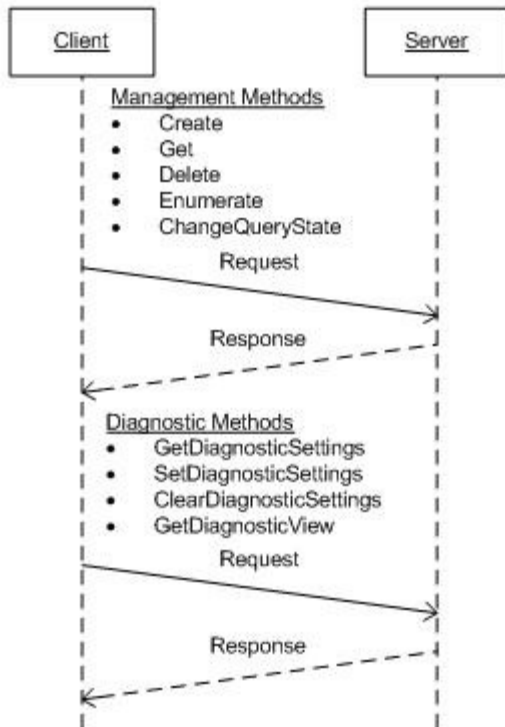


Figure 1: CEPM Web service protocol showing the available methods

1.4 Relationship to Other Protocols

The CEPM protocol uses SOAP over HTTP, as shown in the following layering diagram.

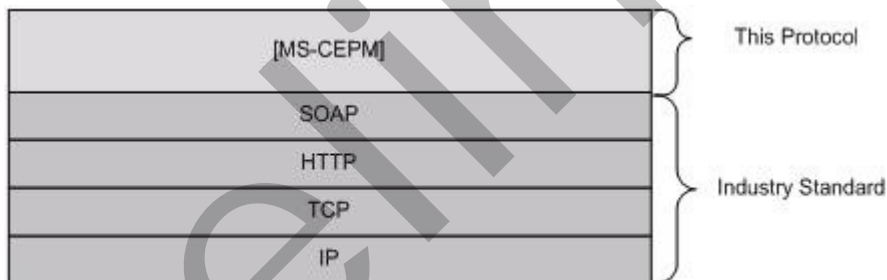


Figure 2: SOAP over HTTP

1.5 Prerequisites/Preconditions

To implement the CEPM protocol successfully, a running instance of the CEP engine in a stand-alone or embedded configuration is the only prerequisite.

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

- **Supported Transports:** This protocol uses multiple transports with SOAP, as specified in section [2.1](#) later in this document.
- **Localization:** This protocol allows text characters in any language, but it does not support localization of text strings into multiple languages. The protocol supports creating expressions that can be properly compared in different cultures, as described in section [2.2.3](#) later in this document.
- **Capability Negotiation:** This is the first released version of this protocol. No protocol capability negotiation is supported.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol messages MUST be formatted as a SOAP envelope as specified in [\[SOAP1.2/1\]](#).

Protocol servers MUST support SOAP [\[SOAP1.2/1\]](#) over Hypertext Transfer Protocol (HTTP) [\[RFC2616\]](#).

The message format is clear-text XML [\[XML10\]](#).

No authentication is supported by this protocol at this time.

2.2 Messages

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

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
meta data	http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Metadata	http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Metadat
msc	http://schemas.microsoft.com/ws/2005/12/wsd/contract	http://schemas.microsoft.com/ws/2005/12/wsd/contract
s	http://www.w3.org/2003/05/soap-envelope	http://www.w3.org/2003/05/soap-envelope
soap	http://schemas.xmlsoap.org/wsd/soap/	http://schemas.xmlsoap.org/wsd/soap/
soap encoding	http://schemas.xmlsoap.org/soap/encoding/	http://schemas.xmlsoap.org/soap/encoding/
soap 12	http://schemas.xmlsoap.org/wsd/soap12/	http://schemas.xmlsoap.org/wsd/soap12/
tns	http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management	http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing	http://schemas.xmlsoap.org/ws/2004/08/addressing
wsam	http://www.w3.org/2007/05/addressing/metadata	http://www.w3.org/2007/05/addressing/metadata
wsap	http://schemas.xmlsoap.org/ws/2004/08/addressing/policy	http://schemas.xmlsoap.org/ws/2004/08/addressing/policy

Prefix	Namespace URI	Reference
wsaw	http://www.w3.org/2006/05/addressing/wsd	http://www.w3.org/2006/05/addressing/wsd/
wsa10	http://www.w3.org/2005/08/addressing	http://www.w3.org/2005/08/addressing/
wSDL	http://schemas.xmlsoap.org/wsd/	http://schemas.xmlsoap.org/wsd/
wsp	http://schemas.xmlsoap.org/ws/2004/09/polic	http://schemas.xmlsoap.org/ws/2004/09/policy/
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
wsx	http://schemas.xmlsoap.org/ws/2004/09/mex	http://schemas.xmlsoap.org/ws/2004/09/mex
xs	http://www.w3.org/2001/XMLSchema	http://www.w3.org/2001/XMLSchema
ser	http://schemas.microsoft.com/2003/10/Serialization/	http://schemas.microsoft.com/2003/10/Serialization/
sera	http://schemas.microsoft.com/2003/10/Serialization/Arrays	http://schemas.microsoft.com/2003/10/Serialization/Arrays

2.2.2 Methods

The following table summarizes the set of method definitions defined by this specification.

Message	Description
Create	Used to create CEP metadata objects that comprise a complex event processing (CEP) application instance, such as event types, input and output adapters, query templates, and queries.
Get	Retrieves the definition of an object that has already been created.
Delete	Used to delete an object that has been created.
Enumerate	Used to return the names in a collection of like objects that have already been created, such as a collection of names of event types.
ChangeQueryState	Used to change the state of a query from stopped to started, or vice versa.
GetDiagnosticSettings	Retrieves the diagnostic settings that are currently in effect on a specific object in the system.
SetDiagnosticSettings	Sets diagnostic settings for monitoring a specific object in the CEP system.
ClearDiagnosticsSettings	Clears any previously set diagnostic settings on a specific object in the CEP system.
GetDiagnosticView	Used to return the observed values for a set of DiagnosticView properties that have been previously defined. The list of properties that are returned is variable; it depends on the settings that are set by using the SetDiagnosticSettings message, and on the type of object for which the diagnostic view is being retrieved.

2.2.2.1 Metadata Methods

These methods are used to create, remove, and manage metadata objects on the CEP server.

2.2.2.1.1 Create Message

A **Create** message is used to create objects on a CEP server, and to receive the response to the message.

2.2.2.1.1.1 CreateRequest Message

The **CreateRequest** message is used to create all CEP metadata objects within an application. The **Application** object is the top-level object that scopes a CEP application. Each **Application** object includes zero or more of the following metadata objects:

- **EventType** objects each for input and output, each of which may contain multiple **Field** objects.
- **InputAdapter** objects
- **OutputAdapter** objects
- **QueryTemplate** objects
- **Query** objects

A [Create](#) message MUST set elements in both the **SOAP header** and the **SOAP body**, as described in the following sections.

2.2.2.1.1.1.1 CreateRequest SOAP Header

A [Create](#) message MUST set the following elements in the SOAP header.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/Create .
tns:Name	xs:anyURI	This string value MUST be set to the name of a currently existing object that is the parent of the object being created. The server is the highest-level object and it is represented by "cep:/Server".

2.2.2.1.1.1.2 CreateRequest SOAP Body

The following elements MUST be present in the SOAP body of a [CreateRequest](#) message.

Element	Type	Description
CreateRequest	CreateRequest	The CreateRequest element contains the definition of the object being created as a child of the object referenced in the tns:Name element of the SOAP header. For the definition of the CreateRequest type, see section 2.2.3.1.1 .

2.2.2.1.1.2 CreateResponse Message

The **CreateResponse** message MUST be sent by the server in response to a received [CreateRequest](#) message, unless there is a fault or an exception.

2.2.2.1.1.2.1 CreateResponse SOAP Header

The following elements MUST be set in the SOAP header of a **CreateResponse** message.

Element	Type	Description
wsa10:Action	xs:string	The wsa10:Action element MUST be set to the following value: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/CreateResponse
tns:Name	xs:anyURI	The value of the tns:Name element MUST be the name of the object that was created. It is constructed by the system and MUST be the concatenation of the following items to form a single string and MUST contain a "/" character to separate each entity in the string: <ul style="list-style-type: none">▪ The name of the parent object that was specified in the Create message in the tns:Name element.▪ A string that represents the type of the object that was just created within the containing parent object.▪ The name of the object of this type that was assigned to the newly created object within its definition. In forming this string element, the sub-string component that represents the type of object MUST be one of the following strings: <ul style="list-style-type: none">▪ "Application"▪ "EventType"▪ "InputAdapter"▪ "OutputAdapter"▪ "QueryTemplate"▪ "Query"
S:ResourceAddress	wsa:ResourceAddress	A standard SOAP ResourceAddress element. For more information, see [WSADDR] .

2.2.2.1.1.2.2 CreateResponse SOAP Body

The SOAP body for a [CreateResponse](#) message MUST be empty.

2.2.2.1.1.2.3 Faults

The response to the [CreateRequest](#) message may be one of the following faults:

- [InvalidNameFault](#)
- [InvalidDefinitionFault](#)

- [ManagementFault](#)
- [RuntimeFault](#)

For a description of the content of the fault return result, see section [2.2.2.3](#).

2.2.2.1.1.3 Create Example

The following examples show a client's **CreateRequest** message, and the CEP server's **CreateResponse** message that is sent in response to the received **CreateRequest** message.

2.2.2.1.1.3.1 CreateRequest

The following example **CreateRequest** message is an instruction from the client to create an **EventType** object with a name of EventType1, which contains one field (named Field1) of type System.Int32.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/Create
    </a:Action>
    <h:Name
      xmlns:h="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
      cep:/Server</h:Name>
    <a:MessageID>urn:uuid:a0a32c60-9387-45a4-91b8-1abb2d22e641</a:MessageID>
    <ActivityId CorrelationId="27af68c7-2469-464f-aeb7-e6df2a27e4bd"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">http://localhost:8090</a:To>
  </s:Header>
  <s:Body>
    <CreateRequest
      xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
      <EventType Name="cep:/Server/Application/app1/EventType/EventType1"
        xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Metadata">
        <Field Name="Field1" Type="System.Int32" Nullable="false"></Field>
      </EventType>
    </CreateRequest>
  </s:Body>
</s:Envelope>
```

2.2.2.1.1.3.2 CreateResponse

The following example shows the **CreateResponse** message that is sent by the server in response to the preceding **CreateRequest** message.

```
<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.microsoft.com/ComplexEventProcessing/2009/05/Management/CreateResponse
</a:Action>
<h:ResourceAddress xmlns:h=
"http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
<a:Address>http://localhost:8090</a:Address>
<a:ReferenceParameters>
<h:Name>cep:/Server/Application/app1/EventType/EventType1</h:Name>
</a:ReferenceParameters>
</h:ResourceAddress>
</s:Header>
<s:Body></s:Body>
</s:Envelope>

```

2.2.2.1.2 Get Message

A **Get** message is used to request and receive the definition of a CEP metadata object that has already been created.

2.2.2.1.2.1 GetRequest Message

A **GetRequest** message is used to fetch the definition of a CEP metadata object that has already been created.

2.2.2.1.2.1.1 GetRequest SOAP Header

The SOAP header for a **GetRequest** message MUST contain the following elements.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/Get .
tns:Name	xs:anyURI	This string value MUST be set to the fully qualified name of the object that is being requested. This is a URI with "cep:/server" as the root, followed by a series of "object type" and "object name" pairs, with "/" as the separator character.

2.2.2.1.2.1.2 GetRequest SOAP Body

The SOAP body for a [GetRequest](#) message MUST be empty.

2.2.2.1.2.2 GetResponse Message

The **GetResponse** message MUST be sent by the server in response to a received [GetRequest](#) message, unless there is a fault or an exception.

2.2.2.1.2.2.1 GetResponse SOAP Header

The SOAP header for a [GetResponse](#) message MUST contain the following elements.

Element	Type	Description
wsa10:Act	xs:stri	The wsa10:Action element MUST be set to the following value:

Element	Type	Description
ion	ng	http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/GetResponse

2.2.2.1.2.2.2 GetResponse SOAP Body

The SOAP body for a [GetResponse](#) message MUST set the following elements.

Element	Type	Description
GetResponse	GetResponse	The GetResponse element contains the definition of the CEP metadata object that was requested in the tns:Name element of the SOAP header of the GetRequest message. For more information, see section 2.2.3.1.2 for the definition of the GetResponse type.

2.2.2.1.2.2.3 Faults

The response to the [GetRequest](#) message may be the following fault:

- [InvalidNameFault](#)

For a description of the content of the fault return result, see section [2.2.2.3](#).

2.2.2.1.2.3 Get Examples

The following examples show a client's [GetRequest](#) message, and the CEP server's [GetResponse](#) message that is sent in response to the received **GetRequest** message.

2.2.2.1.2.3.1 GetRequest

The following example **GetRequest** message is an instruction from the client to get the definition of the URI given in the **h:Name** element of the SOAP header, "cep:/Server/Application/app1/EventType/EventType2".

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/Get</a:Action>
    <h:Name
      xmlns:h="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      cep:/Server/Application/app1/EventType/EventType2</h:Name>
    <a:MessageID>urn:uuid:8e5f6f0b-b571-41a9-8b80-6f6e6454d2b1</a:MessageID>
    <ActivityId CorrelationId="f213cad7-b282-45a3-979c-0e4be5b03e3a"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">http://localhost:8090</a:To>
  </s:Header>
  <s:Body></s:Body>
</s:Envelope>
```

2.2.2.1.2.3.2 GetResponse

The following example shows the **GetResponse** message that is sent by the server in response to the preceding [GetRequest](#) message.

```
<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.microsoft.com/ComplexEventProcessing/2009/05/management/GetResponse
    </a:Action>
  </s:Header>
  <s:Body>
    <GetResponse
      xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      <EventType Name="EventType1" xmlns=
        "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Metadata">
        <Field Name="CountSegmentHitLogicId" Type="System.Int32"
          Nullable="true" />
        <Field Name="SegmentHitLogicId" Type="System.Int32"
          Nullable="true" />
        <Field Name="UserId" Type="System.Int32" Nullable="true" />
      </EventType>
    </GetResponse>
  </s:Body>
</s:Envelope>
```

2.2.2.1.3 Delete Message

The **Delete** message is used to delete a CEP metadata object that was previously created in an instance of the CEP server, and to receive the response to the message.

2.2.2.1.3.1 DeleteRequest Message

A **DeleteRequest** message is used to request the deletion of a CEP metadata object that was previously created, and which currently exists on the server.

2.2.2.1.3.1.1 DeleteRequest SOAP Header

The following elements **MUST** be set in the SOAP header of a [DeleteRequest](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/Delete .
tns:Name	xs:anyURI	This string value MUST be set to the fully qualified name of the CEP metadata object to be deleted.

2.2.2.1.3.1.2 DeleteRequest SOAP Body

The SOAP body for a [DeleteRequest](#) message **MUST** be empty.

2.2.2.1.3.2 DeleteResponse Message

A **DeleteResponse** message MUST be sent in response to a received [DeleteRequest](#) message, unless there is a fault or an exception.

2.2.2.1.3.2.1 DeleteResponse SOAP Header

The following elements MUST be set in the SOAP header of a [DeleteResponse](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/DeleteResponse .
tns:Name	xs:anyURI	This string value MUST be set to the fully qualified name of the object that has been deleted.

2.2.2.1.3.2.2 DeleteResponse SOAP Body

The SOAP body for a [DeleteResponse](#) message MUST be empty.

2.2.2.1.3.2.3 Faults

The response to the [DeleteRequest](#) message may be one of the following faults:

- [InvalidNameFault](#)
- [ManagementFault](#)

For a description of the content of the fault return result, see section [2.2.2.3](#).

2.2.2.1.3.3 Delete Examples

The following examples show a client's [DeleteRequest](#) message, and the CEP server's [DeleteResponse](#) message that is sent in response to the received **DeleteRequest** message.

2.2.2.1.3.3.1 DeleteRequest

The following example **DeleteRequest** message is an instruction from the client to delete the URI contained in the **h:Name** element, "cep:/Server/Application/app1".

```
<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.microsoft.com/ComplexEventProcessing/2009/05/management/Delete</a:Action>
    <h:Name
      xmlns:h="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      cep:/Server/Application/app1</h:Name>
    <a:MessageID>urn:uuid:8ac5ee16-6701-437e-b873-1ef63375660d</a:MessageID>
    <ActivityId CorrelationId="333d64ba-132f-4441-94d6-8c02dfcda6b8"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
```

```

    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
  </s:Header>
  <s:Body></s:Body>
</s:Envelope>

```

2.2.2.1.3.3.2 Delete Response

The following example shows the **DeleteResponse** message that is sent by the server in response to the preceding received [DeleteRequest](#) message.

```

<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.microsoft.com/ComplexEventProcessing/2009/05/management/DeleteResponse
    </a:Action>
    <h:Name
      xmlns:h="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      cep:/Server/Application/app1</h:Name>
    </s:Header>
    <s:Body></s:Body>
  </s:Envelope>

```

2.2.2.1.4 Enumerate Message

An **Enumerate** message is used to request and receive the names of a collection of CEP metadata objects that has already been created.

2.2.2.1.4.1 EnumerateRequest Message

The **EnumerateRequest** message is used to request the enumeration of definitions for a collection of CEP metadata objects with a common parent (for example, a collection of **EventType** objects with a common application object parent).

2.2.2.1.4.1.1 EnumerateRequest SOAP Header

The following elements **MUST** be set in the SOAP header of the [EnumerateRequest](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/Enumerate .
tns:Name	xs:anyURI	This string value MUST be set to a URI that represents the collection to be retrieved. This consists of the fully qualified URI of the parent CEP metadata object that contains the collection of objects to be enumerated, plus one of the following strings that represents the collection to be enumerated: <ul style="list-style-type: none"> “Application”

Element	Type	Description
		<ul style="list-style-type: none"> ▪ "EventType" ▪ "InputAdapter" ▪ "OutputAdapter" ▪ "QueryTemplate" ▪ "Query" ▪ "Operator" ▪ "Branch" ▪ "Stream" ▪ "OutputStream"

2.2.2.1.4.1.2 EnumerateRequest SOAP Body

The SOAP body for an [EnumerateRequest](#) message MUST be empty.

2.2.2.1.4.2 EnumerateResponse Message

An **EnumerateResponse** message MUST be sent in response to a received [EnumerateRequest](#) message, unless there is an exception or a fault.

2.2.2.1.4.2.1 EnumerateResponse SOAP Header

The following elements MUST be set in the SOAP header of an [EnumerateResponse](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/EnumerateResponse .

2.2.2.1.4.2.2 EnumerateResponse SOAP Body

The following elements MUST be set in the SOAP body of an [EnumerateResponse](#) message.

Element	Type	Description
ResourceNames	sera:ArrayOfanyURI	A collection of CEP metadata object names that are contained in the collection specified in the tns:Name element of the SOAP header of the EnumerateRequest message.

2.2.2.1.4.2.3 Faults

The response to the [EnumerateRequest](#) message may contain the following fault:

- [InvalidNameFault](#)

For a description of the content of the fault return result, see section [2.2.2.3](#).

2.2.2.1.4.3 Enumerate Examples

The following examples show a client's [EnumerateRequest](#) message, and the CEP server's [EnumerateResponse](#) message that is sent in response to the received **EnumerateRequest** message.

2.2.2.1.4.3.1 EnumerateRequest

The following example **EnumerateRequest** message is request from the client to retrieve the collection of **EventType** objects with parent "cep:/Server/Application/app1/" as specified in the **h:Name** element of the SOAP header.

```
<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.microsoft.com/ComplexEventProcessing/2009/05/Management/Enumerate
    </a:Action>
    <h:Name xmlns:h=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
      cep:/Server/Application/app1/EventType</h:Name>
    <a:MessageID>urn:uuid:6b1e7c14-3117-43e4-994c-20fe17913e6b</a:MessageID>
    <ActivityId CorrelationId="b5159221-74c6-4c70-aa08-0b43e461d27e"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
  </s:Header>
  <s:Body></s:Body>
</s:Envelope>
```

2.2.2.1.4.3.2 EnumerateResponse

The following example shows the **EnumerateResponse** message that is sent by the server in response to the preceding received [EnumerateRequest](#) message.

```
<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.microsoft.com/ComplexEventProcessing/2009/05/Management/EnumerateResponse
    </a:Action>
  </s:Header>
  <s:Body>
    <ResourceNames
      xmlns:d3p1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
      xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
      <d3p1:anyURI>cep:/Server/Application/app1/EventType/EventType1</d3p1:anyURI>
      <d3p1:anyURI>cep:/Server/Application/app1/EventType/EventType2</d3p1:anyURI>
    </ResourceNames>
  </s:Body>
</s:Envelope>
```

```
</s:Body>
</s:Envelope>
```

2.2.2.1.5 ChangeQueryState Message

The **ChangeQueryState** message is used to start and stop a **Query** object that has been created on a CEP server, and to receive the response to the message.

2.2.2.1.5.1 ChangeQueryStateRequest Message

The **ChangeQueryStateRequest** message is used to start a **Query** object running or stop it while it's running.

2.2.2.1.5.1.1 ChangeQueryStateRequest SOAP Header

The following elements MUST be set in the SOAP header of a [ChangeQueryStateRequest](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/ChangeQueryState .
tns:Name	xs:anyURI	The URI to the Query object for which it is desired to change the query state.

