[W3C](http://www.w3.org/)

**Web Services Addressing 1.0 - Core**

**W3C 推荐 2006年5月9日**

**本版本:**

<http://www.w3.org/TR/2006/REC-ws-addr-core-20060509>

**最新版本:**

<http://www.w3.org/TR/ws-addr-core>

**上一版本:**

<http://www.w3.org/TR/2006/PR-ws-addr-core-20060321>

**编辑:**

Martin Gudgin, Microsoft Corp

Marc Hadley, Sun Microsystems, Inc

Tony Rogers, Computer Associates International, Inc

**摘要**

Web Services Addressing以一种独立于传输的方法来寻址Web服务和信息。 Web Services Addressing 1.0 -核心（本文档）定义了一套抽象的方法和一个XML　Infoset[ XML信息集]的代表，由此来给Web服务提供参考并且方便终端的端到端的寻址。此规则以一种独立于传输的方法使信息系统支持通过网络传递信息，其中包括如终端管理，防火墙，和网关之类的处理节点**．**

**本文的状态**

*本章描述了本文档在发布时的状态.别的档可能会取代本文档. 当前W3C发布文档的列表和这个技术报告的最新版本能够在*[*http://www.w3.org/TR/*](http://www.w3.org/TR/)*中的* [*W3C technical reports index*](http://www.w3.org/TR/) *中找到。*

这是Web Services Addressing 1.0 – Core说明的推荐.由 [Web Services Addressing Working Group](http://www.w3.org/2002/ws/addr/) (WG)发布，WG是[W3C Web Services Activity](http://www.w3.org/2002/ws/Activity)的一部分.

本文档被W3C成员、软件开发者和别的W3C小组以及有关的当事人检阅，并已被W3C Director批准为 W3C推荐标准。本文档是稳定的（stable）。这意味着本文档可被用作参考资料或被其他文档引有。W3C 制定推荐标准的任务是使之受到关注，并促使其被广泛使用。这将增强Web的功能性（functionality）与互操作性（interoperablity）。

WG根据评论对本文档的建议推荐(Proposed Recommendation)版本做了以下编辑上的改动：更清楚地标明了标准的并且提供信息的引用，修正一些典型的错误。执行情况报告和相关测试显示，候选推荐标准已达到了退出标准，并失去了可用标准。 本文档具有对以前版本的标明差异的版本。

如果发现本文档[译注//指英文原始文档]的错误，请发送错误报告至[public-ws-addressing-comments@w3.org](mailto:public-ws-addressing-comments@w3.org)的邮件列表（公开）。

本文档是由[5 February 2004 W3C Patent Policy](http://www.w3.org/Consortium/Patent-Policy-20040205/).下的一组人员完成的。W3C具有公开专利的列表（[public list of any patent disclosures](http://www.w3.org/2004/01/pp-impl/36696/status#specs)）；该列表页面还包括一些公开专利的说明。

**目录**

目录

[**1. 说明** 3](#_Toc201077386)

[**1.1 标识转换** 4](#_Toc201077387)

[**1.2 命名空间** 5](#_Toc201077388)

[**2. 终端引用** 6](#_Toc201077389)

[**2.1 终端引用的信息模型** 6](#_Toc201077390)

[**2.2 终端引用的 XML Infoset 表示** 8](#_Toc201077391)

[**2.3 终端引用对比** 10](#_Toc201077392)

[**2.4 终端引用的生存周期** 10](#_Toc201077393)

[**2.5 终端引用的可扩展性** 10](#_Toc201077394)

[**2.6 在端点引用中识别资源** 11](#_Toc201077395)

[**3. 消息寻址属性** 11](#_Toc201077396)

[**3.1 抽象属性定义** 12](#_Toc201077397)

[**3.2消息寻址属性的 XML Infoset表示** 13](#_Toc201077398)

[**3.3 向一个EPR发送消息** 15](#_Toc201077399)

[**3.4 明确地描述一个应答消息** 15](#_Toc201077400)

[**4. 安全考虑** 17](#_Toc201077401)

[**4.1 附加的安全考虑** 18](#_Toc201077402)

[**5. 引用** 18](#_Toc201077403)

[**5.1 标准的引用** 18](#_Toc201077404)

[**5.2 其它引用** 19](#_Toc201077405)

[**A. 致谢 (非正式)** 20](#_Toc201077406)

**附录**

A. [致谢](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/#acknowledgments) (非正式)

**1. 说明**

《Web Services Addressing 1.0 -核心》（WS-Addressing）定义了两个结构体（constructs）:消息寻址属性(message addressing properties)和终端的引用(endpoint references)，这些使通常由传输协议和信息系统以一种独立于任何特定传输和通迅系统的方式提供的信息标准化.

