The Workflow Management Coalition Specification

Workflow Management Coalition Workflow Standard - Interoperability Internet e-mail MIME Binding

Document Number WFMC-TC-1018

31~~20~~-July-1998
Version 1.1fe (draft)

Workflow Management Coalition
2 Crown Walk
Winchester
SO22 5XE
United Kingdom
Tel: +44 1962 873401
Fax: +44 1962 868111
Email: wfmc@wfmc.org
web: http://www.wfmc.org

First printing July 1998

Table of Content

1 CHANGE HISTORY 8

2 PURPOSE 8

3 AUDIENCE 8

4 SCOPE 8

5 INTRODUCTION 8

6 DEFINED TERMS 8

7 MIME MESSAGE 9

7.1 MESSAGE SUBJECT 10
7.1.1 Message type 10
7.1.2 Request or Notification Messages 11
7.1.3 Response Messages 11
7.1.4 Error Messages 11
7.2 PROTOCOL DATA 11
7.2.1 Head 14
7.2.2 Tail 15
7.2.3 Special Characters 15
7.2.4 Lines 17
7.3 ENSEMBLE THE PROTOCOL DATA 18
7.4 ATTACHMENTS 19
7.5 EXAMPLES 19
7.5.1 Multipart Message 19
7.5.2 Simple Message 20

8 PROTOCOL 20

8.1 FLOW CONTROL 21
8.1.1 Partial response messages 22
8.2 RETRANSMISSION STRATEGY 22
8.2.1 Resending Messages ..... 23~~22~~
8.3 DUPLICATION DETECTION 23
8.3.1 Duplicate Within the Same Conversation 23
8.3.2 Duplicate Outside the Conversation 23
8.4 CONVERSATION ESTABLISHMENT AND TERMINATION 23
8.5 CRASH RECOVERY ..... 24~~23~~

9 DATA TYPES & DECLARATIVES 24

9.1 BASIC ATTRIBUTE TYPES 24
9.1.1 Attachments Values ..... 25~~24~~
9.1.2 Boolean Values 25
9.1.3 Binary Values 25
9.1.4 Date and Time Values 25
9.1.5 Numerical Values 26
9.2 OPERATION FIELDS ..... 27~~26~~
9.2.1 Activity Id 27

9.2.2 Business Process Definition Name 27
9.2.3 Contract Id ..... 28~~27~~
9.2.4 Conversation id 28
9.2.5 Language ..... 29~~28~~
9.2.6 Message Id ..... 29~~28~~
9.2.7 Name-Type-Value Group 30
9.2.8 Node Id ..... 31~~30~~
9.2.9 Operation Id ..... 31~~30~~
9.2.10 Process Definition Id 31
9.2.11 Process Id 31
9.2.12 Product Id ..... 32~~31~~
9.2.13 Profile ..... 32~~31~~
9.2.14 Return Error Codes 32
9.2.15 Role Id ..... 34~~33~~
9.2.16 State ..... 34~~33~~
9.2.17 Timestamp ..... 36~~35~~
9.2.18 User Id ..... 36~~35~~
9.2.19 Version ..... 36~~35~~

10 CONFORMANCE 36

10.1 CONFORMANCE PROFILES 37
10.1.1 Simple Chains 37
10.1.2 Nested Sub-process ..... 39~~38~~

11 OPERATIONS ..... 44~~43~~

11.1 CHANGE PROCESS INSTANCE STATE ..... 46~~44~~
11.2 CREATE PROCESS INSTANCE ..... 48~~46~~
11.3 GET PROCESS INSTANCE ATTRIBUTES ..... 50~~48~~
11.4 GET PROCESS INSTANCE STATE ..... 52~~50~~
11.5 PROCESS INSTANCE ATTRIBUTES CHANGED ..... 54~~52~~
11.6 PROCESS INSTANCE STATE CHANGED ..... 56~~54~~
11.7 SET PROCESS INSTANCE ATTRIBUTES ..... 58~~56~~
11.8 START CONVERSATION ..... 61~~59~~
11.9 STOP CONVERSATION ..... 63~~61~~

12 REFERENCES ..... 64~~62~~

13 APPENDIX A -- AUDIT DATA ..... 65~~63~~

13.1 AUDIT DATA TYPES ..... 65~~63~~
13.2 AUDIT INFORMATION ..... 65~~63~~
13.2.1 Change Process Instance State ..... 66~~64~~
13.2.2 Create Process Instance ..... 72~~70~~
13.2.3 Get Process Instance Attributes ..... 78~~76~~
13.2.4 Get Process Instance State ..... 83~~81~~
13.2.5 Process Instance Attribute Changed ..... 85~~83~~
13.2.6 Process Instance State Changed ..... 88~~86~~
13.2.7 Set Process Instance Attributes ..... 91~~89~~
13.2.8 Start Conversation ..... 97~~95~~
13.2.9 Stop Conversation ..... 99~~97~~

14 APPENDIX B – CHARACTER SET ..... 101~~99~~

15 APPENDIX C – IMPLEMENTATION HINTS (NON-NORMATIVE) ..... 102~~100~~

15.1 CONTRACTS ..... 102~~100~~

15.2 PROCESS AND CONTRACT INFORMATION ..... 103~~101~~
15.2.1 Incoming Requests ..... *103~~101~~*
15.2.2 Outbound Requests ..... *104~~102~~*
15.3 MAILBOX SETUP ..... 105~~102~~

~~**1 CHANGE HISTORY ..... 8**~~

~~**2 PURPOSE ..... 8**~~

~~**3 AUDIENCE ..... 8**~~

~~**4 SCOPE ..... 8**~~

~~**5 INTRODUCTION ..... 8**~~

~~**6 DEFINED TERMS ..... 8**~~

~~**7 MIME MESSAGE ..... 9**~~

~~7.1 MESSAGE SUBJECT ..... 9~~
~~*7.1.1 Message type ..... 10*~~
~~*7.1.2 Request or Notification Messages ..... 10*~~
~~*7.1.3 Response Messages ..... 10*~~
~~*7.1.4 Error Messages ..... 10*~~
~~7.2 PROTOCOL DATA ..... 10~~
~~*7.2.1 Head ..... 14*~~
~~*7.2.2 Tail ..... 14*~~
~~*7.2.3 Special Characters ..... 14*~~
~~*7.2.4 Lines ..... 17*~~
~~7.3 ENSEMBLE THE PROTOCOL DATA ..... 18~~
~~7.4 ATTACHMENTS ..... 18~~
~~7.5 EXAMPLES ..... 19~~
~~*7.5.1 Multipart Message ..... 19*~~
~~*7.5.2 Simple Message ..... 19*~~

~~**8 PROTOCOL ..... 20**~~

~~8.1 FLOW CONTROL ..... 20~~
~~8.2 RETRANSMISSION STRATEGY ..... 21~~
~~8.3 DUPLICATION DETECTION ..... 21~~
~~*8.3.1 Duplicate Within the Same Conversation ..... 21*~~
~~*8.3.2 Duplicate Outside the Conversation ..... 21*~~
~~8.4 CONVERSATION ESTABLISHMENT AND TERMINATION ..... 22~~
~~8.5 CRASH RECOVERY ..... 23~~

~~**9 DATA TYPES & DECLARATIVES ..... 23**~~

~~9.1 BASIC ATTRIBUTE TYPES ..... 23~~
~~*9.1.1 Boolean Values ..... 23*~~
~~*9.1.2 Date and Time Values ..... 24*~~
~~*9.1.3 MIME Values ..... 24*~~
~~9.2 OPERATION FIELDS ..... 24~~
~~*9.2.1 Activity Id ..... 25*~~
~~*9.2.2 Business Definition Process Name ..... 25*~~
~~*9.2.3 Contract Id ..... 25*~~
~~*9.2.4 Conversation id ..... 26*~~
~~*9.2.5 Language ..... 26*~~

9.2.6 Message Id *27*
9.2.7 Name_Type_Value Group *28*
9.2.8 Node Id *28*
9.2.9 Process Definition Id *28*
9.2.10 Process Id *29*
9.2.11 Product Id *29*
9.2.12 Profile *29*
9.2.13 Return Error Codes *30*
9.2.14 Role Id *31*
9.2.15 State *31*
9.2.16 Timestamp *32*
9.2.17 User Id *32*
9.2.18 Version *32*

10 CONFORMANCE **32**

10.1 CONFORMANCE PROFILES 34
10.1.1 Simple Chains *34*
10.1.2 Nested Sub-process (Polling) *35*
10.1.3 Nested Sub-process (Suspended animation) *36*
10.1.4 Nested Sub-process (Deferred synchronous) *37*
10.1.5 Nested Sub-process (Synchronized enactment) *38*
10.1.6 Process Administration *39*

11 OPERATIONS **40**

11.1 CHANGE PROCESS INSTANCE STATE 41
11.2 CREATE PROCESS INSTANCE 43
11.3 GET PROCESS INSTANCE ATTRIBUTES 45
11.4 GET PROCESS INSTANCE STATE 47
11.5 LIST PROCESS INSTANCES 48
11.6 PROCESS INSTANCE ATTRIBUTES CHANGED 49
11.7 PROCESS INSTANCE STATE CHANGED 51
11.8 SET PROCESS INSTANCE ATTRIBUTES 53
11.9 START CONVERSATION 55
11.10 STOP CONVERSATION 57

12 REFERENCES **58**

13 APPENDIX A -- AUDIT DATA **59**

13.1 AUDIT DATA TYPES 59
13.2 AUDIT INFORMATION 59
13.2.1 Change Process Instance State *60*
13.2.2 Create Process Instance *66*
13.2.3 Get Process Instance Attributes *72*
13.2.4 Get Process Instance State *77*
13.2.5 List Process Instances *78*
13.2.6 Process Instance Attribute Changed *79*
13.2.7 Process Instance State Changed *82*
13.2.8 Set Process Instance Attributes *85*
13.2.9 Start Conversation *91*
13.2.10 Stop Conversation *93*

14 APPENDIX B – CHARACTER SET **95**

15 APPENDIX C – IMPLEMENTATION HINTS (NON-NORMATIVE) **96**

15.1 CONTRACTS 96
15.2 PROCESS AND CONTRACT INFORMATION 97
15.2.1 Incoming Requests 97
15.2.2 Outbound Requests 98
15.3 UNIX SENDMAIL 98

1 Change History

Version 1.1 – Editor: Mike Marin (mmarin@filenet.com)

- Formalized the specification and defined missing details.

Version 1.0 – Editor: Mike Anderson (mja@process.icl.co.uk)

- Initial Version

2 Purpose

This document maps to the Workflow Management Workflow Standard - Interoperability Abstract Specification [WfMC1012], which provides an abstract specification that define the functionality necessary to achieve a defined level of interoperability between two or more workflow engines. This document defines a binding that give concrete type definition and message format for the realization of the abstract specification, using Internet e-mail with MIME encoding as the transport mechanism. This version of the document corrects a number of issues, errors and omissions in the version 1.0 document, first published in October 1996.

3 Audience

The intended audience for this document is primarily vendor organizations who seek to implement the MIME Binding of the Workflow Management Coalition Standard - Interoperability. It may also be of interest to those who prepare binding specifications for other transports, and for those seeking to assess conformance claims made by vendors for their products.

4 Scope

The scope of this document is defined by the abstract specification of the interoperability standard [WfMC1012].

5 Introduction

This document gives concrete definition of the messages that flow between two workflow engines in order to effect interoperability as defined in [WfMC1012]. Abstract definitions of the messages given in [WfMC1012] are mapped onto a simple text interface that uses Internet e-mail as the transport method.

The examples in this version of the specification were manually created. They were reviewed for misspelling and correctness. In any discrepancy between the examples and the specification, the second is normative.

6 Defined Terms

The following terms, used in this document, are defined in the Workflow Management Coalition Glossary [WfMC1011].

Byte	A byte corresponds to an octet in the MIME specification [RFC-2045]. In US-ASCII, each byte corresponds to a character.
Character	Refers to the character in the encoding being used. In single-byte encoding, a character corresponds to a byte. In multi-byte encoding, a character may correspond to one or more bytes.

Dialog	The exchange of a set of messages that occurs between two workflow engines to effect a complete interoperation relating to the enactment of two designated process instances.
Source workflow engine	An engine that requests an enactment of a process from another workflow engine. The source workflow engine always starts its dialog by issuing a StartConversation and concludes with a StopConversation. This engine is sometimes called the initiating engine.
Target workflow engine	The workflow engine that receives the request from the source workflow engine. An engine that accepts a StartConversation assumes the role of target workflow engine. The process running on this engine is sometimes called enacted process instance.
Receiving workflow engine	The engine that is processing the incoming message.
Sending workflow engine	The engine that is ~~doing~~ making the request or that is creating the outbound message.
Protocol	The way messages are interchanged between the source workflow engine and the target workflow engine The protocol specifies how these workflow engines deal with the idiosyncrasies~~y~~ ~~in~~arising from the mail system, including lost, duplicated, and out-of-~~order~~ sequence messages.
Protocol data	The part of the message that contains the information the workflow engine interprets

7 MIME Message

This specification is based on MIME (Multipurpose Internet Mail Extension - RFC-2045 to RFC-2049). It uses either a text/plain content type or a multipart MIME message in which the first part must be text/plain.

The content type of the message is either multipart/mixed or text/plain, depending on whether attachments are used. When using attachments, the content type must be multipart/mixed. A simple example of the MIME header is:

```
Content-type: multipart/mixed
```

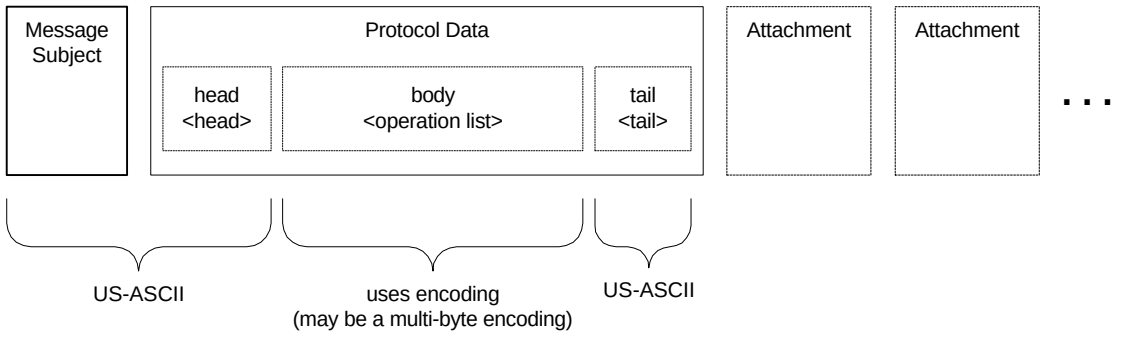
The preamble and epilogue in a multipart message are ignored.

If attachments are not used, the content type should be text/plain. A simple example of the MIME header, in this case is:

```
Content-type: text/plain
```

Details on the MIME header are described in the Multipurpose Internet Mail Extension - RFC-2045.

The rest of this section describes how interoperability information is placed in an electronic mail message. MIME messages have the following parts which are described below:



7.1 Message subject

The subject of a message implementing this specification ~~follows~~ conforms to the following ~~the~~ BNF-like (Backus, Naur Form) syntax:

```
<message Subject>  ::=  <message type> <sequence> <conversation iId> "&&"

<message type>     ::=   "wfmc-if4-request"
                       |  "wfmc-if4-response"
                       |  "wfmc-if4-error"   "(" <message error> ")"

<sequence>         ::=   "[" <message id> "]"
                       |  "[" <message id> "." <op range>"]"

<op range>         ::=   <operation id> "-" <operation id> "/"
                         <operation id>
~~<sequence>~~         ~~::=   "[" <message Id> "]"~~
```

Where <conversation Id>, <message error>, ~~and~~ <message i~~I~~d> and <operation id> are described in the next section. The message subject is always encoded in US-ASCII.

Messages with a subject that does not start with a valid <message type> and <sequence> must be ignored by the receiving engine.

It is possible that some electronic mail systems will truncate the message subject. A message subject without the "&&" at the end is considered truncated. Truncated messages can still be processed by the receiving engine, by using the message body as described in the next session, as long as it has a valid <message type> and <sequence>. The subject <message type> and <sequence> in a truncate message must match the <message type> and <sequence> in the ~~first part~~protocol data <head> (see section 7.2.1 Head). The subject <conversation id> of a truncated message must be discarded by the receiving engine, the <conversation id> of the ~~first part~~protocol data <head> must be used instead.

Example:

```
wfmc-if4-request[12]495+AB010DFF&&
```

The initial request, the one with the StartConversation operation on it, must have zero for <message id>, and it must have a partial <conversation id> (without the TargetConversationID, see section 9.2.4 Conversation id), as showed below~~in here~~:

```
wfmc-if4-request[0]123ABCD+&&
```

7.1.1 Message type

There are three distinct messages that an engine can send. The <message type> indicates the kind of message the electronic mail contains.

Messages must contain operations or responses (see section 11 Operations), but not both. Therefore, the two main message types are request and response.

7.1.2 Request or Notification Messages

A message containing operations must have a message type of wfmc-if4-request.

A source workflow engine uses operations to request work from a target workflow engine. A target workflow engine uses operations to notify the source workflow engine of its progress. In both cases, a wfmc-if4-request is used.

A wfmc-if4-request may contain several operations. The order of the operations indicates the order in which the sending engine expect the receiving engine to process the operations.

7.1.3 Response Messages

A message containing responses to operations must have a message type of wfmc-if4-response.

Response messages are used by source and target engines to respond to operations they receive. All the operations in the request message must be attempted in the order they appear in the request message.

There are two kind of wfmc-if4-response messages:

- A response message in which all the operations requested in the request message (wfmc-if4-request) are being accounted for. In this case, ~~A~~the wfmc-if4-response must contain the same number of responses, as there were operations in the wfmc-if4-request message, which is being answered. The order of the responses, in the message, must follow the order of the operations in the request or notification.
- Partial response messages, in which not all the operations of the request message (wfmc-if4-request) are present. In this case, only a subset of the requested operations is present in the response. The order of execution of the operations must still be honor, and the partial response must contain a complete range of operations. The <sequence> variance with <op range> must be used to indicate this kind of response message.

7.1.4 Error Messages

Message interchange errors must have a message type of wfmc-if4-error.

Message interchange errors are those in which the receiving engine is unable to process the operations or responses in the message, because the message do not conform to a valid message.

7.2 Protocol Data

In a multipart message, the first part must be of content type text/plain and it contains the protocol data. The rest of the MIME parts are considered attachments. In a single-part message, the only part present is of content type text/plain and it contains the protocol.

The body of the first part contains one or more operation requests/responses with appropriate fields, encoded using the concepts defined by the CGI (Common Gateway Interface) and the URL encoding scheme [RFC-1738].

Each operation starts with the operation name which is terminated by a "?" character. The fields are defined as name-value pairs and are separated by the "&" character. No positionality of fields is assumed. The operation ends with "&&".

Extended attributes in the form of name-value pairs separated by the "&" character can be appended at the end of the line, after the following marker "&extended&". Extended attributes may be used by workflow engines to extend the number of fields ~~of~~supporting an operation. The receiving workflow engine may ignore the extended attribute. Extended attributes are optional and workflow engines must not depend on them to be present.

The following ~~the~~ BNF-like (Backus, Naur Form) syntax~~,~~ formally describes the body of the first part of the message. Non-terminal tokens enclosed in quotes must be spelled exactly as defined.

```
<message Body>    ::=   <head> <operation list> <tail>

<head>            ::=   <message type> <sequence> <conversation id>
                        "&" <timestamp> <encoding> <terminator>

<message type>    ::=   "wfmc-if4-request"
                    |   "wfmc-if4-response"
                    |   "wfmc-if4-error"   "(" <message error> ")"

<sequence>         ::=  "[" <message id> "]"
                    |   "[" <message id> "." <op range>"]"

<op range>         ::=  <operation id> "-" <operation id> "/"
                        <operation id>
~~<sequence>~~        ~~::=  "[" <message id> "]"~~

<encoding>        ::=   "," "encoding" "=" <encoding name>
                    |   <empty>

<operation list>  ::=   <op>
                    |   <op> <operation list>

<op>              ::=   <operation> "?" <name-value list> <extended>
                        <terminator>

<extended>        ::=   "&extended&"  <name-value list>
                    |   <empty>

<name-value list> ::=   <name> "=" <value>
                    |   <name> "=" <value> "&" <name-value list>
                    |   <empty>

<terminator>      ::=   "&&"

<tail>            ::=   "end" "(" <checksum> ")"
```

Where:

<checksum>	This is the message checksum from the first byte of this part (always a "w" of either wfmc-if4-request or wfmc-if4-response) to the last byte before the "(" (always the "d" of "end"). See section 7.2.2 Tail. This is a string representation of an integer value.
<conversation Id>	Identifies the conversation to which this message belongs. This is a string value. See section 9.2.4 Conversation id. Note that "+" and "&" must not appear in <conversation id>. If they are necessary, they must be escaped.
<empty>	This is a syntactical placeholder, which indicates no input.
<encoding name>	Implementations may recognize several encodings, including Unicode and JIS. This is an US-ASCII string value. The encoding name values are case insensitive. For Unicode (ISO/IEC 10646) use the following values: UTF-8 UTF-16

ISO-10646-UCS-2
ISO-10646-UCS-4

For JIS X-0208-1997 use the following values:

ISO-2022-JP
Shift_JIS
EUC-JP

US-ASCII and the ISO-8859 family should not be used as encoding values because they are part of the text/plain charset

`<message error>` A message interchange error. Used to tell the remote engine that it last message was not processed because of a message interchange error. This is an US-ASCII string representation of an integer value.

wfmc-if4-error indicates an interchange error relatingof to either a wfmc-if4-request or wfmc-if4-response. The `<message id>` of the wfmc-if4-error must be the same as the `<message id>` of the message in error.