2.2.2.1.5.1.2 ChangeQueryStateRequest SOAP Body

The following elements MUST be set in the SOAP body of a [ChangeQueryStateRequest](#) message.

Element	Type	Description
QueryState	QueryState	The QueryState element is an enumeration that sets the query state. For more information, see section 2.2.3.1.3 .

2.2.2.1.5.2 ChangeQueryStateResponse Message

The **ChangeQueryStateResponse** message MUST be sent by the server in response to a received [ChangeQueryStateRequest](#) message, unless there is an exception or a fault.

2.2.2.1.5.2.1 ChangeQueryStateResponse SOAP Header

The following elements MUST be set in the SOAP header of a [ChangeQueryStateResponse](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/ChangeQueryStateResponse .

2.2.2.1.5.2.2 ChangeQueryStateResponse SOAP Body

The following elements MUST be set in the SOAP body of a [ChangeQueryStateResponse](#) message.

Element	Type	Description
QueryState	QueryState	The QueryState element is an enumeration that sets the query state. For more information, see section 2.2.3.1.3 .

2.2.2.1.5.2.3 Faults

The response to the [ChangeQueryStateRequest](#) message may be one of the following faults:

- [InvalidNameFault](#)
- [RuntimeFault](#)

For a description of the content of the fault return result, see section [2.2.2.3](#).

2.2.2.1.5.3 ChangeQueryState Examples

The following examples show a client's [ChangeQueryStateRequest](#) message, and the CEP server's [ChangeQueryStateResponse](#) message that is sent in response to the received **ChangeQueryStateRequest** message.

2.2.2.1.5.3.1 ChangeQueryStateRequest

The following example **ChangeQueryStateRequest** message is an instruction from the client to set the query state to **QueryStateStarted** for the query with the URI in the **h:Name** element, "cep:/Server/Application/app1/Query/Select1".

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/ChangeQueryState
    </a:Action>
    <h:Name xmlns:h=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
      cep:/Server/Application/app1/Query/Select1</h:Name>
    <a:MessageID>urn:uuid:96c5dfd0-1b1f-456b-bd89-56ea9eb14c7e</a:MessageID>
    <ActivityId CorrelationId="c69e97ff-99e2-44bd-b0f4-b8fe2c46a17f"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">http://localhost:8090</a:To>
  </s:Header>
  <s:Body>
    <QueryState xmlns=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/QueryStateStarted
    </QueryState>
  </s:Body>
</s:Envelope>
```

2.2.2.1.5.3.2 ChangeQueryStateResponse

The following example shows the **ChangeQueryStateResponse** message that is sent by the server in response to the preceding received [ChangeQueryStateRequest](#) message.

```
<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.microsoft.com/ComplexEventProcessing/2009/05/
      Management/ChangeQueryStateResponse
    </a:Action>
    <h:Name xmlns:h=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
      cep:/Server/Application/app1/Query/Select1</h:Name>
    </s:Header>
    <s:Body>
      <QueryState xmlns=
        "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
        http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
        Management/QueryStateStarted
      </QueryState>
    </s:Body>
  </s:Envelope>
```

2.2.2.2 Diagnostic Methods

The following methods are used for diagnosing system health or system performance by monitoring the individual objects and their resource usage from the CEP engine.

2.2.2.2.1 GetDiagnosticSettings Message

This message is used to request the retrieval and to receive the response of the current diagnostic settings that are in effect.

2.2.2.2.1.1 GetDiagnosticSettingsRequest

The **GetDiagnosticSettingsRequest** message is used to request the retrieval of the current diagnostic settings that are in effect for a specific named CEP metadata object.

2.2.2.2.1.1.1 GetDiagnosticSettingsRequest SOAP Header

The following elements **MUST** be set in the SOAP header of a [GetDiagnosticSettingsRequest](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/GetDiagnosticSettings .
tns:Name	xs:anyURI	This URI MUST be set to the name of the Query object for which the diagnostic settings will be retrieved.

2.2.2.2.1.1.2 GetDiagnosticSettingsRequest SOAP Body

The SOAP body for a [GetDiagnosticSettingsRequest](#) message MUST be empty.

2.2.2.2.1.2 GetDiagnosticSettingsResponse

The **GetDiagnosticSettingsResponse** message is sent by the CEP server in response to a received [GetDiagnosticSettingsRequest](#) message.

2.2.2.2.1.2.1 GetDiagnosticSettingsResponse SOAP Header

The following elements MUST be set in the SOAP header of the [GetDiagnosticSettingsResponse](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/GetDiagnosticSettingsResponse .

2.2.2.2.1.2.2 GetDiagnosticSettingsResponse SOAP Body

The following elements MUST be contained in the SOAP body of the [GetDiagnosticSettingsResponse](#) message.

Element	Type	Description
GetDiagnosticSettingsResponse	GetDiagnosticSettingsResponse	This element contains the diagnostic settings currently in effect. For more information, see section 2.2.3.2.2 .

2.2.2.2.1.2.3 Faults

The response to the [GetDiagnosticSettingsRequest](#) message may be one of the following faults:

- [InvalidNameFault](#)
- [GetDiagnosticSettingsNotSupportedFault](#)

For a description of the content of the fault return result, see . section [2.2.2.3](#).

2.2.2.2.1.3 GetDiagnosticSettings Examples

The following examples show a client's [GetDiagnosticSettingsRequest](#) message, and the CEP server's [GetDiagnosticSettingsResponse](#) message that is sent in response to the received **GetDiagnosticSettingsRequest** message.

2.2.2.2.1.3.1 GetDiagnosticSettingsRequest

The following example **GetDiagnosticSettingsRequest** message is a request to retrieve the Diagnostic settings for the URI specified in the **h:Name** element, "cep:/Server/Application/app1/Query/Select1".

```

<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.microsoft.com/ComplexEventProcessing/2009/05/management/GetDiagnosticSettings
    </a:Action>
    <h:Name s:mustUnderstand="1">

xmlns:h="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
  cep:/Server/Application/app1/Query/Select1</h:Name>
  <a:MessageID>urn:uuid:2fb6989f-7078-4f84-89da-23c6135142e1</a:MessageID>
  <ActivityId CorrelationId="a4afc40d-4927-45a4-84c1-fd2295137fbd"
            xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
    00000000-0000-0000-0000-000000000000</ActivityId>
  <a:ReplyTo>
    <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
  </a:ReplyTo>
</s:Header>
  <s:Body></s:Body>
</s:Envelope>

```

2.2.2.2.1.3.2 GetDiagnosticSettingsResponse

The following example shows the **GetDiagnosticSettingsResponse** message that is sent by the server in response to the preceding received [GetDiagnosticSettingsRequest](#) message.

```

<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.microsoft.com/ComplexEventProcessing/2009/05/management/GetDiagnosticSettingsR
esponse
    </a:Action>
  </s:Header>
  <s:Body>
    <GetDiagnosticSettingsResponse
      xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      <DiagnosticAspects>Memory</DiagnosticAspects>
      <DiagnosticLevel>Critical</DiagnosticLevel>
    </GetDiagnosticSettingsResponse>
  </s:Body>
</s:Envelope>

```

2.2.2.2.2 SetDiagnosticSettings

A **SetDiagnosticSettings** message is used to set the diagnostic settings on a specified CEP metadata object, and to receive the response.

2.2.2.2.2.1 SetDiagnosticSettingsRequest

The **SetDiagnosticSettingsRequest** message is used to set diagnostic settings on a CEP metadata object.

2.2.2.2.2.1.1 SetDiagnosticSettingsRequest SOAP Header

The following elements MUST be set in the SOAP header of the [SetDiagnosticSettingsRequest](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/SetDiagnosticSettings .
tns:Name	xs:anyURI	This URI MUST be set to the name of the CEP metadata object for which the diagnostic settings are being set. Valid objects are the server object or a query object.

2.2.2.2.2.1.2 SetDiagnosticSettingsRequest SOAP Body

The following elements MUST be contained in the SOAP body of the [SetDiagnosticSettingsRequest](#) message.

Element	Type	Description
SetDiagnosticSettings	SetDiagnosticSettings	This element contains the settings that will be instantiated on the CEP server. For more information, see section 2.2.3.2.1.

2.2.2.2.2.2 SetDiagnosticSettingsResponse

The **SetDiagnosticSettingsResponse** message is sent by the CEP server in response to a received [SetDiagnosticSettingsRequest](#) message.

2.2.2.2.2.2.1 SetDiagnosticSettingsResponse SOAP Header

The following elements MUST be set in the SOAP header of the [SetDiagnosticSettingsResponse](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/SetDiagnosticSettingsResponse .

2.2.2.2.2.2.2 SetDiagnosticSettingsResponse SOAP Body

The SOAP body for the [SetDiagnosticSettingsResponse](#) message MUST be empty.

2.2.2.2.2.2.3 Faults

The response to the [SetDiagnosticSettings](#) message may be one of the following faults:

- [InvalidNameFault](#)
- [SetDiagnosticSettingsNotSupportedFault](#)

For a description of the content of the fault return result, see section [2.2.2.3](#).

2.2.2.2.3 SetDiagnosticSettings Examples

The following examples show a client's [SetDiagnosticSettingsRequest](#) message, and the CEP server's [SetDiagnosticSettingsResponse](#) message that is sent in response to the received [GetDiagnosticSettingsRequest](#) message.

2.2.2.2.3.1 SetDiagnosticSettings Request

The following example [SetDiagnosticSettingsRequest](#) message is an instruction from the client to set the **DiagnosticSettings** values for the URI specified in the **h:Name** element of the SOAP header, "cep:/Server/Application/app1/Query/MulticastUnionQuery".

```
<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.microsoft.com/ComplexEventProcessing/2009/05/management/SetDiagnosticSettings
    </a:Action>
    <h:Name s:mustUnderstand="1" xmlns:h=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      cep:/Server/Application/app1/Query/MulticastUnionQuery</h:Name>
    <a:MessageID>urn:uuid:3447fce4-6a9e-477b-9a03-balb5781937d</a:MessageID>
    <ActivityId CorrelationId="8ab2b897-04dd-443d-98f8-7bcef440995a"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
  </s:Header>
  <s:Body>
    <SetDiagnosticSettings
      xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      <DiagnosticAspects>Memory</DiagnosticAspects>
      <DiagnosticLevel>Critical</DiagnosticLevel>
    </SetDiagnosticSettings>
  </s:Body>
</s:Envelope>
```

2.2.2.2.3.2 SetDiagnosticSettings Response

The following example shows the [SetDiagnosticSettingsResponse](#) message that is sent by the server in response to the preceding received [SetDiagnosticSettingsRequest](#) message.

```
<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.microsoft.com/ComplexEventProcessing/2009/05/management/SetDiagnosticSettingsR
      esponse
    </a:Action>
  </s:Header>
  <s:Body></s:Body>
```


</s:Envelope>

2.2.2.2.3 ClearDiagnosticSettings

A **ClearDiagnosticSettings** message is used to request and receive the response for the clearing of the diagnostic settings from an object for which **DiagnosticSettings** values had previously been set.

2.2.2.2.3.1 ClearDiagnosticSettingsRequest

The **ClearDiagnosticSettingsRequest** message is used to clear the diagnostic settings that were previously instantiated for a **Query** object. The new settings in effect are inherited from the parent object; otherwise, the default settings are used.

2.2.2.2.3.1.1 ClearDiagnosticSettingsRequest SOAP Header

The following elements MUST be set in the SOAP header of a [ClearDiagnosticSettingsRequest](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/ClearDiagnosticSettings .
tns:Name	xs:anyURI	This URI MUST be set to the name of the object for which the diagnostic settings will be cleared. Valid objects are the server object or a query object.

2.2.2.2.3.1.2 ClearDiagnosticSettingsRequest SOAP Body

The SOAP body for a [ClearDiagnosticSettingsRequest](#) message MUST be empty.

2.2.2.2.3.2 ClearDiagnosticSettingsResponse

The **ClearDiagnosticSettingsResponse** message is sent by the CEP server in response to a received [ClearDiagnosticSettingsRequest](#) message.

2.2.2.2.3.2.1 ClearDiagnosticSettingsResponse SOAP Header

The following elements MUST be set in the SOAP header of a [ClearDiagnosticSettingsResponse](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/ClearDiagnosticSettingsResponse .

2.2.2.2.3.2.2 ClearDiagnosticSettingsResponse SOAP Body

The SOAP body for a [ClearDiagnosticSettingsResponse](#) message MUST be empty.

2.2.2.2.3.2.3 Faults

The response to the [ClearDiagnosticSettings](#) message may be one of the following faults:

- [InvalidNameFault](#)
- [ClearDiagnosticSettingsNotSupportedFault](#)

For a description of the content of the Fault return result see section [2.2.2.3](#).

2.2.2.2.3.3 ClearDiagnosticSettings Examples

The following examples show a client's [ClearDiagnosticSettingsRequest](#) message, and the CEP server's [ClearDiagnosticSettingsResponse](#) message that is sent in response to the received **ClearDiagnosticSettingsRequest** message.

2.2.2.2.3.3.1 ClearDiagnosticSettingsRequest

The following example **ClearDiagnosticSettingsRequest** message is an instruction from the client to clear the diagnostic settings from the URI specified in **h:Name**, "cep:/Server/Application/app1/Query/MulticastUnionQuery," which had previously been set.

```
<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.microsoft.com/ComplexEventProcessing/2009/05/management/ClearDiagnosticSettings
    </a:Action>
    <h:Name s:mustUnderstand="1"
      xmlns:h="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      cep:/Server/Application/app1/Query/MulticastUnionQuery</h:Name>
    <a:MessageID urn:uuid:f609118e-0ed4-46ca-b955-61c3028cbb7a</a:MessageID>
    <ActivityId CorrelationId="77a69c8d-fa71-41dc-ae72-c5b14635b192"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
  </s:Header>
  <s:Body></s:Body>
```

2.2.2.2.3.3.2 ClearDiagnosticSettingsResponse

The following example shows the **ClearDiagnosticSettingsResponse** message that is sent by the server in response to the preceding received ClearDiagnosticSettingsRequest message.

```
<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.microsoft.com/ComplexEventProcessing/2009/05/management/ClearDiagnosticSetting
sResponse.
  </a:Action>
</s:Header>
<s:Body></s:Body>
</s:Envelope>

```

2.2.2.2.4 GetDiagnosticView

A **GetDiagnosticView** message is used to request and receive the content of a diagnostic view that had gathered diagnostic statistics.

2.2.2.2.4.1 GetDiagnosticViewRequest

The **GetDiagnosticViewRequest** message is used to return the observed values for a set of **DiagnosticView** properties that have been previously defined. The list of properties that are returned is variable and depends on the settings that are set by using the [SetDiagnosticSettings](#) message and on the type of object for which the diagnostic view is being retrieved.

2.2.2.2.4.1.1 GetDiagnosticViewRequest SOAP Header

The following elements MUST be set in the SOAP header of a [GetDiagnosticViewRequest](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/GetDiagnosticView .
tns:Name	xs:anyURI	This URI MUST be set to the name of the object for which the diagnostic view statistics are being requested.

2.2.2.2.4.1.2 GetDiagnosticViewRequest SOAP Body

The SOAP body for a [GetDiagnosticViewRequest](#) message MUST be empty.

2.2.2.2.4.2 GetDiagnosticViewResponse

The **GetDiagnosticViewResponse** message is sent by the CEP server in response to a received [GetDiagnosticViewRequest](#) message.

2.2.2.2.4.2.1 GetDiagnosticViewResponse SOAP Header

The following elements MUST be set in the SOAP header of a [GetDiagnosticViewResponse](#) message.

Element	Type	Description
wsa10:Action	xs:string	The value of the wsa10:Action element in the SOAP header MUST be set to http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/GetDiagnosticViewResponse .

2.2.2.2.4.2.2 GetDiagnosticViewResponse SOAP Body

The following elements MUST be contained in the SOAP body of a [GetDiagnosticViewResponse](#) message.

Element	Type	Description
GetDiagnosticViewResponse	GetDiagnosticViewResponse	This element contains the diagnostic view that shows the properties of a specific object for which the diagnostics are being retrieved. Each property is a name/value pair that provides information about some performance or statistical property of the object. For more information, see section 2.2.3.2.3 .

2.2.2.2.4.2.3 Faults

The response to the [GetDiagnosticViewRequest](#) message may be one of the following faults:

- [InvalidNameFault](#)
- [GetDiagnosticViewNotSupportedFault](#)

For a description of the content of the fault return result, see section [2.2.2.3](#).

2.2.2.2.4.3 GetDiagnosticView Examples

The following examples show a client's [GetDiagnosticViewRequest](#) message, and the CEP server's [GetDiagnosticViewResponse](#) message that is sent in response to the received **GetDiagnosticViewRequest** message.

2.2.2.2.4.3.1 GetDiagnosticViewRequest

The following example **GetDiagnosticViewRequest** message is an instruction from the client to retrieve the diagnostic view for the URI in the **h:Name** element of the SOAP header, "cep:/Server/Query".

```
<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.microsoft.com/ComplexEventProcessing/2009/05/management/GetDiagnosticView
    </a:Action>
    <h:Name s:mustUnderstand="1"
      xmlns:h="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      cep:/Server/Query</h:Name>
    <a:MessageID>urn:uuid:80128549-da22-4cb5-b04d-f3236aeb12fe</a:MessageID>
    <ActivityId CorrelationId="042b3829-5b49-42f8-9f74-bfae8d517d9a"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
```

```

    </a:ReplyTo>
  </s:Header>
  <s:Body></s:Body>
</s:Envelope>

```

2.2.2.2.4.3.2 GetDiagnosticViewResponse

The following example shows the **GetDiagnosticViewResponse** message that is sent by the server in response to the preceding received [GetDiagnosticViewRequest](#) message.

```

<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.microsoft.com/ComplexEventProcessing/2009/05/management/GetDiagnosticViewResponse

    </a:Action>
  </s:Header>
  <s:Body>
    <GetDiagnosticViewResponse
      xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
      <View xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Name>cep:/Server/Query</Name>
        <Properties>
          <Property>
            <Name>TotalOperatorCount</Name>
            <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
              i:type="d7p1:long">0</Value>
          </Property>
          <Property>
            <Name>TotalStreamCount</Name>
            <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
              i:type="d7p1:long">0</Value>
          </Property>
          <Property>
            <Name>CurrentEventCountInStream</Name>
            <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
              i:type="d7p1:long">0</Value>
          </Property>
          <Property>
            <Name>TotalEventCountInStream</Name>
            <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
              i:type="d7p1:long">0</Value>
          </Property>
          <Property>
            <Name>TotalStreamMemoryInKB</Name>
            <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
              i:type="d7p1:long">0</Value>
          </Property>
          <Property>
            <Name>CurrentEventCountInOperatorSynopsis</Name>
            <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
              i:type="d7p1:long">0</Value>
          </Property>
          <Property>

```

```

        <Name>TotalEventCountProcessedByOperator</Name>
        <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
            i:type="d7p1:long">22</Value>
    </Property>
    <Property>
        <Name>TotalEventCountOutputedByOperator</Name>
        <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
            i:type="d7p1:long">22</Value>
    </Property>
    <Property>
        <Name>TotalOperatorMemoryInKB</Name>
        <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
            i:type="d7p1:long">0</Value>
    </Property>
    <Property>
        <Name>TotalOperatorCpuUsage</Name>
        <Value xmlns:d7p1="http://www.w3.org/2001/XMLSchema"
            i:type="d7p1:long">0</Value>
    </Property>
    </Properties>
</View>
</GetDiagnosticViewResponse>
</s:Body>
</s:Envelope>

```

2.2.2.3 Faults

All faults in this protocol return the **s:Fault** element [\[SOAP1.2/1\]](#) in the SOAP body.

2.2.2.3.1 InvalidNameFault

An **InvalidNameFault** message is returned when the system tries to dereference a CEP metadata object name or a CEP metadata object type that does not exist.

2.2.2.3.1.1 InvalidNameFault SOAP Header

The following elements **MUST** be set in the SOAP header of an [InvalidNameFault](#).

Element	Type	Description
wsa10:Action	xs:string	The wsa10:Action element MUST be set to the following value: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/InvalidName

2.2.2.3.1.2 InvalidNameFault SOAP Body

The **InvalidNameFault** SOAP body **MUST** contain an **s:Fault** element as defined in [\[SOAP1.2/1\]](#). The **s:Fault** element for this protocol **MUST** contain an **s:Code** element, an **s:Reason** element, and an **s:Detail** element. The following table provides additional information about the elements contained in the **s:Fault** element.

Element	Description
s:Code	The s:Value element of the s:Subcode element of the s:Code element MUST be set to the value a:InvalidNameFault .

Element	Description
s:Reason	The s:Text element of the s:Reason element is set to a human-readable description of the reason for the fault.
s:Detail	The s:Detail element MUST contain an InvalidNameFault element of type InvalidNameFault . For more information, see section 2.2.2.2.2 .

2.2.2.3.1.3 InvalidNameFault Example

The following example shows an **InvalidNameFault** element.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/InvalidName
    </a:Action>
    <a:RelatesTo>urn:uuid:d097f723-0a5b-476e-8e55-39472ea6eefd
    </a:RelatesTo>
    <ActivityId CorrelationId="e96ffc8b-dd1c-4e31-b090-f39d2136bc50"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    </s:Header>
    <s:Body>
      <s:Fault>
        <s:Code>
          <s:Value>s:Sender</s:Value>
          <s:Subcode>
            <s:Value xmlns:a=
"
http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
              a:InvalidNameFault</s:Value>
            </s:Subcode>
          </s:Code>
          <s:Reason>
            <s:Text xml:lang="en-US">The argument cannot be null.</s:Text>
          </s:Reason>
          <s:Detail>
            <InvalidNameFault xmlns=
"
http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management"
              xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
              <Message>The argument cannot be null.</Message>
            </InvalidNameFault>
          </s:Detail>
        </s:Fault>
      </s:Body>
    </s:Envelope>
```

2.2.2.3.2 InvalidDefinitionFault

An **InvalidDefinitionFault** message is returned when an attempt is made to create a CEP metadata object and the attempted definition is invalid.