Web服务终端是一个（可引用的）实体、处理器或资源，通过它找到Web服务的迅息。终端引用传达了另一个终端所需要的信息。请注意， WSDL 2.0有一个终端部分(见《WSDL 2.0的核心语言，第2.15章,终端》)，该部分连同其他的WSDL 2.0部分可以用来描述一个Web服务终端。 事实上Web服务终端可能有多个这样的说明。同样地，多个终端引用（ EPRs ）可以用来传达某一特定的Web服务终端寻址所需的信息。 EPR的用意是传达Web服务终端寻址所需要的信息，而WSDL2.0描述的用意是在描述一个Web服务。

这个规范定义了一套的信息寻址属性（properties），这些属性传达端－端的信息具有的特征，包括源和目的端点的引用、使独立于传输工具的信息寻址统一起来的消息标识。

这两个结构体（construct）均设计为可扩展的和可再用的，以便于其他规则能够建立在这之上并且能处理终端的引用和消息寻址属性之间的平衡。

下面的例子示例了在SOAP1.2上使用这些机制在一个从<http://example.com/business/client1>被发送到http://example.com/fabrikam/purchasing 的消息（见《Web Services Addressing 1.0 –SOAP绑定》在使用SOAP 上的WS-Addressing的更多信息）:

*例子 1-1.在符合SOAP 1.2的消息中使用消息寻址属性。*

(01) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"

xmlns:wsa="http://www.w3.org/2005/08/addressing">

(02) <S:Header>

(03) <wsa:MessageID>http://example.com/6B29FC40-CA47-1067-B31D-00DD010662DA</wsa:MessageID>

(04) <wsa:ReplyTo>

(05) <wsa:Address>http://example.com/business/client1</wsa:Address>

(06) </wsa:ReplyTo>

(07) <wsa:To>http://example.com/fabrikam/Purchasing</wsa:To>

(08) <wsa:Action>http://example.com/fabrikam/SubmitPO</wsa:Action>

(09) </S:Header>

(10) <S:Body>

(11) ...

(12) </S:Body>

(13) </S:Envelope>

第 (02) 到 (09)行是SOAP消息的消息头,在这一部分使用了说明中的规则.第 (10) 到(12)行,是消息体.

第 (03) 到 (08)行包含了消息寻址的文件头（header blocks）. 具体来说, 第 (02)行指定此消息的标识符，第（ 04 ）至（ 06 ）行指定了把该消息做为一个终端引用发送到的终端。第（ 07 ）行指定最终接收这个消息的地址的URI。第（ 08 ）行指明一个Action URI以确定预期语义。

**1.1 标识转换**

本文中以下关键词的解译请见《[*IETF RFC 2119*](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/#RFC2119)》:"MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY"和"OPTIONAL"。

在描述抽象数据模型时，此规范使用《XML Infoset》所使用的标记约定。具体来说，抽象属性名称总是出现在方括号内（例如， [一些属性] ） 。

当描述实体XML模型 《*XML Schema Structures*》,《 *XML Schema Datatypes*》, 该规范使用WS-Security《*WS-Security 2004*.》的标记约定。具体地说即,元素信息项的[childeren]或[attributes]属性的每个成员都使用伪XPath（XPath-like）标记描述 (例如 /x:MyHeader/x:SomeProperty/@value1). 使用{any} 表示元素通配符(<xs:any/>).使用 @{any}表示属性通配符(<xs:anyAttribute/>).

这些伪模式（pseudo-schemas）使用的是BNF样式来转换属性和元素: "?"表示可选(如.0或1), "\*" 表示0或更多, "+"表示一或更多, "[" 和 "]"成对使用, "|"表示选择. 正如标准模式中定义的,属性被赋于个和它们的类型相关的值.内容类型为基本类型的元素被赋于个和它们的内容类型相同的值，为了简洁,伪模式不包括扩展属性.

在定义终端引用属性和消息寻址属性的基数时,本规定使用下面的标志: (*n*..*m*), n代表允许的最小属性数,m代表允许的最大属性数.当n等于m时,相关的终端引用或消息的属性数必须相同.

**1.2 命名空间**

本规定使用的全部命名空间前缀都被列在表1-1中. 注意:命名空间前缀的选择是任意的没有任何语法意义 (参见 《*XML 命名空间*》).

|  |  |
| --- | --- |
| *表1-1. 在本规则中使用的前缀和命名空间* | |
| **前缀** | **命名空间** |
| wsa | http://www.w3.org/2005/08/addressing |
| S | http://www.w3.org/2003/05/soap-envelope |
| xs | http://www.w3.org/2001/XMLSchema |

正如《Web Services Addressing 1.0 – SOAP Binding》里说明的WS-Addressing 可以和 SOAP 一起使用。也如《Web Services Addressing 1.0 - WSDL Binding》里说明的WS-Addressing也可以和WSDL一起用来描述服务。本文档中的示例使用的是XML 1.0的表示,但这点并不是要要求的.

本规定中定义的所有信息项目,被XML 命名空间URI：http://www.w3.org/2005/08/addressing标识.通过打开该XML 命名URI,可以看到一个标准的XML模式文档.

**2. 终端引用**

本章定义了终端引用的信息模型和语法.

