GUIa para los Name SPACES

Tabla de contenido

[1 Nota de Lenguaje 4](#_Toc23766768)

[2 Generalidades 4](#_Toc23766769)

[3 Elementos, atributos y vocabularios 4](#_Toc23766770)

[3.1 Elementos 4](#_Toc23766771)

[3.2 Atributos 5](#_Toc23766772)

[3.3 Vocabularios 5](#_Toc23766773)

[4 Namespace y sus elementos 5](#_Toc23766774)

[4.1 Namespace y atributos 9](#_Toc23766775)

[4.2 Trabajando con NAMESPACE 10](#_Toc23766776)

[4.3 Terminología de NAMESPACE 10](#_Toc23766777)

[4.3.1 Local Name 11](#_Toc23766778)

[4.3.2 Expanded Name 11](#_Toc23766779)

[4.3.3 Namespace Prefix 11](#_Toc23766780)

[4.3.4 Default Namespace 11](#_Toc23766781)

[4.3.5 Qualified Name 12](#_Toc23766782)

[4.3.6 Target Namespace 12](#_Toc23766783)

[4.4 Industry Standard Namespace 12](#_Toc23766784)

[5 Namespace Syntax 13](#_Toc23766785)

[5.1.1 Declaring a Namespace Prefix 13](#_Toc23766786)

[5.2 OVERRIDING A PREFIX 13](#_Toc23766787)

[5.3 VALID PREFIX NAMES 17](#_Toc23766788)

[6 Namespace and Web Service Contracts 18](#_Toc23766789)

[7 Common Namespace in the Web Service Contract 18](#_Toc23766790)

[7.1 WSDL NAMESPACE 19](#_Toc23766791)

[7.2 XML SCHEMA NAMESPACE 19](#_Toc23766792)

[7.3 SOAP NAMESPACE 19](#_Toc23766793)

[7.4 WS-POLICY NAMESPACE 20](#_Toc23766794)

# Nota de Lenguaje

Por ser este un documento altamente tecnico en su contenido y con el objetivo de no perde el contesxto de algunas frases se a decidido mantener en lenguaje ingles algunos términos, tales como Namespace para “espacio de nombre” o foward-slash para “plica” WSDL y SOAP tambien se mantendran en ingles asi como el ejemplo con que se trata de ilustrar el desarrollo del documento.

# Generalidades

Antes de comenzar primero necesitamos establecer Namespace una parte de las plataformas de tecnología de servicios web y XML que es tan fundamental que su comprensión es crítica para diseñar, usar y gobernar el servicio web contratos

Los Namespace han sido tradicionalmente una de las consideraciones de tiempo de diseño más olvidadas, principalmente porque simplemente no se han entendido lo suficientemente bien.

# Elementos, atributos y vocabularios

Comencemos con una introducción a elementos, atributos y vocabularios para que podamos preparar el escenario de cómo los namespace se ajustan al mundo XML.

## Elementos

Cada vez que le pide a alguien que nombre las partes más fundamentales del lenguaje XML, es probable que responda "elementos y atributos". Por supuesto, es cierto que un documento XML está compuesto por una serie de elementos, que existen como etiquetas (texto rodeado de corchetes angulares) de la siguiente manera:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Lo importante de los elementos es que pueden tener contenido. Este contenido puede ser texto u otros elementos, o ambos. Por lo tanto, es útil ver los elementos como contenedores de información. Para determinar qué tan grande es un contenedor (elemento) dado, necesitamos saber dónde comienza y dónde termina. Esta es la razón por la cual los elementos se definen generalmente en pares, donde cada par se compone de un elemento de apertura y un elemento de cierre que juntos establecen una construcción.

Como se muestra aquí, el elemento de cierre se identifica mediante la barra diagonal “/” que se coloca después de su primer paréntesis angular:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## Atributos

Para proporcionar información complementaria sobre un elemento específico sin tener que agregar texto o elementos adicionales, simplemente puede pegar un nombre de atributo y su valor entre los corchetes angulares justo después del elemento, de la siguiente manera:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## Vocabularios

Un vocabulario XML es simplemente un conjunto de elementos y atributos relacionados. Por ejemplo, puede crear una serie de elementos y atributos que podrían usarse para definir la estructura de un documento de factura. Esto probablemente constituiría un vocabulario.