2.2.2.3.2.1 InvalidDefinitionFault SOAP Header

The following elements MUST be set in the SOAP header of an [InvalidDefinitionFault](#).

Element	Type	Description
wsa10:Action	xs:string	The wsa10:Action element MUST be set to the following value: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/InvalidDefinition

2.2.2.3.2.2 InvalidDefinitionFault SOAP Body

The [InvalidDefinitionFault](#) SOAP body MUST contain an **s:Fault** element as defined in [\[SOAP1.2/1\]](#). The **s:Fault** element for this protocol MUST contain an **s:Code** element, an **s:Reason** element, and an **s:Detail** element. The following table provides additional information about the elements contained in the **s:Fault** element.

Element	Description
s:Code	The s:Value element of the s:Subcode element of the s:Code element MUST be set to the value a:InvalidDefinitionFault .
s:Reason	The s:Text element of the s:Reason element is set to a human-readable description of the reason for the fault.
s:Detail	The s:Detail element MUST contain an InvalidDefinitionFault element of type InvalidDefinitionFault . For more information, see section 2.2.2.2.1.2 .

2.2.2.3.2.3 InvalidDefinitionFault Example

The following example shows an **InvalidDefinitionFault** element.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/InvalidDefinition
    </a:Action>
    <a:RelatesTo>urn:uuid:4e484014-b7c7-42b0-9a60-c335441bd1e7</a:RelatesTo>
    <ActivityId CorrelationId="519e87a1-edb0-4f8a-9825-be09882a2934"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    </s:Header>
    <s:Body>
      <s:Fault>
        <s:Code>
          <s:Value>s:Sender</s:Value>
          <s:Subcode>
            <s:Value xmlns:a=
              "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
              a:InvalidDefinitionFault</s:Value>
            </s:Subcode>
          </s:Code>
        </s:Fault>
      </s:Body>
    </s:Envelope>
```



```

<s:Reason>
  <s:Text xml:lang="en-US">The definition is not valid: The
'http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Metadata:Applications'
  element is not declared.-->The
'http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Metadata:Applications'
  element is not declared.</s:Text>
</s:Reason>
<s:Detail>
  <InvalidDefinitionFault
    xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <Message>The
'http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Metadata:Applications'
    element is not declared.</Message>
  </InvalidDefinitionFault>
</s:Detail>
</s:Fault>
</s:Body>
</s:Envelope>

```

2.2.2.3.3 ManagementFault

A **ManagementFault** message is returned whenever a generic error happens in any of the manageability operations on objects in the CEP system.

2.2.2.3.3.1 ManagementFault SOAP Header

The following elements MUST be set in the SOAP header of a [ManagementFault](#) element.

Element	Type	Description
wsa10:Action	xs:string	The wsa10:Action element MUST be set to the following value: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/Fault

2.2.2.3.3.2 ManagementFault SOAP Body

The [ManagementFault](#) SOAP body MUST contain an **s:Fault** element as defined in [\[SOAP1.2/1\]](#). The **s:Fault** element for this protocol MUST contain an **s:Code** element, an **s:Reason** element, and an **s:Detail** element. The following table provides additional information about the elements contained in the **s:Fault** element.

Element	Description
s:Code	The s:Value element of the s:Subcode element of the s:Code element MUST be set to the value a:ManagementFault .
s:Reason	The s:Text element of the s:Reason element is set to a human-readable description of the reason for the fault.
s:Detail	The s:Detail element MUST contain a ManagementFault element of type ManagementFault . For more information, see section 2.2.2.4.2 .

2.2.2.3.3 ManagementFault Example

The following example shows a **ManagementFault** element.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/Fault</a:Action>
    <a:RelatesTo urn:uuid:dbcfedb6-4198-437a-a086-541e07860aba</a:RelatesTo>
    <ActivityId CorrelationId="49acac6d-423c-4f2a-8747-d0b7dedb5056"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    </s:Header>
    <s:Body>
      <s:Fault>
        <s:Code>
          <s:Value>s:Sender</s:Value>
          <s:Subcode>
            <s:Value xmlns:a=
              "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
              a:ManagementFault</s:Value>
            </s:Subcode>
          </s:Code>
          <s:Reason>
            <s:Text xml:lang="en-US">The management service encountered an error:
              Delete operation failed.-->The address 'bogus:/app/address' is not valid.
              Additional information: Invalid scheme 'bogus'.</s:Text>
          </s:Reason>
          <s:Detail>
            <ManagementFault
              xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management"
              xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
              <Message>Delete operation failed.</Message>
            </ManagementFault>
          </s:Detail>
        </s:Fault>
      </s:Body>
    </s:Envelope>
```

2.2.2.3.4 RuntimeFault

A **RuntimeFault** message is returned whenever a generic error happens during the runtime operation of the CEP system.

2.2.2.3.4.1 RuntimeFault SOAP Header

The following elements MUST be set in the SOAP header of a [RuntimeFault](#).

Element	Type	Description
wsa10:Action	xs:string	The wsa10:Action element MUST be set to the following value: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/RuntimeFailure

2.2.2.3.4.2 RuntimeFault SOAP Body

The [RuntimeFault](#) SOAP body MUST contain an **s:Fault** element as defined in [\[SOAP1.2/1\]](#). The **s:Fault** element for this protocol MUST contain an **s:Code** element, an **s:Reason** element, and an **s:Detail** element. The following table provides additional information about the elements contained in the **s:Fault** element.

Element	Description
s:Code	The s:Value element of the s:Subcode element of the s:Code element MUST be set to the value a:RuntimeFault .
s:Reason	The s:Text element of the s:Reason element is set to a human-readable description of the reason for the fault.
s:Detail	The s:Detail element MUST contain a RuntimeFault element of type RuntimeFault . For more information, see section 2.2.3.3.1 .

2.2.2.3.4.3 RuntimeFault Example

The following example shows a **RuntimeFault** element.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management/RuntimeFailure</a:Action>
    <a:RelatesTo urn:uuid:b65a1aab-4673-4e51-92a7-97f46c59031e</a:RelatesTo>
    <ActivityId CorrelationId="f1f8f11f-3b9f-4501-98a7-7e2ea485f412"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    </s:Header>
    <s:Body>
      <s:Fault>
        <s:Code>
          <s:Value>s:Sender</s:Value>
          <s:Subcode>
            <s:Value
              xmlns:a="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management">
              a:RuntimeFault</s:Value>
            </s:Subcode>
          </s:Code>
          <s:Reason>
            <s:Text xml:lang="en-US">There was an error in the runtime:
              Get operation failed.-->The address 'bogus:/app/address' is not valid.
              Additional information: Invalid scheme 'bogus'.</s:Text>
          </s:Reason>
          <s:Detail>
            <RuntimeFault
              xmlns="http://schemas.microsoft.com/ComplexEventProcessing/2009/05/management"
              xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
              <Message>Get operation failed.</Message>
            </RuntimeFault>
          </s:Detail>
        </s:Fault>
```

```

    </s:Body>
  </s:Envelope>

```

2.2.2.3.5 GetDiagnosticSettingsNotSupportedFault

A **GetDiagnosticSettingsNotSupportedFault** message is returned when an attempt to get diagnostic settings from objects for which this operation is not supported is attempted.

2.2.2.3.5.1 GetDiagnosticSettingsNotSupportedFault SOAP Header

The following elements MUST be set in the SOAP header of a **GetDiagnosticSettingsNotSupportedFault** element.

Element	Type	Description
wsa10:Action	xs:string	The wsa10:Action element MUST be set to the following value: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/GetDiagnosticSettingsNotSupported

2.2.2.3.5.2 GetDiagnosticSettingsNotSupportedFault SOAP Body

The [GetDiagnosticSettingsNotSupportedFault](#) SOAP body MUST contain an **s:Fault** element as defined in [\[SOAP1.2/1\]](#). The **s:Fault** element for this protocol MUST contain an **s:Code** element, an **s:Reason** element, and an **s:Detail** element. The following table provides additional information about the elements contained in the **s:Fault** element.

Element	Description
s:Code	The s:Value element of the s:Subcode element of the s:Code element MUST be set to the value a:GetDiagnosticSettingsNotSupportedFault .
s:Reason	The s:Text element of the s:Reason element is set to a human-readable description of the reason for the fault.
s:Detail	The s:Detail element MUST contain a GetDiagnosticSettingsNotSupportedFault element of type GetDiagnosticSettingsNotSupportedFault . For more information, see section 2.2.3.4.5 .

2.2.2.3.5.3 GetDiagnosticSettingsNotSupportedFault Example

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/
      GetDiagnosticSettingsNotSupported</a:Action>
    <a:RelatesTo>urn:uuid:9f6b03b5-a6c1-475b-b900-819bf34b8bf8</a:RelatesTo>
    <ActivityId CorrelationId="de4aa3e7-da65-4661-ab49-f9aabef3ed4a"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    </s:Header>
  <s:Body>
    <s:Fault>
      <s:Code>

```

```

    <s:Value>s:Sender</s:Value>
    <s:Subcode>
      <s:Value xmlns:a=
        "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
        a:GetDiagnosticSettingsNotSupported</s:Value>
      </s:Subcode>
    </s:Code>
    <s:Reason>
      <s:Text xml:lang="en-US">Getting the diagnostic settings for 'cep:/Server'
        is not supported.</s:Text>
    </s:Reason>
    <s:Detail>
      <GetDiagnosticSettingsNotSupported xmlns=
        "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Message>Getting the diagnostic settings for 'cep:/Server'
          is not supported.</Message>
        <Name>cep:/Server</Name>
      </GetDiagnosticSettingsNotSupported>
    </s:Detail>
  </s:Fault>
</s:Body>
</s:Envelope>

```

2.2.2.3.6 SetDiagnosticSettingsNotSupportedFault

A **SetDiagnosticSettingsNotSupportedFault** message is returned when an attempt to set diagnostic settings on an object for which this operation is not supported is attempted.

2.2.2.3.6.1 SetDiagnosticSettingsNotSupportedFault SOAP Header

The following elements MUST be set in the SOAP header of a [SetDiagnosticSettingsNotSupportedFault](#) element.

Element	Type	Description
wsa10:Action	xs:string	The wsa10:Action element MUST be set to the following value: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/SetDiagnosticSettingsNotSupported

2.2.2.3.6.2 SetDiagnosticSettingsNotSupportedFault SOAP Body

The [SetDiagnosticSettingsNotSupportedFault](#) SOAP body MUST contain an **s:Fault** element as defined in [\[SOAP1.2/1\]](#). The **s:Fault** element for this protocol MUST contain an **s:Code** element, an **s:Reason** element, and an **s:Detail** element. The following table provides additional information about the elements contained in the **s:Fault** element.

Element	Description
s:Code	The s:Value element of the s:Subcode element of the s:Code element MUST be set to the value a:SetDiagnosticSettingsNotSupportedFault .
s:Reason	The s:Text element of the s:Reason element is set to a human-readable description of the reason for the fault.

Element	Description
s:Detail	The s:Detail element MUST contain a SetDiagnosticSettingsNotSupportedFault element of type GetDiagnosticSettingsNotSupportedFault . For more information, see section 2.2.3.4.5 .

2.2.2.3.6.3 SetDiagnosticSettingsNotSupportedFault Example

The following example shows a **SetDiagnosticSettingsNotSupportedFault** element.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
      Management/SetDiagnosticSettingsNotSupported</a:Action>
    <a:RelatesTo urn:uuid:51b46784-947e-49da-89d8-2d69bee43e6d</a:RelatesTo>
    <ActivityId CorrelationId="3aa972ea-0ba3-4e67-82e0-b0726d883412"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    </s:Header>
    <s:Body>
      <s:Fault>
        <s:Code>
          <s:Value>s:Sender</s:Value>
          <s:Subcode>
            <s:Value xmlns:a=
              "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
              a:SetDiagnosticSettingsNotSupported</s:Value>
            </s:Subcode>
          </s:Code>
          <s:Reason>
            <s:Text xml:lang="en-US">Setting the diagnostic settings for
              'cep:/Server/Application/appl/EventType' is not supported.</s:Text>
          </s:Reason>
          <s:Detail>
            <GetDiagnosticSettingsNotSupported xmlns=
              "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management"
              xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
              <Message>Setting the diagnostic settings for
                'cep:/Server/Application/appl/EventType' is not supported.</Message>
              <Name>cep:/Server/Application/appl/EventType</Name>
            </GetDiagnosticSettingsNotSupported>
          </s:Detail>
        </s:Fault>
      </s:Body>
    </s:Envelope>
```

2.2.2.3.7 ClearDiagnosticSettingsNotSupportedFault

A **ClearDiagnosticSettingsNotSupportedFault** message is returned when an attempt to clear diagnostic settings from objects for which this operation is not supported is attempted.

2.2.2.3.7.1 ClearDiagnosticSettingsNotSupportedFault SOAP Header

The following elements MUST be set in the SOAP header of a [ClearDiagnosticSettingsNotSupportedFault](#).

Element	Type	Description
wsa10:Action	xs:string	The wsa10:Action element MUST be set to the following value: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/ClearDiagnosticSettingsNotSupported

2.2.2.3.7.2 ClearDiagnosticSettingsNotSupportedFault SOAP Body

The [ClearDiagnosticSettingsNotSupportedFault](#) SOAP body MUST contain an **s:Fault** element as defined in [\[SOAP1.2/1\]](#). The **s:Fault** element for this protocol MUST contain an **s:Code** element, an **s:Reason** element, and an **s:Detail** element. The following table provides additional information about the elements contained in the **s:Fault** element.

Element	Description
s:Code	The s:Value element of the s:Subcode element of the s:Code element MUST be set to the value a:ClearDiagnosticSettingsNotSupportedFault .
s:Reason	The s:Text element of the s:Reason element is set to a human-readable description of the reason for the fault.
s:Detail	The s:Detail element MUST contain a ClearDiagnosticSettingsNotSupportedFault element of type ClearDiagnosticSettingsNotSupportedFault . For more information, see section 2.2.3.4.6 .

2.2.2.3.7.3 ClearDiagnosticSettingsNotSupportedFault Example

The following example shows a **ClearDiagnosticSettingsNotSupportedFault** element.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/
      ClearDiagnosticSettingsNotSupported</a:Action>
    <a:RelatesTo urn:uuid:056eaafc-40f3-4757-94db-deea457220a9</a:RelatesTo>
    <ActivityId CorrelationId="ef5b4a0f-8102-41e9-bafe-f8cdf5f80b6e"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
      00000000-0000-0000-0000-000000000000</ActivityId>
    </s:Header>
    <s:Body>
      <s:Fault>
        <s:Code>
          <s:Value>s:Sender</s:Value>
          <s:Subcode>
            <s:Value xmlns:a=
              "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management">
              a:ClearDiagnosticSettingsNotSupported</s:Value>
            </s:Subcode>
          </s:Code>
        </s:Fault>
      </s:Body>
    </s:Envelope>
```

```

<s:Reason>
  <s:Text xml:lang="en-US">Clearing the diagnostic settings for
  'cep:/Server/Application/appl/EventType' is not supported.</s:Text>
</s:Reason>
<s:Detail>
  <ClearDiagnosticSettingsNotSupported xmlns=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <Message>Clearing the diagnostic settings for
    'cep:/Server/Application/appl/EventType' is not supported.</Message>
    <Name>cep:/Server/Application/appl/EventType</Name>
  </ClearDiagnosticSettingsNotSupported>
</s:Detail>
</s:Fault>
</s:Body>
</s:Envelope>

```

2.2.2.3.8 GetDiagnosticViewNotSupportedFault

A **GetDiagnosticViewNotSupportedFault** message is returned when an attempt to get a diagnostic view from objects for which this operation is not supported is attempted.

2.2.2.3.8.1 GetDiagnosticViewNotSupportedFault SOAP Header

The following elements MUST be set in the SOAP header of a [GetDiagnosticViewNotSupportedFault](#).

Element	Type	Description
wsa10:Action	xs:string	The wsa10:Action element MUST be set to the following value: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/GetDiagnosticViewNotSupported

2.2.2.3.8.2 GetDiagnosticViewNotSupportedFault SOAP Body

The [GetDiagnosticViewNotSupportedFault](#) SOAP body MUST contain an **s:Fault** element as defined in [SOAP1.2/1]. The **s:Fault** element for this protocol MUST contain an **s:Code** element, an **s:Reason** element, and an **s:Detail** element. The following table provides additional information about the elements contained in the **s:Fault** element.

Element	Description
s:Code	The s:Value element of the s:Subcode element of the s:Code element MUST be set to the value a:GetDiagnosticViewNotSupportedFault .
s:Reason	The s:Text element of the s:Reason element is set to a human-readable description of the reason for the fault.
s:Detail	The s:Detail element MUST contain a GetDiagnosticViewNotSupportedFault element of type GetDiagnosticViewNotSupportedFault . For more information, see section 2.2.3.4.7 .

2.2.2.3.8.3 GetDiagnosticViewNotSupportedFault Example

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
```



```

        xmlns:a="http://www.w3.org/2005/08/addressing">
<s:Header>
  <a:Action s:mustUnderstand="1">
    http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management
    /GetDiagnosticViewNotSupported
  </a:Action>
  <a:RelatesTo>urn:uuid:056eaafc-40f3-4757-94db-deea457220a9</a:RelatesTo>
  <ActivityId CorrelationId="ef5b4a0f-8102-41e9-bafe-f8cdf5f80b6e"
    xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">
    00000000-0000-0000-0000-000000000000
  </ActivityId>
</s:Header>
<s:Body>
  <s:Fault>
    <s:Code>
      <s:Value>s:Sender</s:Value>
      <s:Subcode>
        <s:Value xmlns:a=
"http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management "
a:GetDiagnosticViewNotSupported
        </s:Value>
        </s:Subcode>
      </s:Code>
      <s:Reason>
        <s:Text xml:lang="en-US">
          Request the diagnostic view for
          'cep:/Server/Application/appl/EventType' is not supported.
        </s:Text>
      </s:Reason>
      <s:Detail>
        <GetDiagnosticViewNotSupported xmlns=
"http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management "
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
          <Message>
            Request the diagnostic view for
            'cep:/Server/Application/appl/EventType' is not supported.
          </Message>
          <Name>cep:/Server/Application/appl/EventType</Name>
        </GetDiagnosticViewNotSupported>
      </s:Detail>
    </s:Fault>
  </s:Body>
</s:Envelope>

```

2.2.3 Types

The following table summarizes the set of type definitions that are defined by this specification.

Element	Description
CreateRequest	The CreateRequest type forms the SOAP body of the CreateRequest message. For more information, see section 2.2.3.1.1 .
GetResponse	The GetResponse type forms the SOAP body of the GetResponse message. For more information, see section 2.2.3.1.2 .
QueryState	The QueryState type forms the SOAP body of the

Element	Description
	ChangeQueryStateRequest message and the ChangeQueryStateResponse message. For more information, see section 2.2.3.1.3 .
SetDiagnosticSettings	The SetDiagnosticSettings type forms the SOAP body of the SetDiagnosticSettings message. For more information, see section 2.2.3.3.1 .
GetDiagnosticSettingsResponse	The GetDiagnosticSettingsResponse type forms the SOAP body of the GetDiagnosticSettingsResponse message. For more information, see section 2.2.3.3.2 .
GetDiagnosticViewResponse	The GetDiagnosticViewResponse type forms the SOAP body of the GetDiagnosticViewResponse message. For more information, see section 2.2.3.3.3 .

2.2.3.1 Metadata Method Types

2.2.3.1.1 CreateRequest

The following code is the XSD for the **CreateRequest** complex type.

```
<xs:complexType name="CreateRequest">
  <xs:choice>
    <xs:element minOccurs="1" maxOccurs="1" name="InputAdapter"
      type="metadata:InputAdapterType" />
    <xs:element minOccurs="1" maxOccurs="1" name="OutputAdapter"
      type="metadata:OutputAdapterType" />
    <xs:element minOccurs="1" maxOccurs="1" name="Application"
      type="metadata:ApplicationType" />
    <xs:element minOccurs="1" maxOccurs="1" name="EventType"
      type="metadata:EventType" />
    <xs:element minOccurs="1" maxOccurs="1" name="Query"
      type="metadata:QueryType" />
    <xs:element minOccurs="1" maxOccurs="1" name="QueryTemplate"
      type="metadata:QueryTemplateType" />
  </xs:choice>
</xs:complexType>
```

The following table describes the elements and attributes that are referenced in the XSD.

Element	Type	Description
InputAdapter	InputAdapterType	The definition of an InputAdapter object.
OutputAdapter	OutputAdapterType	The definition of an OutputAdapter object.
Application	ApplicationType	The definition of an Application object.
EventType	EventType	The definition of an EventType object.
Query	QueryType	The definition of a Query object.
QueryTemplate	QueryTemplateType	The definition of a QueryTemplate object.

2.2.3.1.2 GetResponse

The following code is the XSD for the **GetResponse** type.