此规定介绍终端引用，旨在支持以下使用情况：

This specification introduces the endpoint reference, a construct designed to support the following usage scenarios:

* 动态生成和定制服务终端描述。
* 具体的服务实例的引用和描述被用来做为已知状态的交互的结果而创立。
* 在紧耦合的沟通各方共享了一套共同的关于在交互中使用的具体的政策或协议假想的环境下的灵活的和动态的终端信息交换。
* 在紧耦合沟通各方共享了一套共同的关于在交互中使用的具体的策略或协议的环境下，灵活并且动态的终端信息交流。

**2.1 终端引用的信息模型**

一个终端引用是一个抽象属性的集合。本规定定义了一套核心的属性，其他文档也可以扩大这些属性或者添加其它属性。语义以及任何的此类属性的XML Infoset表示将会在它们的定义文档里给予描述。 一个终端引用由以下抽象属性组成：

**[address] : IRI (1..1)**

一个绝对的 IRI代表了终端的地址.表2-1列出了本规则介绍的两个预定义[地址]值.

|  |  |
| --- | --- |
| *表 2-1. 预定义 [地址]值* | |
| **URI** | **描述** |
| "http://www.w3.org/2005/08/addressing/anonymous" | 一些终端不能被一个有意义的IRI定位;这个URI用来允许这些终端发送/接收消息.  这些URI的确切意义是通过到特定协议的地址的绑定或使用的EPR的内容来定义的。 |
| "http://www.w3.org/2005/08/addressing/none" | 送往[address]为此值的EPRs的消息一定会被丢弃 (如,以不发送该消息的方式丢弃).该URI的典型应用是在EPRs上使用,指定一个回复或故障终端（见第3.1章 抽象属性的定义）指出在没有回复或故障时，消息的发送地址. |

**[reference parameters] : xs:any (0..unbounded).**

一个引用可能为了方便某一特定交互而包含一些和终端有联系的单独参数。

与终端的妥善互动要求引用参数是具有命名空间资格(namespace-qualified)的元素信息项.引用参数由发行引用的终端提供，并假设对其它终端引用的用户不透明。引用参数与消息之间的绑定，取决于与终端交互使用的绑定协议，《Web Services Addressing 1.0-SOAP Binding》描述了SOAP协议的默认绑定.

引用参数没有被规定。关于这点可能出现的任何规定都没有意义，因为引用参数可能以一种不遵守这种规定的方式被消息限定。

**[metadata] : xs:any (0..unbounded)**

引用可能包含终端描述行为，策略和能力的元数据。元数据可能被包括在一个终端引用中，以便通过一个终端引用来减轻处理过程，或是因为该元数据是动态生成的。

内嵌在EPR的元数据不是必须是终端的元数据的一个完整声明,而且, 虽然在EPR 被初始化的时候,元数据必须是有效的,但以后该元数据仍有可能无效

处理有相同[address]的两个EPR的内嵌元数据之间的矛盾,或处理内嵌元数据与具有不同来源的元数据之间的矛盾,或确定内嵌元数据的当前有效性,超出了本文档的范围,解决上述问题可以使用像EPR生存周期信息(参见第2.4章端点引用生存周期)或元数据从一个被授权的范围内复活的规则

**2.2 终端引用的 XML Infoset 表示**

本节定义了对一个终端引用的一个基于XML Infoset的表示,既有XML类型(wsa:EndpointReferenceType),又有XML元素(<wsa:EndpointReference>)的表示 .为了简洁性,使用了XML短语,如使用了'element' 而不是'element information item' – 这并不是打算强制使用本章定义的这种构造方法来做为XML的文本描述.

wsa:EndpointReferenceType,这个标识使用于任何引用Web 服务终端的地方.下面描述了该类型的内容:

<**wsa:EndpointReference**>

<wsa:Address>*xs:anyURI*</wsa:Address>

<wsa:ReferenceParameters>*xs:any*\*</wsa:ReferenceParameters> ?

<wsa:Metadata>*xs:any*\*</wsa:Metadata>?

</**wsa:EndpointReference**>

下面说明上面列出的各种属性和元素:

**/wsa:EndpointReference**

这表示一些wsa:EndpointReferenceType类型的元素. 本例中使用 <wsa:EndpointReference>预定义元素,实际中,任何 wsa:EndpointReferenceType 类型的元素都可以使用.

**/wsa:EndpointReference/wsa:Address**

这是必须的元素 (它的内容是xs:anyURI类型),它指明终端引用的[address]属性.

**/wsa:EndpointReference/wsa:Address/@{any}**

这是一个可扩展的方法,以便允许指定额外的属性..

**/wsa:EndpointReference/wsa:ReferenceParameters**

这是个可选的元素,该元素可以包含任何命名空间的元素,像引用的[reference parameters]里的元素.

**/wsa:EndpointReference/wsa:ReferenceParameters/@{any}**

这是一个可扩展的机制,以便指定别的属性

**/wsa:EndpointReference/wsa:ReferenceParameters/{any}**

任何在[reference parameters](包括[children],[attributes]和[in-scope namespaces]元素)中的元素信息都具有这种形式.

**/wsa:EndpointReference/wsa:Metadata**

该可选元素可以包含任何命名空间的元素。即与终端的交互有关的元数据的元素.