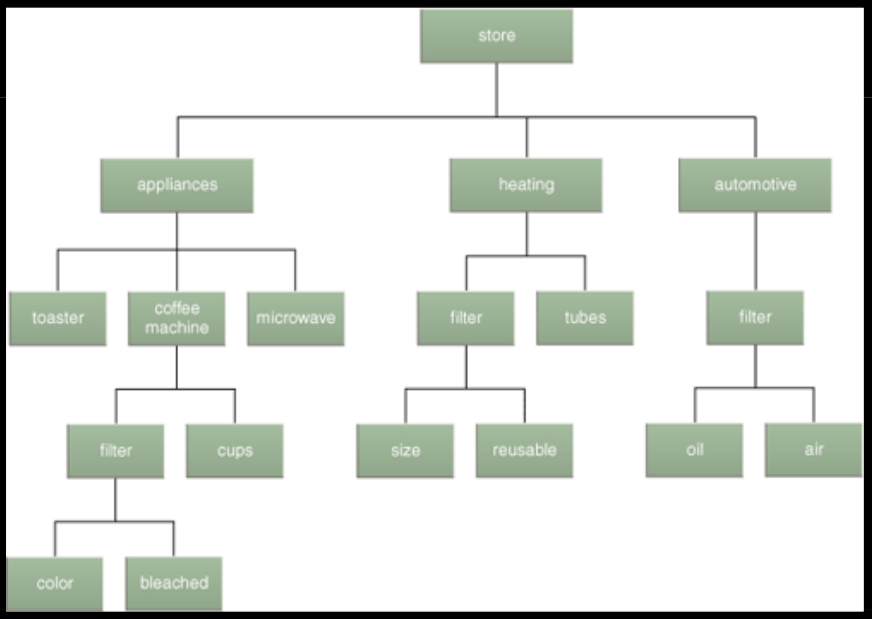
Del mismo modo, es posible que deba trabajar con lenguajes basados ​​en XML (como XML Schema, WSDL o SOAP). Si bien estas representan tecnologías estandarizadas, también existen como colecciones de elementos y atributos XML relacionados. Por lo tanto, cada uno también se considera un vocabulario separado.

Tenga en cuenta que el uso del término "vocabulario" es subjetivo.

# Namespace y sus elementos

Con solo mirar un elemento, puede ser difícil entender lo que está describiendo. Por ejemplo, en un documento donde los elementos se usan para representar productos que se pueden comprar en una ferretería, un elemento llamado filtro podría referirse a uno de varios tipos diferentes de filtros, cada uno con su propio conjunto de características únicas.

La Figura 1 proporciona un ejemplo de vocabulario de elementos XML para una tienda que vende electrodomésticos, equipos de calefacción y piezas de automóviles. Notará en la tercera fila de la jerarquía de elementos que hay tres elementos llamados "filtro", cada uno usado para un propósito diferente (una máquina de café, un horno de calefacción y un motor de automóvil).



Tener diferentes tipos de productos de filtro que usan el mismo elemento de filtro puede plantear un problema de representación de datos bastante significativo. Por ejemplo, es probable que un filtro de café deba representarse de manera diferente que un filtro de automóvil.

Entonces, ¿qué podemos hacer sobre este conflicto de nombres? Bueno, la respuesta fácil es simplemente cambiar los nombres. ¿Por qué no simplemente crear elementos separados coffee\_filter y car\_filter?

Esa es ciertamente una solución y un enfoque que en realidad es bastante común. Pero, ¿y si no fueras tú quien nombrara los elementos? Supongamos que tuvo que trabajar con vocabularios XML predefinidos por organizaciones de estándares o diferentes grupos en su empresa de TI. En este caso, es posible que tenga un elemento llamado filtro proporcionado por la Federación Internacional de Fabricantes de Filtros de Café Federation of Coffee Filter Makers (IFCFM) y otro elemento llamado filtro definido por la International Federation of Automobile Filter Makers (IFAFM).

