C H A P T E R **4**

ADS PEP Attributes Reference

Revised: March, 2007, OL-11798-01

This chapter presents detailed reference information that you need to use Cisco AON Development Studio (ADS) to assign Policy Execution Plan (PEP) attributes.

Note For more information on implementing an AON network, see the following:

- Other chapters in this guide:
    - Getting Started with Cisco ADS
    - Setting Bladelet Properties, Variables, and Rules
    - ADS Bladelets Reference
    - ADS Message Types Reference
    - E-Mail to Cisco ADS Support
- Other guides in the AON library:
    - *AON Installation and Administration Guide* (for information on the AMC server and nodes)
    - *AON Programming Guide* (for information on custom Bladelets, custom adapters, and application program interfaces)

# Contents

Information About PEP Attributes

PEP attributes are items such as name, package, description, interaction style, and operating variables that apply to the entire collection of Bladelets and paths that constitute a PEP.

PEP Attribute Window and Dialog Boxes

Common tasks involving creating PEPs are discussed in "Getting Started with Cisco ADS" chapter. This section describes how to assign PEP attributes, when you start to create a PEP or at any later time.

You assign PEP attributes by means of the PEP Attributes window (Figure 4-1) and subsequent dialog boxes. (To open this window, follow the procedure in the "Creating PEPs" section on page 1-11.)

Figure 4-1   PEP Attributes Window
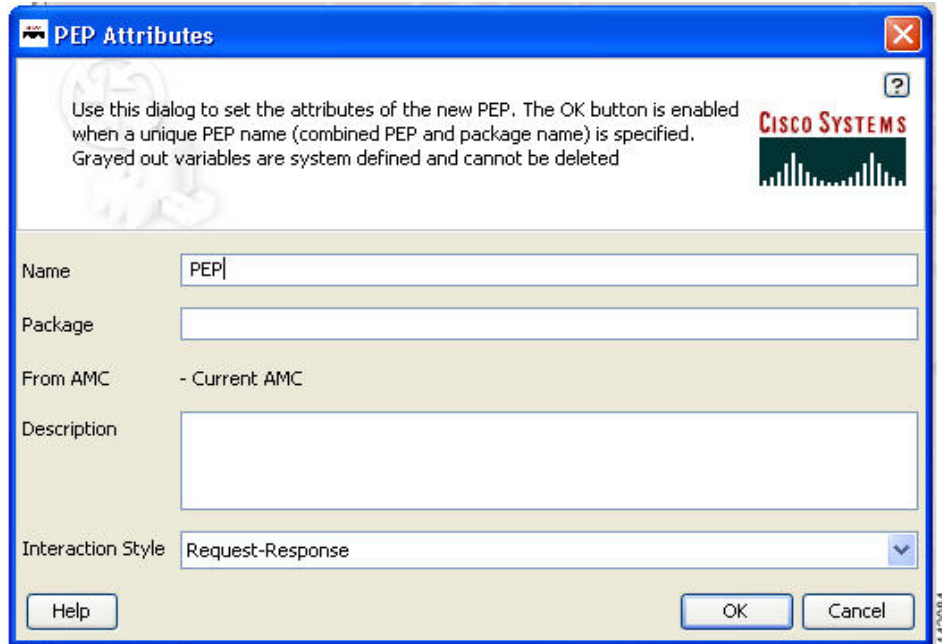
PEP Attribute Variable-Type Choices

PEP attributes are listed in Table 1.

Table 1   Variable Definition: Available Types

Type	Description
AONSSubject	Subject of the AON message.
AONSSubjectListIterator	Iterator for a list of AONSubject objects.

Table 1   Variable Definition: Available Types (continued)