**/wsa:EndpointReference/wsa:Metadata/{any}**

每一个元数据的孩子元素（child element）表示一个元数据的独立的一部分

**/wsa:EndpointReference/wsa:Metadata/{@any}**

这是一个为了允许指定额外属性的可扩展部分.本文档的一些示例说明了该可扩展部分包含一个wsdlLocation属性来提供服务在终端布署的WSDL的位置示意时的使用.

**/wsa:EndpointReference/{any}**

这是一个为了允许指定额外元素的可扩展部分.

**/wsa:EndpointReference/@{any}**

这是一个为了允许指定额外元素的可扩展部分.

**注意:**

描述任何扩展元素或属性的规则经常争论上面的模型会解译这此扩展在抽象属性中可能会出现的影响.就像在第2.1章终端引用的信息模型中描述的那样它们可能即影响核心属性又影响扩展属性.

下面为一个终端引用的示例.这个元素引用了URI为 <http://example.com/fabrikam/acct>的终端.

*例子2-1.终端引用示例.*

<wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">

<wsa:Address>http://example.com/fabrikam/acct</wsa:Address>

</wsa:EndpointReference>

**2.3 终端引用对比**

本规则没有提供确定终端的概念,因此没有提供任何的方法来认定EPR的平等或不平等,并且不会说明他们的平不平等带来的结果.但是,要注意的是,有可能别的文档会提供一个在一定范围内适用的比较功能

**2.4 终端引用的生存周期**

本文档没有为终端引用定义生命周期模型,并且处理终端引用的生存时间问题.别的建立于或者使用WS-Addressing的规则可能为依据该规则建立的终端引用定义了生存周期模型

**2.5 终端引用的可扩展性**

正如 **2.2 使用XML Infoset 表示的终端引用**中所注明,终端引用具有可扩展性.当扩展的属性或元素作为终端引用的一部分出现时,这种扩展的处理模型在这此扩展的规则中定义.处理终端引用的软件可以安全地忽略任何它认识不了或不能理解的扩展.

扩展的元素或属性可以为一个终端引用添加这个终端引用在**2.1终端引用的信息模型**之外的属性。终端引用扩展可能修改终端引且具有的一个或多个现存的属性值.该扩展可能修改绑定终端引用属性到消息地址属性的规则,甚至可以指定使用别的绑定.

当面对软件能够忽略认识不了或不能理解的扩展时,请注意，这种可以修改现有的属性和绑定行办的能力可能会导致在行为上的差异,这种差异取决于是否这种扩展终端引用是被理解这种扩展的软件处理的.在设计终端引用扩展时,设计者应该考虑更多依据本规则的标准处理过程,以防他们的扩展不被软件认识和理解。

**2.6 在端点引用中识别资源**

《Architecture of the World Wide Web》（下面简称为[*AoWWW*](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/#AoWWW)）卷一第2节介绍了使用 URI来识别资源.使用EPR的绝对属性而不是[destination]来识别资源，这点和《AoWWW》推荐的相反.在特定的环境下,这种额外属性的使用可能会是方便的或是有益的; 然而,在设计系统时,应该在使用引用参数识别资源的好处或方便处与像在《AoWWW》第 2.1节中所讲的使用URI唯一标识资源的好处之间做出权衡.

**3. 消息寻址属性**

本节定义了消息寻址属性的信息模型和语法.

消息寻址属性为在交互中出现的终端提供引用. 利用这些特征以支持特定的交互一般被属性本身的语义和控制消息交换的隐藏的不明显的协议来同时定义. 如果明确可用的，本协议可以采取不同的形式，包括，但不限于WSDL 定位和接口;业务流程和电子商贸的规格，除此之外，也可以被用来明确的定义双方之间的协议。

在一个单向互动的模式中,源终端传送消息至目的地，没有定义任何进一步的交互。 “请求-应答”是一种常见的互动模式,该模式由一个源终端发出的初始信息(the request) ，以及随后发出的信息经过目的终端回到源终端（the response） 。反回的消息可以是一个应用消息，一个故障消息，或任何别的消息。不过，请注意，应答消息也可能会作为其它交互信息的一部分发出，并不仅限于一般的单一请求，单一反应模式，或以某一特定的WSDL传输原始数据或定位信息。终端之间的交互协议，可指定发送多个或一个回复。

本规则中定义的信息地址属性集供很多简单变化的单向的并且是请求/应答模式的交互使用.更多的交互模式需额外的消息寻址属性来扩大本文提供的属性，以提高交互能力.

**3.1 抽象属性定义**

消息地址属性集用下面的抽象属性来扩大消息,以支持单向模式,请求/应答模式,以及别的交互模式：

**[destination] : IRI (1..1)**

一个绝对IRI用来表示要接收该消息的地址.

**[source endpoint] : endpoint reference (0..1)**

消息来源终端的引用.

**[reply endpoint] : endpoint reference (0..1)**

要应答该消息的终端的引用.

**[fault endpoint] : endpoint reference (0..1)**

接收该消息相关的故障消息的终端的引用.

**[action] : IRI (1..1)**

一个用来唯一地标识消息的语法含义的绝对的IRI.

推荐[action]的值是一个在WSDL接口或端口类型之内标识一个输入,输出,或故障消息的IRI.一个action可能直接或间接地和相应的WSDL定义有关联.《Web Services Addressing 1.0 - WSDL Binding》描述了联系的机制.