Incorporar estos elementos en un solo documento XML causará problemas y confusión, sin el uso de algún tipo de identificador adicional. Entonces, ¿cómo podemos crear, por ejemplo, un documento de pedido flexible para que el mecánico de automóviles con adicción a la cafeína pueda comprar un filtro de café y un filtro de automóvil al mismo tiempo? La respuesta, es Namespace .

Los Namespace fueron creados por el W3C en una especificación separada que complementa el estándar XML y prácticamente todos los estándares tecnológicos asociados con los contratos de servicios web. (Esta especificación se denomina "Namespaceen XML", y puede acceder a ella a través de www.soaspecs.com).

Un espacio de nombres simplemente nos da una forma de organizar elementos relacionados en grupos. Dentro de un grupo, cada nombre de elemento es idealmente único. De esta forma, diferentes tipos de elementos pueden tener el mismo nombre cuando pertenecen a diferentes grupos.

Creemos algunos grupos para representar los tres tipos de productos de filtro que vende la tienda:

* CoffeeFilterNamespace
* AutoFilterNamespace
* HeatingFilterNamespace

Ahora necesitamos una forma práctica de asociar cada uno de estos grupos con un elemento de filtro individual. Los Namespaceproporcionan un medio para relacionar un grupo con un elemento extendiendo su nombre para incluir una referencia a su grupo.

El siguiente ejemplo agrega un espacio de nombres al elemento de filtro:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

También notará la adición de un nuevo atributo llamado "xmlns", abreviatura de "espacio de nombres XML". Este atributo esencialmente nos permite establecer un espacio de nombres dándole un nombre ("CoffeeFilterNamespace") y un identificador especial ("cfn" ) que actúa como una referencia al espacio de nombres.

Este identificador se denomina prefijo porque (junto con el símbolo de dos puntos) prefija un elemento para indicar a qué espacio de nombre pertenece el elemento, de la siguiente manera:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Con el prefijo y el elemento combinados, ahora terminas con un nombre calificado (también llamado "QName").

El siguiente es el ejemplo correspondiente para el filtro del automóvil, donde el nombre calificado es afn: filter:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Debido a que el uso de un espacio de nombres nos permite crear un elemento completamente nuevo con el mismo nombre, podemos darle al elemento afn: filter un atributo adicional que no existe con el elemento cfn: filter.

El uso de nombres calificados nos permite colocar elementos con el mismo nombre pero que pertenecen a diferentes grupos en un solo documento XML. Mostraremos muchos ejemplos de esto en capítulos posteriores, pero aquí hay una vista previa de cómo se vería la sección de detalles de un documento de pedido con tres elementos de filtro calificados diferentes:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

En este ejemplo, los atributos xmlns se movieron al elemento raíz para que podamos tener múltiples ocurrencias de cada elemento calificado sin tener que repetir este atributo (como lo demuestran los dos cfn: filtes elements).

Además, es posible que haya captado el hecho de que los elementos de orden y detalle no tienen un prefijo. Esto se debe a que podemos establecer un espacio de nombres "predeterminado" para el que no necesitamos proporcionar un prefijo. Explicaremos los prefijos y los Namespacepredeterminados con más detalle en breve.

La Figura 2 muestra el vocabulario de elementos XML que hemos ido construyendo gradualmente. Dentro de nuestra tienda, hemos identificado tres tipos de productos que requieren un elemento de filtro. Para evitar conflictos de nombres, creamos tres Namespace separados para representar los elementos de filtro (y sus elementos secundarios).

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## Namespace y atributos

Otra cosa interesante que puede hacer con XML es asociar un espacio de nombres con un atributo. Esto le permite agrupar atributos relacionados en su propio espacio de nombres tal como puede hacerlo con los elementos. En el siguiente ejemplo, el atributo productID está en un espacio de nombres diferente que el elemento del juego:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Y, los atributos y elementos pueden pertenecer al mismo espacio de nombres:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## Trabajando con NAMESPACE

Cada elemento y atributo solo puede pertenecer a un máximo de un espacio de nombres. Alternativamente, un elemento o un atributo no necesita pertenecer a ningún espacio de nombres. Los primeros ejemplos de código que mostramos al comienzo de la sección Elementos, Atributos y Espacios de nombres, por ejemplo, todavía eran fragmentos XML legítimos sin asociación con espacios de nombres.