Valid values are:

201 -- Truncated message
202 -- Modified Message (Bad Checksum)
203 -- Invalid or unsupported encoding
204 -- Expired Message
205 -- Invalid Message
206 -- Invalid ConversationID
207 -- Invalid Sequence
208 -- Invalid Timestamp
209 -- Invalid Escape Sequence

`<message id>` Message number. This is a string representation of an integer value. See section 9.2.6 Message Id.

`<name>` This is the name of a parameter, as defined in the operation, or the name of an extended attribute (if it appears after “&extended&”). The name values are case insensitive.

Valid values are:

ActivityID
ContractID
ErrorCode
ErrorText
Language
MessageID
Name
Number
ProcessDefinitionID
ProcessID
ProductID
Profile
RootPID
SourceAID
SourceBDefName
SourceNodeID
SourcePID

SourceRoleID
SourceConversationID
SourceUserID
State
TargetNodeID
TargetBDefName
TargetPID
TargetRoleID
TargetConversationID
TargetUserID
Timestamp
Type
Value
Version

`<operation>`	This is the name of an operation as described in section 11 Operations. The operation values are case insensitive. Valid values are: ChangeProcessInstanceState CreateProcessInstance GetProcessInstanceAttribute GetProcessInstanceState ListProcessInstances ProcessInstanceAttributeChanged ProcessInstanceStateChanged SetProcessInstanceAttributes StartConversation StopConversation
`<operation id>`	See 9.2.9 Operation Id.
`<op range>`	Indicates the range of operations that are included in this message. The range must only appear in response messages that do not include responses for all the original operations in the request message. It is composed of three operation-ids: the first operation-id in the response message, the last operation-id in the response message, and the last operation-id in the request message (the total of operations).
`<sequence>`	There are two variants of <sequence>, one with a range and one without. Request messages do not use the range variant. The range variant is only used by response messages in which not all the operations are being responded in a single message.
`<timestamp>`	Time stamp of the message. It must use a UTC time value. See section 9.1.4.3~~9.1.4.3~~9.1.2.3 DateTime Format (WMTDateTime)~~DateTime Format (WMTDateTime)~~~~Timestamp Format (WMTTimestamp)~~.
`<value>`	This is the value of the field or attribute. The type of the value is predefined by the operation.

Note that `<encoding name>`, `<name>`, and `<operation>` are case insensitive. For example: ErrorCode, errorcode, errorCODE, and ERRORcode are all valid `<name>s`.

7.2.1 Head

The head, as described above, is always encoded in US-ASCII. Any other encoding starts immediately after the head terminator ("&&"). When using a particular encoding, the non-terminal tokens enclosed in quotes in the grammar must be encoded. Therefore, the "?", token used after the <operation>, uses one byte in US-ASCII, but two in a double byte encoding.

For example, if Unicode (a double-byte encoding) is used, the head will still be US-ASCII as follows:

```
wfmc-if4-request[4]ABC+123&1998-04-25T04:25:39Z,encoding=UTF-16&&
```

The rest of the message will be encoded in UTF-16. This convention allows the receiving engine to recognize the encoding.

7.2.2 Tail

The tail contains the word "end" followed by a checksum in parenthesis. The tail must be encoded in US-ASCII. This allows the receiving engine to detect message modification. The mechanism does not detect all modification, but it is useful to detect those introduced by the mail system (for example: extra blanks or extra ~~carry~~carriage returns). The tail follows the encoding defined in the head. For example:

```
end(40388)
```

If the tail is not present or does not finish in ")", then the message is considered truncated. Truncated messages must be reported as errors to the transmitting engine. If the checksum calculated by the receiving engine is different from the checksum in the tail, then the message is considered modified. Modified messages must be reported as errors to the transmitting engine. Truncated or modified messages must not be processed by the receiving engine.

The checksum is calculated from the first byte (always a "w" of either wfmc-if4-request or wfmc-if4-response) to the last byte before the checksum in parenthesis (always the "d" of "end"). The following C code implements the variant of the Fletcher checksum algorithm [Fletcher82] used in this specification:

```
unsigned int checksum(const unsigned char *s)
/* Calculates the WfMC message checksum.
 * s is a null terminated string containing the first part of the message.
 * for example: s = "wfmc-if4-request . . . &&end";
 */
{
    unsigned int a, b;

    /* the first byte of the message is always the 'w' in "wfmc-if4-" */
    assert('w' == *s);
    a = b = 0;
    while(*s)
    {
        a = (a + *s++) % 255;
        b = (a + b) % 255;
    }
    /* the last byte is the 'd' in "end" */
    assert('d' == *(--s));
    return (a * 256) + b;
}
```

7.2.3 Special Characters

The MIME text/plain content type, used in this protocol, uses US-ASCII as the default character set, but it also supports the ISO 8859 family (8859-1 Latin 1 (West European), 8859-2 Latin2 (East European), 8859-3 Latin3 (South European), 8859-4 Latin4 (North European), 8859-5 Cyrillic, 8859-6 Arabic, 8859-7 Greek, 8859-8 Hebrew, and 8859-9 Latin5 (Turkish)). US-ASCII characters from 0 to 127 (hexadecimal 00 to 7F) are the same in all of these character sets.

The first part of the message (protocol data) is sent using the set of bytes {10, 13, 32 to 126} (hexadecimal {0A, 0D, 20 to 7E}). This is achieved by escaping bytes that are outside this set. Escaping has two objectives:

- It guarantees that only bytes in the set {10, 13, 32 to 126} are included in the protocol data.
- It guarantees that bytes in the set {10 (line feed), 13 (~~carry~~carriage return), 37 (%), 38 (&)} (hexadecimal {0A, 0D, 25, 26}) are used accordingly to the protocol. These bytes have specific functions within the protocol data and any other appearance must be escaped.

Escaping depends on the byte encoding (<encoding name>). Single-byte encodings, such as US-ASCII and UTF-8 are escaped in one pass. Multi-byte encodings must be escaped in two passes.

7.2.3.1 Escaping Single-Byte Encodings

In single-byte ~~encodings~~encoding, bytes are encoded based on the following table:

dec	hex	Comment
0-31	00-1F	Always escaped
32-36	20-24	Clear (not escaped)
37	25	Used by the protocol to escape bytes, otherwise escaped.
38	26	Used by the protocol to indicate the end of a field value, otherwise escaped.
39-126	27-7E	Clear (not escaped)
127-255	7F-FF	Always escaped

Escaping is achieved by replacing the byte with its hexadecimal representation. The percent sign (%) is used to identify byte escaping. It must be followed by:

- The pair of hexadecimal digits representing the byte being escaped. For ~~example~~example, “K” can be escaped as %4B.
- Another ”%” when the character being escaped is the percent sign. Therefore, a % in the message is escaped as %%.
- A “[“ followed by a series of hexadecimal pairs and closed by a “]” to escape a block of bytes. This modality is useful to escape binary fields. For example, the string “WfMC” can be escaped as “%[57664D43]”.

Any other use of the “%” byte results in an invalid escape sequence (error code 209).

Examples

%7E	Tilde “~”
%7K	Invalid escape sequence. K is not a hexadecimal digit.
%7	Invalid escape sequence. There is only one digit after the “%”.
%%	Percent sign “%”
%$	Invalid escape sequence. “$” is not an hexadecimal digit.
%[55]	Capital letter “U”
%[676F6F64]	String “good”
%[555]	Invalid escape sequence. Missing one hexadecimal digit.

7.2.3.2 Escaping Multi-Byte Encodings~~s~~

Multi-byte ~~encodings~~encoding must be escaped in two steps. The first step is called character escaping, because it is done using the characters of the encoding. The second step is to apply a byte escaping to the encoded message.

7.2.3.2.1 Multi-Byte Character Escaping

Character escaping is used to escape the characters "%" and "&", when they appear in a field value (<value>). These characters are used as markers by the protocol, and so must be escaped in the field value to avoid confusion. Character escaping must be done when placing field values in the message. The following table shows how these characters are escaped:

Character	Encoded as	Comments
%	%%	Four bytes in a double byte encoding.
&	%26	Six bytes in a double byte encoding.

For example, the following segment of a SetProcessInstanceAttributes:

```
name=exp&type=WMTText&value=23 && ( foo % 2)&
```

Is wrong, because the && and % within the value must be encoded. The correct segment is:

```
name=exp&type=WMTText&value=23 %26%26 ( foo %% 2)&
```

7.2.3.2.2 Multi-Byte Byte Escaping

After all the operations (<operation list>) had been placed in the message, a byte escaping is conducted. The byte escaping consist on escaping each byte that is in the set {0-31, 37, 127-255}. This process is conducted as specified in 7.2.3.1 Escaping Single-Byte Encoding~~Escaping Single-Byte Encoding~~~~Escaping Single-Byte Encodings~~.

In some encodings, it maybe practical to escape the whole <operation list> in between "%[" and "]".

Note that this procedure will encode all occurrences of the percent sign (%) byte. In some encodings, this will result in a double escaping of the percent sign. That is an intended effect.

7.2.4 Lines

Following the MIME's text media type, a line is terminated by a carriage return (CR) followed by a line feed (LF). This specification does not rely on the use of lines; however, the email systems processing plain/text do need lines. Although the plain/text MIME specification allows lines of 988 bytes, this specification uses 60 bytes (similar to UUENCODE).

A line that exceeds 60 bytes must be broken, by inserting a ~~carry~~carriage return (CR) and line feed (LF), in that order. Therefore, In this specification a line does not exceed 62 bytes (60 bytes plus ~~carry~~carriage return and line feed). The ~~carry~~carriage return (CR) and line feed (LF) must be removed by the receiving engine before processing the message.

Line breaking must be done after character encoding and byte encoding.

Neither the sender nor the receiver should add any extra bytes around the ~~carry~~carriage return (CR) and line feed (LF), so that the reconstructed message is identical to the original message ("messCRLFage" becomes "message"). Note, that any CR or LF inside the message must be converted to %13 or %10, as any other special byte.

Example:

The following two fragments, are exactly the same (with exception of the checksum, which is calculated including the CRLFs):

```
wfmc-if4-request[0]K1234+&1998-04-25T04:25:39Z&&
StartConversation?ContractID=Nice Group&Version=1.1&
SourceNodeID=xyz@wfmc.org&RootPID=24&
ProductID=MagicWorkflow/5.0&OpID=1&
SourcePID=24&SourceConversationID=K1234&&
CreateProcessInstance?ProcessDefinitionID=Open Account&
Profile=chain&OpID=2&&
SetProcessInstanceAttributes?OpID=3&Number=2&
Name=copies&Type=WMTINT8&Value=3&
NAME=description&TYPE=WMTTEXT&VALUE=Two lines of text, thi
s is the first line%13%10This is the second line.&&
ChangeProcessInstanceState?OpID=4&State=~~start~~open.running&&
StopConversation?OpID=5&&
end(~~47168~~38341)
```

It is the same as:

```
wfmc-if4-request[0]K1234+&1998-04-25T04:25:39Z&&StartConvers
ation?ContractID=Nice Group&Version=1.1&SourceNodeID=xyz@wfm
c.org&RootPID=24&ProductID=MagicWorkflow/5.0&OpID=1&SourcePI
D=24&SourceConversationID=K1234&&CreateProcessInstance?Proce
ssDefinitionID=Open Account&Profile=chain&OpID=2&&SetProcess
InstanceAttributes?OpID=3&Number=2&Name=copies&Type=WMTINT8&
Value=3&NAME=description&TYPE=WMTTEXT&VALUE=Two lines of tex
t, this is the first line%13%10This is the second line.&&Cha
ngeProcessInstanceState?OpID=4&State=open.running&&StopConve
rsation?OpID=5&&end(14814)
~~ation?ContractID=Nice Group&Version=1.1&SourceNodeID=xyz@wfm
c.org&RootPID=24&ProductID=MagicWorkflow/5.0&SourcePID=24&So
urceConversationID=K1234&&CreateProcessInstance?ProcessDefin
itionID=Open Account&Profile=chain&&SetProcessInstanceAttrib
utes?Number=2&Name=copies&Type=WMTINT8&Value=3&NAME=descript
ion&TYPE=WMTTEXT&VALUE=Two lines of text, this is the first
line%13%10This is the second line.&&ChangeProcessInstanceSta
te?&State=start&&StopConversation?&&end(17828)~~
```

7.3 Ensemble the Protocol Data

The order in which the first part (protocol data) is ~~ensemble~~ensemble into the message is important. In order for character escaping, multi-byte encoding, line breaking, truncation, and modification detection to work, the protocol data must be ensemble in the following order:

1. Add the operations with all its fields and values, using the encoding that will be used by the message (<encoding name>). If the encoding is a multi byte encoding, each character may use multiple bytes.
2. In a single byte encoding, the escaping of single byte encodings must be applied (section 7.2.3.1).
3. In multi-byte encoding, when adding field values, the multi-byte character escaping must be applied (section 7.2.3.2.1).
4. In multi-byte encoding, after all the operations and fields had been added, the multi-byte byte escaping must be applied (section 7.2.3.2.2).
5. Prefix the message with the <head> in US-ASCII (section 7.2.1).
6. Postfix the message with the “end” of <tail> in US-ASCII (section 7.2.2).
7. Break the message in lines of no more than 60 bytes (section 7.2.4). Use ~~with~~ CRLF ~~(section 7.2.4)~~to break the lines.~~.~~
8. Calculate the checksum (section 7.2.2).
9. Append the checksum enclosed in parenthesis to the end of the message (as specified in <tail>) (section 7.2.2).

The receiving engine should ~~disassemble~~disassemble the message in the following order:

1. Verify that the message is not expired. If it is, then ignore the message. Note that only US-ASCII is required for this check.

~~1.~~2. Verify that <head> matches the message subject otherwise issue an invalid message (error code 205). Note that only US-ASCII is required for this check.

~~2.~~3. Verify the message is not truncated, otherwise issue truncated message (error code 201). The message must contain the ")" of <tail>. Note that only US-ASCII is required for this check.

~~3.Verify that the message is not expired. If it is, then ignore the message.~~

4. Verify the message checksum by recalculating the checksum. If checksum is not identical issue modified message (error code 202). Note that only US-ASCII is required for this check.

5. Verify that this engine support the encoding specified in the <head> of the message. If the engine does not support the encoding then issue invalid or unsupported encoding (error code 203). Note that only US-ASCII is required for this check.

6. Strip the message of all the CRLFs.

7. In a multi-byte encoding, undo the byte escaping. If an invalid escaping sequence is encounter, then report it (error code 209).

8. In multi-byte encoding, when extracting the value of each field, undo the character escaping. If an invalid escaping sequence is encounter, then report it (error code 209).

9. In a single-byte encoding, undo the byte encoding while tokens are extracted.

7.4 Attachments

Attachments can be used to pass files between workflow engines. When using attachments, the MIME content type must be multipart/mixed and the first part must be text/plain. The first part contains the protocol data, as defined above, the other parts are considered attachments.

Attachments may be referenced by part number or, optionally, a filename. The use of filenames may be supported by workflow engines with access to an electronic mail system that implements content disposition [RFC 2183]. The business contract must specify which kind of attachment referencing is supported.

Each part of the multipart message is assigned a part number. The preamble is assigned zero, the text/plain part that contains the protocol data is assigned one, and any other attachments get progressively higher numbers. An attachment maybe referenced by using the part number.

To reference an attachment by a filename, the Content-Disposition header field must be provided for that part of the message. The sending workflow engine must ensure that all filenames it sends out in a particular message are unique. A sample Content-Disposition header is:

```
Content-Disposition: attachment; filename="i4prop2.doc"
```

~~Attachments are referenced by part number. Each part of the multipart message is assigned a number. Zero is assigned to the preamble, one to the first part, and so on. The assigned numbers refer to the attachment, therefore, an attachment number is always greater than one.~~

Note that checksum does not apply to attachments. This protocol does not address truncated or modified attachments.

7.5 Examples

This section shows two examples of the MIME messages. The first requires some attachments, so it is a multipart message. The second does not require attachments, so it is considered a simple message.

7.5.1 Multipart Message

When attachments are used, a multipart/mixed context type message must be used. The following example shows this case:

```
From: Workflow Engine X <xyz@wfmc.org>
To: Workflow Engine Y <y@wfmc2.org>
Subject: wfmc-if4-request[0]K1234+&
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple boundary"

This is the preamble. It is part Zero, and it is ignored.

--simple boundary
Content-type: text/plain
wfmc-if4-request[0]K1234+&1998-04-25T04:25:39Z&&
StartConversation?ContractID=Nice Group&Version=1.1&
SourceNodeID=xyz@wfmc.org&RootPID=24&OpID=1&
ProductID=MagicWorkflow/5.0&
SourcePID=24&SourceConversationID=K1234&&
CreateProcessInstance?ProcessDefinitionID=Open Account&
Profile=chain&OpID=2&&
SetProcessInstanceAttributes?OpID=3&Number=4&
Name=copies&Type=WMTINT8&Value=3&
NAME=description&TYPE=WMTTEXT&VALUE=Two lines of text, thi
s is the first line%13%10This is the second line.&
name=file1&type=~~wmtmime~~WMTAttachment&value=2&
name=file2&type=~~WMTMIME~~WMTATTACHMENT&value=3&&
ChangeProcessInstanceState?OpID=4&State=~~start~~open.running&&
StopConversation?OpID=5&&
end(~~31082~~7224)
--simple boundary
Content-type: text/plain; charset=us-ascii

This is an attachment. It is attachment number two (name=file1&).
The attachment could be any context type, but in this case it is text/plain.

--simple boundary
Content-type: text/plain; charset=us-ascii

This is another attachment. It is attachment number three (name=file3&).
This attachment could be any context type, but in this case it is text/plain.

--simple boundary—

This is the epilogue. It is part four, and it is also ignored.
```

7.5.2 Simple Message

If attachments are not required, then a text/plain context type message must be used. The following example shows such a message:

```
From: Workflow Engine X <x@wfmc1.org>
To: Workflow Engine Y <y@wfmc2.org>
Subject: wfmc-if4-response[1]K1234+ABC&&
MIME-Version: 1.0
Content-type: text/plain
wfmc-if4-response[1]K1234+ABC&1998-04-25T05:00:33Z&&
StartConversation?ErrorCode=0&SourceConversationID=K1234&
TargetConversationID= ABC&TargetNodeID=abc@wfmc.org&
Version=1.1&ProductID=BetterMagicWorkflow/8.1&OpID=1&&
CreateProcessInstance?ErrorCode=0&OpID=2&
TargetProcessID=32&state=open.notRunning.NotStarted&&
SetProcessInstanceAttributes?ErrorCode=0&OpID=3&
Number=4&name=copies&name=description&
Name=file1&name=file2&&
ChangeProcessInstanceState?ErrorCode=0&state=~~start~~open.running&OpID=4&&
StopConversation?ErrorCode=0&OpID=5&&
end(~~51136~~27234)
```

8 Protocol

This interoperability protocol is accomplished over electronic mail, which is an unreliable communication channel. In this specification, each workflow engine is uniquely identified by the mailbox to which it "listens" (i.e. its mail address). Each message that one workflow engine sends to another is numbered. Moreover, each workflow engine uniquely identifies a message by its type (request, response, or error) and by its ~~number~~sequence.

Even with these precautions, several problems can occur on an unreliable communication channel. This protocol accounts for messages that are:

- Out of sequence
- Duplicate
- Lost
- Truncated
- Modified

The protocol provides for out of sequence and duplicate messages by establishing a conversation in which all messages are enumerated. This specification includes ~~a retransmission strategy that manage~~a retransmission strategy that manages lost messages. The checksum mechanism, described in the previous section, manage truncated and modified messages. This section addresses the following issues:

- Retransmission
- Duplication
- Conversation establishment and termination
- Flow control
- Crash recovery

8.1 Flow Control

This specification assumes that there is no limit on the number of messages a workflow engine can receive, and it does not consider mail overflow.