**[message id] : IRI (0..1)**

一个唯一地标识消息的绝对IRI. 当它出现时，它就是有责任的发送方，以确保每个消息都是被唯一地标识的。当接收到的消息最初就包含相同的[message id]时,接收方的行为是不受本规则限制的.

**[relationship] : (IRI, IRI) (0..unbounded)**

用来指定消息之间的关系的两个值。关系的类型由一个绝对IRI来指定.

相关的消息被一个相当于相关消息的[message id]属性的绝对的IRI标识来标识。消息标志IRI可能指向一个特定的消息,或下面预义的URI来表示”没有特定的消息”: "http://www.w3.org/2005/08/addressing/unspecified"

该规则指定下面表3-1列出的一种预定义的关系类型.

|  |  |
| --- | --- |
| *表 3-1. 预定义 [relationship]值* | |
| **URI** | **描述** |
| "http://www.w3.org/2005/08/addressing/reply" | 指示这是用[message id]IRI指定的消息的回复. |

**[reference parameters] : xs:any (0..unbounded).**

相当于消息寻址到的终端引用的[reference parameters]属性值.

[destination]和[action]属性指示了目标处理位置和各个消息的行为和预定行为.这些属性的值可以被用来方便消息的发送.

一个 WS-Addressing消息寻址属性的绑定必须具有上面列出的属性.《Web Services Addressing 1.0 - SOAP Binding》[[*WS-Addressing SOAP Binding*](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/#WSADDR-SOAP)]为SOAP协议定义了如此的绑定.

**3.2消息寻址属性的 XML Infoset表示**

下面展示了在第3.1章绝对属性定义中定义的消息寻址属性的XML Infoset表示：

<**wsa:To**>*xs:anyURI*</**wsa:To**> ?

<**wsa:From**>*wsa:EndpointReferenceType*</**wsa:From**> ?

<**wsa:ReplyTo**>*wsa:EndpointReferenceType*</**wsa:ReplyTo**> ?

<**wsa:FaultTo**>*wsa:EndpointReferenceType*</**wsa:FaultTo**> ?

<**wsa:Action**>*xs:anyURI*</**wsa:Action**>

<**wsa:MessageID**>*xs:anyURI*</**wsa:MessageID**> ?

<**wsa:RelatesTo** RelationshipType="*xs:anyURI*"?>*xs:anyURI*</**wsa:RelatesTo**> \*

<**wsa:ReferenceParameters**>*xs:any*\*</**wsa:ReferenceParameters**> ?

下面描述上面的模式中的属性和元素:

**/wsa:To**

可选元素(它的内容是xs:anyURI类型),它提供[destination]属性值.如果没有该元素,那么[目的]属性就是:<http://www.w3.org/2005/08/addressing/anonymous>。

**/wsa:From**

可选元素(类型为:wsa:EndpointReferenceType),它提供了[source endpoint]的属性值。

**/wsa:ReplyTo**

可选元素(类型为: wsa:EndpointReferenceType),它提供[reply endpoint]的属性值.如果没有该元素,则[reply endpoint]的[address]属性的EPR为： <http://www.w3.org/2005/08/addressing/anonymous>。

**/wsa:FaultTo**

可选元素(类型为 wsa:EndpointReferenceType),它提供[fault endpoint]的属性值.

**/wsa:Action**

必需元素(它的内容为xs:anyURI类型),它传达了[action]属性的值。

**/wsa:MessageID**

可选元素(它的内容为 xs:anyURI类型),它传达了[message id]属性。

**/wsa:RelatesTo**

可重复选元素信息项,它以 (IRI,IRI)对的形式组成了一个抽象[relationship]属性值.该元素的内容 ( xs:anyURI类型) 传达了与 [message id]相关的消息。

**/wsa:RelatesTo/@RelationshipType**

可选属性( xs:anyURI类型),它以传达了一个IRI形式的关系类型.当不用该属性是,该关系类型的暗含属性值是:<http://www.w3.org/2005/08/addressing/reply>。

**/[reference parameters]\***

[reference parameters]中的每个元素信息项(包括所有的 [children], [attributes] 和 [in-scope namespaces]元素)都是这种表现形式.

请注意，上面列出的每个元素信息项都允许属性通配符以便未来的扩展.信息处理器可以安全地忽略任何它不能认识的扩展属性.扩展属性允许在需要时使用xml:id[[*xml:id*](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/#XMLID)]来标识这些元素。

**3.2.1 IRI的对比**

如 [action], [message id]与 [relationship]类的消息寻址属性值是绝对的IRI。这些IRI的目的与其说是取回资源不如说是识别资源。因此, 足够用在《[*IETF RFC 3987*](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/#RFC3987)》第5.3.1章讲述的简单的字符串匹配来决定这些IRI之间的等价性的。

[destination]属性值的比较除了使用简单的字符串匹配来判断属性值是否是匿名的,即[destination]是否有<http://www.w3.org/2005/08/addressing/anonymous>这个值之外,均是超出范围的。