Sin embargo, en el mundo del contrato de servicios web y el diseño de mensajes, los Namespaceestán en todas partes. Los encontrará en definiciones WSDL, definiciones de mensajes SOAP e incluso definiciones de políticas. Cierre cualquiera de estos documentos y abra una definición de Esquema XML, y habrá más espacios de nombres, directamente en su cara. Como no puedes escapar de ellos, es mejor abrazarlos.

Como se mencionó anteriormente, algunas organizaciones de estándares externas nos imponen algunos espacios de nombres. Con frecuencia, debemos trabajar "tal cual", construyendo nuestra arquitectura de contrato de servicios web a su alrededor. Sin embargo, en cuanto a los demás, su diseño depende de nosotros.

Debido a que los Namespaceeventualmente se vuelven tan profundos y muy arraigados en todos sus documentos XML relacionados con contratos de servicios web, nunca debe permitir que se creen "ad-hoc". Generalmente se requieren convenciones de diseño y una estrategia de gobernanza de espacio de nombres a largo plazo.

Una de las decisiones de diseño más importantes con las que debe lidiar primero al elaborar una estrategia es determinar cuántos espacios de nombre debe crear. Debido a que cada elemento y atributo solo puede pertenecer a uno, debe asegurarse de definir los tipos correctos de Namespaceque establezcan grupos y categorías con el alcance apropiado de lo que se supone que representan.

Aquí hay algunas cosas a considerar:

* la cantidad de documentos comerciales que espera crear o trabajar con
* los modelos de servicio que planea usar
* el alcance y el tamaño de su inventario de servicios planificado (y si otros inventarios de dominio residirán junto a él)
* otros dominios funcionales o comerciales que pueden existir en su empresa
* Estándares y convenciones existentes relacionados con el modelado de datos y negocios

## Terminología de NAMESPACE

Según lo que hemos descrito hasta ahora, ahora podemos definir algunos términos que se utilizan dentro de la industria para describir y trabajar con espacios de nombres.

### Local Name

El nombre local es simplemente el nombre de un elemento o atributo sin el espacio de nombres al que pertenece.

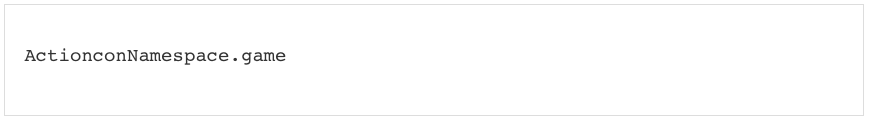
En el siguiente ejemplo, "game" y "productID" son nombres locales:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

### Expanded Name

La combinación de un elemento o nombre de atributo y su espacio de nombre se denomina nombre expandido, como se muestra aquí:



### Namespace Prefix

Debido a que los nombres de Namespacepueden ser bastante largos y pueden contener caracteres que no están permitidos en los nombres XML, no es práctico usar siempre nombres expandidos. En cambio, podemos crear un alias llamado prefijo de espacio de nombres.

En el ejemplo familiar que sigue, el prefijo ns: se declara a través del atributo xmlns y luego lo usa el elemento del juego y el atributo productID.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

### Default Namespace

Como su nombre indica, el espacio de nombres predeterminado representa el espacio de nombres dentro de un documento XML que no requiere un prefijo (porque es el "predeterminado"). En otras palabras, todavía existe un espacio de nombres, pero los elementos que pertenecen a él no requieren un prefijo porque simplemente estamos afirmando que "cualquier elemento sin prefijo pertenece a este espacio de nombres".

Imagen que contiene captura de pantalla

Descripción generada automáticamente

El espacio de nombres predeterminado se usa más comúnmente por conveniencia. En los próximos capítulos, notará que el espacio de nombres predeterminado se utiliza para representar el espacio de nombres WSDL para que los elementos relacionados con WSDL en los contratos de servicios web no requieran un prefijo.

### Qualified Name

Como ya hemos explicado en las secciones anteriores, el nombre calificado de un elemento es la combinación de su nombre local y espacio de nombres (como se representa con el prefijo), como se muestra aquí:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

En la lista, tanto ns: game como ns: productID son nombres calificados. Si se ha declarado un espacio de nombres predeterminado, el nombre del elemento sin prefijo se considera el nombre calificado.