Incoming mail, within a conversation, must be processed in the order specified by the m~~M~~essage-id~~ID~~ (see 9.2.6 Message Id). Each message is numbered as follows:

Message	**Numeration** (message-id)
Request messages originating from the source workflow engine	- Start with 0 (StartConversation) - Last message number sent + 4
Response messages originating from the target workflow engine	- Received message number + 1
Request messages originating from the target workflow engine	- Start with 2 - Last message number sent + 4
Response messages originating from the source workflow engine	- Received message number + 1

Within each request message, individual operations are numbered starting from 1, so that they can be accounted for. This is accomplished by using an operation-id (see 9.2.9 Operation Id).

A receiving engine is always expecting one of the following m~~M~~essage-id~~ID~~s (from a particular n~~N~~ode-id~~ID~~ within a c~~C~~onversation-id~~ID~~) to arrive:

- A wfmc-if4-request with a m~~M~~essage-id~~ID~~ that is four more than the last request from the other engine

  The first message from the other engine is either zero (during conversation establishment), or two, which is the first target engine notification (wfmc-if4-request) message.

- A wfmc-if4-response with a m~~M~~essage-id~~ID~~ that is one more that the request this (receiving) engine just sent

- A wfmc-if4-error with the same m~~M~~essage-id~~ID~~ as the last message sent by this (receiving) engine.

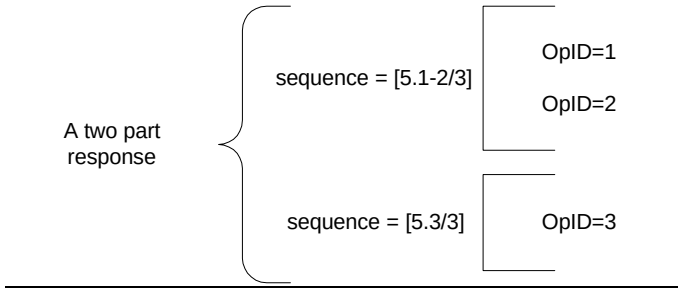
A received message that does not comply with the expected m~~M~~essage-id~~ID~~ is duplicated or out-of-sequence or it has an invalid m~~M~~essage-id~~ID~~. Duplicate messages are treated as specified in 8.3 Duplication Detection. An invalid m~~M~~essage-id~~ID~~ is one that does not comply with the sequencing rules as specified in 9.2.6 Message Id. An invalid m~~M~~essage-id~~ID~~ must be answered with a wfmc-if4-error.

An engine that receives an out-of-sequence message must initially ignore it. The receiving engine ~~may~~ should wait until it processes new messages, then process this message in the correct sequence~~, or it may ignore the message completely, in which case the message becomes lost (the sending engine will resent it)~~.

8.1.1 Partial response messages

In the case in which a workflow engine need to send a partial response message, the <sequence> variance with <op range> should be used (see 7.1.3 Response Messages, 7.2 Protocol Data, and 9.2.9 Operation Id). In this case, the receiving engine must be able to order the messages using the <op range> in addition to the message-id. For example, in a normal situation, the response message has a <sequence> that is composed by a message-id alone, as follows:

In a partial response, the <sequence> is composed by the message-id and the <op range> as is shown in this partial response:



8.2 Retransmission Strategy

The retransmission strategy handles lost messages. A timer is associated with each message sent. If the timer expires before the response is received, the request must be sent again.

The parties of a business agreement assign an arbitrary timer value that is large enough to allow a message round trip (the time that elapses between sending an operation and receiving its response). The timer granularity must be large (several hours or days) to avoid clock synchronization problems.

The sending engine resends a request if the timer expires before it receives the response. The receiving engine must ignore messages for which the <timestamp> plus the timer value is less than the current UTC (Coordinated Universal Time) time, because they are expired messages.

The sending engine must retry lost messages as many times as specified in its business contract. The sending engine must notify a human administrator if it is unable to contact the other engine after the specified number of retries.

Timers solve the problem of a duplicated and delayed StartConversation, as described in the next section.

8.2.1 Resending Messages

The workflow engines must keep copies of the messages they send out, so that if the message is lost it can be resent. These copies must be kept until the workflow engine receives a new message from the other engine on the same conversation. The StopConversation or it response must be kept for double of the timer period.

The workflow engine must update the timestamp of a message before it is resent, otherwise the receiving engine will ignore the message, because it will be expired.

8.3 Duplication Detection

A message from a particular n~~N~~ode-id~~ID~~ (see 9.2.8 Node Id) that has the c~~C~~onversation~~ID~~-id (see 9.2.4 Conversation id) and the m~~M~~essage-id~~ID~~ (see 9.2.6 Message Id) of a~~n~~ message already processed ~~message~~ is a duplicated message. Duplicated messages occur in two situations:

- During the current conversation.
- After the conversation finished. In particular, a duplicated start conversation message that arrives days after the conversation completed must be handled correctly.

8.3.1 Duplicate Within the Same Conversation

In this case, the receiving engine receives a message that has an active c~~C~~onversation-id~~ID~~, but the m~~M~~essage-id~~ID~~ has been used, in one of the following ways:

- Message--id is the same as the last message--id received, in which case the receiver must assume that its last message was lost and, therefore, must resend all messages after that message
- Message--id is less than the last processed message--id, in which case, the receiver should ignore the message.

8.3.2 Duplicate Outside the Conversation

Most of the time, an invalid conversation--id exposes out-of-conversation message. However, a delayed and duplicate start conversation presents a special problem, because it looks like as a valid message to the receiving engine. It looks like a valid message because, it has a conversation--id that is not in use, it has a message--id of zero, and it contains a start conversation.

The retransmission timer value solves this problem. If the message has expired (expiration time is calculated by adding the timer value to the message <timestamp>) the receiving engine must ignore it.

8.4 Conversation Establishment and Termination

StartConversation marks the beginning of a message interchange between two workflow engines. After the StartConversation, any number of operations and electronic mail messages may be exinterchanged between by the workflow engines. Enactment of a workflow and all the electronic mail interchange required between the source and the target workflow engines must occur within a single conversation.

StopConversation marks the end of the message interchange. StopConversation does not force the workflow engine to terminate its process; rather, it just indicates that the source engine wants to stop the message interchange. It tells the target workflow engine that:

1. It should release all the data structures that pertain to the conversation.
2. It must not send notification (wfmc-if4-request) messages concerning that particular process enactment to the source engine.
3. The process enacted may continue execution, but the source workflow engine has no further interest in it.

After the StopConversation, the cConversation-idID (source or target) is invalid. After the StopConversation, both workflow engines are free to reassign the pProcess-idID (source or target). Note that ProcessIDs can be reassigned, but cConversation-idIDs must not be reassigned. Process-idIDs are not guaranteed across conversation boundaries.

8.5 Crash Recovery

This specification assumes that requests operations are transactional and atomic.

Messages must always be sent after the workflow transaction is committed. Otherwise, they may be received as false messages.

The sending engine must handle the case in which it sends a message, then fails. Whether the message is a request or a response, it must be sent after the workflow transaction is committed. If the engine fails after committing the transaction, but before sending the message, the message is just lost. In a reply, if the engine fails before committing the transaction, the original request message becomes lost. A lost message is preferable to a false messages.

Messages must always be sent after the workflow transaction is committed. Otherwise, they may be received as false messages.

9 Data Types & Declaratives

9.1 Basic Attribute Types

The following table describes all the basic types used in this specification. Type names are case insensitive.

Type	C type	Comments
WMTAttachment		See **Attachments Values** below for details
WMTBinary		See **Binary Values** below for details
WMTBoolean	char	See **Boolean Values** below for details
WMTDate		See **Date and Time Values** below for details
WMTDateTime		See **Date and Time Values** below for details
WMTDouble	double	See **Numerical Values** below for details
WMTFloat	float	See **Numerical Values** below for details

WMTInt16	short	See **Numerical Values** below for details
WMTInt32	long	See **Numerical Values** below for details
WMTInt8	char	See **Numerical Values** below for details
WMTText	char*	See **Numerical Values** below for details
WMTTime		See **Date and Time Values** below for details
WMTUInt16	unsigned short	See **Numerical Values** below for details
WMTUInt32	unsigned long	See **Numerical Values** below for details
WMTUInt8	unsigned char	See **Numerical Values** below for details

9.1.1 Attachments Values

The value placed in the operation is a reference to the MIME attachment. Attachments are only present in multipart MIME messages. Each part of the multipart message is enumerated for purposes of this protocol. The preamble is assigned number zero, the first part (the protocol) is assign number one, and so on. For example, the following segment of a SetProcessInstanceAttributes refers to the first attachment (part number two):

```
name=attach1&type=WMTAttachment&value=2&
```

Alternative, the use of filenames may be supported by workflow engines with access to an electronic mail system that implements content disposition [RFC 2183]. For example, the following segment of a SetProcessInstanceAttributes refers to the file called proposal.doc:

```
name=attach2&type=WMTAttachment&value=proposal.doc&
```

Note that the business contract must specify which kind of attachment referencing is supported.

~~9.1.1~~9.1.2 Boolean Values

Boolean values passed between workflow engines in messages are as follows:

Boolean Value	~~Encoding~~use
True	1
False	0

For example:

```
name=boolField&type=WMTBoolean&value=1&
```

9.1.3 Binary Values

Binary values are placed in the message byte by byte independent of the encoding. A byte in a binary value represents an octet independent of the encoding and independent of the character set being used.

For example, the following binary field:

Binary				hexadecimal
0000 0000	1111 1111	1001 0110	0011 1100	00 FF 96 C3
0101 1010	1010 1010	1111 0000	0111 1110	A5 55 F0 E7

Is placed in the operation, independent of the encoding and independent of the character set, as:

```
name=binaryBlock&type=WMTBinary&value=%[00FF96C3A555F0E7]&
```

~~9.1.2~~9.1.4 Date and Time Values

Valid values for the date and time types are a subset of the ISO 8601 date/time representations. The subset is based on the complete representation of the date and time formats.

Using 8:25:39p.m (twenty-five minutes and thirty-nine seconds after eight o'clock in the evening) of April 24, 1998 in California (Pacific Standard Time) as an example, we obtain the following table of valid date and time formats.

WMTDate	WMTTime	WMTDateTime~~Timestamp~~
1998-04-24	20:25:39	1998-04-24T20:25:39
	04:25:39Z	1998-04-25T04:25:39Z

~~9.1.2.1~~9.1.4.1 Calendar Date Format (WMTDate)

Use the following format: YYYY-MM-DD, where YYYY is the year in the Gregorian calendar; MM is the month of the year, from 01 (January) to 12 (December); and DD is the day of the month, from 01 to 31.

~~9.1.2.2~~9.1.4.2 Time Format (WMTTime)

The following formats are permissible: HH:MM:SS or HH:MM:SSZ where HH is the hours, from 00 to 24; MM is the minutes, from 00 to 59; SS is the seconds, from 00 to 59; and "Z" is the symbol that indicates Coordinated Universal Time (UTC). Midnight may be expressed as 00:00:00 or 24:00:00.

Times in the format HH:MM:SS (without "Z") is considered local. Use of local time is discouraged, because there is no guarantee that two interoperating workflow engines are in the same time zone.

Times in the format HH:MM:SSZ is represented in Coordinated Universal Time (UTC). For practical purposes (when fractions of a second are not important), UTC is ~~similar~~ the same as~~to~~ Greenwich Mean Time (GMT), i.e., the local time at the Greenwich meridian (zero degrees longitude).

~~9.1.2.3~~9.1.4.3 Date~~Timestamp~~ Format (WMTDateTime~~Timestamp~~)

The symbol "T" separates the date and time part of a time-stamp format. The date and time components must follow the formats given in the sections above. The use of UTC time may, depending on the time of day, affect the date in a combined format.

9.1.5 Numerical Values

Numerical values are placed in the message using a string representation of the value. This specification uses the ANSI-C printf/scanf conversions as follows:

Type	**C type**	**ANSI-C printf/scanf format specification**
WMTDouble	double	%lf
WMTFloat	float	%f
WMTInt16	short	%hd
WMTInt32	long	%ld
WMTInt8	char	%d
WMTUInt16	unsigned short	%hu

WMTUInt32	unsigned long	%lh
WMTUInt8	unsigned char	%u

The decimal separator for WMTDouble and WMTFloat must be a period (“.”). Note that some ANSI-C implementations will use the locale specification to determine the decimal separator. Independent of the locale specification, the sending engine must guarantee the separator to be a period, and the receiving engine must accept period as separator.

~~9.1.3MIME Values~~

~~The value placed in the operation is a reference to the MIME attachment. Attachments are only present in multipart MIME messages. Each part of the multipart message is enumerated for purposes of this protocol. The preamble is assigned number zero, the first part (the protocol) is assign number one, and so on. For example, the following segment of a SetProcessInstanceAttributes refers to the first attachment (part number two):~~

~~`name=attach1&type=WMTMIME&value=2&`~~

9.2 Operation Fields

Fields are defined as name-value pairs, separated by the “&” character. There is no positionality of fields in the operations (unless it is specifically stated). A field passed with no given value (null field) passes as the field name only, viz:

```
nullparameter=&
```

This forces the receiving engine to set the field to null.

Optional fields may be omitted from the message. When an optional field is not present, the receiving engine may use a default value or null at its discretion. Default values must be agreed as part of the business agreement.

Operation field names are case insensitive, so contractid, ContractID, and CONTRACTid are all valid spellings.

9.2.1 Activity Id

Name	Type	Variations
ActivityID	WMTText	SourceAID

An activity~~ I~~id identifies a process step. An activity is defined in [WfMC1011] as “a piece of work that forms one logical step within a process”.

The source activity~~ ~~id (SourceAID) variation is used when notifying the source engine of changes in attribute values or state. In this case, it must contain the same value as the received in the StartConversation’s ActivityID. If the StartConversation’s ActivityID was not present or it was null, then this field becomes optional; otherwise, it must contain the appropriate activity-id.~~that value.~~

Example:

```
ActivityID=1.2.3&
```

9.2.2 Business Process Definition ~~Process~~ Name

Name	Type	Variations
BusinessPName	WMTText	SourceBDefName TargetBDefName

This describes the business name of the process definition. The business name maybe different thanfrom the process definition--id. This is an optional field, used for auditing purposes.

Example:

```
SourceBDefName=Credit Verification Process&
```

9.2.3 Contract Id

Name	Type	Variations
ContractID	WMTText	

When workflow engines interoperate,; they do so based on a business agreement. This specification calls the business agreement ContractID. It is an identifier, of type WMTText, which has been agreed upon by the administrators of both workflow engines. This allows the target workflow engine to check the ContractID against a list of business agreements that it is willing to honor. (This specification does not define the mechanism by which administrators to define business agreements.).

The initiating (source) workflow engine sends a ContractID with which the target workflow engine agrees. The source engine may maintain a list of valid business agreements contexts for each workflow engine that is used as a target. The target workflow engine checks the ContractID against a list of valid business agreements contexts for the source engine. It may reject the message if it finds no match. The ContractID allows the target workflow engine to implement several business agreement polices. For example, it accepts requests that are:

- Independent of the SourceNodeID and the ContractID
- From "SourceNodeID=x@wfmc.org", independent of the ContractID
- From "SourceNodeID=x@wfmc.org" that has "ContractID=Contract1234"
- With "ContractID=Charity" independent of the SourceNodeID

It is the responsibility of the target engine to verify that the source engine is adhering to the contract. The target workflow engine must return "not authorized" error when the source workflow engine fails to adhere to the contract.

Example:

```
ContractID=equipment suppliers AB12&
```

9.2.4 Conversation id

Name	Type	Variations
ConversationID	WMTText	SourceConversationID TargetConversationID <conversation id>

Each conversation has a two-part conversation identifier. Each engine in the conversation assigns an arbitrary but unique identifier for its part of the conversation. The initiating workflow engine assigns a SourceConversationID and the target workflow engine (the one with which the initiating workflow engine wishes to start a conversation) assigns a TargetConversationID. If the source engine alone sets the conversation--id, there is a remote the possibility that a target workflow engine working with multiple source engines will receive the same conversation--id from more than one source. Similarly, if the target engine alone sets the conversation--id, a source workflow engine, working with multiple targets, could receive back the same conversation--id from more than one engine. To overcome this problem, a truly unique conversation--id can be achieved by having both engines provide a conversation--id component that is unique to each. The combined source/target pairing of conversation ids is then guaranteed to be unique for both engines.

ConversationID is a field composed of two variants. It is defined as SourceConversationID "+" TargetConversationID. The character "+" is used in ConversationID to separate the source and the target conversation ids. For example:

```
~~        TargetConversationID=1AB01234&~~
        SourceConversationID=210&
        TargetConversationID=1AB01234&
Then
        ConversationID=210+1AB01234&
```

Note that "+" is not a valid character neither for SourceConversationID nor TargetConversationID, because it is used as a separator in ConversationID.

9.2.5 Language

Name	Type	Variations
Language	WMTText	

The language uses the two-character ISO 639-1 code. It indicates which language the source workflow engine is using to communicate with the end user. The target workflow engine may use the same language to set the ErrorText.

Note that language does not specify the encoding. It is only used to specify the language preference for error description (ErrorText).

Language is an optional field in StartConversation, in which case the language of the target engine is assumed.

Example:

```
Language=en&
```

9.2.6 Message Id

Name	Type	Variations
MessageID	WMTInt32	<message id>

Messages within a conversation are uniquely numbered. This allows the workflow engines to protect against messages being lost, duplicated, or delivered out of sequence. The receiving workflow engine must be able to identify:

- Messages it has received and acted on
- Messages received out-of-sequence, thus they are not yet eligible for processing
- Messages not received (e.g., if only messages 0, 4 and 16 have been received, it is possible to deduce that messages 8 and 12 are missing).

The semantic of the message number is either *"Message n I have sent to engine X during conversation S"* or *"Message n I have received from engine X during conversation S"*. To allow each engine to distinguish messages it has sent and messages it has received, both engines numbers the messages they send to the other engine in increments of four. Responses to request/notification (wfmc-if4-request) messages are numbered by adding 1 to the request/notification message number.

The source workflow engine starts its requests (wfmc-if4-request messages) with message-id zero that corresponds to the StartConversation. The target workflow engine replies (wfmc-if4-response) with message-id one (zero plus one). The next source workflow engine message is four (zero plus four), to which the target workflow engine responds with five (four plus one).

This specification provides two notification operations: ProcessInstanceAttributeChanged and ProcessInstanceStateChanged. These two notification operations are initiated by the target workflow engine by using a wfmc-if4-request message.

The target workflow engine starts its notifications (wfmc-if4-request messages) with message-id two. The source workflow engine responds (wfmc-if4-response message) with one more (in this case three).

Interchange error messages (wfmc-if4-error) use the message-id of the bad message, so that the sequencing is not disturbed. An interchange error message may be caused by either a wfmc-if4-request or a wfmc-if4-response.

Example:

Source			Target		
wfmc-if4-request	Wfmc-if4-response	Wfmc-if4-error	wfmc-if4-request	Wfmc-if4-response	Wfmc-if4-error
0					
				1	
4					
				5	
8					
					8
8					
				9	
		9			
				9	
			2		
	3				
			6		
	7				
12					
				13	

Note that:

- Each request has an even message-id.
- Each response has an odd message-id.
- The message-id of a request coming from the source is divisible by four.
- The message-id of a request coming from the target is not divisible by four, but it minus two is divisible by four.

Using the above concepts, it is possible to guarantee that:

- In an environment where there may be many workflow engines conducting interoperability conversations with each other, every conversation is uniquely identifiable
- Within such an environment, every message is uniquely identifiable.

9.2.7 Name-Type-Value Group

Name	Type	Variations
Number	WMTInt16	
Name	WMTText	
Type	WMTText	
Value		

The name-type-value group is used when a list of attributes is required. The name-type-value group is always preceded by number. Number indicates how many name-type-value groups are present in the operation. This is the only case in which field positionality is important within an operation.

Example:

```
number=2&name=first&type=WMTInt8&value=2&name=~~first~~second&type=WMTText&value=test&
```

9.2.8 Node Id

Name	Type	Variations
NodeID	WMTText	SourceNodeID TargetNodeID

The node-id is the electronic mail address of a workflow engine. The receiving engine always ignores the return email address, and instead, uses the node-id to address the other engine (except in the case of message interchange errors).

The node-id in conjunction with the contract-id ~~must~~ will be used by the receiving engine to ~~validate~~ authenticate against its contract information.