**3.3 向一个EPR发送消息**

本节描述了怎么根据一个EPR构成一个消息的过程 。

1. 如果 EPR的 [address]属性是"http://www.w3.org/2005/08/addressing/none" ,就丢弃该消息,否则,就填入消息的消息寻址属性:
   * [action]:该属性为必需的,但不是从EPR得到的。
   * [destination]:该属性即为EPR的 [address]属性值。
   * [reference parameters]:该属性即选择的 EPR的 [reference parameters]属性值。

**3.4 明确地描述一个应答消息**

本节说明了创建一个和别的消息相关的应答或故障消息的WS-Addressing-specific规则。

1. 选择合适的EPR：
   * 如果应答是个普通的消息,从相关的消息的[reply endpoint]消息寻址属性中选择EPR。

**注意:**

在使用XML Infoset表示时,[reply endpoint]消息寻址属性将一直存在,如果没有 wsa:ReplyTo元素, [reply endpoint]的消息地址属性值将默认为一个[address]属性为”http://www.w3.org/2005/08/addressing/anonymous" 的EPR-见 [**3.2 消息地址属性的XML Infoset表示**](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/#msgaddrpropsinfoset)。

如果没有[reply endpoint]消息寻址属性,处理时一定会出错.除了使用消息寻址属性的替代表示.

* + 如果答复是故障信息并且相关信息的[fault endpoint]信息地址属性是不为空，那么从该属性选择EPR。如果[fault endpoint]属性为空，从相关讯息的[reply endpoint]信息地址属性选择EPR;如果[reply endpoint]属性是空的，相关讯息的接近者的行为是由不受本规则约束的。
  + 在上述任何情况下，如果相关讯息，缺乏一个[message id]属性，处理器要报错.

1. 发送的消息除了包含上面几点外,也需包括：
   * [relationship]: 该属性必须包含像下面所示的一对IRI;关系类型是预定义的应答URI "http://www.w3.org/2005/08/addressing/reply" 并且相关消息的标识符是从被回复的消息中得到的 [message id]属性值;本属性中也可能表示别的关系。

下面的例子示例了以SOAP1.2消息的形式，且被序列为标头块的包含消息寻址属性的消息：

*例子 3-1. 示例消息。*

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

xmlns:wsa="http://www.w3.org/2005/08/addressing">

<S:Header>

<wsa:MessageID>http://example.com/someuniquestring</wsa:MessageID>

<wsa:ReplyTo>

<wsa:Address>http://example.com/business/client1</wsa:Address>

</wsa:ReplyTo>

<wsa:To>mailto:fabrikam@example.com</wsa:To>

<wsa:Action>http://example.com/fabrikam/mail/Delete</wsa:Action>

</S:Header>

<S:Body>

<f:Delete xmlns:f="http://example.com/fabrikam">

<maxCount>42</maxCount>

</f:Delete>

</S:Body>

</S:Envelope>

该消息应该具有下列的属性值：

* [destination]: "mailto:fabrikam@example.com"
* [reply endpoint]: [address]为 <http://example.com/business/client1>的终端
* [action]: "http://example.com/fabrikam/mail/Delete"
* [message id]: "http://example.com/someuniquestring"

下面举了一个上面消息的应答的例子：

*例子 3-2. 应答消息示范.*

<S:Envelope

xmlns:S="http://www.w3.org/2003/05/soap-envelope"

xmlns:wsa="http://www.w3.org/2005/08/addressing">

<S:Header>

<wsa:MessageID>http://example.com/someotheruniquestring</wsa:MessageID>

<wsa:RelatesTo>http://example.com/someuniquestring</wsa:RelatesTo>

<wsa:To>http://example.com/business/client1</wsa:To>

<wsa:Action>http://example.com/fabrikam/mail/DeleteAck</wsa:Action>

</S:Header>

<S:Body>

<f:DeleteAck xmlns:f="http://example.com/fabrikam"/>

</S:Body>

</S:Envelope>

该消息应该具有下列属性值：

* [destination]: "http://example.com/business/client1"
* [action]: "http://example.com/fabrikam/mail/DeleteAck"
* [message id]: "http://example.com/someotheruniquestring"
* [relationship]: ("http://www.w3.org/2005/08/addressing/reply", "http://example.com/someuniquestring")

**4. 安全考虑**

如果接收方不认为该消息是对处理器安全的,本文档不要求消息接收方在该消息上遵守 WS-Addressing 架构。

Conformance to this specification does not require a message receiver to honor the WS-Addressing constructs within a message if the receiver is not satisfied that the message is safe to process.