```
<xs:complexType name="GetResponse">
  <xs:choice>
    <xs:element minOccurs="1" maxOccurs="1" name="InputAdapter"
      type="metadata:InputAdapterType" />
    <xs:element minOccurs="1" maxOccurs="1" name="OutputAdapter"
      type="metadata:OutputAdapterType" />
    <xs:element minOccurs="1" maxOccurs="1" name="Application"
      type="metadata:ApplicationType" />
    <xs:element minOccurs="1" maxOccurs="1" name="EventType"
      type="metadata:EventType" />
    <xs:element minOccurs="1" maxOccurs="1" name="Query"
      type="metadata:QueryType" />
    <xs:element minOccurs="1" maxOccurs="1" name="QueryTemplate"
      type="metadata:QueryTemplateType" />
  </xs:choice>
</xs:complexType>
```

The types and descriptions for the **GetResponse** type are identical to those for the **CreateRequest** type. For more information, see section [2.2.3.1.1](#) for further information.

2.2.3.1.3 QueryState

The following code is the XSD for the **QueryState** type.

```
<xs:simpleType name="QueryState">
  <xs:restriction base="xs:string">
    <xs:enumeration value=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
      Management/QueryStateStarted" />
    <xs:enumeration value=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
      Management/QueryStateStopped" />
  </xs:restriction>
</xs:simpleType>
```

The following table describes the elements and attributes that are referenced in the XSD.

Element	Type	Description
QueryState	xs:string (restriction)	This enumeration allows the starting and stopping of a query. The possible values are as follows: http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/QueryStateStarted : The query is started and begins consuming and emitting events. http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management/QueryStateStopped : The query is stopped.

2.2.3.2 Metadata Definition Types

2.2.3.2.1 Metadata Object Types

2.2.3.2.1.1 QueryType

A query of type **QueryType** is used to bind together an input stream, an output stream, and a [QueryTemplate](#).

The following code is the XSD for the **QueryType** type.

```
<xs:complexType name="QueryType">
  <xs:annotation>
    <xs:documentation>The schema of a CreateQuery command. It contains
    information to bind a query template's input and output streams to
    stream sources and sinks.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded"
      name="OutputStreamBinding"
      type="tns:OutputStreamBindingType" />
    <xs:element minOccurs="1" maxOccurs="unbounded"
      name="InputStreamBinding"
      type="tns:InputStreamBindingType" />
  </xs:sequence>
  <xs:attribute name="Name" type="xs:anyURI" use="required" />
  <xs:attribute name="QueryTemplate" type="xs:anyURI" use="required" />
  <xs:attribute name="Description" type="xs:string" use="optional" />
</xs:complexType>
```

The following tables describe the elements and attributes for the **QueryType** type.

Element	Type	Description
OutputStreamBinding	OutputStreamBindingType	This element associates an event sink with a stream export operator in a query template.
InputStreamBinding	InputStreamBindingType	This element associates an event source (an input adapter or another query) with a stream import operator in a query template.

Attribute	Type	Description
Name	xs:anyURI	The query name by which this query will be referenced.
QueryTemplate	xs:anyURI	The name of the QueryTemplate object to which this query is bound.
Description	xs:string	A human-readable description that is not processed by the CEP server.

2.2.3.2.1.1.1 OutputStreamBindingType

The following code is the XSD for the **OutputStreamBindingType** type.

```
<xs:complexType name="OutputStreamBindingType">
  <xs:annotation>
    <xs:documentation>Output Stream Binding. Pairs a stream sink with
      a query template.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:annotation>
      <xs:documentation>The contained XML element will be passed to the
        output adapter as initialization information.</xs:documentation>
    </xs:annotation>
    <xs:any minOccurs="0" maxOccurs="1" namespace="##any"
      processContents="skip" />
  </xs:sequence>
  <xs:attribute name="OutputStream" type="xs:anyURI" use="required">
    <xs:annotation>
      <xs:documentation>Reference to an export operator name.
    </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="OutputStreamTarget" type="xs:anyURI" use="required">
    <xs:annotation>
      <xs:documentation>Reference to an output adapter.
    </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="StreamEventShape" type="tns:StreamEventShapeType"
    use="optional">
    <xs:annotation>
      <xs:documentation>Desired stream shape at the output.
    </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="StreamEventOrdering"
    type="tns:StreamEventOrderingType"
    use="optional">
    <xs:annotation>
      <xs:documentation>Desired time ordering at the output.
    </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

The following tables describe the elements and attributes for the **OutputStreamBindingType** type.

Element	Type	Description
(any XML)	xs:any	The XML element contained in the OutputStreamBindingType type is not interpreted by the CEP server or by the CEPM protocol. This XML element is passed to the OutputAdapter component, which is pointed to by the OutputStreamTarget XML attribute on the OutputStreamBinding element. This XML element acts as a query startup parameter that can be used to initialize the adapter at run time. It is up to the adapter author as to whether and how to

Element	Type	Description
		process this piece of XML.

Attribute	Type	Description
OutputStream	xs:anyURI	The name of the export operator in a QueryTemplate object.
OutputStreamTarget	xs:anyURI	The name of the stream sink. This can be either another query or an output adapter.
StreamEventOrdering	base=xs:string	This enumeration specifies the desired temporal ordering of events in the query's output stream. "ChainOrdered": An insert event and its associated chain of retraction events are in order in relation to one another, but different inserts can be out of order. "FullyOrdered" (default): All events will be fully ordered.
StreamEventShape	base=xs:string	The type of events that an output stream will contain. "Insert Only" – Insert events and associated retraction events are consolidated so that only the final insert events are output. "PointInTime" – Like "InsertOnly," but with "PointInTime" only event start times are kept, whereas event end times are set to the start times. "Signal" (default) – Each insert event can be followed by no more than one retraction event.

2.2.3.2.1.1.2 InputStreamBindingType

The **InputStreamBindingType** type specifies the input stream that the query binds to.

The following code is the XSD for the **InputStreamBindingType** type.

```
<xs:complexType name="InputStreamBindingType">
  <xs:annotation>
    <xs:documentation>Input Stream Binding. Pairs a stream source
      with a query template.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:annotation>
      <xs:documentation>The contained XML element will be passed to the
        output adapter as initialization information.</xs:documentation>
    </xs:annotation>
    <xs:any minOccurs="0" maxOccurs="1" namespace="##any"
      processContents="skip" />
  </xs:sequence>
  <xs:attribute name="InputStream" type="xs:anyURI" use="required">
    <xs:annotation>
      <xs:documentation>Reference to an import operator name.
    </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

```

</xs:attribute>
<xs:attribute name="InputStreamSource" type="xs:anyURI" use="required">
  <xs:annotation>
    <xs:documentation>Reference to an input adapter.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:complexType>

```

Element	Type	Description
(any XML)	xs:any	The XML element contained in the InputStreamBindingType is not interpreted by the CEP server and is passed to the InputAdapter component, which is pointed to by the InputStreamTarget XML attribute on the InputStreamBinding element. This XML element acts as a query startup parameter that can be used to initialize the adapter at run time. It is up to the adapter author as to whether and how to process this piece of XML.

Attribute	Type	Description
InputStream	xs:anyURI	The InputStream name. The name of an import operator in the query template.
InputStreamSource	xs:anyURI	The name of the stream source. This can be either another query or an input adapter.

2.2.3.2.1.2 QueryTemplateType

The **QueryTemplateType** type defines how to compute an output stream from one or more input streams.

The following code is the XSD for the **QueryTemplateType** type.

```

<xs:complexType name="QueryTemplateType">
  <xs:annotation>
    <xs:documentation>A Query template has n import and one export operator.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="Import"
      type="tns:ImportOperatorType" />
    <xs:element minOccurs="1" maxOccurs="1" name="Export"
      type="tns:ExportOperatorType" />
    <xs:group minOccurs="0" maxOccurs="unbounded" ref="tns:AnyOperator" />
  </xs:sequence>
  <xs:attribute name="Name" type="xs:anyURI" />
</xs:complexType>

```

The following tables describe the elements and attributes for the **QueryTemplateType** type.

Element	Type	Description
Import	ImportOperatorType	Defines a stream entry point of the query template.
Export	ExportOperatorType	Defines a stream exit point of the query template.
(group)	AnyOperator	A set of operators that defines the query graph and thus the query operation.

Attribute	Type	Description
Name	xs:anyURI	The name given to this QueryTemplate , by which it will be referenced.

2.2.3.2.1.2.1 ImportOperatorType

The following code is the XSD for the **ImportOperatorType** type.

```
<xs:complexType name="ImportOperatorType">
  <xs:annotation>
    <xs:documentation>Import Operator. Denotes the query's import stream.
    The Name attribute identifies the stream. Refers to a single operator
    as its output. The attribute Type refers to the stream type using the
    type's name.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:TerminatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
          type="tns:StreamDefinitionType" />
      </xs:sequence>
      <xs:attribute name="Name" type="xs:anyURI" use="required" />
      <xs:attribute name="Type" type="xs:anyURI" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following tables describe the elements and attributes for the **ImportOperatorType** type.

Element	Type	Description
OutputStream	StreamDefinitionType	Defines a stream entry point to be used by one or more operators in the specified query template.

Attribute	Type	Description
Name	xs:anyURI	The name by which this import operator will be referenced.
Type	xs:anyURI	The URI of the EventType object of the events that will be available on the stream.

2.2.3.2.1.2.2 ExportOperatorType

The following code is the XSD for the **ExportOperatorType** type.

```
<xs:complexType name="ExportOperatorType">
  <xs:annotation>
    <xs:documentation>Export Operator. Makes the query's outgoing
      stream explicit. The Name attribute identifies the stream. Refers
      to a single operator as its input.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:TerminatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="InputStream"
          type="tns:StreamReferenceType" />
      </xs:sequence>
      <xs:attribute name="Name" type="xs:anyURI" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following tables describe the elements and attributes for the **ExportOperatorType** type.

Element	Type	Description
InputStream	StreamReferenceType	References a stream exit point that was defined by some other operator in the specified query template.

Attribute	Type	Description
Name	xs:anyURI	The name by which this export operator will be referenced.

2.2.3.2.1.3 ApplicationType

The **ApplicationType** type defines an application object. The **Application** object is the top-level container of the system. A defined **Application** object acts as a namespace for other metadata entities that belong together.

The following code is the XSD for the **ApplicationType** type.

```
<xs:complexType name="ApplicationType">
  <xs:annotation>
    <xs:documentation>Application object.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="Name" type="xs:anyURI" use="required" />
</xs:complexType>
```

The following table describes the attributes for the **ApplicationType** type.

Attribute	Type	Description
Name	xs:anyURI	Application name. The application will be referenced by this name.

2.2.3.2.1.4 Adapter Types

Adapters are binary files compiled from user-written code, and they represent an input or output stream source. They convert proprietary event data into CEP event format for input, or convert CEP event format into a proprietary format for output.

2.2.3.2.1.4.1 AdapterBaseType

In the CEPM protocol, the complex types [InputAdapterType](#) and [OutputAdapterType](#) are defined as extensions to the type **AdapterBaseType**.

The following code is the XSD for the **AdapterBaseType** type.

```
<xs:complexType name="AdapterBaseType">
  <xs:annotation>
    <xs:documentation>Adapter base type. The common attributes of input
    and output adapter.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="Name" type="xs:anyURI" use="required" />
  <xs:attribute name="Class" type="xs:string" use="required" />
  <xs:attribute name="Description" type="xs:string" use="optional" />
</xs:complexType>
```

The following table describes the attributes for the **AdapterBaseType** type.

Attribute	Type	Description
Name	xs:anyURI	The name given to the adapter. The adapter will be referenced by this name.
Class	xs:string	This string represents an assembly qualified name of a .NET class implementing the adapter or the GUID that represents an adapter class in unmanaged code. For more information about how an assembly-qualified name is constructed, see [MSDN-TAQNP] . This assembly contains the code that represents the adapter at run time.
Description	xs:string	Adapter description.

2.2.3.2.1.4.2 InputAdapterType

Input adapters are binary files compiled from user-written code, and they represent an input stream source. They convert proprietary event data into CEP event format. The **InputAdapterType** type is a reference to that user-written code.

The **InputAdapterType** type is an extension of the [AdapterBaseType](#) type. It adds no elements or attributes. The following code is the XSD for the **InputAdapterType** type.

```

<xs:complexType name="InputAdapterType">
  <xs:annotation>
    <xs:documentation>Input adapter.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:AdapterBaseType">
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

2.2.3.2.1.4.3 OutputAdapterType

Output adapters are binary files compiled from user-written code. They receive the events that are produced by the CEP engine. The **OutputAdapterType** type references that user-written code. The **OutputAdapterType** type is an extension of the [AdapterBaseType](#) type. It adds no elements or attributes. The following code is the XSD for the **OutputAdapterType** type.

```

<xs:complexType name="OutputAdapterType">
  <xs:annotation>
    <xs:documentation>Output adapter.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:AdapterBaseType" />
  </xs:complexContent>
</xs:complexType>

```

2.2.3.2.1.5 EventType

Objects of type **EventType** represent the transient data items in a complex event processing system. The **EventType** type is used to define the structure of an event, consisting of one or more fields.

The following code is the XSD for the **EventType** type.

```

<xs:complexType name="EventType">
  <xs:annotation>
    <xs:documentation>An event has a name and 0..n fields.
  </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Field"
      type="tns:FieldType" />
  </xs:sequence>
  <xs:attribute name="Name" type="xs:anyURI" />
</xs:complexType>

```

The following tables describe the elements and attributes for the **EventType** type.

Element	Type	Description
Field	FieldType	A collection of objects of type FieldType that form the fields of the specified

Element	Type	Description
		event.

Attribute	Type	Description
Name	xs:anyURI	The event name by which the specified event will be referenced.

2.2.3.2.1.5.1 FieldType Type

A field contains data values of a defined type. The values in each field are processed by the various operators in a query.

The following code is the XSD for the **FieldType** type.

```
<xs:complexType name="FieldType">
  <xs:annotation>
    <xs:documentation>A field is of a built-in type and has a number
      of attributes.</xs:documentation>
  </xs:annotation>
  <xs:sequence />
  <xs:attribute name="Name" type="xs:string" use="required" />
  <xs:attribute name="Type" type="tns:BuiltinType" use="required" />
  <xs:attribute name="Culture" type="xs:string" use="optional" />
  <xs:attribute name="Nullable" type="xs:boolean" use="required" />
  <xs:attribute name="MaxSize" type="xs:int" use="optional">
    <xs:annotation>
      <xs:documentation>MaxSize is only applicable to string and byte
        array types. For string, this is the number of characters, for
        byte array this is the number of bytes.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="SizeFixed" type="xs:boolean" use="optional">
    <xs:annotation>
      <xs:documentation>SizeFixed is only applicable to string and
        byte array types. It denotes a field of a fixed size.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

The following table describes the elements and attributes for the **FieldType** type.

Attribute	Type	Description
Name	xs:string	The name by which the specified field will be referenced.
Type	BuiltinType	An enumeration value that represents the type of the field. The enumeration values correspond to the Microsoft.System namespace for .NET. For more information, see [MSDN-SYSNAME] .
Culture	xs:string	The culture of the specified field. The format of this string is specified in [ISO 639-2] and [ISO 3166] . For more information, see [MSDN-MPCEP] . The

Attribute	Type	Description
		culture is used as the default culture for comparison operators.
Nullable	xs:boolean	The attribute that indicates whether this field is nullable.
MaxSize	xs:int	An attribute that is applicable only to string or byte array types. For strings, it is the maximum number of characters. For byte fields, it is the maximum number of bytes.
SizeFixed	xs:boolean	An attribute that is applicable only to string or byte array types. SizeFixed defines whether the specified field is a fixed-size field or a variable-size field.

2.2.3.2.2 AnyOperator Group

The **AnyOperator** group contains the top-level operator types that may be contained in a [QueryTemplate](#) object.

The following code is the XSD for the **AnyOperator** group.

```
<xs:group name="AnyOperator">
  <xs:annotation>
    <xs:documentation>Placeholder for exactly one operator element of
    any type.</xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element name="QueryTemplateReference"
      type="tns:QueryTemplateReferenceOperatorType" />
    <xs:element name="MultiCast" type="tns:MultiCastOperatorType" />
    <xs:element name="Project" type="tns:ProjectOperatorType" />
    <xs:element name="Select" type="tns:SelectOperatorType" />
    <xs:element name="Join" type="tns:JoinOperatorType" />
    <xs:element name="Union" type="tns:UnionOperatorType" />
    <xs:element name="Aggregate" type="tns:AggregationOperatorType" />
    <xs:element name="AlterLifeTime" type="tns:AlterLifeTimeOperatorType" />
    <xs:element name="GroupAndApply" type="tns:GroupAndApplyOperatorType" />
    <xs:element name="TopK" type="tns:TopKOperatorType" />
  </xs:choice>
</xs:group>
```

The following table describes the elements for the **AnyOperator** group.

Element	Type	Description
QueryTemplateReference	QueryTemplateReferenceOperatorType	This operator is used to embed another QueryTemplate object within a current QueryTemplate object.
Multicast	MulticastOperatorType	The Multicast operator broadcasts an input stream to multiple output streams.
Project	ProjectOperatorType	The Project operator applies an arbitrary number

Element	Type	Description
		of expressions to an input stream and produces a single output stream.
Select	SelectOperatorType	The Select operator is used to filter inputs and to select a subset of the inputs for output.
Join	JoinOperatorType	The Join operator is used to join two inputs based on an expression.
Union	UnionOperatorType	The Union operator provides the definition for combining multiple input streams and placing them on a single output stream.
Aggregate	AggregateOperatorType	The Aggregate operator defines an operation that represents the arithmetic aggregation of inputs to produce an output stream.
AlterLifeTime	AlterLifeTimeOperatorType	The AlterLifeTime operator is used to define time windows for events, such that subsequent operations can be performed on the events within the defined windows.
GroupAndApply	GroupAndApplyOperatorType	The GroupAndApply operator is used to partition a stream into subsets and to perform operations on each subset.
TopK	TopKOperatorType	The TopK operator specifies that observed values are ranked and only the top K of them are placed on an output stream.

2.2.3.2.2.1 QueryTemplateReferenceOperatorType

The **QueryTemplateReferenceOperatorType** type is used to embed another query template within a specified query template.

The following code is the XSD for the **QueryTemplateReferenceOperatorType** type.

```
<xs:complexType name="QueryTemplateReferenceOperatorType">
  <xs:annotation>
    <xs:documentation>Embeds another query template in the query.
  </xs:documentation>
</xs:complexType>
```

```

</xs:annotation>
<xs:complexContent mixed="false">
  <xs:extension base="tns:OperatorBaseType">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded"
        name="InputStream" type="tns:QTrefInputStreamType" />
      <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
        type="tns:QTrefOutputStreamType" />
    </xs:sequence>
    <xs:attribute name="TemplateName" type="xs:anyURI"
      use="required" />
  </xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following tables describe the elements and attributes for the **QueryTemplateReferenceOperatorType** type.

Element	Type	Description
InputStream	QTrefInputStreamType	The input stream for this query operator. This element refers to an import operator in the specified query template.
OutputStream	QTrefOutputStreamType	The output stream for this query operator. This element refers to the export operator in the specified query template.

Attribute	Type	Description
TemplateName	xs:anyURI	This URI references an already existing query template.

2.2.3.2.2.1.1 QTrefInputStreamType

The **QTrefInputStreamType** type is used to reference an input operator in another **QueryTemplate** object.

This type is an extension to the [StreamReferenceType](#) type. Thus, it refers to a stream defined somewhere else in a specified query template and feeds it into the specified input operator in another query template. The following code is the XSD for the **QTrefInputStreamType** type.

```

<xs:complexType name="QTrefInputStreamType">
  <xs:annotation>
    <xs:documentation>Type for the input stream in an QT reference operator.
    In addition to the local stream name, it also needs to refer to the
    respective endpoint in the other query template. This is done via
    the attribute "ExternalName". It refers to the stream name that is used
    in the Import in the embedded query template.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:StreamReferenceType">
      <xs:attribute name="ExternalName" type="xs:anyURI" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

The following table describes the attributes for the **QTrefInputStreamType** type.

Attribute	Type	Description
ExternalName	xs:anyURI	A reference to the endpoint in the QueryTemplate object that is referenced in this QueryTemplate object. This is the stream name in the Import operator of the referenced QueryTemplate object.

2.2.3.2.2.1.2 QTrefOutputStreamType

The **QTrefOutputStreamType** type is used to reference the embedded query template's output stream.

This type is an extension to the [StreamReferenceType](#). Thus, it receives the outgoing stream from another query template and makes it available in this query template.

The following code is the XSD for the **QTrefOutputStreamType** type.

```

<xs:complexType name="QTrefOutputStreamType">
  <xs:annotation>
    <xs:documentation>Type for the output stream in an QT reference
    operator. In addition to the local stream name, it also needs to refer
    to the respective endpoint in the other query template. This is done
    via the attribute "ExternalName". It refers to the stream name that is
    used in a Export in the embedded query template.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:StreamDefinitionType">
      <xs:attribute name="ExternalName" type="xs:anyURI" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table describes the attributes for the **QTrefOutputStreamType** type.

Attribute	Type	Description
ExternalName	xs:anyURI	Reference to the endpoint in the QueryTemplate object that is referenced in this QueryTemplate object. This is the stream name in the export operator of the referenced QueryTemplate object.