### Target Namespace

Este es un tipo especial de espacio de nombres que se utiliza para establecer un valor de espacio de nombres asociado con un documento XML específico.

## Industry Standard Namespace

Muchos de los Namespace con los que necesitará trabajar han sido predefinidos en especificaciones tecnológicas producidas por organizaciones de estándares, como el W3C y OASIS. Casi todos estos Namespace estandarizados siguen convenciones de nomenclatura como la que acabamos de presentar en la sección anterior.

* WSDL = http://schemas.xmlsoap.org/wsdl/
* XML Schema = http://www.w3.org/2001/XMLSchema
* SOAP = http://schemas.xmlsoap.org/wsdl/soap/

# Namespace Syntax

Hasta ahora, hemos estado explorando para qué namespace se utilizan y cómo se pueden nombrar y estructurar. Ahora es el momento de observar más de cerca cómo los elementos y atributos en un documento XML pueden asociarse con los namespace.

## Declaring a Namespace Prefix

Ha visto cómo se declaran los prefijos de espacio de nombres mediante el uso del atributo especial xmlns, como se muestra nuevamente en nuestro ejemplo familiar:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

En este caso, la declaración de un prefijo se produce dentro del elemento del juego. Una vez establecido, el prefijo se puede usar dentro de ese mismo elemento, así como cualquiera de sus elementos secundarios. Y, por supuesto, también se puede usar con cualquier atributo que exista en el elemento y cualquiera de los elementos secundarios.

Donde en el elemento ocurre la declaración del espacio de nombres no importa. Por ejemplo, en el ejemplo de código anterior, el atributo productID usa el prefijo ns: aunque ese prefijo se declare después del atributo. Esto sigue siendo perfectamente válido.

El siguiente ejemplo muestra cómo la declaración del espacio de nombres también se aplica a un elemento secundario:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Como es de esperar, los elementos gtin y title forman parte del namespace llamado http://actioncon.com/schema/gameinfo porque llevan el prefijo ns: con ellos.

## OVERRIDING A PREFIX

Un prefijo no necesariamente tiene que aplicarse a un documento XML completo. Un elemento secundario puede anular una declaración de espacio de nombres asignando un prefijo a un espacio de nombres diferente que luego se aplica a un subconjunto del documento.

Veamos un ejemplo para este caso:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Aunque el elemento gtin está usando el mismo prefijo, ahora pertenece a un espacio de nombres diferente (texto en negrita) porque el prefijo ns: se "vuelve a declarar" con un nuevo atributo xmlns.

Este ejemplo sería menos confuso si simplemente estableciéramos un nuevo prefijo en lugar de reutilizar el existente, de la siguiente manera:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Un prefijo de namespace se aplica solo dentro del alcance establecido por el elemento que contiene su declaración. Esto significa que en el ejemplo anterior, el prefijo ns2: solo puede ser usado por el elemento gtin y cualquier atributo o elemento hijo que pueda caer dentro de él.

Por lo tanto, el siguiente ejemplo no es válido:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Debido a que el elemento de título no se encuentra dentro del elemento ns2: gtin, no puede usar el prefijo ns2:.

Sin embargo, este ejemplo es válido:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Aunque declaramos el prefijo ns2: en el elemento gtin, esto no significa que todos los atributos y elementos secundarios deben usar este prefijo. De hecho, en la mayoría de los documentos de contrato de servicios web verá todo tipo de elementos de diferentes vocabularios mezclados con diferentes prefijos.

Por lo tanto, a menudo tiene más sentido declarar simplemente todos los prefijos de espacio de nombres en el elemento raíz de un documento dado, como se muestra aquí:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Para una representación visual de una arquitectura de espacio de nombres de muestra, revisemos brevemente el ejemplo de la ferretería, donde hay diferentes tipos de filtros. Originalmente creamos Namespace eparados para los filtros para que el mismo elemento de filtro pudiera usarse para representar diferentes tipos de productos de filtro.