WS-Addressing支持消息发送方具有选择指定一个消息接收方发送额外的非请求信息到别的接收方.为了扩展该非请求信息的内容,也可以使用初始消息发送方提供的引用参数.因为这些能力,使用WS-Addressing的通迅的是足够安全的,并且在接收方在一个消息内处理WS-Addressing构造时实现了通迅双方的足够安全.下面是消息安全性的几个方面:

1. 为了防止EPR和消息地址属性被篡改,EPR和消息地址属性应该是绝对受保护的.传输时应该通过使用消息级别签名,或使用一个EPR内的XML数字签名来提供这种私有保护。
2. EPR的使用者应该通过考虑下面几个方面在使用前验证EPR的可信度:
   1. 该EPR的来源是否可靠
   2. 该EPR的来源是否被授权表示该EPR的 [address]属性
   3. 该EPR的 [address]是否是个可信的目的地址

举例来说， 信息的接收方可能依赖于由一个可信任的一方发署的对消息寻址属性的可核查的签名,以确定该信息是否具有可信任的来源，并进一步要求确认[reply endpoint]和[fault endpoint]是否签署了一项原则，有权代表这些EPR的[address]，以确保没有发送非请求信息。建立信任的另外一个含义是可用来确定是否有特别的EPR是可信任的。

**4.1 附加的安全考虑**

为了防止信息泄露,EPR发行者没有足够的保护时,不应该把敏感信息放在 [address]或[reference parameters]属性里.

一些处理器可能会使用[message id]作为唯一性标准的一部分来检测信息回复.为了回复检测应该仔细确认[message id]是原数据里的消息编号,比如时间戳,这样就能把合法的转发与回复攻击相区别开来。也可以使用不可预先得到的[message id]在构造时防止攻击,并且在没有收到实际的消息时,发送一个非请求的回复到消息地址那里同样也是明智之举。

**5. 引用**

**5.1 标准的引用**

**[IETF RFC 2119]**

[*Key words for use in RFCs to Indicate Requirement Levels*](http://www.ietf.org/rfc/rfc2119.txt), S. Bradner, Author. Internet Engineering Task Force, June 1999. 地址为http://www.ietf.org/rfc/rfc2119.txt.

**[IETF RFC 3987]**

[*Internationalized Resource Identifiers (IRIs)*](http://www.ietf.org/rfc/rfc3987.txt) M. Duerst, and M. Suignard, Authors. Internet Engineering Task Force, January 2005. 地址为 http://www.ietf.org/rfc/rfc3987.txt.

**[XML 1.0]**

[*Extensible Markup Language (XML) 1.0 (Third Edition)*](http://www.w3.org/TR/2004/REC-xml-20040204), T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, Editors. World Wide Web Consortium, 4 February 2004. XML 1.0 Recommendation的本版本在 <http://www.w3.org/TR/2004/REC-xml-20040204>.最新版本在 http://www.w3.org/TR/REC-xml.

**[XML Namespaces]**

[*Namespaces in XML*](http://www.w3.org/TR/1999/REC-xml-names-19990114), T. Bray, D. Hollander, and A. Layman, Editors. World Wide Web Consortium, 14 January 1999. XML Information Set Recommendation 的本版本在<http://www.w3.org/TR/1999/REC-xml-names-19990114>.最新版本在http://www.w3.org/TR/REC-xml-names.

**[XML Information Set]**

[*XML Information Set (Second Edition)*](http://www.w3.org/TR/2004/REC-xml-infoset-20040204/), J. Cowan and R. Tobin, Editors. World Wide Web Consortium, 4 February 2004。XML Information Set 的本版本在http://www.w3.org/TR/2004/REC-xml-infoset-20040204. 最新版本在http://www.w3.org/TR/xml-infoset.

**[XML Schema Structures]**

[*XML Schema Part 1: Structures Second Edition*](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/), H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, Editors. World Wide Web Consortium, 28 October 2004. XML Schema Part 1 Recommendation 的本版本在 http://www.w3.org/TR/2004/REC-xmlschema-1-20041028. 最新版本在http://www.w3.org/TR/xmlschema-1.

**[XML Schema Datatypes]**

[*XML Schema Part 2: Datatypes Second Edition*](http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/), P. Byron and A. Malhotra, Editors. World Wide Web Consortium, 28 October 2004. XML Schema Part 2 Recommendation的本版本在<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>.最新版本在http://www.w3.org/TR/xmlschema-2.

**5.2 其它引用**

**[AoWWW]**

[*Architecture of the World Wide Web, Volume One*](http://www.w3.org/TR/2004/REC-webarch-20041215/), I. Jacobs, and N. Walsh, Editors. World Wide Web Consortium, 15 December 2004. Architecture of the World Wide Web, Volume One Recommendation的本版本在<http://www.w3.org/TR/2004/REC-webarch-20041215/>。最新版本在 http://www.w3.org/TR/webarch/

**[SOAP 1.1]**

[*Simple Object Access Protocol (SOAP) 1.1*](http://www.w3.org/TR/2000/NOTE-SOAP-20000508/), D. Box, et al, Editors. World Wide Web Consortium, 8 May 2000. 地址为http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

**[SOAP 1.2 Messaging Framework]**

[*SOAP Version 1.2 Part 1: Messaging Framework*](http://www.w3.org/TR/2003/REC-soap12-part1-20030624/), M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, and H. Frystyk Nielsen, Editors. W3C Recommendation, World Wide Web Consortium, 24 June 2003. SOAP Version 1.2 Part 1: Messaging Framework Recommendation的本版本在http://www.w3.org/TR/2003/REC-soap12-part1-20030624/. 最新版本在http://www.w3.org/TR/soap12-part1/.