2.2.3.2.2.1.3 Example

```

<QueryTemplateReference Name="QTReference1"
  QueryTemplateName="cep:/Server/Application/app1/QueryTemplate/Inner">
  <InputStream Name="import1" ExternalName="InputStreamSource1"/>
  <InputStream Name="import2" ExternalName="InputStreamSource2"/>
  <OutputStream Name="qtref1" ExternalName="OutputStreamSource1"/>

```


</QueryTemplateReference>

2.2.3.2.2.2 MultiCastOperatorType

The **MultiCastOperatorType** type defines an operator that replicates a single input stream to multiple output streams.

The following code is the XSD for the **MultiCastOperatorType** type.

```
<xs:complexType name="MultiCastOperatorType">
  <xs:annotation>
    <xs:documentation>A multicast creates multiple named streams out of
    a single input stream. The input events are simply replicated to all
    outputs.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:OperatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="InputStream"
          type="tns:StreamReferenceType" />
        <xs:element minOccurs="2" maxOccurs="unbounded"
          name="OutputStream" type="tns:StreamDefinitionType" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table describes the elements for the **MultiCastOperatorType** type.

Element	Type	Description
InputStream	StreamReferenceType	A reference to the input stream for this MultiCast element.
OutputStream	StreamDefinitionType	The definitions for the multiple output streams.

2.2.3.2.2.2.1 Example

```
<MultiCast Name="MulticastOperator">
  <InputStream Name="import1"></InputStream>
  <OutputStream Name="Multicast1"></OutputStream>
  <OutputStream Name="Multicast2"></OutputStream>
</MultiCast>
```

2.2.3.2.2.3 ProjectOperatorType

The **ProjectOperatorType** type is used as a container for defining an arbitrary number of project expressions on fields of an input stream to produce a single output.

The following code is the XSD for the **ProjectOperatorType** type.

```

<xs:complexType name="ProjectOperatorType">
  <xs:annotation>
    <xs:documentation>A project operator applies an arbitrary number of
    project expressions to a single input stream and yields a single output
    stream.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:OperatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="InputStream"
          type="tns:StreamReferenceType" />
        <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
          type="tns:StreamDefinitionType" />
        <xs:element minOccurs="0" maxOccurs="unbounded"
          name="ProjectExpression"
          type="tns:ProjectExpressionContainerType" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table describes the elements for the **ProjectOperatorType** type.

Element	Type	Description
InputStream	StreamReferenceType	The input stream for this Project element.
OutputStream	StreamDefinitionType	The output stream defined by this Project element.
ProjectExpression	ProjectExpressionContainerType	An arbitrary number of project expressions may be specified. Each expression can contain multiple operations. Each project expression evaluates the value of one field in the resulting output event.

2.2.3.2.2.3.1 ProjectExpressionContainerType

The **ProjectExpressionContainerType** type contains a single expression that is used by the **ProjectOperatorType** type.

The following code is the XSD for the **ProjectExpressionContainerType** type.

```

<xs:complexType name="ProjectExpressionContainerType">
  <xs:annotation>
    <xs:documentation>A project expression contains a single expression
    that determines the value of a new event field. It extends the base
    container type by adding an attribute to assign a name to that new
    field. This is also a base class for other operators' expressions that
    result in new event fields.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ExpressionContainerType">
      <xs:attribute name="OutputField" type="xs:anyURI" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    </xs:complexContent>
  </xs:complexType>

```

The following table describes the attributes for the **ProjectExpressionContainerType** type.

Attribute	Type	Description
OutputField	xs:anyURI	The name to be assigned to the new field in the event payload that contains the result of the application of the project expressions specified in the containing Project element.

2.2.3.2.2.3.2 Example

```

<Project Name="p1">
  <InputStream Name="timeavg" />
  <OutputStream Name="Project.avg" />
  <ProjectExpression OutputField="powerPhaseA">
    <InputField Name="Power Phase A" />
  </ProjectExpression>
  <ProjectExpression OutputField="powerPhaseB">
    <InputField Name="Power Phase B" />
  </ProjectExpression>
  <ProjectExpression OutputField="powerPhaseC">
    <InputField Name="Power Phase C" />
  </ProjectExpression>
  <ProjectExpression OutputField="phaseAverage">
    <Divide>
      <Add>
        <Add>
          <InputField Name="powerPhaseA" />
          <InputField Name="powerPhaseB" />
        </Add>
        <InputField Name="powerPhaseC" />
      </Add>
      <Constant Type="cep:/Server/Application/system/EventType/System.Single"
        Value="3" />
    </Divide>
  </ProjectExpression>
</Project>

```

2.2.3.2.2.4 SelectOperatorType

The **SelectOperatorType** type is used to filter inputs and to select a subset of the inputs for output.

The following code is the XSD for the **SelectOperatorType** type.

```

<xs:complexType name="SelectOperatorType">
  <xs:annotation>
    <xs:documentation>A select expression contains exactly one filter
      expression.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">

```

```

<xs:extension base="tns:OperatorBaseType">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="InputStream"
      type="tns:StreamReferenceType" />
    <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
      type="tns:StreamDefinitionType" />
    <xs:element minOccurs="1" maxOccurs="1" name="FilterExpression"
      type="tns:ExpressionContainerType" />
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following table describes the elements for the **SelectOperatorType** type.

Element	Type	Description
InputStream	StreamReferenceType	The input stream for this Select element.
OutputStream	StreamDefinitionType	The output stream defined by this Select element.
FilterExpression	ExpressionContainerType	A filter expression. This expression defines which of the inputs will be selected to be placed on the output stream. Only events that fulfill the filter expression will be output by the Select operator.

2.2.3.2.2.4.1 Example

```

<Select Name="SelectOperator1">
  <InputStream Name="import1"></InputStream>
  <OutputStream Name="select1"></OutputStream>
  <FilterExpression>
    <Equal>
      <Modulo>
        <InputField Name="Field1"></InputField>
        <Constant Value="3"
          Type="cep:/Server/Application/system/
Type/System.Int32">
          </Constant>
        </Modulo>
        <Constant Value="0"
          Type="cep:/Server/Application/system/EventType/System.Int32">
          </Constant>
        </Equal>
      </FilterExpression>
    </Select>

```

2.2.3.2.2.5 JoinOperatorType

The **JoinOperatorType** type is used to join two inputs based on an expression.

The following code is the XSD for the **JoinOperatorType** type.

```

<xs:complexType name="JoinOperatorType">
  <xs:annotation>
    <xs:documentation>A Join element has two inputs and one output. The
    join predicate is specified as a child element. The join can include
    zero or more ProjectExpressions
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:OperatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="2" maxOccurs="2" name="InputStream"
          type="tns:StreamReferenceType" />
        <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
          type="tns:StreamDefinitionType" />
        <xs:element minOccurs="1" maxOccurs="1" name="JoinPredicate"
          type="tns:ExpressionContainerType" />
        <xs:element minOccurs="0" maxOccurs="unbounded"
          name="ProjectExpression"
          type="tns:ProjectExpressionContainerType" />
      </xs:sequence>
      <xs:attribute name="JoinType">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="LeftOuter" />
            <xs:enumeration value="RightOuter" />
            <xs:enumeration value="FullOuter" />
            <xs:enumeration value="LeftAnti" />
            <xs:enumeration value="RightAnti" />
            <xs:enumeration value="LeftSemi" />
            <xs:enumeration value="RightSemi" />
            <xs:enumeration value="LeftAntiSemi" />
            <xs:enumeration value="RightAntiSemi" />
            <xs:enumeration value="Inner" />
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="PointEvents" type="xs:boolean" use="optional"
        default="false" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following tables describe the elements and attributes for the **JoinOperatorType** type.

Element	Type	Description
InputStream	StreamReferenceType	The input stream for the specified Join element. A join has exactly two input streams.
OutputStream	StreamDefinitionType	The output stream defined by the specified Join element.
JoinPredicate	ExpressionContainerType	The element that contains the expression on which to base the join.

Element	Type	Description
ProjectExpression	ProjectExpressionContainerType	The element with which the join operator can optionally define project expressions on the join result. Each project expression computes the value for a single field in the output event from the values of the two input events. The result event of the join operator contains exactly the set of all project expressions.

Attribute	Type	Description
JoinType	xs:string (restriction)	This enumeration indicates the type of join that will be performed. <ul style="list-style-type: none"> ▪ "LeftOuter" ▪ "RightOuter" ▪ "FullOuter" ▪ "LeftAnti" ▪ "RightAnti" ▪ "LeftSemi" ▪ "RightSemi" ▪ "LeftAntiSemi" ▪ "RightAntiSemi" ▪ "Inner"
PointEvents	xs:boolean	This attribute can be set to true if both input streams of a join operator contain only point events. Setting it to true will result in a different Join implementation being used by the engine but will not affect the correctness of the results.

2.2.3.2.2.5.1 Example

```

<Join Name="join1" JoinType="Inner" PointEvents="true">
  <InputStream Name="alterlifetime1"></InputStream>
  <InputStream Name="alterlifetime2"></InputStream>
  <OutputStream Name="join1"></OutputStream>
  <JoinPredicate>
    <Equal>
      <Compare>
        <InputField Name="GroupId" StreamName="alterlifetime1"></InputField>
        <InputField Name="GroupId" StreamName="alterlifetime2"></InputField>
      </Compare>
      <Constant Type="cep:/Server/Application/system/EventType/System.Int32"
        Value="0"></Constant>
    </Equal>
  </JoinPredicate>
</Join>

```

```

    </Equal>
  </JoinPredicate>
  <ProjectExpression OutputField="UserId">
    <InputField Name="UserId" StreamName="alterlifetime2"></InputField>
  </ProjectExpression>
  <ProjectExpression OutputField="SegmentHitLogicId">
    <InputField Name="SegmentHitLogicId" StreamName="alterlifetime2">
      </InputField>
    </ProjectExpression>
  <ProjectExpression OutputField="CountSegmentHitLogicId">
    <InputField Name="CountSegmentHitLogicId" StreamName="alterlifetime2">
      </InputField>
    </ProjectExpression>
  </Join>

```

2.2.3.2.2.6 UnionOperatorType

A **UnionOperatorType** type takes multiple input streams and places them on a single output stream.

The following code is the XSD for the **UnionOperatorType** type.

```

<xs:complexType name="UnionOperatorType">
  <xs:annotation>
    <xs:documentation>A union operator funnels multiple input stream into
    one output stream.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:OperatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="2" maxOccurs="unbounded"
          name="InputStream"
          type="tns:StreamReferenceType" />
        <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
          type="tns:StreamDefinitionType" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table describes the elements for the **UnionOperatorType** type.

Element	Type	Description
InputStream	StreamReferenceType	The input streams for this Union element.
OutputStream	StreamDefinitionType	The output stream defined by this Union element.

2.2.3.2.2.6.1 Example

```

<Union Name="UnionOperator">
  <InputStream Name="Multicast1"></InputStream>
  <InputStream Name="Multicast2"></InputStream>

```

```

    <OutputStream Name="union1"></OutputStream>
  </Union>

```

2.2.3.2.2.7 AggregationOperatorType

The **AggregationOperatorType** type is used to define an arithmetic aggregation of inputs to produce an output stream.

The following code is the XSD for the **AggregationOperatorType** type.

```

<xs:complexType name="AggregationOperatorType">
  <xs:annotation>
    <xs:documentation>An aggregate element has one or more aggregate
      expressions, each yielding a new column that represents the
      aggregation result.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:OperatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="InputStream"
          type="tns:StreamReferenceType" />
        <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
          type="tns:StreamDefinitionType" />
        <xs:element minOccurs="1" maxOccurs="unbounded"
          name="AggregateExpression"
          type="tns:AggregateExpressionContainerType" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table describes the elements for the **AggregationOperatorType** type.

Element	Type	Description
InputStream	StreamReferenceType	The input stream for this Aggregate element.
OutputStream	StreamDefinitionType	The output stream defined by this Aggregate element.
AggregateExpression	AggregateExpressionContainerType	A collection of elements of type AggregateExpressionContainer . The aggregation operator can define one or more aggregation expressions.

2.2.3.2.2.7.1 AggregateExpressionContainerType

The **AggregateExpressionContainerType** type is used to contain an aggregation expression.

The following code is the XSD for the **AggregateExpressionContainerType** type.


```

<xs:complexType name="AggregateExpressionContainerType">
  <xs:annotation>
    <xs:documentation>An aggregate expression specifies the aggregate
    operation as an attribute and the expression to apply the function
    on (usually one of the input fields).</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:group minOccurs="0" maxOccurs="1" ref="tns:AnyExpression" />
  </xs:sequence>
  <xs:attribute name="OutputField" type="xs:anyURI" use="required" />
  <xs:attribute name="AggregateFunction" type="tns:AggregateFunctionType" />
</xs:complexType>

```

The following tables describe the elements and attributes for the **AggregateExpressionContainerType** type.

Element	Type	Description
(group)	AnyExpression	Zero or one element of the AnyExpression group. The aggregation will be carried out on the values resulting from the expression evaluation over the input events. The count aggregate does not operate on an expression; rather, it counts entire events. Thus, no expression is required if the aggregation function is count.

Attribute	Type	Description
OutputField	xs:anyURI	The output field that will carry the aggregation result of this aggregation expression.
AggregateFunction	AggregateFunctionType	The function that will be used to aggregate the values computed from the input using the expression.

2.2.3.2.2.7.1.1 AggregateFunctionType

The **AggregateFunctionType** type is an enumeration that enumerates the available functions for numeric aggregation.

The following code is the XSD for the **AggregateFunctionType** type.

```

<xs:simpleType name="AggregateFunctionType">
  <xs:annotation>
    <xs:documentation>The type of aggregation.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Min" />
    <xs:enumeration value="Max" />
    <xs:enumeration value="Avg" />
    <xs:enumeration value="Sum" />
    <xs:enumeration value="Count" />
  </xs:restriction>
</xs:simpleType>

```

The enumeration values of the **AggregateFunctionType** type represent the common numeric aggregation functions, minimum, maximum, average, sum, and count, respectively.

2.2.3.2.2.7.2 Example

```
<Aggregate Name="avg1">
  <InputStream Name="appin" />
  <OutputStream Name="appout" />
  <AggregateExpression OutputField="Average power consumption"
    AggregateFunction="Avg">
    <InputField Name="Power" />
  </AggregateExpression>
</Aggregate>
```

2.2.3.2.2.8 AlterLifeTimeOperatorType

The **AlterLifeTimeOperatorType** type is used to create a time window of events that are passed to the output. It does that through the alteration of the events' start timestamp and lifetime period.

The following code is the XSD for the **AlterLifeTimeOperatorType** type.

```
<xs:complexType name="AlterLifeTimeOperatorType">
  <xs:annotation>
    <xs:documentation>An AlterLifetime operator defines two expressions:
    One for the new start time and one for the new life time of the event.
    At least one of these must be specified.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:OperatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="InputStream"
          type="tns:StreamReferenceType" />
        <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
          type="tns:StreamDefinitionType" />
        <xs:element minOccurs="0" maxOccurs="1"
          name="StartTimeExpression"
          type="tns:ExpressionContainerType" />
        <xs:element minOccurs="0" maxOccurs="1"
          name="LifeTimeExpression"
          type="tns:ExpressionContainerType" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table describes the elements for the **AlterLifeTimeOperatorType** type.

Element	Type	Description
InputStream	StreamReferenceType	The input stream for this AlterLifeTime element.
OutputStream	StreamDefinitionType	The output stream defined by this

Element	Type	Description
		AlterLifeTime element.
StartTimeExpression	ExpressionContainerType	The expression that is used to compute the new start time of the specified event.
LifeTimeExpression	ExpressionContainerType	The expression for the new lifetime of the specified event. At least one of StartTimeExpression and LifeTimeExpression MUST be specified.

2.2.3.2.2.8.1 Example

```

<AlterLifeTime Name="alt2">
  <InputStream Name="altin" />
  <OutputStream Name="onehour" />
  <StartTimeExpression>
    <ValidStartTime />
  </StartTimeExpression>
  <LifeTimeExpression>
    <Constant Type="cep:/Server/Application/system/EventType/System.TimeSpan"
      Value="PT3600S" />
  </LifeTimeExpression>
</AlterLifeTime>

```

2.2.3.2.2.9 GroupAndApplyOperatorType

The **GroupAndApplyOperatorType** type is used to divide inputs into groups and then apply the same sub-query to each group.

The following code is the XSD for the **GroupAndApplyOperatorType** type.

```

<xs:complexType name="GroupAndApplyOperatorType">
  <xs:annotation>
    <xs:documentation>
      Implements the Group and Apply operator. One or more grouping
      expressions determine the event partitions. The operator graph in
      the Apply element will be applied to each group separately. The
      grouping expression is of the same type as the project expression:
      it can contain any expression, but it must assign a field name to
      that expression result.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:OperatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="InputStream"
          type="tns:StreamReferenceType" />
        <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
          type="tns:StreamDefinitionType" />
        <xs:element minOccurs="1" maxOccurs="unbounded"
          name="GroupingExpression"
          type="tns:ProjectExpressionContainerType" />
        <xs:element minOccurs="1" maxOccurs="1" name="Apply" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        type="tns:ApplyBranchType">
<xs:key name="ApplyStreamKey">
  <xs:annotation>
    <xs:documentation>Stream identifier to be used in the
operators of that apply element.</xs:documentation>
  </xs:annotation>
  <xs:selector xpath="./*/tns:OutputStream" />
  <xs:field xpath="@Name" />
</xs:key>
<xs:keyref name="ApplyStreamKeyref"
refer="tns:ApplyStreamKey">
  <xs:annotation>
    <xs:documentation>Stream reference for operators.
A stream reference has to match a stream identifier
in order to connect operators.</xs:documentation>
  </xs:annotation>
  <xs:selector xpath="./*/tns:InputStream" />
  <xs:field xpath="@Name" />
</xs:keyref>
</xs:element>
</xs:sequence>
<xs:attribute name="AddGroupingFields" type="xs:boolean"
use="optional" default="false" />
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following table describes the elements for the **GroupAndApplyOperatorType** type.

Element	Type	Description
InputStream	StreamReferenceType	The input stream for the specified operator.
OutputStream	StreamDefinitionType	The output stream defined by the specified operator.
GroupingExpression	ProjectExpressionContainerType	The expression that defines which events are partitioned into the same group. For each distinct result of the grouping expression, a separate group is created.
Apply	ApplyBranchType	The sub-query that will be applied to each group separately.

Attribute	Type	Description
AddGroupingField	xs:boolean	If this flag is set, the result of the grouping expressions will be added to the events' payloads in the operator's output stream.

2.2.3.2.2.9.1 ApplyBranchType

The **ApplyBranchType** type is used to specify the operations that are to be applied to each branch of the **Group and Apply** operator.

The following code is the XSD for the **ApplyBranchType** type.

```
<xs:complexType name="ApplyBranchType">
  <xs:annotation>
    <xs:documentation>The Apply element encapsulates the apply operator
    graph of the Group and Apply operator. It must have exactly one input
    and one output, which are terminated by elements of type ApplyInputType
    and ApplyOutputType. These elements are named ImportOperator and
    ExportOperator to be able to re-use existing query templates as apply
    branches. However, their type here is different from
    query-template-level imports and exports in that they do not require a
    type specification.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="ApplyInput"
      type="tns:ApplyInputType" />
    <xs:element minOccurs="1" maxOccurs="1" name="ApplyOutput"
      type="tns:ApplyOutputType" />
    <xs:group minOccurs="0" maxOccurs="unbounded"
      ref="tns:AnyOperator" />
  </xs:sequence>
</xs:complexType>
```

The following table describes the elements for the **ApplyBranchType** type.

Element	Type	Description
ApplyInput	ApplyInputType	The input stream definition for the apply query graph.
ApplyOutput	ApplyOutputType	The output stream definition for the apply query graph.
(group)	AnyOperator	The set of operators that define the apply sub-query.

2.2.3.2.2.9.1.1 ApplyInputType

The **ApplyInputType** type is used to define the input operator for an apply branch.

The following code is the XSD for the **ApplyInputType** type.

```
<xs:complexType name="ApplyInputType">
  <xs:annotation>
    <xs:documentation>Input terminator of the apply operator graph.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:TerminatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
          type="tns:StreamDefinitionType" />
      </xs:sequence>
      <xs:attribute name="Name" type="xs:anyURI" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following tables describe the elements and attributes for the **ApplyInputType** type.

Element	Type	Description
OutputStream	StreamDefinitionType	The output stream defined by this ApplyInput operator. Further operators in the apply branch can now refer to this stream with their input streams.

Attribute	Type	Description
Name	xs:anyURI	The assigned name by which this ApplyInput operator will be referenced.

2.2.3.2.2.9.1.2 ApplyOutputType

The **ApplyOutputType** type is used to define the output operator for an apply branch.

The following code is the XSD for the **ApplyOutputType** type.

```
<xs:complexType name="ApplyOutputType">
  <xs:annotation>
    <xs:documentation>Output terminator of the apply operator graph.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:extension base="tns:TerminatorBaseType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="InputStream"
          type="tns:StreamReferenceType" />
      </xs:sequence>
      <xs:attribute name="Name" type="xs:anyURI" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following tables describe the elements and attributes for the **ApplyOutputType** type.

Element	Type	Description
InputStream	StreamReferenceType	The input stream into this ApplyOutput operator. Refers to another operator's OutputStream object in the apply branch.

Attribute	Type	Description
Name	xs:anyURI	The assigned name by which this ApplyOutput operator will be referenced.