Example:

```
NodeID=xyz@wfmc.org&
```

9.2.9 Operation Id

Name	Type	Variations
OpID	WMTUInt16	<operation id>

The operation-id is an enumeration of operations within a request message. Each request message starts with operation-id number =1. The operations in a response message have operation-ids that correspond to the operation-id of the request operation.

A range of operation-ids is composed of the first operation-id in the response message, a hyphen, the last operation-id in the response message, a slash, and the total number of operations in the original request message. A range is only used only when a partial response messages is generated.

Example:

```
OpID=2&
```

~~9.2.9~~9.2.10 Process Definition Id

Name	Type	Variations
ProcessDefinitionID	WMTText	

Process definition-id is the name of a workflow process that can be executed in a target engine by using a CreateProcessInstance operation. It often corresponds to the name of a business process, for example, expense report.

Example:

```
ProcessDefinitionID=expense report&
```

~~9.2.10~~9.2.11 Process Id

Name	Type	Variations
ProcessID	WMTText	RootPID SourcePID TargetPID

The process-id is a unique identifier for a process enactment. This arbitrary and unique identifier is assigned by the workflow engine that enact the process. The source workflow engine is executing a process enactment and, so, has a process-id. A CreateProcessInstance enacts a process in the target workflow engine; so, it has a process-id.

The RootPID is the process-id of the top-level process ~~id~~. Several levels of process enactment occur~~s~~ when a top-level process creates a process instance, that in turn creates another process instance., ~~and so forth.~~ In this case, it is important to know which was the original process-id (process-id of the top-level process).

If the process establishing a conversation has a RootPID, then it should pass it as the RootPID of the conversation. If the process establishing the conversation does not have a RootPID (if it has been enacted locally in the source workflow engine), then the RootPID must be the same as the SourcePID.

~~9.2.11~~9.2.12 Product Id

Name	Type	Variations
ProductID	WMTText	

The product-id identifies the product making the request. This allows the other engine to know which product implements the other end of the conversation. Based on that information, the engines may use extended attributes. Product-id is of the form: <product name>/<version number>.

Example:

```
ProductID=MagicWorkflow/5.1&
```

~~9.2.12~~9.2.13 Profile

Name	Type	Variations
Profile	WMTText	

The profile allows the source workflow engine to indicate what kind of conformance profile it is intended to engage with the target workflow engine. See conformance profiles for details on each profile.

Profile can take one of the following, case insensitive, values:

c~~C~~hain	Simple chains
~~Polling~~nested	N~~Polling n~~ested sub-process

Example:

```
Profile=chain&
```

~~9.2.13~~9.2.14 Return Error Codes

Name	Type	Variations
ErrorCode	WMTInt16	AErrorCode
ErrorText	WMTText	

Request and notification (wfmc-if4-request) messages sent by one workflow engine to another are answered by a response message which tells the sending workflow engine what happened on the receiving workflow engine as a result of the message being sent. There are four possible outcomes to such interactions:

- The request/notification message was not processed due to a preceding error.
- The request/notification message was not processed by the receiving workflow engine because the required functionality was not supported or not implemented.
- The request/notification message was processed successfully.
- An error occurred in the processing of the request/notification message

These four cases are identified to the sending workflow engine via the ErrorCode and ErrorText field in the response message.

Error codes in the range 0 to 255 are used by this specification. Unassigned values in the range 0 to 255 are reserved for future use. The ErrorCode can have the following values:

ErrorCode	Description
0	Success
1	Operation not implemented
2	Not authorized
3	Unknown ProcessID
4	Invalid state
5	Invalid or unknown attribute name
7	Invalid attribute type for this attribute
8	Attribute value does not corresponds to attribute type
10	Invalid or expired ContractID
11	Invalid or unsupported language
12	Invalid or unsuported MIME version
20	There is no conversation in process
30	Operation failed see AErrorCode for details
40	Invalid ProcessDefinitionID
41	Invalid or not implemented profile
201	Truncated message
202	Modified Message (Bad Checksum)
203	Invalid or unsupported encoding
204	Expired Message
205	Invalid Message
206	Invalid ConversationID
207	Invalid Sequence
208	Invalid Timestamp
209	Invalid Escape Sequence
255	Operation not performed. This code is used when a~~n~~ ~~early~~ preceding operation (in the same message) fails
256 to 32767	Failure. Implementation specific error code.

The ErrorText field allows for a short explanatory message that explains the error code to be passed back (this is an optional field). When possible, it should use the StartConversation language argument to decide which language to use for the error message. The StartConversation language field indicates which language the source workflow engine is using to interact with the end user. This scheme allows the target workflow engine the possibility of returning a meaningful value for the failure state, which might convey to the source workflow engine what went wrong, rather than just the fact that something went wrong. The values for Failure may be workflow engine-specific or they may be the result of an agreement between the designers of the two interoperating workflow definitions.

In a message with several operations, if one operation fails, the rest of the operations are ~~ignored~~not attempted (including the StopConversation). Their error code is set to 255 (operation not performed). The following message segment illustrate that case:

```
StartConversation?ErrorCode=0&SourceConversationID=K1234&
TargetConversationID= ABC&TargetNodeID=abc@wfmc.org&
Version=1.1&ProductID=BetterMagicWorkflow/8.1&OpID=1&&
CreateProcessInstance?ErrorCode=40&OpID=2&&
SetProcessInstanceAttributes?ErrorCode=255&OpID=3&&
ChangeProcessInstanceState?ErrorCode=255&OpID=4&&
StopConversation?ErrorCode=255&OpID=5&&
```

~~9.2.14~~9.2.15 Role Id

Name	Type	Variations
RoleID	WMTText	SourceRoleID TargetRoleID

A role is defined in [WfMC1011] as "A group of participants exhibiting a specific set of attributes, qualifications and/or skills" and it is normally associated with security privileges. The use of roles is optional in this specification and caution is recommended when sending role information to third party organizations.

Example:

```
RoleID=FinaceClerk&
```

~~9.2.15~~9.2.16 State

Name	Type	Variations
State	WMTText	

~~State can take one of the following, case insensitive, values:~~

~~Abort~~	~~Process was stopped before completion. Subordinate work is being allowed to continue.~~
~~Complete~~	~~Process finished normally.~~
~~Start~~	~~Process is executing. Start can be used to both to start a NotStarted process, and to resume a suspended process.~~
~~Terminate~~	~~Process and all subordinate work was stopped before completion.~~
~~Suspend~~	~~Process is temporarily paused.~~
~~NotStarted~~	~~Process had been created but it has not started execution.~~

States are organized into several levels of granularity, lower level states refining higher-level ones. An implementation might choose to support states on any level of granularity, omit states and add additional states to the list defined below. A state is identified by its name that includes its super-state parents using dot notation.

The top level of states for a Process Instance distinguishes two states, *open* and *closed*. The open state has two sub-states, *running* and *notRunning*; notRunning in turn has two sub-states, *notStarted* and *suspended.* The following list describes the states in detail:

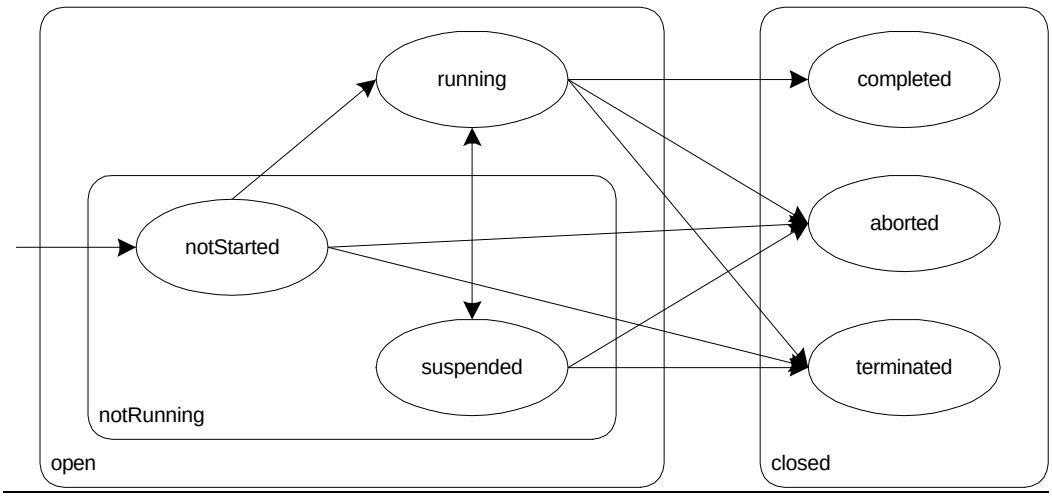
State	Description
open	The Process Instance is enacted
open.running	The Process Instance is executing

open.notRunning	The Process Instance is temporarily not executing
open.notRunning.notStarted	The Process Instance has been created, but was not started yet
open.notRunning.suspended	Execution of the Process Instance was temporarily suspended
closed	Enactment of the Process Instance has been finished
closed.aborted	Enactment of the Process Instance has been aborted. It is an abnormal termination with no attempt to terminate sub-processes. It is used in catastrophic circumstances where nothing except clearing the process away can be done.
closed.terminate	Enactment of the Process Instance has been terminated. It is an abnormal but graceful termination, in which an attempt to terminate all running activities and sub-processes is attempted.
closed.completed	Enactment of the Process Instance has completed normally.

An implementation might decide to support refinement of states to a certain level only or omit certain states; valid sets of states include for example:

- *open* and *closed*
- *notRunning*, *running* and *closed*
- *notStarted*, *running*, *completed* and *terminated*
- ...

The following diagram shows the states and potential state-transitions; transitions are shown for the bottom-level states only, transitions between the higher-level states can be deduced from that easily; e.g., there is a transition from *open* to *closed* or from *notRunning* to *running*, but no transition backwards in both cases.



Note that:

- State names are case insensitive.
- Transitions can be made from *notRunning* states to the *running* state; transitions from the *running* to the *notRunning* super-state can be made to the *suspended* sub-state only.
- When enactment of a Process Instance is finished, its state will take one of the flavours of the *closed* state, depending on the way of ending enactment (normally *completed*, *terminated* or *aborted*). The *completed* state can only be reached from the *running* state since it represents normal completion of

the Process Instance; the other *closed* sub-states are reached via the WMAbrtProcessInstance or WMTerminateProcessInstance operations.

- The *closed* state is a final state, i.e., there is no transition from a *closed* state to an *open* state.

Example:

```
State=~~start~~open.running&
```

~~9.2.16~~9.2.17 Timestamp

Name	Type	Variations
Timestamp	WMTDateTime~~stamp~~	<timestamp>

The timestamp when the message was sent. The time stamp used for <timestamp> must use Coordinated Universal Time (UTC).

~~9.2.17~~9.2.18 User Id

Name	Type	Variations
UserID	WMTText	SourceUserID TargetUserID

A user (participant) is defined in [WfMC1011] as "A resource which performs the work represented by a workflow activity instance" and it is normally associated with security privileges. The use of users is optional in this specification and caution is recommended when sending user information to third party organizations.

Example:

```
UserID=jdoe&
```

~~9.2.18~~9.2.19 Version

Name	Type	Variations
Version	WMTText	

Version of MIME binding used to construct messages. For this specification, it must contain the value: 1.1

Example:

```
version=1.1&
```

10 Conformance

A vendor can claim conformance to this or any other WfMC specification only when specifically authorized to make that claim by the WfMC. A vendor of a workflow engine who claims conformance with this interface shall clearly indicate:

- Mechanism/binding required to effect interoperability e.g., Internet e-mail MIME.
- the style of interoperability dialogue supported:
  - atomic
  - batched
  - both
- Style of dialogue supported:
  - half duplex

    - full duplex
- Conformance profile supported:
    - Simple Chains
    - Polling nested sub-process
    - Suspended animation nested sub-process
    - Deferred synchronous nested sub-process
    - Synchronized enactment nested sub-process
    - Process administration
- Audit data implementation:
    - WfMC audit data
    - Product specific audit data
    - No audit data

A conforming implementation of this Functional Area of the Workflow Management Coalition specification must support implementation of the relevant portion of the Administration and Monitoring specification through capturing and maintaining the audit data specified. The audit data specified in this document indicates the information that a conforming workflow engine must be able to present, in relation to the status and history of interoperation with other workflow engines, relating to enactment of a process instance. It is not intended as a specification of how or where that information is stored and maintained.

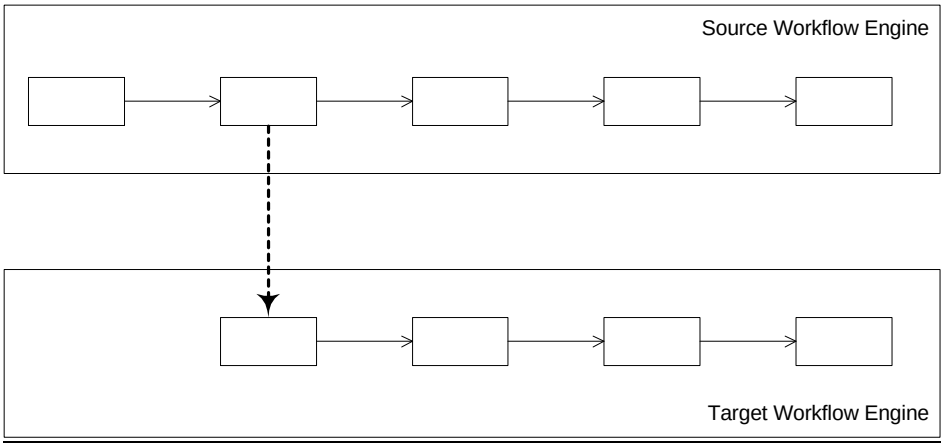
It should be noted that workflow engine interoperability implies more than just the ability to have one workflow engine do things at the behest of another. It requires that both workflow engines are capable of doing things at the behest of the other. Ideally the set of things that each can do for the other will coincide.

The WfMC Interoperability Proving Framework [WFMC1021] outlines a scheme against which vendors may test and which systems integrators and end users can use as a basis for assess compatibility of products that claim to use this binding as a means of interoperability with other workflow products.

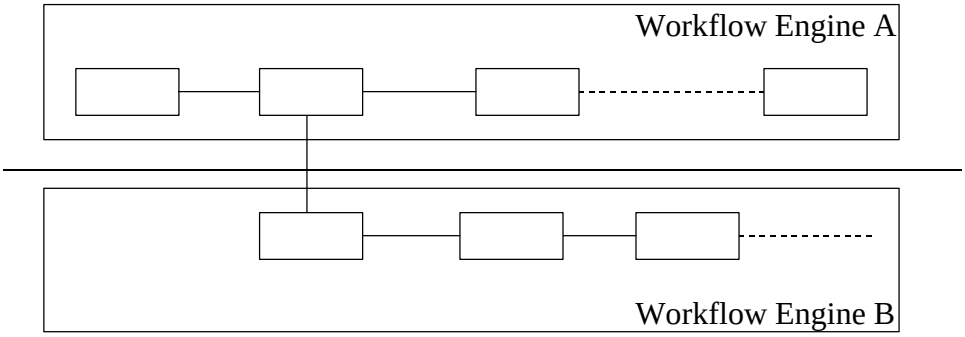
10.1 Conformance Profiles

This version of the MIME Binding supports implementation of interoperability scenarios characterised by the conformance profiles given below. More complex interoperability scenarios may be achievable, but are not defined here as they are the subject of ongoing work by the Workflow Management Coalition.

10.1.1 Simple Chains



10.1.1.1 Intended Use



The simple chains conformance profile is typified by the ability of: one workflow engine to

- Create a process instance on another according to a known process definition;
- Instantiate a process instance created on another given the appropriate process identifier and permissions;
- Cause enactment of an identified, instantiated process instance on another workflow engine.

~~The simple chains conformance profile allows one workflow engine to create a process instance on another according to a known process definition; to instantiate the process instance and to cause enactment of the instantiated process instance.~~

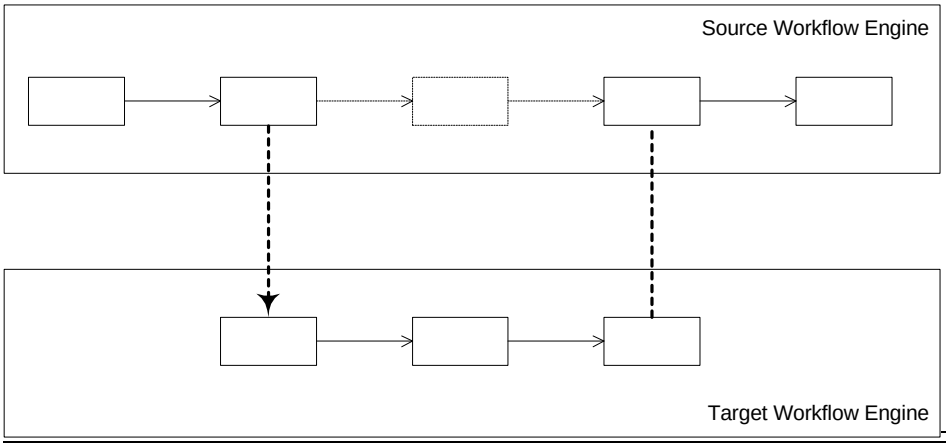
10.1.1.2 Operations

Source Workflow Engine	**Target Workflow Engine**
StartConversation	
StopConversation	
CreateProcessInstance	
SetProcessInstanceAttributes	
ChangeProcessInstanceState (~~Start~~open.running)	

There is no presumed ordering of operations beyond the requirement that the interoperation must begin with a StartConversation and finish with a StopConversation. However, if the process instance on the target workflow engine is to be created, the first operation following StartConversation must be CreateProcessInstance (you cannot set attribute values or initiate enactment for a process instance that does not exist).

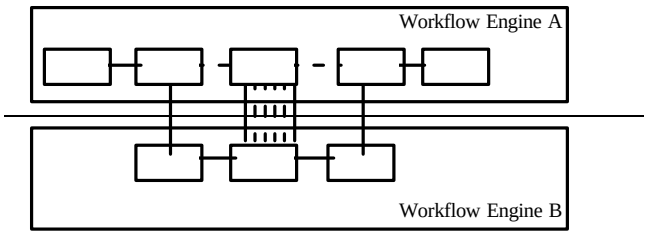
Permission to interoperate with a designated process instance on another workflow engine is established through authentication of the request messages according to rules in the prevailing interoperability contract.

10.1.2 Nested Sub-process



~~(Polling)~~

10.1.2.1 Intended Use



Nested sub-process is characterised by the ability of

- One workflow engine to create a process instance on another according to a known process definition;
- One workflow engine to instantiate a process instance created on another given the appropriate process identifier and permissions;
- One workflow engine to cause enactment of an identified, instantiated process instance on another workflow engine
- The source workflow engine to know when the sub-process has completed and optionally to take back resulting attribute values to be used in the ongoing enactment of the parent process instance.

~~The nested sub-process (polling) conformance profile allows one workflow engine to create a process instance on another according to a known process definition; to instantiate the process instance and to cause enactment of the instantiated process instance. The invoking workflow engine carries on with its own enactment until it reaches a point where it needs to effect a rendezvous with its child sub-process. At this stage it polls the enacting workflow engine to determine when the sub-process has reached completion. It is also possible that the sub-process on the enacting engine is prematurely terminated before~~

~~***the parent process reaches its rendezvous point. Polling is effected using the GetProcessInstanceState operation, which will return the state, one of:***~~

~~- NotStarted~~

~~- Start (up and running)~~

~~- Suspend~~

~~- Completed~~

~~- Terminated~~

~~- Aborted~~

~~The invoking process can use the GetProcessInstanceAttribute operator to retrieve any consequent values from the sub-process before releasing it.~~

10.1.2.2 Operations

Source Workflow Engine	**Target Workflow Engine**
StartConversation	
StopConversation	
CreateProcessInstance	
SetProcessInstanceAttributes	
ChangeProcessInstanceState (open.running)	
ChangeProcessInstanceState (closed.aborted)	
ChangeProcessInstanceState (closed.terminated)	
ProcessInstanceStateChanged (closed.aborted)	
ProcessInstanceStateChanged (closed.terminated)	
	ProcessInstanceStateChanged (open.running)
	ProcessInstanceStateChanged (closed.aborted)
	ProcessInstanceStateChanged (close.terminated)
	ProcessInstanceStateChanged (closed.completed)
	ProcessInstanceAttributeChanged
GetProcessInstanceAttribute	
GetProcessInstanceState	

There is no presumed ordering of operations beyond the requirement that the interoperation must begin with a StartConversation and finish with a StopConversation. However, if the process instance on the target workflow engine is to be created, the first operation following StartConversation must be CreateProcessInstance (you cannot set attribute values or initiate enactment for a process instance that does not exist). Similarly, the target workflow engine cannot notify the source workflow engine of changes in the values of designated process instance attributes or of the state of the sub-process instance before enactment of the sub-process instance has commenced.