Ahora podemos agregar nuevos Namespacepara establecer los siguientes dominios dentro de la jerarquía de datos lógicos:

* Namespaceespecíficos de la línea de productos para electrodomésticos, calefacción y automotriz
* un espacio de nombres de la Tienda raíz que representa el alcance de toda la tienda

La Figura 3 muestra cómo estas nuevas definiciones de espacio de nombres se aplican al vocabulario XML, creando dominios donde algunos Namespaceanulan a otros.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## VALID PREFIX NAMES

Analicemos brevemente las reglas para nombrar prefijos. Como probablemente ya pueda adivinar, tiene sentido mantenerlos cortos dado que su propósito principal es actuar como un acceso directo para el valor real del espacio de nombres para reducir la longitud del documento XML en su conjunto.

El primer carácter de un prefijo debe ser una letra o el guión bajo (‘\_’). Las letras permitidas incluyen caracteres alfabéticos pero no números o caracteres especiales como ‘(‘ o ‘&’. Para el resto del prefijo, se aplican casi las mismas restricciones, excepto que se permiten números, puntos y guiones.

Los siguientes prefijos de espacio de nombres son válidos:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

... y estos no son válidos:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

# Namespace and Web Service Contracts

Un contrato de servicio web se compone de múltiples tecnologías que se expresan mediante diferentes estándares basados en XML y que las definiciones de contrato resultantes se expresan usando XML. Cuando tiene tantos elementos y atributos XML que se originan en diferentes vocabularios y ahora necesitan unirse para definir un contrato de servicio web, solo sabe que los Namespace erán una gran parte de la ecuación.

Investiguemos esto más a fondo agrupando los tipos de Namespaceinvolucrados en un contrato de servicio web en las siguientes dos categorías:

* namespace que se aplican al contrato en sí
* namespace que se aplican a los mensajes a nivel de cable que se intercambian al usar el contrato

Ahora agreguemos otra dimensión a esto agregando otras dos categorías:

* namespace estandarizados que están predefinidos por los estándares de tecnología de contrato de servicios web
* namespace personalizados definidos por usted, el diseñador del contrato de servicios web

Los dos últimos tipos de Namespace e aplican a cada uno de los primeros. En otras palabras, utilizará Namespaceestandarizados y personalizados para diseñar el contrato de servicio web y cualquier mensaje relacionado.

En esta etapa temprana del libro, no podemos describir de dónde provienen estos Namespaceo cómo se usan. Los capítulos posteriores introducirán Namespacepara nuevos vocabularios e idiomas de forma individual y proporcionarán más orientación para crear valores de espacio de nombres personalizados.

Sin embargo, solo para darle una idea de lo que está por venir, veamos brevemente algunos Namespacemostrando ejemplos de código que incluyen declaraciones de prefijos de espacio de nombres para una definición WSDL y una definición de mensaje SOAP.

# Common Namespace in the Web Service Contract

Comencemos con el siguiente ejemplo que muestra un elemento de definiciones WSDL que contiene una serie de declaraciones de namespace

Imagen que contiene captura de pantalla

Descripción generada automáticamente

El texto en negrita resalta el código que pertenece al espacio de nombres de destino.

## WSDL NAMESPACE

Tanto el resumen como las descripciones concretas se definen utilizando el Lenguaje de descripción de servicios web (WSDL), que se compone de un vocabulario de elementos y atributos asociados con este namespace:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## XML SCHEMA NAMESPACE

Se requiere el siguiente Namespace para los elementos del esquema XML que se utilizan en el contrato de servicio web:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## SOAP NAMESPACE

Se explicó anteriorente cómo la parte vinculante de una descripción concreta se puede utilizar para asociar un contrato de servicio web con un protocolo de comunicación real. Cuando deseamos enviar mensajes de entrada y salida utilizando el protocolo SOAP, debemos indicar esto en la parte vinculante del contrato agregando elementos del lenguaje SOAP. Por eso también se necesita el siguiente namespace:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

## WS-POLICY NAMESPACE

Puede recordar que las políticas se pueden aplicar a varias partes del contrato de servicio web. Para expresar y adjuntar estas políticas, necesitamos usar elementos del vocabulario WS-Policy, que por lo tanto introduce la necesidad de agregar también este espacio de nombres:

Imagen que contiene captura de pantalla

Descripción generada automáticamente