Type	Description
Content	Content of the AON. An object of this type is created by the CreateContent Bladelet and is consumed by the CreateMessage Bladelet. You do not create this directly; rather, it is created by the CreateContent Bladelet. The message type has a content attribute that returns the content of the message (example: REQUEST_MESSAGE.content() gives the content of the incoming message). Content has the following attributes: • numParts—If the underlying content is a MIME content, returns the number of parts in the content. The number of parts of a MIME content in the incoming message is given by REQUEST_MESSAGE.content().numParts(). • document—If the underlying content is of the XML type, returns the Document object corresponding to the actual content. The document representation of the content in the incoming message is given by REQUEST_MESSAGE.content().document().
ContentListIterator	List of content values (see Content for details) that can be accessed one at a time.
Document	DOM document. An object of this type can be extracted from the Content object (see Content for details) if the content is XML. You do not create this directly.
FindContentListIterator	Iterator for content list search results (see Content for details).
FindResult	Collection of search results for one Xpath. You do not create this directly; rather, it is generated by the Find Bladelet. It has the following attributes: • String value—For a single node, returns the string value of the node; for a list of nodes, returns the string value of the first node. • String node Value(*i*)—Returns the string value of the *i*th node • int size—Returns the size of the result set.
FindResultMapIterator	Iterator for a map of Xpath/Regex search results. The key for the map is the name of the input Xpath. The value of the map is the FindResult corresponding to that Xpath. This is generated by the Find Bladelet. It has the following attributes: • FindResult first • FindResult last • Findresult elementAt
FindResultMapListIterator	Iterator for a map list of Xpath/Regex search results (see above).

Table 1   Variable Definition: Available Types (continued)

Type	Description
Message	AON message. The PEP variable REQUEST_MESSAGE of this type is available in the request-action and represents the incoming message. The PEP variable REQUEST_MESSAGE of this type is available in the response-action and represents the outgoing message. The CreateMessage Bladelet can create an object of this type in the PEP. You do not create this directly. It has the following attributes: • messageId—Returns the id of the message. The id of an incoming message is given by: REQUEST_MESSAGE.messageId() • timeStamp—Time at which the message was created. The timestamp of the incoming message is given by REQUEST_MESSAGE.timeStamp() • srcIP—IP address of the message source. The source IP of the incoming message is given by REQUEST_MESSAGE.srcIP() • srcPort—Port number of the message source. The source port of the incoming message is given by REQUEST_MESSAGE.srcPort() • destIP—IP address of the message destination. The destination IP of the incoming message is given by REQUEST_MESSAGE.destIP() • destPort—Port number of the message destination. The destination port of the incoming message is given by REQUEST_MESSAGE.destPort() • destProtocol—String representation of the message protocol. The protocol name of the incoming message is given by REQUEST_MESSAGE.destProtocol() • header—Value of the header in the message. The User-Agent header of the incoming message is given by REQUEST_MESSAGE.header(User-Agent) • content—AON Content of the message. The content of the incoming message is given by REQUEST_MESSAGE.content() • URI—Destination uniform resource identifier (URI) of the message. The URI of the incoming message is given by REQUEST_MESSAGE.URI()
SearchResult	Maps a search specifier to a list of content. In each case, the search specifier is determined by a previously specified search criteria. You use the search specifier to locate the corresponding result for a particular search criteria.
SearchResultListIterator	Iterator over a list of SearchResult objects.
SecurityContext	Store of subject and credential information for certain message or content.
SecurityContextListIterator	Iterator for a list of SecurityContext objects.
Boolean	Value of either true or false.
byte	Value that you can input directly. Range: –128 to 127.

Table 1   Variable Definition: Available Types (continued)

Type	Description
double	Value that you can input directly. Range: 4.9E–324 to 1.7976931348623157E308.
float	Value that you can input directly. Range: 1.401298464324817E–45 to 3.4028234663852886E38.
int	Value that you can input directly. AON uses this in its Bladelets. Range: –2147483648 to 2147483647.
iterator	Keeps the state of a position in a list. It can be used for accessing items from the data structure and can be viewed as a list, array, or stream. It is able to do so one at a time.
list	List or collection of objects. The element of a list can be any object (such as a string). It has the following attributes: • Size—Returns the size of the list. For example, if a PEP variable "I" of type list is defined in the PEP, I.size() gives the number of elements in the list. To retrieve the *n*th element in a list, specify list[*n*]. For example, to get the first element in a list, specify list[0]
long	Value that you can input directly. Range: –9223372036854775808 to 9223372036854775807.
map	Collection of name-value entries. AON supports maps of string-typed keys to string-typed values. It has the following values: • Size—Returns the size of the map. For example, if a PEP variable "m" of type map is defined in the PEP, m.size() gives the number of entries in the map. To retrieve the value in a map corresponding to key "K", use map{K}. In case of a list of maps, to access the value corresponding to key "K" in the first map in the list, use list [0]{K}.
object	Any object. You do not provide this directly. If input to a Bladelet is of this type, any PEP variable can be bound to the input.
short	Value that you can input directly. Range: –32768 to 32767.
string	String of characters. Type the string in the ADS text box or text area.