2.2.3.2.2.9.2 Example

```

<GroupAndApply Name="GroupAndApply1">
  <InputStream Name="import"></InputStream>
  <OutputStream Name="ga"></OutputStream>
  <GroupingExpression OutputField="GroupExpr1">
    <Modulo>
      <InputField Name="Field1"></InputField>
      <Constant Value="3" Type=
        "cep:/Server/Application/system/EventType/System.Int32">
      </Constant>
    </Modulo>
  </GroupingExpression>
  <Apply>
    <ApplyInput Name="appin">
      <OutputStream Name="applyin"></OutputStream>
    </ApplyInput>
    <ApplyOutput Name="appout">
      <InputStream Name="select"></InputStream>
    </ApplyOutput>
    <Select Name="SelectOperator">
      <InputStream Name="applyin"></InputStream>
      <OutputStream Name="select"></OutputStream>
      <FilterExpression>
        <Equal>
          <Modulo>
            <InputField Name="Field1"></InputField>
            <Constant Value="4" Type=
              "cep:/Server/Application/system/EventType/System.Int32">
            </Constant>
          </Modulo>
          <Constant Value="0" Type=
            "cep:/Server/Application/system/EventType/System.Int32">
          </Constant>
        </Equal>
      </FilterExpression>
    </Select>
  </Apply>
</GroupAndApply>

```

2.2.3.2.2.10 TopKOperatorType

The **TopKOperatorType** type performs a ranking based on observed or computed field values and returns only the top K in number, where K is user-specified in the definition. In the case of a tie, all events with the same rank are output so that the operation is always deterministic.

The following code is the XSD for the **TopKOperator** type.

```

<xs:complexType name="TopKOperatorType">
  <xs:annotation>
    <xs:documentation>TopK operator. The K is specified by the required
      RankDepth attribute. The calculated rank can be projected in the output
      of the operator by specifying a field name through the attribute
      RankOutputField. The rank is calculated according to the value of the
      rank expression, its datatype, and the specified ordering. If more
      than one rank expression is specified, they are evaluated subsequently,
      i.e., if one rank expression evaluates for a tie for any two events,

```

```

        the next expression in the sequence is evaluated, etc.
    </xs:documentation>
</xs:annotation>
<xs:complexContent mixed="false">
    <xs:extension base="tns:OperatorBaseType">
        <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="InputStream"
                type="tns:StreamReferenceType" />
            <xs:element minOccurs="1" maxOccurs="1" name="OutputStream"
                type="tns:StreamDefinitionType" />
            <xs:element minOccurs="1" maxOccurs="unbounded"
                name="RankExpression"
                type="tns:RankExpressionContainerType" />
        </xs:sequence>
        <xs:attribute name="RankDepth" type="xs:int" use="required" />
        <xs:attribute name="RankOutputField" type="xs:anyURI" use="optional"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following tables describe the elements and attributes for the **TopKOperator** type.

Element	Type	Description
InputStream	StreamReferenceType	The input stream for this TopK operator.
OutputStream	StreamDefinitionType	The output stream defined by this TopK operator.
RankExpression	RankExpressionContainerType	The expression whose result will be used to determine the rank of the input events.

Attribute	Type	Description
RankDepth	xs:int	An integer that specifies the number of top-ranked events to include in the output.
RankOutputField	xs:anyURI	The attribute that specifies the name of the optional output field in which to include the rank in the output events' payload.

2.2.3.2.2.10.1 RankExpressionContainerType

The **RankExpressionContainerType** type is used to specify one or more ranking expressions.

The following code is the XSD for the **RankExpressionContainerType** type.

```

<xs:complexType name="RankExpressionContainerType">
    <xs:annotation>
        <xs:documentation>A rank expression contains a single expression that
            is to be used to determine the rank in a TopK operator. It extends the
            base container type by adding an attribute to specify the ordering.
        </xs:documentation>
    </xs:annotation>

```



```

<xs:complexContent mixed="false">
  <xs:extension base="tns:ExpressionContainerType">
    <xs:attribute name="Order" type="tns:RankOrderType"
      use="required" />
  </xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following table describes the attributes for the **RankExpressionContainerType** type.

Attribute	Type	Description
Order	RankOrderType	The attribute that specifies whether values are ordered in ascending or descending order.

2.2.3.2.2.10.1.1 RankOrderType

The **RankOrderType** type is an enumeration containing the values on which the ranking can be ordered.

The following code is the XSD for the **RankOrderType** type.

```

<xs:simpleType name="RankOrderType">
  <xs:annotation>
    <xs:documentation>The ordering of a rank expression can be ascending
      or descending.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Ascending" />
    <xs:enumeration value="Descending" />
  </xs:restriction>
</xs:simpleType>

```

The enumeration values for the **RankOrderType** type are as follows.

Value	Description
Ascending	Ranked in ascending order.
Descending	Ranked in descending order.

2.2.3.2.2.10.2 Example

```

<TopK Name="TopK1" RankDepth="3" RankOutputField="Field3">
  <InputStream Name="import1"></InputStream>
  <OutputStream Name="TopKOutput1"></OutputStream>
  <RankExpression Order="Ascending">
    <InputField Name="Field1" StreamName="import1"></InputField>
  </RankExpression>
  <RankExpression Order="Ascending">
    <InputField Name="Field2" StreamName="import1"></InputField>
  </RankExpression>

```

</TopK>

2.2.3.2.3 Additional Types, Groups, and AttributeGroups

2.2.3.2.3.1 BuiltinType

The **BuiltinType** type contains an enumeration of data types that are used for other elements within the system.

The following code is the XSD for the **BuiltinType** type.

```
<xs:simpleType name="BuiltinType">
  <xs:annotation>
    <xs:documentation>List of all natively supported types, as relative URI.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="System.Boolean" />
    <xs:enumeration value="System.Char" />
    <xs:enumeration value="System.SByte" />
    <xs:enumeration value="System.Int16" />
    <xs:enumeration value="System.Int32" />
    <xs:enumeration value="System.Int64" />
    <xs:enumeration value="System.Byte" />
    <xs:enumeration value="System.UInt16" />
    <xs:enumeration value="System.UInt32" />
    <xs:enumeration value="System.UInt64" />
    <xs:enumeration value="System.Decimal" />
    <xs:enumeration value="System.Single" />
    <xs:enumeration value="System.Double" />
    <xs:enumeration value="System.Guid" />
    <xs:enumeration value="System.DateTime" />
    <xs:enumeration value="System.TimeSpan" />
    <xs:enumeration value="System.String" />
    <xs:enumeration value="System.Byte[]" />
  </xs:restriction>
</xs:simpleType>
```

The values of the enumeration represent the types that are used in **EventField** definitions.

2.2.3.2.3.2 OperatorBaseType

The **OperatorBaseType** type is a base type on which other operators are defined with extension or restriction.

The following code is the XSD for the **OperatorBaseType** type.

```
<xs:complexType name="OperatorBaseType">
  <xs:annotation>
    <xs:documentation>Operator base type. Every operator has a name.
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="Name" type="xs:anyURI" use="required" />
</xs:complexType>
```

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

The following table describes the attributes for the **OperatorBaseType** type.

Attribute	Type	Description
Name	xs:anyURI	The name of the specified operator.

2.2.3.2.3.3 StreamReferenceType

The **StreamReferenceType** type is used to refer to a name that has been defined by a **StreamDefinition** element in another operator.

The following code is the XSD for the **StreamReferenceType** type.

```
<xs:complexType name="StreamReferenceType">
  <xs:annotation>
    <xs:documentation>ID that refers to a stream.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="Name" type="xs:anyURI" use="required" />
</xs:complexType>
```

The following table describes the attributes for the **StreamReferenceType** type.

Attribute	Type	Description
Name	xs:anyURI	Name of the stream that is referred to.

2.2.3.2.3.4 StreamDefinitionType

The **StreamDefinitionType** type is used to define a stream. It denotes an output stream from an operator in a **QueryTemplate**. It defines a name that can be referenced by the **StreamReference** element in another operator.

The following code is the XSD for the **StreamDefinitionType** type.

```
<xs:complexType name="StreamDefinitionType">
  <xs:annotation>
    <xs:documentation>ID that defines a stream. Stream here denotes the
    connection between operators.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="Name" type="xs:anyURI" use="required" />
</xs:complexType>
```

The following table describes the attributes for the **StreamDefinitionType** type.

Attribute	Type	Description
Name	xs:anyURI	Name of the stream being defined.

2.2.3.2.3.5 ExpressionContainerType

The **ExpressionContainerType** type represents a base container for one of the expressions in the **AnyExpression** group.

The following code is the XSD for the **ExpressionContainerType** type.

```
<xs:complexType name="ExpressionContainerType">
  <xs:annotation>
    <xs:documentation>Expression container type. An element of this type
    must contain exactly one expression of any type.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:group minOccurs="1" maxOccurs="1" ref="tns:AnyExpression" />
  </xs:sequence>
</xs:complexType>
```

The following table describes the elements for the **ExpressionContainerType** type.

Element	Type	Description
(group)	AnyExpression	The expression contained in the specified container. This MUST be one of the expressions defined in the AnyExpression group.

2.2.3.2.3.6 TerminatorBaseType

The **TerminatorBaseType** type is used as a base type for stream termination elements, such as **Import**, **Export**, **ApplyInput**, and **ApplyOutput**.

The following code is the XSD for the **TerminatorBaseType** type.

```
<xs:complexType name="TerminatorBaseType">
  <xs:annotation>
    <xs:documentation>Base type for stream termination elements.
  </xs:documentation>
  </xs:annotation>
</xs:complexType>
```

The **TerminatorBaseType** type defines no elements or attributes.

2.2.3.2.3.7 AnyExpression Group

The **AnyExpression** group is the group that contains all of the expressions that are available for use in operations.

The following code is the XSD for the **AnyExpression** group.

```
<xs:group name="AnyExpression">
  <xs:annotation>
    <xs:documentation>Placeholder for exactly one expression element of
    any type.</xs:documentation>
  </xs:annotation>
</xs:group>
```

```

</xs:annotation>
<xs:choice>
  <xs:element name="Abs" type="tns:UnaryArithmeticExpression" />
  <xs:element name="Add" type="tns:BinaryArithmeticExpression" />
  <xs:element name="And" type="tns:BinaryExpression" />
  <xs:element name="Compare" type="tns:ComparisonExpression" />
  <xs:element name="Condition" type="tns:ConditionExpression" />
  <xs:element name="Constant" type="tns:ConstantExpression" />
  <xs:element name="Convert" type="tns:ConvertExpression" />
  <xs:element name="Divide" type="tns:BinaryArithmeticExpression" />
  <xs:element name="Equal" type="tns:ComparisonExpression" />
  <xs:element name="EventKind" type="tns:SystemFieldExpression" />
  <xs:element name="GreaterThan" type="tns:ComparisonExpression" />
  <xs:element name="GreaterThanOrEqual"
    type="tns:ComparisonExpression" />
  <xs:element name="Hash" type="tns:HashExpression" />
  <xs:element name="InputField" type="tns:InputFieldExpression" />
  <xs:element name="LessThan" type="tns:ComparisonExpression" />
  <xs:element name="LessThanOrEqual" type="tns:ComparisonExpression" />
  <xs:element name="Max" type="tns:NaryArithmeticExpression" />
  <xs:element name="MethodCall" type="tns:MethodCallExpression" />
  <xs:element name="Min" type="tns:NaryArithmeticExpression" />
  <xs:element name="Modulo" type="tns:BinaryArithmeticExpression" />
  <xs:element name="Multiply" type="tns:BinaryArithmeticExpression" />
  <xs:element name="NewValidEndTime" type="tns:SystemFieldExpression" />
  <xs:element name="Negate" type="tns:UnaryArithmeticExpression" />
  <xs:element name="Not" type="tns:UnaryExpression" />
  <xs:element name="NotEqual" type="tns:ComparisonExpression" />
  <xs:element name="Or" type="tns:BinaryExpression" />
  <xs:element name="Subtract" type="tns:BinaryArithmeticExpression" />
  <xs:element name="ValidStartTime" type="tns:SystemFieldExpression" />
  <xs:element name="ValidEndTime" type="tns:SystemFieldExpression" />
</xs:choice>
</xs:group>

```

The following table describes the elements for the **AnyExpression** group.

Element	Type	Description
Abs	UnaryArithmeticExpression	The absolute value of a single operand.
Add	BinaryArithmeticExpression	An expression that performs an addition operation on two operands.
And	BinaryExpression	An expression that performs a logical And operation on two operands.
Compare	ComparisonExpression	An expression that compares two operands. The result is negative if the first operand is less than the second operand, positive if the first operand is greater than the second, and zero if the two operands are equal.
Condition	ConditionExpression	An expression that performs a conditional evaluation. The first expression is evaluated to a Boolean value. If true, then the condition expression is assigned the second expression; otherwise, it is assigned the third expression.

Element	Type	Description
Constant	ConstantExpression	An expression that specifies a constant.
Convert	ConvertExpression	An expression that converts a value to a new type.
Divide	BinaryArithmeticExpression	An expression that performs a division operation on two operands.
Equal	ComparisonExpression	An expression that compares two operands for equality.
EventKind	SystemFieldExpression	An expression that returns the event type as a number: 1 - Event is a CTI 2 - Event is an Insert 3 - Event is a Retract 4 - Event is an Expand
GreaterThan	ComparisonExpression	An expression that tests for Greater Than condition on two operands.
GreaterThanOrEqual	ComparisonExpression	An expression that tests for the Greater Than Or Equal condition on two operands.
Hash	HashExpression	An expression that specifies a hash expression. It computes a hash value on n child operands.
InputField	InputFieldExpression	An expression that returns the value of a field in an event's payload.
LessThan	ComparisonExpression	An expression that tests for the Less Than condition on two operands.
LessThanOrEqual	ComparisonExpression	An expression that tests for the Less Than Or Equal condition on two operands.
Max	NaryArithmeticExpression	An expression that determines the maximum value of n operands.
MethodCall	MethodCallExpression	An expression that calls a method with n child operands as its arguments.
Min	NaryArithmeticExpression	An expression that determines the minimum value of n operands.
Modulo	BinaryArithmeticExpression	An expression that returns the remainder of integer division of one number by another number.
Multiply	BinaryArithmeticExpression	An expression that performs a multiplication operation on two operands.
NewValidEndTime	SystemFieldExpression	An expression that returns the new valid end time system field.
Negate	UnaryArithmeticExpression	An expression that negates a numeric

Element	Type	Description
		expression.
Not	UnaryExpression	An expression that returns the logical Not operation on an operand.
NotEqual	ComparisonExpression	An expression that determines whether two operands are Not Equal.
Or	BinaryExpression	An expression that performs a logical Or operation on two operands.
Subtract	BinaryArithmeticExpression	An expression that performs subtraction on two operands.
ValidEndTime	SystemFieldExpression	An expression that refers to the end time of the specified event.
ValidStartTime	SystemFieldExpression	An expression that refers to the start time of the specified event.

2.2.3.2.3.7.1 UnaryArithmeticExpression

The **UnaryArithmeticExpression** type is used as a base type for unary arithmetic expressions.

The following code is the XSD for the **UnaryArithmeticExpression** type.

```
<xs:complexType name="UnaryArithmeticExpression">
  <xs:annotation>
    <xs:documentation>Unary arithmetic expression. Has 1 child
    expression and no attributes.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:UnaryExpression">
      <xs:sequence>
        <xs:group minOccurs="1" maxOccurs="1" ref="tns:AnyExpression" />
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

The following table describes the elements for the **UnaryArithmeticExpression** type.

Element	Type	Description
(group)	AnyExpression	The single child operand.

2.2.3.2.3.7.2 BinaryArithmeticExpression

The **BinaryArithmeticExpression** type is used as a base type for binary arithmetic expressions.

The following code is the XSD for the **BinaryArithmeticExpression** type.

```

<xs:complexType name="BinaryArithmeticExpression">
  <xs:annotation>
    <xs:documentation>Binary arithmetic expression. Has 2 child
    expressions and no attributes.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:BinaryExpression">
      <xs:sequence>
        <xs:group minOccurs="2" maxOccurs="2" ref="tns:AnyExpression" />
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

The following table describes the elements for the **BinaryArithmeticExpression** type.

Element	Type	Description
(group)	AnyExpression	The two child operands.

2.2.3.2.3.7.3 ComparisonExpression

The **ComparisonExpression** type is used to compare two expressions.

The following code is the XSD for the **ComparisonExpression** type.

```

<xs:complexType name="ComparisonExpression">
  <xs:annotation>
    <xs:documentation>Comparison expression. Compares two child expressions.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:BinaryExpression">
      <xs:sequence>
        <xs:group minOccurs="2" maxOccurs="2" ref="tns:AnyExpression" />
      </xs:sequence>
      <xs:attribute name="Culture" type="xs:string" use="optional" />
      <xs:attribute name="IgnoreCase" type="xs:boolean" use="optional" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

The following tables describe the elements and attributes for the **ComparisonExpression** type.

Element	Type	Description
(group)	AnyExpression	Exactly two expressions for the comparison operation.

Attribute	Type	Description
Culture	xs:string	The culture. The comparison is performed taking into account the culture

Attribute	Type	Description
		setting. The value is inherited from the field if it is not specified on the comparison expression. The comparison will fall into one of the following four cases: If the culture is specified here, that culture is used. If the culture is specified on the fields and they are both the same, that culture will be used. If the culture is specified on the fields and they are different, the operation will fail. If no culture is specified, byte by byte comparison will be used.
IgnoreCase	xs:boolean	The attribute that specifies whether or not to ignore case in the comparison of string values.

2.2.3.2.3.7.4 ConstantExpression

The **ConstantExpression** type is used to specify a constant as an input to another expression.

The following code is the XSD for the **ConstantExpression** type.

```

<xs:complexType name="ConstantExpression">
  <xs:annotation>
    <xs:documentation>Constant expression. Has no child expression.
    Contains type and value attributes.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:NullaryExpression">
      <xs:sequence />
      <xs:attributeGroup ref="tns:TypeIdentifier" />
      <xs:attribute name="Value" type="xs:string" use="optional" />
      <xs:attribute name="NullValue" type="xs:boolean" use="optional"
        default="false" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

The following table describes the attributes for the **ConstantExpression** type.

Attribute	Type	Description
(group)	TypeIdentifier	The type of the specified constant expression.
Value	xs:string	The XML representation of the value according to its type.
NullValue	xs:Boolean	An attribute that specifies whether or not this value is null.

2.2.3.2.3.7.5 ConvertExpression

The **ConvertExpression** type is used to convert an expression to a different type.

The following code is the XSD for the **ConvertExpression** type.

```

<xs:complexType name="ConvertExpression">
  <xs:annotation>
    <xs:documentation>Conversion expression. Converts one child
    expression into a type.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:UnaryExpression">
      <xs:sequence>
        <xs:group minOccurs="1" maxOccurs="1" ref="tns:AnyExpression" />
      </xs:sequence>
      <xs:attributeGroup ref="tns:TypeIdentifier" />
      <xs:attribute name="DateTimeKind" type="tns:DateTimeType"
        use="optional" />
      <xs:attributeGroup ref="tns:ExpressionReturnTypeFacets"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

The following tables describe the elements and attributes for the **ConvertExpression** type.

Element	Type	Description
(group)	AnyExpression	The expression for the unary arithmetic.

Attribute	Type	Description
(group)	TypeIdentifier	The type that the expression is to be converted to.
DateTimeKind	DateTimeType	The type of date-time. This can be either Coordinated Universal Time (UTC) or local time.
(group)	ExpressionReturnTypeFacets	Additional facets about the return type of the expression.

2.2.3.2.3.7.6 HashExpression

The **HashExpression** type is used to perform a hashing operation on any number of operands.

The following code is the XSD for the **HashExpression** type.

```

<xs:complexType name="HashExpression">
  <xs:annotation>
    <xs:documentation>Hash expression. Represents a hash value based on
    1..n child expressions.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:ExpressionBase">
      <xs:sequence>
        <xs:group minOccurs="1" maxOccurs="unbounded"
          ref="tns:AnyExpression" />
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>

```

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

The following table describes the elements for the **HashExpression** type.

Element	Type	Description
(group)	AnyExpression	The expressions that will be used to obtain the values over which the hash function will be computed.

2.2.3.2.3.7.7 InputFieldExpression

The **InputFieldExpression** type is used to identify an Input field from a specific stream.

The following code is the XSD for the **InputFieldExpression** type.

```
<xs:complexType name="InputFieldExpression">
  <xs:annotation>
    <xs:documentation>Input field expression. Has no child expression.
    Refers to a field in a stream by the field identifier.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:NullaryExpression">
      <xs:sequence />
      <xs:attributeGroup ref="tns:FieldIdentifier" />
      <xs:attributeGroup ref="tns:StreamIdentifier" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

The following table describes the attributes for the **InputFieldExpression** type.

Attribute	Type	Description
(group)	FieldIdentifier	The input field to be used.
(group)	StreamIdentifier	The event stream to be used to look up the specified field.

2.2.3.2.3.7.8 NaryArithmeticExpression

The **NaryArithmeticExpression** type is used as a base type for arithmetic expressions with an arbitrary number of operands.

The following code is the XSD for the **NaryArithmeticExpression** type.

```
<xs:complexType name="NaryArithmeticExpression">
  <xs:annotation>
    <xs:documentation>N-ary arithmetic expression. Has 1..n child
    expressions and arbitrary attributes.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:ExpressionBase">
```

```

    <xs:sequence>
      <xs:group minOccurs="1" maxOccurs="unbounded"
        ref="tns:AnyExpression" />
    </xs:sequence>
  </xs:restriction>
</xs:complexContent>
</xs:complexType>

```

The following table describes the elements for the **NaryArithmeticExpression** type.

Element	Type	Description
(group)	AnyExpression	The unbounded set of child operands.