Permission to interoperate with a designated process instance on another workflow engine is established through authentication of the request messages according to rules in the prevailing interoperability contract.

A source engine, in this profile, that is being terminated must terminate all the sub-processes by sending them a ChangeProcessInstanceState (closed.terminated). The “terminate” succeed only if the sub-processes are terminated, otherwise it fails.

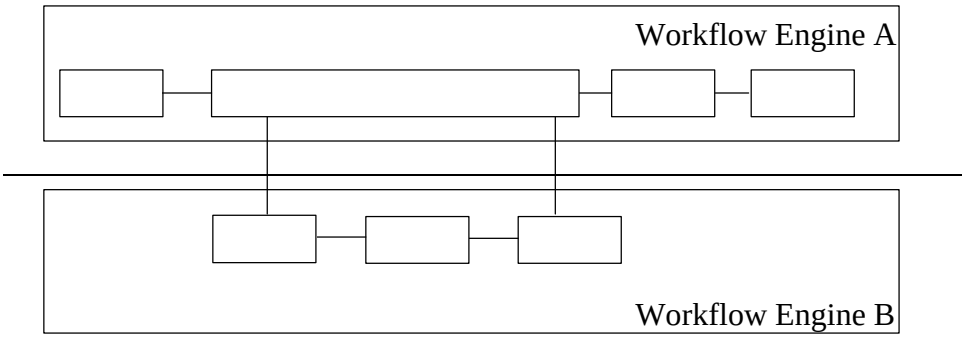
A source engine, in this profile, that is being aborted may do one of the following:

- Attempt to abort all the sub-processes by sending them a ChangeProcessInstanceState (closed.abort).
- Notify all the sub-processes of its state change by sending them a ProcessInstanceStateChanged (closed.abort).

The aborting source engine may not wait for the responses from the sub-processes, to complete its abort.

10.1.3Nested Sub-process (Suspended animation)

10.1.3.1Intended Use



The nested sub-process (suspended animation) conformance profile allows one workflow engine to create a process instance on another according to a known process definition; to instantiate the process instance and to cause enactment of the instantiated process instance. The invoking workflow engine must wait for the sub-process to complete before it can continue its own enactment, i.e. it is in a state of suspended animation. The invoking workflow engine will receive notification of how the sub-process achieves termination. There are three possibilities:

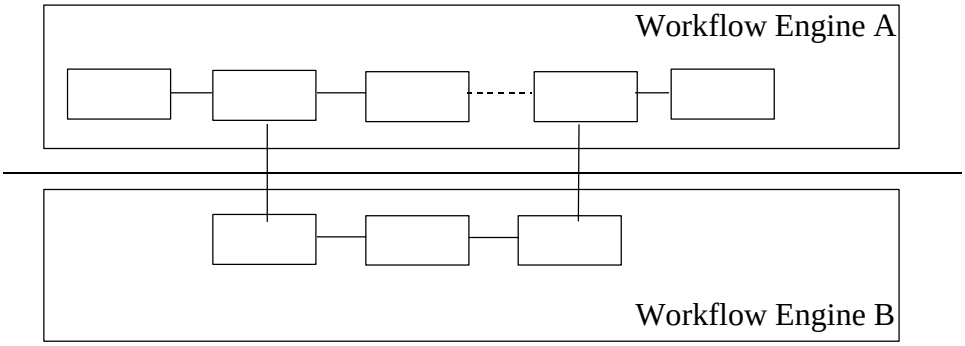
- the sub-process completes normally - Process Instance Completed
- the sub-process is aborted locally - Process Instance Aborted
- the sub-process is terminated locally, gracefully, but before it reaches normal completion - Process Instance Terminated

It is also possible that the suspended process on the invoking engine is prematurely terminated before the sub-process reaches completion. In this eventuality it is assumed that control over what happens to the sub-process (i.e. should it be allowed to complete normally; should it be told to terminate prematurely or should it be aborted?) is determined in the definition of the parent process instance.

10.1.3.2Operations

10.1.4Nested Sub-process (Deferred synchronous)

10.1.4.1Intended Use



The nested sub-process (deferred synchronous) conformance profile allows one workflow engine to create a process instance on another according to a known process definition; to instantiate the process instance

and to cause enactment of the instantiated process instance. The invoking workflow engine does not wait for the sub-process to complete before it continues enactment of the parent process, but will rendezvous with the sub-process when it has completed at some later time in order to retrieve consequent values of elements of process relevant data. The invoking workflow engine will receive notification of how the sub-process achieves termination. There are three possibilities:

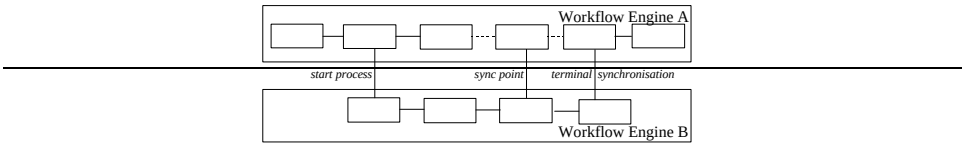
- the sub-process completes normally - Process Instance Completed
- the sub-process is aborted locally - Process Instance Aborted
- the sub-process is terminated locally, gracefully, but before it reaches normal completion - Process Instance Terminated

It is also possible that the suspended process on the invoking engine is prematurely terminated before the sub-process reaches completion. In this eventuality it is assumed that control over what happens to the sub-process (i.e. should it be allowed to complete normally; should it be told to terminate prematurely or should it be aborted?) is determined in the definition of the parent process instance.

10.1.4.2Operations

10.1.5Nested Sub-process (Synchronized enactment)

10.1.5.1Intended Use



The nested sub-process (deferred synchronous) conformance profile allows one workflow engine to create a process instance on another WfMC conformant workflow engine according to a known process definition; to instantiate the process instance; to cause enactment of the instantiated process instance and to synchronize the enactment of the parent process with that of the invoked sub-process. The invoking workflow engine does not wait for the sub-process to complete before it continues enactment of the parent process, but will rendezvous with the sub-process at pre-defined points in their respective enactment and when it has completed at some later time, in order to retrieve consequent values of elements of process relevant data. The invoking workflow engine will receive notification of how the sub-process achieves termination. There are three possibilities:

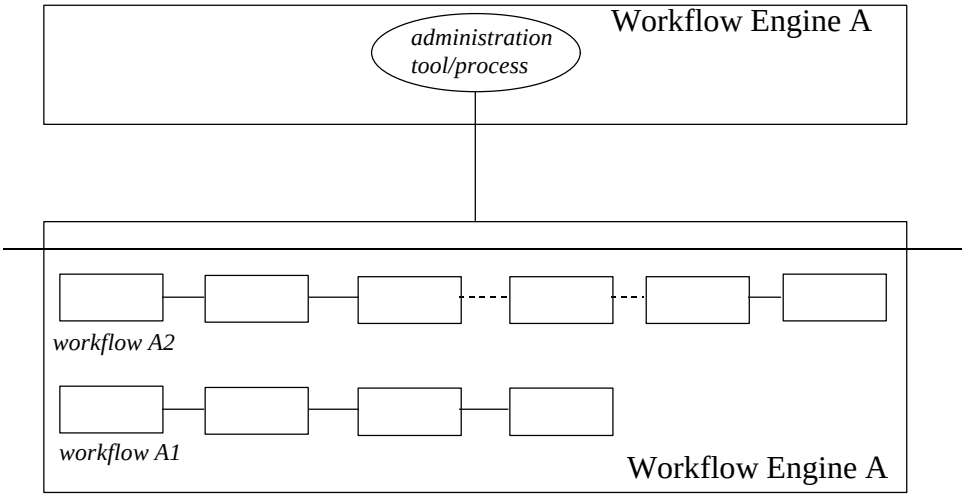
- the sub-process completes normally - Process Instance Completed
- the sub-process is aborted locally - Process Instance Aborted
- the sub-process is terminated locally, gracefully, but before it reaches normal completion - Process Instance Terminated

It is also possible that the suspended process on the invoking engine is prematurely terminated before the sub-process reaches completion. In this eventuality it is assumed that control over what happens to the sub-process (i.e. should it be allowed to complete normally; should it be told to terminate prematurely or should it be aborted?) is determined in the definition of the parent process instance.

10.1.5.2Operations

10.1.6Process Administration

~~*10.1.6.1Intended Use*~~



~~The process administration conformance profile allows workflow administration tools or activities defined within workflows enacted on one workflow engine to conduct the following process administration activities on another WfMC conformant workflow engine:~~

- ~~list process instances currently being enacted on behalf of the querying workflow engine~~
- ~~ascertain the current state of a given process instance being enacted on behalf of the querying workflow engine~~
- ~~start and stop enactment of sub-process instances~~
- ~~monitor the progress of enacted sub-processes~~
- ~~get and set values of process relevant data~~

~~*10.1.6.2Operations*~~

11 Operations

In the following text, the function specifications define operations required to effect interoperability between two (or more) workflow engines. The message specifications define information that must be passed between two workflow engines in order to effect the operations described.

The position of fields within an operation is arbitrary. Therefore, there is no field positionality. The only exception being the name-type-value groups that must appear in the order givenat order.

The position of operations within the message is critical. The receiving workflow engine must execute operations in the order designated in the messageThe operations must be executed by the receiving workflow engine in the order of appearance in the message.

There are situations in which a required operation parameter (e.g. ProcessID) is the result of a previous operation successfully being performed (e.g. CreateProcessInstance). In order to place both operations in a single message, the source workflow engine delegates setting a value for the parameter field to the target workflow engine.

For example, in the following message fragment, ProcessID (required in SetProcessInstanceAttributes and in ChangeProcessInstanceState) is the result of CreateProcessInstance. In this fragment, the source workflow engine did not include the unknown ProcessID field in the message sent, so the target engine is required to use the ProcessID resulting from the CreateProcessInstance.

There are situations in which a required field in an operation is the result of a previous operation. In order to place both operations in a single message, the sending engine must skip the field and the receiving engine must replace its value with the result of the previous operation. The typical case being of ProcessID, which is required in SetProcessInstanceAttributes, but it is the result of a CreateProcessInstance.

For example, in the following message fragment, ProcessID (required in SetProcessInstanceAttributes and in ChangeProcessInstanceState) is the result of CreateProcessInstance. In this fragment, the sending engine did not placed the unknown ProcessID in the message, but the receiving engine will use the ProcessID resulting from the CreateProcessInstance.

```
CreateProcessInstance?OpID=2&ProcessDefinitionID=Open Account&Profile=chain&&
SetProcessInstanceAttributes?OpID=3&Number=1&Name=copies&Type=WMTINT8&Value=3&&
ChangeProcessInstanceState?OpID=4&State=startopen.running&&
```

In this specification, the operation and field names are case insensitive. Most of the values defined in this specification are also case insensitive. The few exceptions to thise case insensitivity rule:

1. Non-terminal tokens enclosed in quotes in the BNF grammar, which must be spelled exactly as defined. Therefore, they are case sensitive.
2. Values supplied by a workflow engine must be considered case sensitive. These are the values supplied in between the "=" and "&" in a field assignment. Therefore, ConversationID, SessionID, etc. are case sensitive.
3. Values defined as part of an interoperability contract must be considered case sensitive. Therefore, ContractID, ProcessDefinitionID, etc. are case sensitive.

1.Non-terminal tokens enclosed in quotes in the BNF grammar must be spelled exactly as defined. Therefore, they are case sensitive.

2.Values supplied by a workflow engine must be considered case sensitive. These are the values supplied in between the "=" and "&" in a field assignment. Therefore, ConversationID, SessionID, etc. are case sensitive.

~~3.Values defined as part of a business contract must be considered case sensitive. Therefore, ContractID, ProcessDefinitionID, etc. are case sensitive.~~

11.1 Change Process Instance State

Abstract Specification Mapping

```
WMAReturnCode WMRequestChangeProcessInstanceState (
                in engine_identifier    WMAEngineID,
                in process_id           WMAObjectID,
                in state                WMAObjectState
                );
WMAReturnCode WMRequestAbortProcessInstance (
                In engine_identifier    WMAEngineID,
                in process_id           WMAObjectID,
                );
WMAReturnCode WMRequestStartProcessInstance (
                in engine_identifier    WMAEngineID,
                in process_id           WMAObjectID
                );
WMAReturnCode WMRequestTerminateProcessInstance (
                in engine_identifier    WMAEngineID,
                in process_id             WMAObjectID
                );
```

Description Change the state of the designated process instance. For example, from suspended to resumed or vice versa.

The receiving workflow engine creates a Process/Sub-process audit record, as specified in [WfMC015], to record that enactment of the suspended process instance was resumed or vice versa.

Rationale It is an operational requirement of users of interoperating workflow systems that it be possible to suspend and resume enactment of sub-process.

Direction This message is always sent from *Source* Engine to *Target* Engine

~~**Process Instances can be changed to the following states:**~~

~~Abort~~	~~Abort enactment of the identified workflow process instance. Sub-processes are not notified.~~
~~Start~~	~~Start the enactment of the identified workflow process instance~~
~~Suspend~~	~~Suspend the process. An start must be issued to reactivate the process again.~~
~~Terminate~~	~~Terminate enactment of the identified workflow process instance. The process should in turn terminate any active sub-process.~~

Request Fields

Field	Comments
ActivityID	The activity on the source workflow engine requesting the change. This field is optional.
OpID	The operation-id of this operation within the message.
SourceRoleID	Role-ID responsible for request. This field is optional.
SourceUserID	User-ID responsible for request. This field is optional.
State	The state to which the designated process instance is to be changed
TargetPID	Process~~-~~-id of process that must be changed. If this operation is the same message than the StartConversation, this field must be omitted and the target workflow engine will use the TargetPID on the StartConversation reply.
TargetRoleID	Role-ID responsible for process instance on target workflow engine.

	This field is optional.
TargetUserID	User-ID responsible for process instance on target workflow engine. This field is optional.

Example of a wfmc-if4-request message:

```
ChangeProcessInstanceState?OpID=2&TargetPID=24&State=~~start~~closed.aborted&&
```

Response Fields

Field	Comments
ActivityID	The ID of the current activity in the target workflow engine at the time the state change occurred. This field is optional.
ErrorCode	
ErrorText	Optional
OpID	The operation-id of this operation within the message.
State	State of the process instance in the target workflow engine. Note that this state maybe different that the requested state. For example, requesting an abort, may result in a termination.
TargetRoleID	Role assumed by primary user. This field is optional.
TargetUserID	User-ID of primary user. This field is optional.

Error Codes

0	Success
1	Operation not implemented
2	Not authorized
3	Unknown ProcessID (TargetPID)
4	Invalid state
255	Operation not performed, because of a failure by an early operation.

Example of a wfmc-if4-response message:

```
ChangeProcessInstanceState?OpID=2&ErrorCode=0&state=~~start~~closed.aborted&&
```

11.2 Create Process Instance

Abstract Specification Mapping

```
WMAReturnCode WMRequestCreateProcessInstance (
                in  engine_identifier        WMAEngineID,
                in  process_definition_id    WMAObjectID,
                in  return_flag              WMABool,
                in  parent_pid               WMAObjectID,
                in  activity_id              WMA_Activity_id,
                out sub_process_id           WMA_Process_instance_id,
                out user_id                  WMAResourceID,
                out role_id                  WMAResourceID
                );
```

Description Identify a process definition that the receiving workflow engine is required to enact.

Rationale A workflow engine must be able to communicate the identity of a process definition to another workflow engine in order for the latter to enact it.

Direction This message is always sent from *Source* Engine to *Target* Engine

Request Fields

Field	Comments
OpID	The operation-id of this operation within the message.
ProcessDefinitionID	Identifier of the process definition that the target workflow engine is required to use to create the process instance
Profile	Indicates the type of dialog to be conducted with the created process.
SourceBDefName	Business Definition Process Name for source process instance. This field is optional.
SourceRoleID	Role-ID responsible for request. This field is optional.
SourceUserID	User-ID responsible for request. This field is optional.
TargetRoleID	Role-ID responsible for process instance on target workflow engine. This field is optional.
TargetUserID	User-ID responsible for process instance on target workflow engine. This field is optional.

Example of a wfmc-if4-request message:

```
CreateProcessInstance?OpID=2&ProcessDefinitionID=Open Account&
Profile=chain&&
```

Response Fields

Field	Comments
ErrorCode	
ErrorText	Optional
OpID	The operation-id of this operation within the message.
TargetProcessID	ID of newly created process instance
State	State of new process instance
TargetBDefName	Business Definition Process Name for target process instance. This

	field is optional.
TargetRoleID	Role assumed by primary user. This field is optional.
TargetUserID	User-ID of primary user. This field is optional.

Error Codes

0	Success
1	Operation not implemented
2	Not authorized
40	Invalid ProcessDefinitionID
41	Invalid or not implemented profile
255	Operation not performed, because of a failure by an early operation.

Example of a wfmc-if4-response message:

```
CreateProcessInstance?OpID=2&ErrorCode=0&
TargetProcessID=32&state=~~NotStart~~open.notRunning.notStarted&&
```

11.3 Get Process Instance Attributes

Abstract Specification Mapping

```
WMAReturnCode WMRequestGetProcessInstanceAttributes (
                in  engine_identifier    WMAEngineID,
                in  process_id           WMAObjectID,
                in  root_pid             WMAObjectID,
                in  activity_id          WMAObjectID,
                out attributes           WMAAttributeList
                );
```

Description Return the value(s) of the requested process instance attributes (process relevant data).

Rationale Checking values of process relevant data is one way in which a workflow engine can check on the progress of a workflow being enacted on another workflow engine.

Direction This message is always sent from *Source* Engine to *Target* Engine

Request Fields

Field	Comments
ActivityID	Identifies the activity on the source workflow engine. This field is optional.
OpID	The operation-id of this operation within the message.
TargetProcessID	The PID of the process instance on the target workflow engine from which attribute values are being requested.
SourceBDefName	Business Definition Process Name for source process instance. This field is optional.
SourceRoleID	Role-ID responsible for request. This field is optional.
SourceUserID	User-ID responsible for request. This field is optional.
TargetRoleID	Role-ID responsible for process instance on target workflow engine. This field is optional.
TargetUserID	User-ID responsible for process instance on target workflow engine. This field is optional.
Number	The number of Attributes for which values are being requested.
Name[1]	Attribute identifier
…	...

Example of a wfmc-if4-request message:

```
GetProcessInstanceAttributes?OpID=2&TargetProcessID=32&
Number=1&Name=copies&&
```

Response Fields

Field	Comments
ErrorCode	
ErrorText	Optional

[1] Information repeated for each attribute value set.

OpID	The operation-id of this operation within the message.
Number	The number of Attributes for which values have been supplied
Name[2]	attribute identifier
Type	new attribute type
Value	new attribute value
...	...

Error Codes

0	Success
1	Operation not implemented
2	Not authorized
3	Unknown ProcessID
5	Invalid or unknown attribute name
255	Operation not performed, because of a failure by an early operation.

Example of a wfmc-if4-response message:

```
GetProcessInstanceAttributes?OpID=2&ErrorCode=0&
Number=1&Name=copies&Type=WMTINT8&Value=3&&
```

[2] Information repeated for each attribute value to be set.

11.4 Get Process Instance State

Abstract Specification Mapping

```
WMAReturnCode WMRequestGetProcessInstanceState  (
                in  engine_identifier      WMAEngineID,
                in  process                WMAObjectID,
                out state                  WMAObjectState
                );
```

Description Get the status of a process instance that another workflow engine is enacting.

Rationale For a long-term sub-process with a life beyond that of the conversation during which it was created, it is important that the invoking workflow engine be able to check that the invoked sub-process is alive and well or that it has completed.

Direction This message is always sent from *Source* Engine to *Target* Engine

Request Fields

Field	Comment
OpID	The operation-id of this operation within the message.
TargetProcessID	PID of process instance on target workflow engine for which state information is requested

Example of a wfmc-if4-request message:

```
GetProcessInstanceState?OpID=2&TargetProcessID=48&&
```

Response Fields

Field	Comment
ErrorCode	
ErrorText	Optional
OpID	The operation-id of this operation within the message.
State	State of process instance

Error Codes

0	Success
1	Operation not implemented
2	Not authorized
3	Unknown ProcessID
255	Operation not performed, because of a failure by an early operation.

Example of a wfmc-if4-response message:

```
GetProcessInstanceState?OpID=2&ErrorCode=0&state=~~start~~open.running&&
```

~~11.5~~List Process Instances

~~Not yet specified~~

~~11.6~~11.5 Process Instance Attributes Changed

Abstract Specification Mapping

```
WMAReturnCode WMNotifyProcessAttributesChanged (
                in engine_identifier      WMAEngineID,
                in process_id             WMAObjectID,
                in attributes             WMAAttributeList
                );
```

Description Notify the source workflow engine of a change in the (sub) process instance attributes (process relevant data).

