

1 **5.5 Designating Optional Behaviors**

2 **5.5.1 Optional behavior in Compact authoring**

3 Optional behaviors represent behaviors that may be engaged by a consumer. When using the compact
4 authoring form for assertions, such behaviors are marked by using `wsp:Optional` attribute with a value
5 of "true". In order to simplify reference to such assertions, we just use the phrase "optional assertions"
6 in this section. During the process of normalization the runtime behavior is indicated by two policy
7 alternatives, one with and one without the assertion. In a consumer/provider scenario, the choice of
8 engaging the runtime behavior is upon the consumer by selecting the appropriate policy alternative.
9 The provider may influence what is possible by choosing whether or not to include policy alternatives
10 in a policy expression, by using the `wsp:Optional` attribute.

11 **5.5.2 Optional behavior at runtime**

12 Since optional behaviors indicate optionality for both the provider and the consumer, behaviors that
13 must always be engaged by a consumer must not be marked as "optional" with a value "true" since this
14 would allow the consumer to select the policy alternative without the assertion, and thus not engaging
15 the behavior.

16 ***Good practice a: Limit use of Optional Assertions***

17 Assertion Authors should not use optional assertions for behaviors that must be present in
18 compatible policy expressions.

19 The target scope of an optional assertion is an important factor for Assertion Authors to consider as it
20 determines the *granularity* where the behavior is optionally engaged. For example, if the assertion is
21 targeted for an endpoint policy subject, it is expected to govern all the messages that are indicated by
22 the specific endpoint when optional behavior is *engaged*. Since the behavior would be applicable to
23 policy subject that is designated, it is important for the Assertion Authors to choose the appropriate
24 level of granularity for optional behaviors, to consider whether a specific message or all messages, etc.
25 are targeted.

27 ***Good practice b: Indicate use of an Optional Assertion***

28 When a given behavior may be optional, it must be possible for both message participants to determine
29 that the assertion is selected by both parties, either out of band or as reflected by the message content.

30
31 When optional behaviors are indicated by attaching assertions with only one side of an interaction, such
32 as an inbound message of a request-response, the engagement of the rest of the interaction will be
33 *undefined*. Therefore, the Assertion Authors are encouraged to consider how the attachment on a
34 message policy subject on a response message should be treated when optional behaviors are specified
35 for message exchanges within a request response for response messages, using message policy subject.
36 Leaving the semantics not specified or incompletely specified may result in providers making
37 assumptions. Similarly, if engagement of a behavior is only specified for an outbound message, the
38 Assertion Authors should consider describing the semantics if the incoming messages also utilized the
39 behavior. This is especially important if the assertion is applicable to more than one specific policy
40 subject. One approach that is currently taken by WS-RM Policy [[Web Services Reliable Messaging](#)
41 [Policy](#)] is to introduce both message and endpoint policy subjects for one of its assertions and require

42 the use of endpoint policy subject when message policy subject is used via attachment.

43

44 ***Good practice c: Consider entire message exchange pattern when specifying Assertions that may be***
45 ***optional***

46 Assertion Authors should associate optional assertions with the appropriate endpoint and use the
47 smallest possible granularity to limit the degree to which optionality applies.

48 Behaviors must be engaged with respect to messages that are targeted to the provider so that the
49 provider can determine that the optional behavior is engaged. In other words, the requirement of self
50 describing nature of messages [[5.3.3 Self Describing Messages](#)] in order to engage behaviors must not
51 be forgotten with regard to the client's ability to detect and select the alternative if it is to participate in
52 the exchange.

53

54 An explicit, out of band mechanism may be necessary to enable a client to indicate that the optional
55 behavior is engaged. Currently such a mechanism is outside the scope of WS-Policy Framework.

56

57 **Example**

58

59 The [Web Services Policy Primer](#) document contains an example that outlines the use of [MTOM](#) as an
60 optional behavior that can be engaged by a consumer. Related to this behavior is an assertion that
61 identifies the use of MIME Multipart/Related serialization [reference]. Policy-aware clients that
62 recognize and engage this policy assertion will use Optimized MIME Serialization for messages.

63

64 The semantics of the MTOM assertion declare that the behavior must be reflected in messages by
65 requiring that they use an obvious wire format (MIME Multipart/Related serialization). Thus, this
66 optional behavior is self describing. For example, an inbound message to a web service that requires
67 MTOM must adhere to Optimized MIME Serialization. By examining the message, the provider can
68 determine whether the policy alternate that contains the MTOM assertion is being obeyed. (Good
69 Practice b)

70

71 Note that if a MTOM assertion were only bound to an inbound message endpoint, then it it would not
72 be clear whether the outbound message from the provider would also utilize the behavior. Thus this
73 assertion should be associated at the granularity of an entire message exchange. The semantics of the
74 assertion should specify this to avoid inappropriate assumptions by implementations. This is important
75 for an optional assertion where it may not be clear whether it is to apply in a message exchange when
76 optionally used in part of that exchange (Good Practice c).