2.2.3.2.3.7.9 MethodCallExpression

The **MethodCallExpression** type is used to call a .NET assembly to perform an operation on any number of operands. The possible methods include user-defined code as well as existing .NET methods.

The following code is the XSD for the **MethodCallExpression** type.

```

<xs:complexType name="MethodCallExpression">
  <xs:annotation>
    <xs:documentation>User-defined function. Its value is defined by a
      method of a class. 0..n input expressions can be passed to the
      method as parameters.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:ExpressionBase">
      <xs:sequence>
        <xs:group minOccurs="0" maxOccurs="unbounded"
          ref="tns:AnyExpression" />
      </xs:sequence>
      <xs:attribute name="Method" type="xs:string" use="required" />
      <xs:attribute name="Class" type="xs:string" use="required" />
      <xs:attribute name="Deterministic" type="xs:boolean" use="optional"
        default="false" />
      <xs:attributeGroup ref="tns:ExpressionReturnTypeFacets"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

```

The following tables describe the elements and attributes for the **MethodCallExpression** type.

Element	Type	Description
(group)	AnyExpression	The values from the evaluation of these expressions will be passed as parameters to the external method in the specified order.

Attribute	Type	Description
Method	xs:string	The name of the method to be called.
Class	xs:string	The name of the class that contains the method in a .NET assembly. This MUST be an assembly-qualified class name.
Deterministic	xs:boolean	An attribute that specifies whether or not the method is deterministic.
(group)	ExpressionReturnTypeFacets	An attribute that specifies additional information about the return type.

2.2.3.2.3.7.10 UnaryExpression

The **UnaryExpression** type is used to specify a single operand on which a unary operation may be performed.

The following code is the XSD for the **UnaryExpression** type.

```
<xs:complexType name="UnaryExpression">
  <xs:annotation>
    <xs:documentation>Unary expression. Has 1 child expression.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:ExpressionBase">
      <xs:sequence>
        <xs:group minOccurs="1" maxOccurs="1"
          ref="tns:AnyExpression" />
      </xs:sequence>
      <xs:anyAttribute namespace="##any" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

The following tables describe the elements and attributes for the **UnaryExpression** type.

Element	Type	Description
(group)	AnyExpression	The expression for the unary expression.

Attribute	Type	Description
anyAttribute	attributeGroup	A placeholder attribute group that enables extensions to the specified class to define specific attributes.

2.2.3.2.3.7.11 BinaryExpression

The **BinaryExpression** type is used as a base type for binary expressions.

The following code is the XSD for the **BinaryExpression** type.

```
<xs:complexType name="BinaryExpression">
  <xs:annotation>
    <xs:documentation>Binary expression. Has 2 child expressions and
      arbitrary attributes.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:ExpressionBase">
      <xs:sequence>
        <xs:group minOccurs="2" maxOccurs="2" ref="tns:AnyExpression" />
      </xs:sequence>
      <xs:anyAttribute namespace="##any" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

The following tables describe the elements and attributes for the **BinaryExpression** type.

Element	Type	Description
(group)	AnyExpression	Exactly two expressions for a binary operation.

Attribute	Type	Description
anyAttribute	attributeGroup	A placeholder attribute group that enables extensions to the specified class to define specific attributes.

2.2.3.2.3.7.12 SystemFieldExpression

The **SystemFieldExpression** type is used to define the elements for the system field access expressions: **ValidStartTime**, **ValidEndTime**, **NewValidEndTime**, and **EventKind**.

The following code is the XSD for the **SystemFieldExpression** type.

```
<xs:complexType name="SystemFieldExpression">
  <xs:annotation>
    <xs:documentation>System field expression. Has no child expression.
      Refers to a system field in a stream.</xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:NullaryExpression">
      <xs:sequence />
      <xs:attributeGroup ref="tns:StreamIdentifier" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

The following table describes the attributes for the **SystemFieldExpression** type.

Attribute	Type	Description
(group)	StreamIdentifier	The stream that contains the system field.

2.2.3.2.3.8 NullaryExpression

The **NullaryExpression** type is the base type for expressions that take no operands.

The following code is the XSD for the **NullaryExpression** type.

```
<xs:complexType name="NullaryExpression">
  <xs:annotation>
    <xs:documentation>Nullary expression. Has no child expressions.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent mixed="false">
    <xs:restriction base="tns:ExpressionBase">
      <xs:sequence />
      <xs:anyAttribute namespace="##any" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

The following table describes the attributes for the **NullaryExpression** type.

Attribute	Type	Description
anyAttribute	attributeGroup	A placeholder attribute group that enables extensions to the specified class to define specific attributes.

2.2.3.2.3.9 TypeIdentifier AttributeGroup

The **TypeIdentifier** attribute group is used to define an expression's type. It contains a single attribute to specify the type name.

The following code is the XSD for the **TypeIdentifier** attribute group.

```
<xs:attributeGroup name="TypeIdentifier">
  <xs:annotation>
    <xs:documentation>Refers to a data type in the Orinoco type system.
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="Type" type="xs:anyURI" use="required" />
</xs:attributeGroup>
```

The following table describes the attributes for the **TypeIdentifier** attribute group.

Attribute	Type	Description
Type	xs:anyURI	The URI that represents a type within a CEP Application object.

2.2.3.2.3.10 DateTimeType

The **DateTimeType** type is an enumeration that indicates whether date and time values are based on local time or on UTC.

The following code is the XSD for the **DateTimeType** type.

```
<xs:simpleType name="DateTimeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Utc" />
    <xs:enumeration value="Local" />
  </xs:restriction>
</xs:simpleType>
```

The following table describes the attributes for the **DateTimeType** type.

Attribute	Description
Utc	UTC time
Local	Local time

2.2.3.2.3.11 ExpressionReturnTypeFacets Attribute Group

The **ExpressionReturnTypeFacets** attribute group is used to specify additional information about types for which the CEP engine would not be able to infer that information on its own.

The following code is the XSD for the **ExpressionReturnTypeFacets** attribute group.

```
<xs:attributeGroup name="ExpressionReturnTypeFacets">
  <xs:annotation>
    <xs:documentation>Refers to a list of attributes that can be specified
    for those expressions where the CEP engine cannot derive return type
    facets</xs:documentation>
  </xs:annotation>
  <xs:attribute name="MaxSize" type="xs:unsignedInt" use="optional"/>
  <xs:attribute name="SizeFixed" type="xs:boolean" use="optional"/>
  <xs:attribute name="Nullable" type="xs:boolean" use="optional"/>
</xs:attributeGroup>
```

The following table describes the attributes for the **ExpressionReturnTypeFacets** attribute group.

Attribute	Type	Description
MaxSize	xs:unsignedInt	The maximum size – in bytes or characters – for byte arrays or strings, respectively.
SizeFixed	xs:boolean	An attribute that indicates whether the specified type is a fixed-size type.
Nullable	xs:boolean	An attribute that indicates whether the specified value is nullable.

2.2.3.2.3.12 StreamIdentifier AttributeGroup

The **StreamIdentifier** attribute group is used to identify a stream that has already been defined. This is necessary, for instance, in join predicate expressions, which can refer to event fields from multiple input streams.

The following code is the XSD for the **StreamIdentifier** attribute group.

```
<xs:attributeGroup name="StreamIdentifier">
  <xs:annotation>
    <xs:documentation>Refers to a stream by the stream name that was
      defined in the corresponding scope.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="StreamName" type="xs:anyURI" use="optional" />
</xs:attributeGroup>
```

The following table describes the attributes for the **StreamIdentifier** attribute group.

Attribute	Type	Description
StreamName	xs:anyURI	The name of a stream that has already been defined.

2.2.3.2.3.13 ExpressionBase

The **ExpressionBase** type is the base type on which other expressions are defined as an extension or a restriction.

The following code is the XSD for the **ExpressionBase** type.

```
<xs:complexType name="ExpressionBase">
  <xs:annotation>
    <xs:documentation>Expression base type. Can have 0..n child expressions.
  </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:group minOccurs="0" maxOccurs="unbounded" ref="tns:AnyExpression" />
  </xs:sequence>
  <xs:anyAttribute namespace="##any" />
</xs:complexType>
```

The following tables describe the elements and attributes for the **ExpressionBase** type.

Element	Type	Description
(group)	AnyExpression	An expression.

Attribute	Type	Description
anyAttribute	attributeGroup	A placeholder attribute group that enables extensions to the specified class to define specific attributes.

2.2.3.2.3.14 FieldIdentifier

The **FieldIdentifier** attribute group is used to identify a field by its name.

The following code is the XSD for the **FieldIdentifier** attribute group.

```
<xs:attributeGroup name="FieldIdentifier">
  <xs:annotation>
    <xs:documentation>Refers to a field within a stream type by its name.
  </xs:documentation>
</xs:annotation>
  <xs:attribute name="Name" type="xs:anyURI" use="required" />
</xs:attributeGroup>
```

The following table describes the attributes for the **FieldIdentifier** attribute group.

Attribute	Type	Description
Name	xs:anyURI	The URI for the name of a field that has already been defined.

2.2.3.3 Diagnostic Method Types

This section contains the types that are used by the diagnostic methods.

2.2.3.3.1 SetDiagnosticSettings

The following code is the XSD for the **SetDiagnosticSettings** element, which contains the in-line complex type definition.

```
<xs:element name="SetDiagnosticSettings">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="DiagnosticAspects"
        type="tns:DiagnosticAspects" />
      <xs:element minOccurs="0" name="DiagnosticLevel"
        type="tns:DiagnosticLevel" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

The following table lists and describes the attributes for the **SetDiagnosticSettings** element.

Attribute	Type	Description
DiagnosticAspects	DiagnosticAspects	An enumeration value that indicates which diagnostic aspects will be included in a diagnostic view, performance counters, tracing, or event tracing.
DiagnosticLevel	DiagnosticLevel	An enumeration value that indicates what level of verbosity of diagnostic events will be included in the diagnostic output.

2.2.3.3.1.1 DiagnosticAspects

The **DiagnosticAspects** type is an enumeration of the possible diagnostic aspects that can be enabled.

The following code is the XSD for the **DiagnosticAspects** type.

```
<xs:simpleType name="DiagnosticAspects">
  <xs:list>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="None">
          <xs:annotation>
            <xs:appinfo>
              <EnumerationValue xmlns=
                "http://schemas.microsoft.com/2003/10/Serialization/">0
              </EnumerationValue>
            </xs:appinfo>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Debug">
          <xs:annotation>
            <xs:appinfo>
              <EnumerationValue xmlns=
                "http://schemas.microsoft.com/2003/10/Serialization/">1
              </EnumerationValue>
            </xs:appinfo>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="DiagnosticViews">
          <xs:annotation>
            <xs:appinfo>
              <EnumerationValue xmlns=
                "http://schemas.microsoft.com/2003/10/Serialization/">2
              </EnumerationValue>
            </xs:appinfo>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="PerformanceCounters">
          <xs:annotation>
            <xs:appinfo>
              <EnumerationValue xmlns=
                "http://schemas.microsoft.com/2003/10/Serialization/">4
              </EnumerationValue>
            </xs:appinfo>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="EndToEndTracing">
          <xs:annotation>
            <xs:appinfo>
              <EnumerationValue xmlns=
                "http://schemas.microsoft.com/2003/10/Serialization/">8
              </EnumerationValue>
            </xs:appinfo>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="CepEventTracing">
          <xs:annotation>
```

```

    <xs:appinfo>
      <EnumerationValue xmlns=
        "http://schemas.microsoft.com/2003/10/Serialization/">16
      </EnumerationValue>
    </xs:appinfo>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="StateChanges">
  <xs:annotation>
    <xs:appinfo>
      <EnumerationValue xmlns=
        "http://schemas.microsoft.com/2003/10/Serialization/">32
      </EnumerationValue>
    </xs:appinfo>
  </xs:annotation>
</xs:enumeration>
<xs:enumeration value="Memory">
  <xs:annotation>
    <xs:appinfo>
      <EnumerationValue xmlns=
        "http://schemas.microsoft.com/2003/10/Serialization/">64
      </EnumerationValue>
    </xs:appinfo>
  </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:list>
</xs:simpleType>

```

The following table describes the enumeration values for the **DiagnosticAspects** type.

Value	Description
None	A value specifying that no diagnostic aspects are included.
Debug	A value specifying that debug tracing is enabled.
DiagnosticViews	A value specifying that diagnostic view data collection is enabled.
PerformanceCounters	A value specifying that performance counter data collection is enabled.
EndToEndTracing	A value specifying that event tracing for windows is enabled. For more information, see [MSDN-IDPTETW] .
CepEventTracing	A value that enables tracing of CEP events as they flow through a query. This tracing is used by the Event Flow Debugger.
StateChange	A value that enables trace events that are emitted when CEP objects are created or destroyed, or when they change their state.
Memory	A value that enables tracing of the memory management subsystem.

2.2.3.3.1.2 DiagnosticLevel

The **DiagnosticLevel** type is an enumeration of the different diagnostic level values. The level specifies the level of verbosity for the enabled diagnostic aspects.

The following code is the XSD for the **DiagnosticLevel** type.

```
<xs:simpleType name="DiagnosticLevel">
  <xs:annotation>
    <xs:appinfo>
      <ActualType Name="unsignedByte" Namespace="http://www.w3.org/2001/XMLSchema"
        xmlns="http://schemas.microsoft.com/2003/10/Serialization/" />
    </xs:appinfo>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Always" />
    <xs:enumeration value="Critical" />
    <xs:enumeration value="Error" />
    <xs:enumeration value="Warning" />
    <xs:enumeration value="Informational" />
    <xs:enumeration value="Verbose" />
  </xs:restriction>
</xs:simpleType>
```

The following table describes the enumeration values for the **DiagnosticLevel** type.

Value	Description
Always	A value that denotes the least verbose level. Only events that are always emitted are included. These values are ordered from least verbose to most verbose.
Critical	A value specifying that events with the Critical level are included.
Error	A value specifying that error events are included.
Warning	A value specifying that warning events are included.
Informational	A value specifying that informational events are included.
Verbose	A value that denotes the most verbose level. All of the preceding values are included.

2.2.3.3.2 GetDiagnosticSettingsResponse

The **GetDiagnosticSettingsResponse** type contains the current values for diagnostic settings that are in effect.

The following code is the XSD for the **GetDiagnosticSettingsResponse** type.

```
<xs:element name="GetDiagnosticSettingsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="DiagnosticAspects"
        type="tns:DiagnosticAspects" />
      <xs:element minOccurs="0" name="DiagnosticLevel"
        type="tns:DiagnosticLevel" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

The following table describes the enumeration values for the **GetDiagnosticSettingsResponse** type.

Element	Type	Description
DiagnosticAspect	DiagnosticAspect	An enumeration value that indicates which diagnostic aspects will be included in the Diagnostic View statistics.
DiagnosticLevel	DiagnosticLevel	An enumeration value that indicates what level of criticality of diagnostic events will be included in the Diagnostic View statistics.

2.2.3.3.3 GetDiagnosticViewResponse

The **GetDiagnosticViewResponse** type contains a report of the accumulated statistics for a Diagnostic View.

The following code is the XSD for the **GetDiagnosticViewResponse** type.

```
<xs:element name="GetDiagnosticViewResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="View" nillable="true"
        type="tns:DiagnosticView" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

The following table describes the elements for the **GetDiagnosticViewResponse** type.

Element	Type	Description
View	DiagnosticView	This element contains the Diagnostic View results that were requested.

2.2.3.3.3.1 DiagnosticView

The **DiagnosticView** type contains the contents of the statistical results for a diagnostic view.

The following code is the XSD for the **DiagnosticView** type.

```
<xs:complexType name="DiagnosticView">
  <xs:sequence>
    <xs:element name="Name" nillable="true" type="xs:anyURI" />
    <xs:element minOccurs="0" name="Properties" nillable="true"
      type="tns:Properties" />
  </xs:sequence>
</xs:complexType>
```

The following table describes the elements for the **DiagnosticView** type.

Element	Type	Description
Name	xs:anyURI	This URI represents the name of the object for which the diagnostic view is being returned.
Properties	Properties	The Properties element contains a collection of Property objects, each of which contains a single name-value pair from the view.

2.2.3.3.3.1.1 Properties

The **Properties** type contains a collection of **Property** elements, each of which is a name/value pair that contains a single value that is part of the diagnostic view.

```
<xs:complexType name="Properties">
  <xs:annotation>
    <xs:appinfo>
      <IsDictionary xmlns="http://schemas.microsoft.com/2003/10/Serialization/">
        true</IsDictionary>
      </xs:appinfo>
    </xs:annotation>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Property">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" nillable="true" type="xs:string" />
            <xs:element name="Value" nillable="true" type="xs:anyType" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```

The following table describes the elements for the **Properties** type.

Element	Type	Description
Property	complexType	This XML element contains a single name/value pair, which together form a statistical result being returned in the Diagnostic View.
Name	xs:string	The name of the observed statistical value that is being returned.
Value	xs:anyType	The value for the statistical value that is being returned.

2.2.3.4 Fault Types

This section contains the definitions for the fault types.

2.2.3.4.1 InvalidNameFault

This complex type defines the type for the **s:Detail** child element of the **s:Fault** element in the **SOAP fault** body.

The following code is the XSD for the **InvalidNameFault** complex type.

```

<xs:complexType name="InvalidNameFault">
  <xs:sequence>
    <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string" />
  </xs:sequence>
</xs:complexType>

```

The following table describes the elements for the **InvalidNameFault** complex type.

Element	Type	Description
Message	xs:string	A descriptive message for the fault.

2.2.3.4.2 InvalidDefinitionFault

This complex type defines the type for the **s:Detail** child element of the **s:Fault** element in the SOAP fault body.

The following code is the XSD for the **InvalidDefinitionFault** complex type.

```

<xs:complexType name="InvalidDefinitionFault">
  <xs:sequence>
    <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string" />
  </xs:sequence>
</xs:complexType>

```

The following table describes the elements for the **InvalidDefinitionFault** complex type.

Element	Type	Description
Message	xs:string	A descriptive message for the fault.

2.2.3.4.3 ManagementFault

This complex type defines the type for the **s:Detail** child element of the **s:Fault** element in the SOAP fault body.

The following code is the XSD for the **ManagementFault** complex type.

```

<xs:complexType name="ManagementFault">
  <xs:sequence>
    <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string" />
  </xs:sequence>
</xs:complexType>

```

The following table describes the elements for the **ManagementFault** complex type.

Element	Type	Description
Message	xs:string	A descriptive message for the fault.

2.2.3.4.4 RuntimeFault

This complex type defines the type for the **s:Detail** child element of the **s:Fault** element in the SOAP fault body.

The following code is the XSD for the **RuntimeFault** complex type.

```
<xs:complexType name="RuntimeFault">
  <xs:sequence>
    <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string" />
  </xs:sequence>
</xs:complexType>
```

The following table describes the elements for the **RuntimeFault** complex type.

Element	Type	Description
Message	xs:string	A descriptive message for the fault.

2.2.3.4.5 GetDiagnosticSettingsNotSupportedFault

This complex type defines the type for the **s:Detail** child element of the **s:Fault** element in the SOAP fault body.

The following code is the XSD for the **GetDiagnosticSettingsNotSupportedFault** complex type.

```
<xs:complexType name="GetDiagnosticSettingsNotSupported">
  <xs:sequence>
    <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string" />
    <xs:element minOccurs="0" name="Name" nillable="true" type="xs:anyURI" />
  </xs:sequence>
</xs:complexType>
```

The following table describes the elements for the **GetDiagnosticSettingsNotSupportedFault** complex type.

Element	Type	Description
Message	xs:string	A descriptive message for a particular fault (error).
Name	xs:anyURI	The URI of the CEP metadata object that caused a particular fault.

2.2.3.4.6 ClearDiagnosticSettingsNotSupportedFault

This complex type defines the type for the **s:Detail** child element of the **s:Fault** element in the SOAP fault body.

The following code is the XSD for the **ClearDiagnosticSettingsNotSupportedFault** complex type.

```
<xs:complexType name="ClearDiagnosticSettingsNotSupported">
  <xs:sequence>
```

```

<xs:element minOccurs="0" name="Message" nillable="true"
            type="xs:string" />
<xs:element minOccurs="0" name="Name" nillable="true"
            type="xs:anyURI" />
</xs:sequence>
</xs:complexType>

```

The following table describes the elements for the **ClearDiagnosticSettingsNotSupportedFault** complex type.

Element	Type	Description
Message	xs:string	A descriptive message for a particular fault (error).
Name	xs:anyURI	The URI of the CEP metadata object that caused a particular fault.

2.2.3.4.7 GetDiagnosticViewNotSupportedFault

This complex type defines the type for the **s:Detail** child element of the **s:Fault** element in the SOAP fault body.

The following code is the XSD for the **GetDiagnosticViewNotSupportedFault** complex type.

```

<xs:complexType name="GetDiagnosticViewNotSupported">
  <xs:sequence>
    <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string" />
    <xs:element minOccurs="0" name="Name" nillable="true" type="xs:anyURI" />
  </xs:sequence>
</xs:complexType>

```

The following table describes the elements for the **GetDiagnosticViewNotSupportedFault** complex type.

Element	Type	Description
Message	xs:string	A descriptive message for a particular fault (error).
Name	xs:anyURI	The URI of the CEP metadata object that caused a particular fault.

2.2.4 SOAP Headers

The following table summarizes the set of SOAP header definitions that are defined by this protocol specification.