Rationale This operation is provided so that a workflow engine that is enacting a sub-process can notify the source workflow engine that the value of a set of elements of workflow-relevant data has been changed. This allows tracking of milestones in the management of workflows enacted in a multi-engine contract.

Direction This message is always sent from *Target* Engine to *Source* Engine

Notification Fields

Field	Comment
OpID	The operation-id of this operation within the message.
TargetPID	The ProcessID of the process instance that changed attributes.
SourceAID	The ID of the activity instance within the source workflow engine that is to be notified that the attribute value has changed. This field may be optional.
SourceRoleID	Role-ID responsible for the process instance that caused the notification request. This field is optional.
SourceUserID	User-ID responsible for the process instance that caused the notification request. This field is optional.
TargetRoleID	Role-ID responsible for the process instance on the target workflow engine which caused the newly started process instance to be created. This field is optional.
TargetUserID	User-ID responsible for the process instance on the target workflow engine which caused the newly started process instance to be created. This field is optional.
Number	Number of attributes that have changed.
Name[3]	attribute identifier
Type	attribute type
Value	new attribute value
…	…

Example of a wfmc-if4-request message:

```
ProcessInstanceAttributeChanged?OpID=2&TargetPID=24&
Number=1&Name=copies&Type=WMTINT8&Value=3&&
```

[3] Information repeated for each attribute value to be set.

Response Fields

Field	Comment
ErrorCode	
ErrorText	Optional
OpID	The operation-id of this operation within the message.

Error Codes

0	Success
1	Operation not implemented
2	Not authorized
3	Unknown ProcessID (TargetPID)
5	Invalid or unknown attribute name
7	Invalid attribute type for this attribute
8	Attribute value does not corresponds to attribute type
255	Operation not performed, because of a failure by an early operation.

Example of a wfmc-if4-response message:

```
ProcessInstanceAttributeChanged?OpID=2&ErrorCode=0&&
```

~~11.7~~11.6 Process Instance State Changed

Abstract Specification Mapping

```
WMAReturnCode WMNotifyProcessInstanceStateChange (
                in engine_identifier      WMAEngineID,
                in process_id             WMAObjectID,
                in new_state              WMAObjectState
                );
WMAReturnCode WMNotifyProcessInstanceAborted (
                in engine_identifier      WMAEngineID,
                in process_id             WMAObjectID
                );
WMAReturnCode WMNotifyProcessInstanceCompleted (
                in engine_identifier      WMAEngineID,
                in process_id             WMAObjectID
                );
WMAReturnCode WMNotifyProcessInstanceStarted (
                in engine_identifier    WMAEngineID,
                in process_id           WMAObjectID
                );
WMAReturnCode WMNotifyProcessInstanceTerminated (
                in engine_identifier      WMAEngineID,
                in process_id             WMAObjectID
                );
```

Description Notify the source workflow engine of a state change in the (sub).

Rationale When the invoking process instance hangs while waiting for the enacted sub-process to complete, the workflow engine enacting the sub-process must have a means of communicating the completion to the invoking engine.

Direction This message is normally sent from *Target* Engine to *Source* Engine; however, during some scenarios (see conformance profiles), it may be sent from the *Source* engine to the *Target* engine.

~~**Process Instances can change to the following states:**~~

~~Abort~~ ~~Notify the invoking workflow engine that enactment of the given workflow instance has been aborted locally on the target workflow engine~~

~~Complete~~ ~~Notify the invoking workflow engine that enactment of the given workflow instance has completed normally on the target workflow engine~~

~~Start~~ ~~Notify the invoking workflow engine that the enactment of the given workflow instance has started on the target workflow engine.~~

~~This state is necessary for situations in which the process instance has been created on the host at the behest of another workflow engine, but has not started. It may be that enactment of such a process instance would be started once certain pre-conditions have been satisfied. In such a circumstance, it would be appropriate for the host workflow engine to notify the other workflow engine when enactment of the process instance had started.~~

~~Suspend~~ ~~Notify the invoking workflow engine that enactment of the given workflow instance has been suspended on the target workflow engine~~

~~Terminate~~ ~~Notify the invoking workflow engine that enactment of the given workflow instance has been terminated locally on the target workflow engine.~~

Notification Fields

Field	Comment
ProcessID	The Process-id of the process instance that changed state.

ActivityID	The ID of the activity instance that is to be notified that the attribute value has changed. This field may be optional.
OpID	The operation-id of this operation within the message.
SourceRoleID	Optional
SourceUserID	Optional
State	New state of process instance
TargetRoleID	Optional
TargetUserID	Optional

Example of a wfmc-if4-request message:

```
ProcessInstanceStateChanged?OpID=2&ProcessID=24&state=closed.completed&&
```

Response Fields

Field	Comment
ErrorCode	
ErrorText	Optional
OpID	The operation-id of this operation within the message.

Error Codes

0	Success
1	Operation not implemented
2	Not authorized
3	Unknown ProcessID
4	Invalid state
255	Operation not performed, because of a failure by an early operation.

Example of a wfmc-if4-response message:

```
ProcessInstanceStateChanged?OpID=2&ErrorCode=0&&
```

~~11.8~~11.7 Set Process Instance Attributes

Abstract Specification Mapping

```
WMAReturnCode WMRequestSetProcessInstanceAttributes (
                in  engine_identifier     WMAEngineID,
                in  root_pid              WMAObjectID,
                in  activity_id           WMAObjectID,
                in  process_id            WMAObjectID,
                out attributes            WMAAttributeList
                );
```

Description Set the value(s) of process instance attributes (process relevant data) in a selected process definition.

The attribute list sent to target workflow engine contains value specifications for one or more process instance attributes to be set. The target workflow engine attempts to set attribute values in the order in which they occur in the list. It returns a response message containing a list of those attributes for which the set operation was successful. In the event that part way through enacting a list of attribute values an error occurs, the attribute for which it was unable to set a value is not contained in the response message and the return code value indicates a failure. The target workflow engine does not enact the list of attribute values beyond the point at which a failure occurs.

Rationale Process definitions are only partial and must be fully (or sufficiently) instantiated before enactment may commence.

Direction This message is normally sent from *Source* Engine to *Target* Engine; however, during some scenarios (see conformance profiles), it may be sent from the *Target* engine to the *Source* engine.

Request Fields

Field	Comment
ProcessID	The process~~-~~id of the process instance for which attribute values are being provided. If this operation is the same message than the StartConversation, this field must be omitted and the target workflow engine will use the TargetPID on the StartConversation reply.
OpID	The operation-id of this operation within the message.
Number	The number of Attributes for which value changes are required.
Name[4]	attribute identifier
Type	attribute type (Refer to “Basic Attribute Types” in section 4.2 Data Types & Declaratives)
Value	new attribute value
...	...

Example of a wfmc-if4-request message:

```
SetProcessInstanceAttributes?OpID=2&ProcessID=24&
Number=1&Name=copies&Type=WMTINT8&Value=3&&
```

[4] Information repeated for each attribute value to be set.

Response Fields

Field	Comment
ErrorCode	Overall operation error. If it is different than zero, then the message may contain attribute errors
ErrorText	Optional
OpID	The operation-id of this operation within the message.
ProcessID	The process~~-~~-id of the process instance for which changes to attribute values were requested.
Number	The number of a~~A~~ttributes ~~for which value changes have succeeded (should be equal to the number of attributes for which changes were requested).~~on error, or zero if there is no attribute errors. It indicates the number of Name-AErrorCode pairs present in this operation response.
Name	The name of an attribute that has an error.
AErrorCode~~Name~~[5]	The error code for the attribute name preceding this field. Valid errors are: 5 -- Invalid or unknown attribute name 7 -- Invalid attribute type ~~The name of an attribute that has had its value changed~~ 8 -- Attribute value does not corresponds to attribute type

Note that SetProcessInstanceAttributes has two possible response formats:

- On success and errors that are not caused by attributes, the field ‘number’ is set to zero. In this case, the fields ‘Name’ and ‘AErrorCode’ are not present.
- On errors caused by attributes, the field ‘number’ indicates the number of attributes on error. For each of those attributes, the fields ‘Name’ and ‘AErrorCode’ provide the details. The overall ErrorCode must be different than zero.

Error Codes

0	Success
1	Operation not implemented
2	Not authorized
3	Unknown ProcessID
5	Invalid or unknown attribute name
7	Invalid attribute type for this attribute
8	Attribute value does not corresponds to attribute type
30	Operation failed see AErrorCode for details
255	Operation not performed, because of a failure by an early operation.

Example of a wfmc-if4-response message:

~~[5] Information repeated for each attribute value set.~~

SetProcessInstanceAttributes?OpID=2&ErrorCode=0&Number=0&&

or

SetProcessInstanceAttributes?OpID=2&ErrorCode=30&number=2&~~**SetProcessInstanceAttributes**?ErrorCode=0&~~
~~Number=1&~~Name=copies&AErrorCode=7&name=name&AErrorCode=8&&

~~11.9~~11.8 Start Conversation

Description Connect to the designated workflow engine, starting a new conversation·

Both workflow engines create a conversation management record as defined in [WfMC015] to mark the start of the conversation.

Rationale For dialogues effecting interoperation, it is necessary that a connection be established between the interoperating workflow engines before interoperation can take place.

Direction This message is always sent from *Source* Engine to *Target* Engine

Request Fields

Field	Comment
ActivityID	Identifies the activity that originates the conversation start request. . This field is optional.
ContractID	Contract~~-~~id for set of workflow engines interoperating within the current business agreement
Language	Two character ISO 639-1 code. This field is optional, in which case the target engine decides the language to use.
OpID	The operation-id of this operation within the message.
ProductID	String, that identifies the product making the request, of the form: <product name>/<version number>
RootPID	Process~~-~~id of top root workflow process instance
SourceConversationID	Conversation-id ~~ID~~ allocated by source workflow ~~engine or by the transport mechanism~~
SourceNodeID	Mail address for source workflow engine
SourcePID	Process instance~~-~~id of process that originates the conversation start request
Version	Version of MIME Binding used to construct messages

Example of a wfmc-if4-request message:

```
StartConversation?OpID=1&ContractID=Nice Group&Version=1.1&Language=de&
SourceNodeID=xyz@wfmc.org&RootPID=730&ProductID=MagicWorkflow/5.0&
SourcePID=24&SourceConversationID=123&&
```

Response Fields

Field	Comments
ErrorCode	
ErrorText	Optional
OpID	The operation-id of this operation within the message.
ProductID	String, identifying the product (answering the request), of the form: <product name>/<version number>
SourceConversationID	Conversation-id ~~ID~~ allocated by source workflow engine ~~or by the transport mechanism~~

TargetConversationID	Conversation-id ~~ID~~ allocated by target workflow engine ~~or by the transport mechanism~~
TargetNodeID	Mail address for target workflow engine
Version	Version of MIME Binding used to construct messages

Error Codes

0	Success
10	Invalid or expired ContractID
11	Invalid or unsupported language
12	Invalid or unsuported MIME version

Example of a wfmc-if4-response message:

```
StartConversation?OpID=1&ErrorCode=0&SourceConversationID=123&
TargetConversationID=456&TargetNodeID=abc@wfmc.org&
Version=1.1&ProductID=BetterMagicWorkflow/8.1&&
```

~~11.10~~11.9 Stop Conversation

Abstract Specification Mapping

```
WMAReturnCode WMRelinquishProcessInstance (
                in engine_identifier      WMAEngineID,
                in process_id             WMAObjectID
                );
```

Description Terminate the current conversation, disconnect from the interoperating workflow engine.

Notify another workflow engine that, as far as this workflow engine is concerned, it may now 1) release all memory that contains data structures that pertain to the given conversation and/or 2) not send notification messages concerning that conversation.

This does not terminate processes that started during the conversation, these continue unaffected. Both workflow engines create a conversation management record as defined in [WfMC015] to mark the end of the conversation.

Rationale Provides a way to close a conversation.

Direction This message is always sent from *Source* Engine to *Target* Engine

Request Fields

Field	Comments
OpID	The operation-id of this operation within the message.

Example of a wfmc-if4-request message:

```
StopConversation?OpID=5&&
```

Response Fields

Field	Comments
ErrorCode	
ErrorText	Optional
OpID	The operation-id of this operation within the message.

Error Codes

0	Success
20	There is no conversation in process
255	Operation not performed, because of a failure by an early operation.

Example of a wfmc-if4-response message:

```
StopConversation?OpID=5&ErrorCode=301&ErrorText=I don't like you&&
```

12 References

[Fletcher82]	Fletcher, J.	An Arithmetic Checksum for Serial Transmissions. IEEE Transactions on Communication, Vol. COM-30, No. 1, pp 247-252, January 1982.
[OMG93]	OMG 93.12.43	The Common Object Request Broker: Architecture and Specification (1.2)
[WfMC000]		Workflow Management Coalition Glossary
[WfMC1009]	WFMC TC-1009	Workflow Management Coalition Interface 2 Application Programming Interface (WAPI) Specification, November 1995
[WfMC1012]	WFMC TC - 1012	Workflow Management Coalition Workflow Standard - Interoperability Abstract Specification, October 1996
[WfMC1013]	WFMC-TC-1013	Workflow Application Programmer's (WAPI) Interface 2 Naming Conventions
[WfMC1015]	WFMC-TC-1015	Workflow Management Coalition Interface 5 Audit Data Specification, May 1996
[WfMC1021]	WFMC-TC-1021	Interoperability Proving Framework, February 1998
[WfMC1011]	WFMC-TC-1011	Terminology and Glossary, June 96

13 Appendix A -- Audit Data

The audit data specification below, define presentation of audit data that must be possible following interoperation between workflow engines using this binding. It is not intended that this part of the specification be taken as a literal description of how such data must be stored, rather it is included as an aid to designers who subsequently implement the Coalition's Audit and Management Interfaces.

13.1 Audit Data Types

The concrete realization (WMTEngineID) of the abstract data type WMTAEngineID is a data structure that identifies a particular workflow engine by:

- ID of the workflow contract within which the interoperability is effected
- Mail address[6] from which it receives messages from other workflow engines operating within the given workflow contract
- Source conversation-id ~~ID~~ it has assigned to the interoperability dialogue it conducts with a specified other workflow engine
- Target conversation-id ~~ID~~ assigned by the other workflow engine to the interoperability dialogue it conducts with this workflow engine[7]

```
typedef
{
    WMTResourceID      contract_id;
    WMTResourceID      node_id;
    WMTResourceID      source_conversation;
    WMTResourceID      target_conversation;
} WMTEngineID;
```

The following types are derived from the audit data specification document [WfMC1015]:

```
Typedef
{
    WMTText            object_id[UNIQUE_ID_SIZE];
}WMTObjectID;  //Object is Process, Activity or Workitem


typedef
{
    WMTText            resource_ID[UNIQUE_ID_SIZE];
}WMTResourceID;
//Resource is Person,Role or Network Resource
```

All other types and declaratives used in this binding specification are in the specification of the Workflow Management API [WfMC1009]

13.2 Audit Information

[6] In this binding mail address equates to node id

[7] Note that it is possible for two workflow engines to be conducting multiple interoperability dialogues with each other, in parallel, within a given time frame.

13.2.1 Change Process Instance State

The following audit data records would be created as a result of changing the state of a process instance being enacted on the source workflow engine on behalf of the target workflow engine.

Source Workflow Engine Audit Data -- Request

Name	M/O	Data Type	Description
InitialProcessInstanceID	M	WMTProcInstID	Unique ID of initial (root) process instance
CurrentProcessInstanceID	M	WMTProcInstID	Unique ID of current process instance
ActivityInstanceID	O	WMTActivityInstID	Unique ID of current activity instance
ProcessState	M	WMTProcInstState	state of current process instance
EventCode	M	WMTEventCode	'WMSentRequestChangeProcessInstanceState'
ContractID	M	WMTResourceID	Contract ID for source workflow engine
NodeID	M	WMTResourceID	Node ID for source workflow engine
UserID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
RoleID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	WMTDateTime~~stamp~~	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
Source Activity Definition Business Name	O	WMTObjectName	Business name of current activity on the source engine originating the request to change process state
Target Process Definition ID	O	WMTObjectID	as supplied to target workflow engine
Target Process Instance ID	O	WMTObjectID	Process instance on target workflow engine
Target Process Definition Business Name	O	WMTObjectName	null | as supplied by target workflow engine
TargetNodeID	M	WMTResourceID	Node ID of Workflow Engine accepting change state request.
TargetUserID	O	WMTResourceID	ID of remote user currently performing the process (may be null)
TargetRoleID	O	WMTResourceID	ID of remote role currently performing the process (may be null)
Target State	O	WMTProcInstState	new state of process instance on remote engine
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data -- Request

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on source workflow engine which caused the sub-process to be created on the target workflow engine
Current Process Instance ID	M	WMTProcInstID	Process Instance ID for sub-process created on the target workflow engine
Activity Instance ID	O	WMTActivityInstID	null
Process State	M	WMTProcInstState	new state of process instance
Event Code	M	WMTEventCode	'WMReceivedRequestChangeProcessInstanceState'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	null or as instantiated
Role ID	M	WMTResourceID	null or as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when request to change process state was received
Information ID	M	WMTObjectID	'WfMC'
MessageID	M	WMTObjectID	Message ID associated with event.
SourceInitialProcessInstance ID	M	WMTProcInstID	Initial Process Instance ID of source workflow engine
SourceCurrentProcess InstanceID	M	WMTProcInstID	Current Process Instance ID of source workflow engine
SourceActivityInstanceID	M	WMTActivityInstID	Activity Instance ID of the source WFE
SourceTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp of source WFE event
SourceNodeID	M	WMTResourceID	Node ID of source Workflow Engine
SourceUserID	O	WMTResourceID	User ID associated with the remote workflow engine request
SourceRoleID	O	WMTResourceID	Role ID associated with the remote workflow engine request
SourceProcessDefinition BusinessName	O	WMTText	Business name of remote workflow engine process that initiated the process instance (may be null)
SourceActivityDefinition BusinessName	O	WMTText	Business name of the remote WFE activity spawning the request
SourceProcess Definition ID	M	WMTObjectID	as supplied by source workflow engine
SourceRequestedState		WMTProcInstState	state process instance requested to change to
Process Definition Business ID	O	WMTObjectName	as supplied by source workflow engine - may be null

Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Source Workflow Engine Audit Data - Operation

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	new state of process instance
Event Code	M	WMTEventCode	'WMChangedProcessInstanceState'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceActivityInstanceID	M	WMTObjectID	Activity ID on source workflow engine
RemoteNodeID	M	WMTResourceID	NodeID of target WFE
RemoteProcessInstanceID	M	WMTObjectID	Process instance on target workflow engine
RemoteTimestamp	M	WMAObjectID	Timestamp for when enactment of the process instance state changed on target workflow engine
RemoteProcessDefinitionBusinessName	O	WMTObjectName	null | as supplied by target workflow engine
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data - Operation

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	new state of process instance
Event Code	M	WMTEventCode	'WMChangedProcessInstanceState'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
PreviousProcessState	M	WMTProcInstState	State of process instance prior to change
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Source Workflow Engine Audit Data - Response

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on source workflow engine which caused the sub-process to be created on the target workflow engine
Current Process Instance ID	M	WMTProcInstID	Process Instance ID for sub-process created on t target workflow engine
Activity Instance ID	O	WMTActivityInstID	null
Process State	M	WMTProcInstState	new state of process instance
Event Code	M	WMTEventCode	'WMReceivedChangedProcessInstanceState'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	null or as instantiated
Role ID	M	WMTResourceID	null or as instantiated
Timestamp	M	~~WMTTimestamp~~W	Timestamp for when notification that process

		MTDateTime	state changed was received
Information ID	M	WMTObjectID	'WfMC'
MessageID	M	WMTObjectID	Message ID associated with event.
SourceInitialProcessInstance ID	M	WMTProcInstID	Initial Process Instance ID of source workflow engine
SourceCurrentProcess InstanceID	M	WMTProcInstID	Current Process Instance ID of source workflow engine
SourceActivityInstanceID	M	WMTActivityInstID	Activity Instance ID of the source WFE
SourceTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp of source WFE event
SourceNodeID	M	WMTResourceID	Node ID of source Workflow Engine
SourceUserID	O	WMTResourceID	User ID associated with the remote workflow engine request
SourceRoleID	O	WMTResourceID	Role ID associated with the remote workflow engine request
SourceProcessDefinition BusinessName	O	WMTText	Business name of remote workflow engine process that initiated the process instance (may be null)
SourceActivityDefinition BusinessName	O	WMTText	Business name of the remote WFE activity spawning the request
SourceProcess Definition ID	M	WMTObjectID	as supplied by source workflow engine
SourceRequestedState		WMTProcInstState	state process instance requested to change to
Process Definition Business ID	O	WMTObjectName	as supplied by source workflow engine - may be null
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data - Response