**[WS-Addressing SOAP Binding]**

[*Web Services Addressing 1.0 - SOAP Binding*](http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509), M. Gudgin, M. Hadley, and T. Rogers, Editors. World Wide Web Consortium, 9 May 2006. WS-Addressing SOAP Binding Recommendation 的本版本在http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509. 最新版本在http://www.w3.org/TR/ws-addr-soap.

**[WS-Addressing WSDL Binding]**

[*Web Services Addressing 1.0 - WSDL Binding*](http://www.w3.org/TR/2006/WD-ws-addr-wsdl-20060216), M. Gudgin, M. Hadley, T. Rogers, and Ü. Yalçinalp, Editors. World Wide Web Consortium, 16 February 2006. WS-Addressing WSDL Binding specification 的本版本在http://www.w3.org/TR/2006/WD-ws-addr-wsdl-20060216. 最新版本在http://www.w3.org/TR/ws-addr-wsdl.

**[WSDL 1.1]**

[*Web Services Description Language (WSDL) 1.1*](http://www.w3.org/TR/2001/NOTE-wsdl-20010315), E. Christensen, et al, Authors. World Wide Web Consortium, March 2001.地址为http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

**[WSDL 2.0 Core Language]**

[*Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*](http://www.w3.org/TR/2006/CR-wsdl20-20060327), R. Chinnici, J. J. Moreau, A. Ryman, and S. Weerawarana, Editors. World Wide Web Consortium, 27 March 2006. WSDL 2.0 specification 的本版本在 http://www.w3.org/TR/2006/CR-wsdl20-20060327. 最新版本在http://www.w3.org/TR/wsdl20.

**[WS-Security 2004]**

[*Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)*](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf), A. Nadalin, C. Kaler, P. Hallam-Baker, and R. Monzillo, Editors. Organization for the Advancement of Structured Information Standards, March 2004. 地址为http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.

**[xml:id]**

[*xml:id Version 1.0*](http://www.w3.org/TR/2005/REC-xml-id-20050909/), J. Marsh, D. Veillard, and N. Walsh, Editors. World Wide Web Consortium, 9 September 2005. xml:id Version 1.0 Recommendation 的本版本在<http://www.w3.org/TR/2005/REC-xml-id-20050909/>.最新版本在http://www.w3.org/TR/xml-id/.

**A. 致谢 (非正式)**

本文档是 [W3C Web Service Addressing Working Group](http://www.w3.org/2002/ws/addr/)集体智慧的结晶。

Working Group的成员如下(按字母顺序排列): Abbie Barbir (Nortel Networks), Andreas Bjärlestam (ERICSSON), Dave Chappell (Sonic Software), Eran Chinthaka (WSO2), Francisco Curbera (IBM Corporation), Glen Daniels (Sonic Software), Vikas Deolaliker (Sonoa Systems, Inc.), Paul Downey (BT), Jacques Durand (Fujitsu Limited), Robert Freund (Hitachi, Ltd.), Marc Goodner (Microsoft Corporation), Arun Gupta (Sun Microsystems, Inc.), Hugo Haas (W3C/ERCIM), Marc Hadley (Sun Microsystems, Inc.), David Hull (TIBCO Software, Inc.), Yin-Leng Husband (HP), David Illsley (IBM Corporation), Anish Karmarkar (Oracle Corporation), Paul Knight (Nortel Networks), Philippe Le Hégaret (W3C/MIT), Amelia Lewis (TIBCO Software, Inc.), Bozhong Lin (IONA Technologies, Inc.), Mark Little (JBoss Inc.), Jonathan Marsh (Microsoft Corporation), Jeff Mischkinsky (Oracle Corporation), Nilo Mitra (ERICSSON), Eisaku Nishiyama (Hitachi, Ltd.), Ales Novy (Systinet Inc.), David Orchard (BEA Systems, Inc.), Gilbert Pilz (BEA Systems, Inc.), Alain Regnier (Ricoh Company, Ltd.), Tony Rogers (Computer Associates), Tom Rutt (Fujitsu Limited), Davanum Srinivas (WSO2), Jiri Tejkl (Systinet Inc.), Mike Vernal (Microsoft Corporation), Steve Vinoski (IONA Technologies, Inc.), Katy Warr (IBM Corporation), Pete Wenzel (Sun Microsystems, Inc.), Steve Winkler (SAP AG), Ümit Yalçinalp (SAP AG), Prasad Yendluri (webMethods, Inc.).

Previous members of the Working Group were: Lisa Bahler (SAIC - Telcordia Technologies), Rebecca Bergersen (IONA Technologies, Inc.), Ugo Corda (Sun Microsystems, Inc.), Michael Eder (Nokia), Yaron Goland (BEA Systems, Inc.), Marc Goodner (SAP AG), Martin Gudgin (Microsoft Corporation), Mark Nottingham (BEA Systems, Inc.), Mark Peel (Novell, Inc.), Harris Reynolds (webMethods, Inc.), Rich Salz (IBM Corporation), Davanum Srinivas (Computer Associates), Greg Truty (IBM Corporation).

同时也非常感谢对[discussions on public-ws-addressing@w3.org](http://lists.w3.org/Archives/Public/public-ws-addressing/)作出贡献的人们。