Header	Description
CreateRequest_Headers	SOAP headers for the CreateRequest message. For more information, see section 2.2.2.1.1.1.1 .
CreateResponse_Headers	SOAP headers for the CreateResponse message. For more information, see section 2.2.2.1.1.2.1 .
GetRequest_Headers	SOAP headers for the GetRequest message. For more

Header	Description
	information, see section 2.2.2.1.2.1.1 .
DeleteRequest_Headers	SOAP headers for the DeleteRequest message. For more information, see section 2.2.2.1.3.1.1 .
DeleteResponse_Headers	SOAP headers for the DeleteResponse message. For more information, see section 2.2.2.1.3.2.1 .
EnumerateRequest_Headers	SOAP headers for the EnumerateRequest message. For more information, see section 2.2.2.1.4.1.1 .
ChangeQueryStateRequest_Headers	SOAP headers for the ChangeQueryStateRequest message. For more information, see section 2.2.2.1.5.1.1 .
ChangeQueryStateResponse_Headers	SOAP headers for the ChangeQueryStateResponse message. For more information, see section 2.2.2.1.5.2.1 .
GetDiagnosticSettingsRequest_Headers	SOAP headers for the GetDiagnosticSettingsRequest message. For more information, see section 2.2.2.2.1.1.1 .
SetDiagnosticSettingsRequest_Headers	SOAP headers for the SetDiagnosticSettingsResponse message. For more information, see section 2.2.2.2.1.1 .
ClearDiagnosticSettingsRequest_Headers	SOAP headers for the ClearDiagnosticSettingsRequest message. For more information, see section 2.2.2.2.3.1.1 .
GetDiagnosticViewRequest_Headers	SOAP headers for the GetDiagnosticViewRequest message. For more information, see section 2.2.2.2.4.1.1 .

3 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in this appendix.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsm="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:tns=
  "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management"
  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:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
  xmlns:wsw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:wsa10="http://www.w3.org/2005/08/addressing"
  xmlns:wsmx="http://schemas.xmlsoap.org/ws/2004/09/mex"
  targetNamespace=
  "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/Management"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
<wsdl:types>
  <xsd:schema targetNamespace=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/Imports">
    <xsd:import namespace=
      "http://schemas.microsoft.com/ComplexEventProcessing
      /2009/05/Management" />
    <xsd:import namespace=
      "http://schemas.microsoft.com/2003/10/Serialization/" />
    <xsd:import namespace=
      "http://schemas.microsoft.com/ComplexEventProcessing
      /2009/05/Metadata" />
    <xsd:import namespace=
      "http://www.w3.org/2005/08/addressing" />
    <xsd:import namespace=
      "http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
  </xsd:schema>
</wsdl:types>
<wsdl:message name="CreateRequest">
  <wsdl:part name="CreateRequest" element="tns:CreateRequest" />
</wsdl:message>
<wsdl:message name="CreateRequest_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="CreateResponse" />
<wsdl:message name="CreateResponse_Headers">
  <wsdl:part name="ResourceAddress" element="tns:ResourceAddress" />
</wsdl:message>
<wsdl:message name="IManagementService_Create_InvalidNameFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidNameFault" />
</wsdl:message>
```

```

<wsdl:message name=
  "IManagementService_Create_InvalidDefinitionFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidDefinitionFault" />
</wsdl:message>
<wsdl:message name="GetRequest" />
<wsdl:message name="GetRequest_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="GetResponse">
  <wsdl:part name="GetResponse" element="tns:GetResponse" />
</wsdl:message>
<wsdl:message name="IManagementService_Get_InvalidNameFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidNameFault" />
</wsdl:message>
<wsdl:message name="DeleteRequest" />
<wsdl:message name="DeleteRequest_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="DeleteResponse" />
<wsdl:message name="DeleteResponse_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="IManagementService_Delete_ManagementFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:ManagementFault" />
</wsdl:message>
<wsdl:message name="IManagementService_Delete_InvalidNameFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidNameFault" />
</wsdl:message>
<wsdl:message name="EnumerateRequest" />
<wsdl:message name="EnumerateRequest_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="EnumerateResponse">
  <wsdl:part name="ResourceNames" element="tns:ResourceNames" />
</wsdl:message>
<wsdl:message name="IManagementService_Enumerate_InvalidNameFaultFault_
  FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidNameFault" />
</wsdl:message>
<wsdl:message name="ChangeQueryStateRequest">
  <wsdl:part name="QueryState" element="tns:QueryState" />
</wsdl:message>
<wsdl:message name="ChangeQueryStateRequest_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="ChangeQueryStateResponse">
  <wsdl:part name="QueryState" element="tns:QueryState" />
</wsdl:message>
<wsdl:message name="ChangeQueryStateResponse_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="IManagementService_ChangeQueryState_InvalidNameFaultFault_
  FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidNameFault" />
</wsdl:message>
<wsdl:message name="IManagementService_ChangeQueryState_RuntimeFaultFault_
  FaultMessage">
  <wsdl:part name="detail" element="tns:RuntimeFault" />
</wsdl:message>

```

```

<wsdl:message name="GetDiagnosticSettingsRequest" />
<wsdl:message name="GetDiagnosticSettingsRequest_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="GetDiagnosticSettingsResponse">
  <wsdl:part name="parameters" element="tns:GetDiagnosticSettingsResponse" />
</wsdl:message>
<wsdl:message name="IManagementService_GetDiagnosticSettings_
  GetDiagnosticSettingsNotSupportedFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:GetDiagnosticSettingsNotSupported" />
</wsdl:message>
<wsdl:message name="IManagementService_GetDiagnosticSettings_
  InvalidNameFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidNameFault" />
</wsdl:message>
<wsdl:message name="SetDiagnosticSettingsRequest">
  <wsdl:part name="parameters" element="tns:SetDiagnosticSettings" />
</wsdl:message>
<wsdl:message name="SetDiagnosticSettingsRequest_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="IManagementService_SetDiagnosticSettings_OutputMessage" />
<wsdl:message name="IManagementService_SetDiagnosticSettings_
  InvalidNameFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidNameFault" />
</wsdl:message>
<wsdl:message name="IManagementService_SetDiagnosticSettings_
  SetDiagnosticSettingsNotSupportedFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:GetDiagnosticSettingsNotSupported" />
</wsdl:message>
<wsdl:message name="ClearDiagnosticSettingsRequest" />
<wsdl:message name="ClearDiagnosticSettingsRequest_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="IManagementService_ClearDiagnosticSettings_OutputMessage" />
<wsdl:message name="IManagementService_ClearDiagnosticSettings_
  InvalidNameFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidNameFault" />
</wsdl:message>
<wsdl:message name="IManagementService_ClearDiagnosticSettings_
  ClearDiagnosticSettingsNotSupportedFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:ClearDiagnosticSettingsNotSupported" />
</wsdl:message>
<wsdl:message name="GetDiagnosticViewRequest" />
<wsdl:message name="GetDiagnosticViewRequest_Headers">
  <wsdl:part name="Name" element="tns:Name" />
</wsdl:message>
<wsdl:message name="GetDiagnosticViewResponse">
  <wsdl:part name="parameters" element="tns:GetDiagnosticViewResponse" />
</wsdl:message>
<wsdl:message name="IManagementService_GetDiagnosticView_
  GetDiagnosticViewNotSupportedFaultFault_FaultMessage">
  <wsdl:part name="detail" element="tns:GetDiagnosticViewNotSupported" />
</wsdl:message>
<wsdl:message name="IManagementService_GetDiagnosticView_InvalidNameFaultFault_
  FaultMessage">
  <wsdl:part name="detail" element="tns:InvalidNameFault" />
</wsdl:message>
<wsdl:portType name="IManagementService">

```

```

<wsdl:operation name="Create">
  <wsdl:input wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/Create"
    name="CreateRequest" message="tns:CreateRequest" />
  <wsdl:output wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/CreateResponse"
    name="CreateResponse" message="tns:CreateResponse" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/InvalidName"
    name="InvalidNameFaultFault"
    message="tns:IManagementService_Create_InvalidNameFaultFault_
    FaultMessage" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/InvalidDefinition"
    name="InvalidDefinitionFaultFault"
    message="tns:IManagementService_Create_InvalidDefinitionFaultFault_
    FaultMessage" />
</wsdl:operation>
<wsdl:operation name="Get">
  <wsdl:input wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/Get"
    name="GetRequest" message="tns:GetRequest" />
  <wsdl:output wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/GetResponse"
    name="GetResponse" message="tns:GetResponse" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/InvalidName"
    name="InvalidNameFaultFault" message=
    "tns:IManagementService_Get_InvalidNameFaultFault_FaultMessage" />
</wsdl:operation>
<wsdl:operation name="Delete">
  <wsdl:input wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/Delete"
    name="DeleteRequest" message="tns>DeleteRequest" />
  <wsdl:output wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/DeleteResponse"
    name="DeleteResponse" message="tns>DeleteResponse" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/Fault"
    name="ManagementFaultFault" message=
    "tns:IManagementService_Delete_ManagementFaultFault_
    FaultMessage" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/InvalidName"
    name="InvalidNameFaultFault" message=
    "tns:IManagementService_Delete_InvalidNameFaultFault_
    FaultMessage" />
</wsdl:operation>

```

```

<wsdl:operation name="Enumerate">
  <wsdl:input wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/Enumerate"
    name="EnumerateRequest" message="tns:EnumerateRequest" />
  <wsdl:output wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/EnumerateResponse"
    name="EnumerateResponse" message="tns:EnumerateResponse" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/InvalidName"
    name="InvalidNameFaultFault" message=
    "tns:IManagementService_Enumerate_InvalidNameFaultFault_
    FaultMessage" />
</wsdl:operation>
<wsdl:operation name="ChangeQueryState">
  <wsdl:input wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing
    /2009/05/Management/ChangeQueryState"
    name="ChangeQueryStateRequest" message=
    "tns:ChangeQueryStateRequest" />
  <wsdl:output wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/ChangeQueryStateResponse"
    name="ChangeQueryStateResponse"
    message="tns:ChangeQueryStateResponse" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/InvalidName"
    name="InvalidNameFaultFault" message=
    "tns:IManagementService_ChangeQueryState_InvalidNameFaultFault_
    FaultMessage" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/RuntimeFailure"
    name="RuntimeFaultFault" message=
    "tns:IManagementService_ChangeQueryState_RuntimeFaultFault_
    FaultMessage" />
</wsdl:operation>
<wsdl:operation name="GetDiagnosticSettings">
  <wsdl:input wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/GetDiagnosticSettings"
    name="GetDiagnosticSettingsRequest"
    message="tns:GetDiagnosticSettingsRequest" />
  <wsdl:output wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/GetDiagnosticSettingsResponse"
    name="GetDiagnosticSettingsResponse"
    message="tns:GetDiagnosticSettingsResponse" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/GetDiagnosticSettingsNotSupported"
    name="GetDiagnosticSettingsNotSupportedFaultFault" message=
    "tns:IManagementService_GetDiagnosticSettings_
    GetDiagnosticSettingsNotSupportedFaultFault_FaultMessage" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/

```



```

        Management/InvalidName"
        name="InvalidNameFaultFault" message=
        "tns:IManagementService_GetDiagnosticSettings_
        InvalidNameFaultFault_FaultMessage" />
</wsdl:operation>
<wsdl:operation name="SetDiagnosticSettings">
  <wsdl:input wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/SetDiagnosticSettings"
    name="SetDiagnosticSettingsRequest" message=
    "tns:SetDiagnosticSettingsRequest" />
  <wsdl:output wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    /Management/SetDiagnosticSettingsResponse"
    message="tns:IManagementService_SetDiagnosticSettings_
    OutputMessage" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/InvalidName"
    name="InvalidNameFaultFault" message=
    "tns:IManagementService_SetDiagnosticSettings_
    InvalidNameFaultFault_FaultMessage" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/SetDiagnosticSettingsNotSupported"
    name="SetDiagnosticSettingsNotSupportedFaultFault" message=
    "tns:IManagementService_SetDiagnosticSettings_
    SetDiagnosticSettingsNotSupportedFaultFault_FaultMessage" />
</wsdl:operation>
<wsdl:operation name="ClearDiagnosticSettings">
  <wsdl:input wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/ClearDiagnosticSettings"
    name="ClearDiagnosticSettingsRequest" message=
    "tns:ClearDiagnosticSettingsRequest" />
  <wsdl:output wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/ClearDiagnosticSettingsResponse"
    message="tns:IManagementService_ClearDiagnosticSettings_
    OutputMessage" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/InvalidName"
    name="InvalidNameFaultFault" message=
    "tns:IManagementService_ClearDiagnosticSettings_
    InvalidNameFaultFault_FaultMessage" />
  <wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/ClearDiagnosticSettingsNotSupported"
    name="ClearDiagnosticSettingsNotSupportedFaultFault" message=
    "tns:IManagementService_ClearDiagnosticSettings_
    ClearDiagnosticSettingsNotSupportedFaultFault_FaultMessage" />
</wsdl:operation>
<wsdl:operation name="GetDiagnosticView">
  <wsdl:input wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/GetDiagnosticView"
    name="GetDiagnosticViewRequest" message=
    "tns:GetDiagnosticViewRequest" />

```

```

<wsdl:output wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/GetDiagnosticViewResponse"
    name="GetDiagnosticViewResponse" message=
    "tns:GetDiagnosticViewResponse" />
<wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/GetDiagnosticViewNotSupported"
    name="GetDiagnosticViewNotSupportedFaultFault" message=
    "tns:IManagementService_GetDiagnosticView_
    GetDiagnosticViewNotSupportedFaultFault_FaultMessage" />
<wsdl:fault wsaw:Action=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/InvalidName"
    name="InvalidNameFaultFault" message=
    "tns:IManagementService_GetDiagnosticView_InvalidNameFaultFault_
    FaultMessage" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="DefaultBinding_IManagementService"
    type="tns:IManagementService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="Create">
        <soap:operation soapAction=
            "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
            Management/Create"
            style="document" />
        <wsdl:input name="CreateRequest">
            <soap:header message="tns:CreateRequest_Headers" part="Name"
                use="literal" />
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output name="CreateResponse">
            <soap:header message="tns:CreateResponse_Headers" part="ResourceAddress"
                use="literal" />
            <soap:body use="literal" />
        </wsdl:output>
        <wsdl:fault name="InvalidNameFaultFault">
            <soap:fault name="InvalidNameFaultFault" use="literal" />
        </wsdl:fault>
        <wsdl:fault name="InvalidDefinitionFaultFault">
            <soap:fault name="InvalidDefinitionFaultFault" use="literal" />
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Get">
        <soap:operation soapAction=
            "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
            Management/Get"
            style="document" />
        <wsdl:input name="GetRequest">
            <soap:header message="tns:GetRequest_Headers" part="Name" use="literal" />
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output name="GetResponse">
            <soap:body use="literal" />
        </wsdl:output>
        <wsdl:fault name="InvalidNameFaultFault">
            <soap:fault name="InvalidNameFaultFault" use="literal" />
        </wsdl:fault>
    </wsdl:operation>

```

```

</wsdl:operation>
<wsdl:operation name="Delete">
  <soap:operation soapAction=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/Delete"
    style="document" />
  <wsdl:input name="DeleteRequest">
    <soap:header message="tns:DeleteRequest_Headers" part="Name" use="literal" />
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output name="DeleteResponse">
    <soap:header message="tns:DeleteResponse_Headers" part="Name"
      use="literal" />
    <soap:body use="literal" />
  </wsdl:output>
  <wsdl:fault name="ManagementFaultFault">
    <soap:fault name="ManagementFaultFault" use="literal" />
  </wsdl:fault>
  <wsdl:fault name="InvalidNameFaultFault">
    <soap:fault name="InvalidNameFaultFault" use="literal" />
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="Enumerate">
  <soap:operation soapAction=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/Enumerate"
    style="document" />
  <wsdl:input name="EnumerateRequest">
    <soap:header message="tns:EnumerateRequest_Headers" part="Name"
      use="literal" />
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output name="EnumerateResponse">
    <soap:body use="literal" />
  </wsdl:output>
  <wsdl:fault name="InvalidNameFaultFault">
    <soap:fault name="InvalidNameFaultFault" use="literal" />
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="ChangeQueryState">
  <soap:operation soapAction=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/ChangeQueryState"
    style="document" />
  <wsdl:input name="ChangeQueryStateRequest">
    <soap:header message="tns:ChangeQueryStateRequest_Headers" part="Name"
      use="literal" />
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output name="ChangeQueryStateResponse">
    <soap:header message="tns:ChangeQueryStateResponse_Headers" part="Name"
      use="literal" />
    <soap:body use="literal" />
  </wsdl:output>
  <wsdl:fault name="InvalidNameFaultFault">
    <soap:fault name="InvalidNameFaultFault" use="literal" />
  </wsdl:fault>
  <wsdl:fault name="RuntimeFaultFault">
    <soap:fault name="RuntimeFaultFault" use="literal" />
  </wsdl:fault>

```

```

    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="GetDiagnosticSettings">
    <soap:operation soapAction=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
      Management/GetDiagnosticSettings"
      style="document" />
    <wsdl:input name="GetDiagnosticSettingsRequest">
      <soap:header message="tns:GetDiagnosticSettingsRequest_Headers" part="Name"
        use="literal" />
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output name="GetDiagnosticSettingsResponse">
      <soap:body use="literal" />
    </wsdl:output>
    <wsdl:fault name="GetDiagnosticSettingsNotSupportedFaultFault">
      <soap:fault name="GetDiagnosticSettingsNotSupportedFaultFault"
        use="literal" />
    </wsdl:fault>
    <wsdl:fault name="InvalidNameFaultFault">
      <soap:fault name="InvalidNameFaultFault" use="literal" />
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="SetDiagnosticSettings">
    <soap:operation soapAction=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
      Management/SetDiagnosticSettings"
      style="document" />
    <wsdl:input name="SetDiagnosticSettingsRequest">
      <soap:header message="tns:SetDiagnosticSettingsRequest_Headers" part="Name"
        use="literal" />
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
    <wsdl:fault name="InvalidNameFaultFault">
      <soap:fault name="InvalidNameFaultFault" use="literal" />
    </wsdl:fault>
    <wsdl:fault name="SetDiagnosticSettingsNotSupportedFaultFault">
      <soap:fault name="SetDiagnosticSettingsNotSupportedFaultFault"
        use="literal" />
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="ClearDiagnosticSettings">
    <soap:operation soapAction=
      "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
      Management/ClearDiagnosticSettings"
      style="document" />
    <wsdl:input name="ClearDiagnosticSettingsRequest">
      <soap:header message="tns:ClearDiagnosticSettingsRequest_Headers"
        part="Name" use="literal" />
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
    <wsdl:fault name="InvalidNameFaultFault">
      <soap:fault name="InvalidNameFaultFault" use="literal" />
    </wsdl:fault>
  </wsdl:operation>

```

```
</wsdl:fault>
<wsdl:fault name="ClearDiagnosticSettingsNotSupportedFaultFault">
  <soap:fault name="ClearDiagnosticSettingsNotSupportedFaultFault"
    use="literal" />
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="GetDiagnosticView">
  <soap:operation soapAction=
    "http://schemas.microsoft.com/ComplexEventProcessing/2009/05/
    Management/GetDiagnosticView"
    style="document" />
  <wsdl:input name="GetDiagnosticViewRequest">
    <soap:header message="tns:GetDiagnosticViewRequest_Headers" part="Name"
      use="literal" />
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output name="GetDiagnosticViewResponse">
    <soap:body use="literal" />
  </wsdl:output>
  <wsdl:fault name="GetDiagnosticViewNotSupportedFaultFault">
    <soap:fault name="GetDiagnosticViewNotSupportedFaultFault" use="literal" />
  </wsdl:fault>
  <wsdl:fault name="InvalidNameFaultFault">
    <soap:fault name="InvalidNameFaultFault" use="literal" />
  </wsdl:fault>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

4 Change Tracking

This section will report content and/or editorial changes, beginning with the next release.

Preliminary

5 Index

A

[Applicability statement](#) 13

C

[ChangeQueryState message](#) 26
[ClearDiagnosticSettings message](#) 33
[ClearDiagnosticSettingsNotSupportedFault message](#) 46
[Create message](#) 16

D

[Delete message](#) 21
[Diagnostic method types](#) 98
[Diagnostic methods](#) 28

E

[Enumerate message](#) 23

F

[Fault types](#) 103
[Faults](#) 38
[Full WSDL](#) 108

G

[Get message](#) 19
[GetDiagnosticSettings message](#) 28
[GetDiagnosticSettingsNotSupportedFault message](#) 44
[GetDiagnosticView message](#) 35
[GetDiagnosticViewNotSupportedFault message](#) 48

I

[Informative references](#) 11
[Introduction](#) 8
[InvalidDefinitionFault message](#) 39
[InvalidNameFault message](#) 38

M

[ManagementFault message](#) 41
Messages ([section 2](#) 14, [section 2.2](#) 14)
[Metadata definition types](#) 52
[Metadata method types](#) 50
[Metadata methods](#) 16
[Methods](#) 15

N

[Namespaces](#) 14
[Normative references](#) 9

O

[Overview](#) 11

P

[Preconditions](#) 12
[Prerequisites](#) 12
[Protocol overview \(synopsis\)](#) 11

R

[References](#) 9
[Relationship to other protocols](#) 12
[RuntimeFault message](#) 42

S

[SetDiagnosticSettings message](#) 30
[SetDiagnosticSettingsNotSupportedFault message](#) 45
[SOAP Headers](#) 106
[Standards assignments](#) 13
[Synopsis](#) 11

T

[Transport](#) 14
[Types](#) 49

V

[Vendor-extensible fields](#) 13
[Versioning and capability negotiation](#) 13