Name	M/O	Data Type	Description
InitialProcessInstanceID	M	WMTProcInstID	Unique ID of initial (root) process instance
CurrentProcessInstanceID	M	WMTProcInstID	Unique ID of current process instance
ActivityInstance ID	O	WMTActivityInstID	Unique ID of current activity instance
ProcessState	M	WMTProcInstState	state of current process instance
EventCode	M	WMTEventCode	'WMSentChangedProcessInstanceState'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would

			consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
Source Activity Definition Business Name	O	WMTObjectName	Business name of current activity on the source engine originating the request to change process state
Target Process Definition ID	O	WMTObjectID	as supplied to target workflow engine
Target Process Instance ID	O	WMTObjectID	Process instance on target workflow engine
Target Process Definition Business Name	O	WMTObjectName	null | as supplied by target workflow engine
TargetNodeID	M	WMTResourceID	Node ID of Workflow Engine accepting change state request.
TargetUserID	O	WMTResourceID	ID of remote user currently performing the process (may be null)
TargetRoleID	O	WMTResourceID	ID of remote role currently performing the process (may be null)
Target State	O	WMTProcInstState	new state of process instance on remote engine
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

13.2.2 Create Process Instance

Source Workflow Engine Audit Data -- Request

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	'open.not-running'
Event Code	M	WMTEventCode	'WMSentStartProcessInstance'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceActivityDefinitionBusinesName	O	WMTObjectName	Business name of current activity on the source engine originating the request to create a new process instance
TargetProcessDefinitionID	M	WMTObjectID	as supplied to target workflow engine
Target Process Instance ID	M	WMTObjectID	Process instance created on target workflow engine
TargetProcessDefinitionBusinessName	O	WMTObjectName	null | as supplied by target workflow engine
TargetNodeID	M	WMTResourceID	Node ID of Workflow Engine accepting the conversation request.
TargetUserID	O	WMTResourceID	ID of remote user requested to perform the process (may be null)
TargetRoleID	O	WMTResourceID	ID of remote role requested to perform the process (may be null)
Target State	O	WMTProcInstState	State of new process instance
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data -- Request

Name	M/O	Data Type	Description
InitialProcessInstanceID	M	WMTProcInstID	Process Instance ID for process instance on source workflow engine which caused the request for the sub-process to be created on the target workflow engine
CurrentProcessInstanceID	M	WMTProcInstID	Process Instance ID for sub-process created on the target workflow engine
ActivityInstanceID	O	WMTActivityInstID	null
ProcessState	M	WMTProcInstState	State of created process instance
EventCode	M	WMTEventCode	'WMReceivedRequestStartProcessInstance'
ContractID	M	WMTResourceID	Contract ID for source workflow engine
NodeID	M	WMTResourceID	Node ID for target workflow engine
UserID	M	WMTResourceID	null or as instantiated
RoleID	M	WMTResourceID	null or as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when request to create process instance was received
InformationID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceInitialProcessInstance ID	M	WMTProcInstID	Initial Process Instance ID of source workflow engine
SourceCurrentProcessInstanceID	M	WMTProcInstID	Current Process Instance ID of source workflow engine
SourceActivityInstanceID	M	WMTActivityInstID	Activity Instance ID of the source WFE
SourceTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp of source WFE event
SourceNodeID	M	WMTResourceID	Node ID of source Workflow Engine
SourceUserID	O	WMTResourceID	User ID associated with the remote workflow engine request
SourceRoleID	O	WMTResourceID	Role ID associated with the remote workflow engine request
SourceProcess Definition ID	M	WMTObjectID	as supplied by source workflow engine
SourceProcessDefinition BusinessName	O	WMTText	Business name of remote workflow engine process that generated the start process request.
SourceActivityDefinition BusinessName	O	WMTText	Business name of the remote WFE activity spawning the request
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Source Workflow Engine Audit Data - Operation

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	'open.not-running'
Event Code	M	WMTEventCode	'WMCreatedProcessInstance'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceActivityInstanceID	M	WMTObjectID	Activity ID on source workflow engine
RemoteNodeID	M	WMTResourceID	NodeID of target WFE
RemoteProcessInstanceID	M	WMTObjectID	Process instance created on target workflow engine
RemoteTimestamp	M	WMAObjectID	Timestamp for when process instance created on target WFE
RemoteProcessDefinitionBusinessName	O	WMTObjectName	null | as supplied by target workflow engine
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data - Operation

Name	M/O	Data Type	Description
InitialProcessInstanceID	M	WMTProcInstID	Process Instance ID for process instance on source workflow engine which caused the request for the sub-process to be created on the target workflow engine
CurrentProcessInstanceID	M	WMTProcInstID	Process Instance ID for sub-process created on th target workflow engine

ActivityInstanceID	O	WMTActivityInstID	null
ProcessState	M	WMTProcInstState	State of created process instance
EventCode	M	WMTEventCode	'WMCreatedProcessInstance'
ContractID	M	WMTResourceID	Contract ID for source workflow engine
NodeID	M	WMTResourceID	Node ID for target workflow engine
UserID	M	WMTResourceID	null or as instantiated
RoleID	M	WMTResourceID	null or as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when request to create process instance was received
InformationID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
ProcessDefinition ID	M	WMTObjectID	Process Definition ID identifying the definition used to create this process instance
ProcessDefinitionBusinessName	O	WMTText	Business name of the process definition
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Source Workflow Engine Audit Data - Response

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	'open.not-running'
Event Code	M	WMTEventCode	'WMReceivedStartedProcessInstance'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceActivityDefinitionBusines	O	WMTObjectName	Business name of current activity on the

Name			source engine originating the request to create a new process instance
TargetProcessDefinitionID	M	WMTObjectID	as supplied to target workflow engine
Target Process Instance ID	M	WMTObjectID	Process instance created on target workflow engine
TargetProcessDefinitionBusinessName	O	WMTObjectName	null | as supplied by target workflow engine
TargetNodeID	M	WMTResourceID	Node ID of Workflow Engine accepting the conversation request.
TargetUserID	O	WMTResourceID	ID of remote user requested to perform the process (may be null)
TargetRoleID	O	WMTResourceID	ID of remote role requested to perform the process (may be null)
Target State	O	WMTProcInstState	State of new process instance
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data - Response

Name	M/O	Data Type	Description
InitialProcessInstanceID	M	WMTProcInstID	Process Instance ID for process instance on source workflow engine which caused the request for the sub-process to be created on the target workflow engine
CurrentProcessInstanceID	M	WMTProcInstID	Process Instance ID for sub-process created on th target workflow engine
ActivityInstanceID	O	WMTActivityInstID	null
ProcessState	M	WMTProcInstState	State of created process instance
EventCode	M	WMTEventCode	'WMSentStartedProcessInstance'
ContractID	M	WMTResourceID	Contract ID for source workflow engine
NodeID	M	WMTResourceID	Node ID for target workflow engine
UserID	M	WMTResourceID	null or as instantiated
RoleID	M	WMTResourceID	null or as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when request to create process instance was received
InformationID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceInitialProcessInstance ID	M	WMTProcInstID	Initial Process Instance ID of source workflow engine
SourceCurrentProcessInstanceID	M	WMTProcInstID	Current Process Instance ID of source

			workflow engine
SourceActivityInstanceID	M	WMTActivityInstID	Activity Instance ID of the source WFE
SourceTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp of source WFE event
SourceNodeID	M	WMTResourceID	Node ID of source Workflow Engine
SourceUserID	O	WMTResourceID	User ID associated with the remote workflow engine request
SourceRoleID	O	WMTResourceID	Role ID associated with the remote workflow engine request
SourceProcess Definition ID	M	WMTObjectID	as supplied by source workflow engine
SourceProcessDefinition BusinessName	O	WMTText	Business name of remote workflow engine process that generated the start process request.
SourceActivityDefinition BusinessName	O	WMTText	Business name of the remote WFE activity spawning the request
Extension Number	M	WMTInt16	‘1’
Extension Type	M	WMTText	‘4MIME’
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

13.2.3 Get Process Instance Attributes

The following audit data records would be created as a result of the target workflow engine successfully providing each requested attribute value at the behest of the source workflow engine.

Source Workflow Engine Audit Data -- Request

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance (may be null)
Process State	M	WMTProcInstState	State of process instance on source workflow engine
Event Code	M	WMTEventCode	'WMSentRequestGetProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
Source Activity Definition Business Name	O	WMTObjectName	Business name of current activity on the source engine (may be null)
Target Process Instance ID	M	WMTObjectID	Process instance on target workflow engine
Target Process Definition Business Name	O	WMTObjectName	null | as declared in process definition
TargetNodeID	M	WMTResourceID	Node ID of Workflow Engine accepting request for attribute value(s).
TargetUserID	O	WMTResourceID	ID of remote user requested to provide attribute value (may be null)
TargetRoleID	O	WMTResourceID	ID of remote role requested to provide attribute value (may be null)
Attribute Name	O	WMTAttributeName	Name of attribute requested
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data -- Request

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on source workflow engine which caused the request for the sub-process to be created on the target workflow engine
Current Process Instance ID	M	WMTProcInstID	Process Instance ID for sub-process created on the target workflow engine
Activity Instance ID	O	WMTActivityInstID	null
Process State	M	WMTProcInstState	State of process instance
Event Code	M	WMTEventCode	'WMReceivedRequestGetProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	null or as instantiated
Role ID	M	WMTResourceID	null or as instantiated
Current Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when attribute value was sent
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceInitialProcessInstance ID	M	WMTProcInstID	Initial Process Instance ID of source workflow engine
SourceCurrentProcess InstanceID	M	WMTProcInstID	Current Process Instance ID of source workflow engine
SourceActivityInstanceID	M	WMTActivityInstID	Activity Instance ID of the source WFE
SourceTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp of source WFE event
SourceNodeID	M	WMTResourceID	Node ID of source Workflow Engine
SourceUserID	O	WMTResourceID	User ID associated with the remote workflow engine request
SourceRoleID	O	WMTResourceID	Role ID associated with the remote workflow engine request
SourceProcessDefinition BusinessName	O	WMTText	Business name of remote workflow engine process that generated the start process request.
SourceActivityDefinition BusinessName	O	WMTText	Business name of the remote WFE activity spawning the request
Attribute Name	O	WMTAttributeName	Name of attribute requested
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the

			transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Source Workflow Engine Audit Data - Response

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on source workflow engine which caused the request for the sub-process to be created on the target workflow engine
Current Process Instance ID	M	WMTProcInstID	Process Instance ID for sub-process created on the target workflow engine
Activity Instance ID	O	WMTActivityInstID	null
Process State	M	WMTProcInstState	State of process instance
Event Code	M	WMTEventCode	'WMReceivedRetrievedProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	null or as instantiated
Role ID	M	WMTResourceID	null or as instantiated
Current Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when attribute value was sent
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceInitialProcessInstance ID	M	WMTProcInstID	Initial Process Instance ID of source workflow engine
SourceCurrentProcess InstanceID	M	WMTProcInstID	Current Process Instance ID of source workflow engine
SourceActivityInstanceID	M	WMTActivityInstID	Activity Instance ID of the source WFE
SourceTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp of source WFE event
SourceNodeID	M	WMTResourceID	Node ID of source Workflow Engine
SourceUserID	O	WMTResourceID	User ID associated with the remote workflow engine request
SourceRoleID	O	WMTResourceID	Role ID associated with the remote workflow engine request
SourceProcessDefinition BusinessName	O	WMTText	Business name of remote workflow engine process that generated the start process request.
SourceActivityDefinition BusinessName	O	WMTText	Business name of the remote WFE activity spawning the request
Attribute Name	O	WMTAttributeName	Name of attribute requested
Attribute Type	O	WMTAttributeType	Type of attribute requested

Attribute Length	O	WMTAttributeLength	Length of requested attribute
Attribute Value	O	WMTAttributeValue	Value
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data - Response

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance (may be null)
Process State	M	WMTProcInstState	State of process instance on source workflow engine
Event Code	M	WMTEventCode	'WMSentRetrievedProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
Source Activity Definition Business Name	O	WMTObjectName	Business name of current activity on the source engine (may be null)
Target Process Instance ID	M	WMTObjectID	Process instance on target workflow engine
Target Process Definition Business Name	O	WMTObjectName	null | as declared in process definition
TargetNodeID	M	WMTResourceID	Node ID of Workflow Engine accepting request for attribute value(s).
TargetUserID	O	WMTResourceID	ID of remote user requested to provide attribute value (may be null)
TargetRoleID	O	WMTResourceID	ID of remote role requested to provide attribute value (may be null)
Attribute Name	O	WMTAttributeName	Name of attribute requested
Attribute Type	O	WMTAttributeType	Type of attribute requested

Attribute Length	O	WMTAttributeLength	Length of requested attribute
Attribute Value	O	WMTAttributeValue	Value
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

13.2.4 Get Process Instance State

Audid data is not specified for this operation.

~~13.2.5List Process Instances~~

~~Audid data is not specified for this operation.~~

~~13.2.6~~13.2.5 Process Instance Attribute Changed

The following audit data records would be created as a result of the source workflow engine having changed the value of a notifiable attribute.

Notifying Workflow Engine Audit Data - Actual Event

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on Source workflow engine which caused the request for the sub-process to be created on the target workflow engine
Current Process Instance ID	M	WMTProcInstID	Process Instance ID for sub-process created on the target workflow engine
Activity Instance ID	O	WMTActivityInstID	null
Process State	M	WMTProcInstState	State of created process instance
Event Code	M	WMTEventCode	'WMAssignedProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	null or as instantiated
Role ID	M	WMTResourceID	null or as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when request to create process instance was received
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
Changed Attribute Name	M	WMTAttributeName	Name of attribute changed
Changed Attribute Type	M	WMTAttributeType	Type of attribute changed
Changed Attribute Length	M	WMTAttributeLength	Changed length of attribute
Changed Attribute Value	M	WMTAttributeValue	New value
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Notifying Workflow Engine Audit Data – Notification

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	state of process instance when event occurred

EventCode	M	WMTEventCode	'WMSentChangedProcessInstanceAttribute'
ContractID	M	WMTResourceID	Contract ID for source workflow engine
NodeID	M	WMTResourceID	Node ID for source workflow engine
UserID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
RoleID	M	WMTResourceID	null | as instantiated
CurrentTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
InformationID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
Source Activity Definition Business Name	O	WMTObjectName	Business name of current activity on the source engine when attribute value changed
Remote Process Instance ID	M	WMTObjectID	Process instance created on target workflow engine
Remote Activity Instance ID	M	WMTActivityInstID	Remote workflow engine activity instance ID
RemoteNodeID	M	WMTResourceID	Node ID of Workflow Engine receiving the notification.
RemoteTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Changed Attribute Name	O	WMTAttributeName	Name of attribute changed
Changed Attribute Type	O	WMTAttributeType	Type of attribute changed
Changed Attribute Length	O	WMTAttributeLength	Changed length of attribute
Changed Attribute Value	O	WMTAttributeValue	New value
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instanceRemoteAID=2
Process State	M	WMTProcInstState	state of current process instance
Event Code	M	WMTEventCode	'WMReceivedChangedProcessInstanceAtribute'

Contract ID	M	WMTResourceID	Contract ID for target workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as instantiated
Current Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when notification was received
Information ID	M	WMTObjectID	'WfMC'
MessageID	M	WMTObjectID	Message ID associated with event.
Source Activity Definition Business Name	O	WMTObjectName	Business name of current activity on the source engine when the process attribute value changed
Remote Process Instance ID	M	WMTObjectID	Process instance ID on source workflow engine
Remote Activity Instance ID	M	WMTActivityInstID	Remote workflow engine activity instance ID
RemoteNodeID	M	WMTResourceID	Node ID of Workflow Engine sending the notification.
RemoteTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded on remote engine
Changed Attribute Name	O	WMTAttributeName	Name of attribute changed
Changed Attribute Type	O	WMTAttributeType	Type of attribute changed
Changed Attribute Length	O	WMTAttributeLength	Changed length of attribute
Changed Attribute Value	O	WMTAttributeValue	New value
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

~~13.2.7~~13.2.6 Process Instance State Changed

The following audit data records would be created as a result of the completion of a process instance being enacted on the source workflow engine on behalf of the target workflow engine.

Notifying Workflow Engine Audit Data - Actual Event

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	new state of process instance
Event Code	M	WMTEventCode	'WMChangedProcessInstanceState'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
PreviousProcessState	M	WMTProcInstState	State of process instance prior to change
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Notifying Workflow Engine Audit Data – Notification

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	new state of process instance
Event Code	M	WMTEventCode	'WMSentChangedProcessInstanceState'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with

			this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as instantiated
Current Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	M	WMTObjectID	Message ID associated with event.
Source Activity Definition Business Name	O	WMTObjectName	Business name of first activity on the source engine
Remote Process Instance ID	M	WMTObjectID	Process instance created on target workflow engine
Remote Activity Instance ID	M	WMTActivityInstID	Remote workflow engine activity instance ID
RemoteNodeID	M	WMTResourceID	Node ID of Workflow Engine receiving the notification.
RemoteTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instanceRemoteAID=2
Process State	M	WMTProcInstState	state of current process instance
Event Code	M	WMTEventCode	'WMReceivedChangedProcessInstanceState'
Contract ID	M	WMTResourceID	Contract ID for target workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as instantiated
Current Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when notification was received
Information ID	M	WMTObjectID	'WfMC'
MessageID	M	WMTObjectID	Message ID associated with event.
Source Activity Definition	O	WMTObjectName	Business name of current activity on the

Business Name			source engine when the process instance state change occurred
Remote Process Instance ID	M	WMTObjectID	Process instance for which state change occurred on source workflow engine
Remote Activity Instance ID	M	WMTActivityInstID	Remote workflow engine activity instance ID
RemoteNodeID	M	WMTResourceID	Node ID of Workflow Engine sending the notification.
RemoteTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded on remote engine
Extension Number	M	WMTInt16	‘1’
Extension Type	M	WMTText	‘4MIME’
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

~~13.2.8~~13.2.7 Set Process Instance Attributes

The following audit data records would be created as a result of the target workflow engine successfully changing each attribute value at the behest of the source workflow engine.

Source Workflow Engine Audit Data -- Request

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	State of process instance on source workflow engine
Event Code	M	WMTEventCode	'WMSentRequestChangeProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceActivityDefinitionBusiness Name	O	WMTObjectName	Business name of current activity on the source engine originating the request to create a new process instance
TargetProcessInstanceID	M	WMTObjectID	Process instance created on target workflow engine
TargetProcessDefinitionBusiness Name	O	WMTObjectName	null | as supplied by target workflow engine
TargetNodeID	M	WMTResourceID	Node ID of Workflow Engine accepting the conversation request.
TargetUserID	O	WMTResourceID	ID of remote user requested to perform the process (may be null)
TargetRoleID	O	WMTResourceID	ID of remote role requested to perform the process (may be null)
Target State	O	WMTProcInstState	State of new process instance
Attribute Name	O	WMTAttributeName	Name of attribute changed
Attribute Type	O	WMTAttributeType	Type of attribute changed
Attribute Length	O	WMTAttributeLength	Changed length of attribute
Attribute Value	O	WMTAttributeValue	New value
Extension Number	M	WMTInt16	'1'

Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data -- Request

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on Source workflow engine which caused the request for the sub-process to be created on the target workflow engine
Current Process Instance ID	M	WMTProcInstID	Process Instance ID for sub-process created on the target workflow engine
Activity Instance ID	O	WMTActivityInstID	null
Process State	M	WMTProcInstState	State of created process instance
Event Code	M	WMTEventCode	'WMReceivedRequestChangeProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	null or as instantiated
Role ID	M	WMTResourceID	null or as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when request to create process instance was received
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceInitialProcessInstance ID	M	WMTProcInstID	Initial Process Instance ID of source workflow engine
SourceCurrentProcess InstanceID	M	WMTProcInstID	Current Process Instance ID of source workflow engine
SourceActivityInstanceID	M	WMTActivityInstID	Activity Instance ID of the source WFE
SourceTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp of source WFE event
SourceNodeID	M	WMTResourceID	Node ID of source Workflow Engine
SourceUserID	O	WMTResourceID	User ID associated with the remote workflow engine request
SourceRoleID	O	WMTResourceID	Role ID associated with the remote workflow engine request
SourceProcessDefinition BusinessName	O	WMTText	Business name of remote workflow engine process that generated the start process request.
SourceActivityDefinition BusinessName	O	WMTText	Business name of the remote WFE activity spawning the request

Attribute Name	O	WMTAttributeName	Name of attribute changed
Attribute Type	O	WMTAttributeType	Type of attribute changed
Attribute Length	O	WMTAttributeLength	Changed length of attribute
Attribute Value	O	WMTAttributeValue	New value
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Source Workflow Engine Audit Data - Operation

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	State of process instance on source workflow engine
Event Code	M	WMTEventCode	'WMAssignedProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceActivityInstanceID	M	WMTObjectID	Activity ID on source workflow engine
RemoteNodeID	M	WMTResourceID	NodeID of target WFE
RemoteProcessInstanceID	M	WMTObjectID	Process instance on target workflow engine
RemoteTimestamp	M	WMAObjectID	Timestamp for when attribute value changed on remote workflow engine
RemoteProcessDefinitionBusinessName	O	WMTObjectName	null | as supplied by target workflow engine
Attribute Name	O	WMTAttributeName	Name of attribute changed
Attribute Type	O	WMTAttributeType	Type of attribute changed
Attribute Length	O	WMTAttributeLength	Changed length of attribute
Attribute Value	O	WMTAttributeValue	New value

Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data - Operation

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on Source workflow engine which caused the request for the sub-process to be created on the target workflow engine
Current Process Instance ID	M	WMTProcInstID	Process Instance ID for sub-process created on the target workflow engine
Activity Instance ID	O	WMTActivityInstID	null
Process State	M	WMTProcInstState	State of created process instance
Event Code	M	WMTEventCode	'WMAssignedProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	null or as instantiated
Role ID	M	WMTResourceID	null or as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when request to create process instance was received
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
Changed Attribute Name	M	WMTAttributeName	Name of attribute changed
Changed Attribute Type	M	WMTAttributeType	Type of attribute changed
Changed Attribute Length	M	WMTAttributeLength	Changed length of attribute
Changed Attribute Value	M	WMTAttributeValue	New value
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Source Workflow Engine Audit Data - Response

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on Source workflow engine which caused the request

			for the sub-process to be created on the target workflow engine
Current Process Instance ID	M	WMTProcInstID	Process Instance ID for sub-process created on the target workflow engine
Activity Instance ID	O	WMTActivityInstID	null
Process State	M	WMTProcInstState	State of created process instance
Event Code	M	WMTEventCode	'WMReceivedChangedProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine
User ID	M	WMTResourceID	null or as instantiated
Role ID	M	WMTResourceID	null or as instantiated
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp for when request to create process instance was received
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceInitialProcessInstance ID	M	WMTProcInstID	Initial Process Instance ID of source workflow engine
SourceCurrentProcess InstanceID	M	WMTProcInstID	Current Process Instance ID of source workflow engine
SourceActivityInstanceID	M	WMTActivityInstID	Activity Instance ID of the source WFE
SourceTimestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp of source WFE event
SourceNodeID	M	WMTResourceID	Node ID of source Workflow Engine
SourceUserID	O	WMTResourceID	User ID associated with the remote workflow engine request
SourceRoleID	O	WMTResourceID	Role ID associated with the remote workflow engine request
SourceProcessDefinition BusinessName	O	WMTText	Business name of remote workflow engine process that generated the start process request.
SourceActivityDefinition BusinessName	O	WMTText	Business name of the remote WFE activity spawning the request
Attribute Name	O	WMTAttributeName	Name of attribute changed
Attribute Type	O	WMTAttributeType	Type of attribute changed
Attribute Length	O	WMTAttributeLength	Changed length of attribute
Attribute Value	O	WMTAttributeValue	New value
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data - Response

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity Instance ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	State of process instance on source workflow engine
Event Code	M	WMTEventCode	'WMSentChangedProcessInstanceAttribute'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | as supplied by target workflow engine
Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
Information ID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
SourceActivityDefinitionBusiness Name	O	WMTObjectName	Business name of current activity on the source engine originating the request to create a new process instance
TargetProcessInstanceID	M	WMTObjectID	Process instance created on target workflow engine
TargetProcessDefinitionBusiness Name	O	WMTObjectName	null | as supplied by target workflow engine
TargetNodeID	M	WMTResourceID	Node ID of Workflow Engine accepting the conversation request.
TargetUserID	O	WMTResourceID	ID of remote user requested to perform the process (may be null)
TargetRoleID	O	WMTResourceID	ID of remote role requested to perform the process (may be null)
Attribute Name	O	WMTAttributeName	Name of attribute changed
Attribute Type	O	WMTAttributeType	Type of attribute changed
Attribute Length	O	WMTAttributeLength	Changed length of attribute
Attribute Value	O	WMTAttributeValue	New value
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

~~13.2.9~~13.2.8 Start Conversation

Source Engine Audit Data

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	The state of the current process instance
Event Code	M	WMTEventCode	'WMStartedConversation'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity. May be null.
Role ID	M	WMTResourceID	null | the role of the user involved in this event
Current Timestamp	M	~~WMTTimestamp~~WMTDateTime	Time when event was recorded
InformationID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
CorrespondentContextID	M	WMAResourceID	ContractID of Workflow Engine accepting the conversation request.
CorrespondentNodeID	M	WMAResourceID	Node ID of Workflow Engine accepting the conversation request.
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTResourceID	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTResourceID	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on source workflow engine which caused the request for the conversation to be started
Current Process Instance ID	M	WMTProcInstID	null
Activity ID	O	WMTActivityInstID	null
Process State	M	WMTProcInstState	null
Event Code	M	WMTEventCode	'WMStartedConversation'
Contract ID	M	WMTResourceID	Contract ID for target workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine

User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity. May be null.
Role ID	M	WMTResourceID	null
Current Timestamp	M	~~WMTTimestamp~~WMTDateTime	Time event was recorded
InformationID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
CorrespondentContextID	M	WMAResourceID	ContractID of source workflow engine source conversation start.
CorrespondentNodeID	M	WMAResourceID	Node ID of source workflow engine source the conversation start.
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

~~13.2.10~~13.2.9 Stop Conversation

Source Workflow Engine Audit Data

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Unique ID of initial (root) process instance
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	The state of the current process instance
Event Code	M	WMTEventCode	'WMStoppedConversation'
Contract ID	M	WMTResourceID	Contract ID for source workflow engine
Node ID	M	WMTResourceID	Node ID for source workflow engine
User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null | the role of the user involved in this event
Current Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp provided by target workflow engine
InformationID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
CorrespondentContextID	M	WMAResourceID	ContractID of Workflow Engine accepting the conversation request.
CorrespondentNodeID	M	WMAResourceID	Node ID of Workflow Engine accepting the conversation request.
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTINT8	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTINT8	As supplied by target workflow engine or by the transport mechanism

Target Workflow Engine Audit Data

Name	M/O	Data Type	Description
Initial Process Instance ID	M	WMTProcInstID	Process Instance ID for process instance on source workflow engine which caused the request for the conversation to be started
Current Process Instance ID	M	WMTProcInstID	Unique ID of current process instance
Activity ID	O	WMTActivityInstID	Unique ID of current activity instance
Process State	M	WMTProcInstState	The state of the current process instance
Event Code	M	WMTEventCode	'WMStoppedConversation'
Contract ID	M	WMTResourceID	Contract ID for target workflow engine
Node ID	M	WMTResourceID	Node ID for target workflow engine

User ID	M	WMTResourceID	ID of user whom the business would consider the primary person involved with this event. This could be a person or entity.May be null.
Role ID	M	WMTResourceID	null
Current Timestamp	M	~~WMTTimestamp~~WMTDateTime	Timestamp event was recorded
InformationID	M	WMTObjectID	'WfMC'
MessageID	O	WMTObjectID	Message ID associated with event.
CorrespondentContextID	M	WMAResourceID	ContractID of Workflow Engine source the conversation stop.
CorrespondentNodeID	M	WMAResourceID	Node ID of Workflow Engine source the conversation stop.
Extension Number	M	WMTInt16	'1'
Extension Type	M	WMTText	'4MIME'
SourceConversation ID	M	WMTResourceID	As supplied by source workflow engine or by the transport mechanism
TargetConversation ID	M	WMTResourceID	As supplied by target workflow engine or by the transport mechanism

14 Appendix B – Character Set

The following table shows the character set used in this specification. It is a subset of US-ASCII (ANSI X3.4-1986).

char	dec	hex	Description	char	dec	hex	Description
( )	32	20	Space	(P)	80	50	Capital letter P
(!)	33	21	Exclamation mark	(Q)	81	51	Capital letter Q
(")	34	22	Quotation mark	(R)	82	52	Capital letter R
(#)	35	23	Pound sign	(S)	83	53	Capital letter S
($)	36	24	Dollar sign	(T)	84	54	Capital letter T
(%)	37	25	Percent sign	(U)	85	55	Capital letter U
(&)	38	26	Ampersand	(V)	86	56	Capital letter V
(')	39	27	Apostrophe	(W)	87	57	Capital letter W
(()	40	28	Left parenthesis	(X)	88	58	Capital letter X
())	41	29	Right parenthesis	(Y)	89	59	Capital letter Y
(*)	42	2A	Asterisk	(Z)	90	5A	Capital letter Z
(+)	43	2B	Plus sign	([)	91	5B	Left square bracket
(,)	44	2C	Comma	(\)	92	5C	Reverse solidus, backslash
(-)	45	2D	Hyphen, minus sign	(])	93	5D	Right square bracket
(.)	46	2E	Period, full stop	(^)	94	5E	Circumflex accent
(/)	47	2F	Solidus, slash	(_)	95	5F	Low line, underline
(0)	48	30	Digit zero	(`)	96	60	Grave accent
(1)	49	31	Digit one	(a)	97	61	Small letter a
(2)	50	32	Digit two	(b)	98	62	Small letter b
(3)	51	33	Digit three	(c)	99	63	Small letter c
(4)	52	34	Digit four	(d)	100	64	Small letter d
(5)	53	35	Digit five	(e)	101	65	Small letter e
(6)	54	36	Digit six	(f)	102	66	Small letter f
(7)	55	37	Digit seven	(g)	103	67	Small letter g
(8)	56	38	Digit eight	(h)	104	68	Small letter h
(9)	57	39	Digit nine	(i)	105	69	Small letter i
(:)	58	3A	Colon	(j)	106	6A	Small letter j
(;)	59	3B	Semicolon	(k)	107	6B	Small letter k
(<)	60	3C	Less-than sign, left angle bracket	(l)	108	6C	Small letter l
(=)	61	3D	Equals sign	(m)	109	6D	Small letter m
(>)	62	3E	Greater-than sign, right angle bracket	(n)	110	6E	Small letter n
(?)	63	3F	Question mark	(o)	111	6F	Small letter o
(@)	64	40	Commercial at sign	(p)	112	70	Small letter p
(A)	65	41	Capital letter A	(q)	113	71	Small letter q
(B)	66	42	Capital letter B	(r)	114	72	Small letter r
(C)	67	43	Capital letter C	(s)	115	73	Small letter s
(D)	68	44	Capital letter D	(t)	116	74	Small letter t
(E)	69	45	Capital letter E	(u)	117	75	Small letter u
(F)	70	46	Capital letter F	(v)	118	76	Small letter v
(G)	71	47	Capital letter G	(w)	119	77	Small letter w
(H)	72	48	Capital letter H	(x)	120	78	Small letter x
(I)	73	49	Capital letter I	(y)	121	79	Small letter y
(J)	74	4A	Capital letter J	(z)	122	7A	Small letter z
(K)	75	4B	Capital letter K	({)	123	7B	Left curly bracket, left brace
(L)	76	4C	Capital letter L	(|)	124	7C	Vertical line, vertical bar
(M)	77	4D	Capital letter M	(})	125	7D	Right curly bracket, right brace
(N)	78	4E	Capital letter N	(~)	126	7E	Tilde
(O)	79	4F	Capital letter O				

15 Appendix C – Implementation Hints (Non-Normative)

This appendix presents a collection of useful hints and ideas to implement the specification. This appendix is non-normative, and the hints and ideas presented in here are not required to claim compliance with the specification; however, they may help implementers to architect their implementations.

15.1 Contracts

Interoperability contracts denote trading agreements across service boundaries, which operate between workflow applications, enacted on different workflow engines. These cooperating workflow engines may be within the same organization or within separate organizations, that collectively operates a value chain. An interoperability contract governing workflow engine interoperability across a service boundary will describe elements from the list presented below according to the nature and requirements of the business process being supported across the service boundary and the requirements of the organizations operating that process.

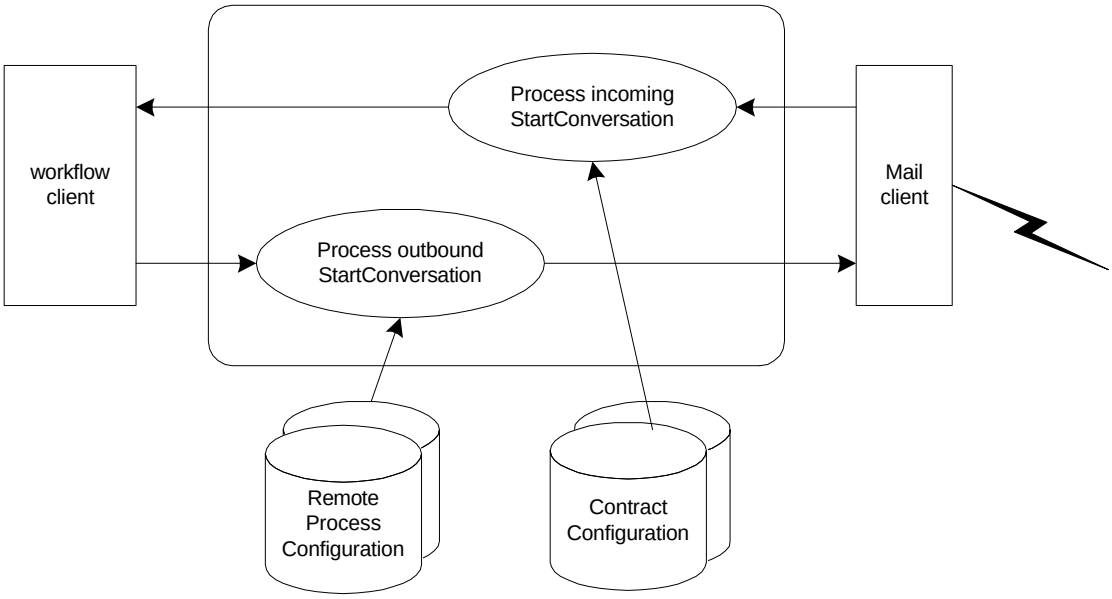
1. Which workflow engines within one service domain are visible to/capable of interoperating with workflow engines in the other service domain
2. Which workflow definitions can be enacted within one service domain at the behest of workflow engines in the other service domain
3. The transport binding supported (e.g. MIME)
4. for each workflow definition identified in the contract:
    - The conformance profile required to effect interoperability
    - Enactment requirements. For example, the ProcessDefinitionID, and attributes required to enact the process.
    - For each traded (shared) element of workflow relevant data
        - Access rights (readable/writable)
        - Value constraints (minimum/maximum values, number of permitted updates/accesses)
    - Outcomes/outputs/returned elements of workflow relevant data
    - Audit data policy
    - Change control policy
5. Security policy and implementation
    - Authentication
    - Support for policy on non-repudiation
    - Shared key cryptography & key management
    - Handling security breaches
6. Exception handling/recovery protocols & transactional behavior
7. The handling of lost messages
    - The length of time a workflow engine might wait for a missing message to turn up
    - The number of resend attempts before giving up
    - What to do in case it gives up

8. Service level agreements, metrication/escalation and performance penalties

Individual interoperability contracts will have a unique ContractID identifier, determined by the organizations trading across the service boundary, which is used to support authentication mechanisms.

15.2 Process and Contract Information

To effect interoperability under an agreed contract, it is necessary for an administrator to configure his or her workflow engine. This configuration includes information on incoming requests as well as information required for outbound requests. This configuration information may be maintained in a database, and configuration tools maybe provided. For sake of simplicity, simple configuration files are presented in here.



15.2.1 Incoming Requests

An incoming request start with a StartConversation. The processing of StartConversation includes a validation against the contract information agreed in the business agreement. This information can be described in a contract-configuration file. There is one contract-configuration file for each contract that can be used from external workflow engines. It contains information on which remotes engines have access to this system, and what processes can they enact under that contract.

This is an example of this configuration file:

```
; Example contract configuration
; There should be one of these for each contract that can be used
; by external workflow engines
ContractName=My Friends

; Protocol being used. The only valid value is I4MIME
; This field is optional, if omitted I4MIME is used.
Protocol=I4MIME

; list of workflow engines that are allow to use this contract.
; An "*" indicates any email address is valid
ValidAddress=abc@xyz.com
ValidAddress=friend@other.com
ValidAddress=qwerty@another.com

; List of valid processes that can be enacted
; by an external workflow engine using this contract
```

```
ValidProcessDefinitionID=Credit validation
ValidProcessDefinitionID=Expense Report
ValidProcessDefinitionID=PinkSlipProcessing

; Message timer. Used for retransmissions and
; to ignore expired messages
Timer=241:300:00

; Attachment support
; 0- attachment are not supported
; 1- part-number referencing
; 2- filename referencing
Attachments=1

; Specify the number of lost message retries
Reties=5


; Other configuration information
; ...
```

15.2.2 Outbound Requests

Outbound requests also start with the creation of a StartConversation. To create a StartConversation it is necessary to have information on how to contact the remote workflow engine and what business agreement to use. This information can be described in a process-configuration file. There is one process-configuration file for each process that can be remotely invoked. It contains information on how to contact the remote workflow engine and what contract to use.

This is an example of this configuration file:

```
; Example process configuration
; There should be one of these for each process that can
; be remotely invoked.

; Protocol being used. The only valid value is I4MIME
; This field is optional, if omitted I4MIME is used.
Protocol=I4MIME

; email address of the remote workflow engine
Address=abc@xyz.com

; Message timer. Used for retransmissions and
; to ignore expired messages
Timer=241:030:00

; Name of the remote contract that accepts messages from
; this workflow engine.
Contract=MyRemoteContract

; Name of the remote Process to be enacted
ProcessDefinitionID=MyRemoteWorkClass

; Attachment support
; 0- attachment are not supported
; 1- part-number referencing
; 2- filename referencing
Attachments=1

; Specify the number of lost message retries
Reties=5


; Other configuration information
; ...
```

15.3 Mailbox Setup~~UNIX sendmail~~

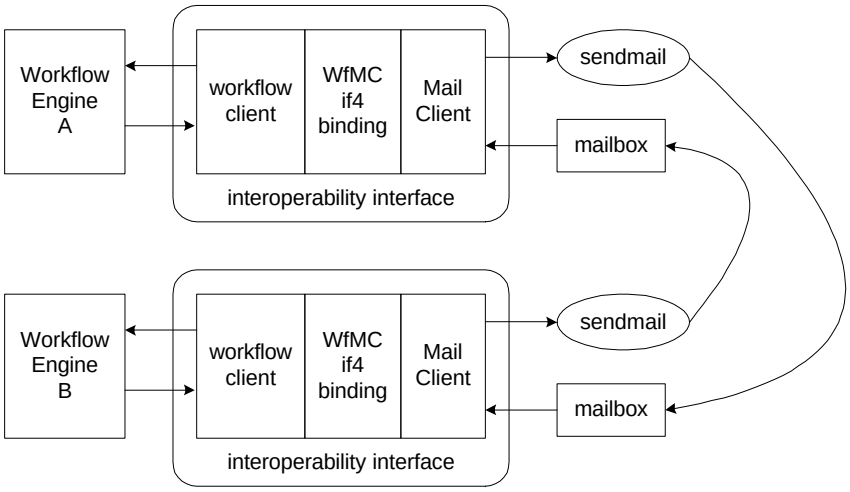
Using a standard utility such as UNIX `sendmail`[8], a user can be set up to receive requests and responses from other workflow engines. By creating a "`.forward`" file in the user's home directory, a developer may specify that an application receive messages, piped into its standard input for processing.

An example of the contents of a "`.forward`" file is:

```
\username,"|/export/home/wfmc/interop"
```

This would automatically invoke the application `/export/home/wfmc/interop` when a mail message is received. The application can interpret requests and responses and make appropriate calls on the vendor's workflow engine.

To send a message to another workflow engine, the message can be written to a file and `sendmail` can be invoked using a system call to dispatch it.



Architecture to effect mail based interoperability between two workflow engines

In this scheme, each workflow engine is uniquely identified by the mailbox it listens to (i.e. its mail address).

8 Although this example of an implementation strategy is based on tools and directory structures available in the UNIX environment, similar strategies are feasible for other

