INTERNATIONAL STANDARD

ISO/IEC 24752-6

First edition
2014-12-15

Information technology — User interfaces — Universal remote console —

Part 6: Web service integration

Technologies de l'information — Interfaces utilisateur — Console à distance universelle —

Partie 6: Intégration du service web

Reference number
ISO/IEC 24752-6:2014(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information.

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 35, *User interfaces*.

ISO/IEC 24752 consists of the following parts, under the general title *Information technology — User interfaces — Universal remote console*:

— *Part 1: Framework*

— *Part 2: User interface socket description*

— *Part 4: Target description*

— *Part 5: Resource description*

— *Part 6: Web service integration*

Introduction

Web services are becoming increasingly ubiquitous in the form of public Internet-wide services and private services in protected environments. Even devices and appliances in the digital home are being made network-accessible by exposing them as Web services.

The universal remote console technology allows for pluggable user interfaces for any kind of devices and services, including web services. For a web service to adopt the URC concepts, it needs to expose a composition of user interface socket elements. This can be achieved in two ways: either the Web service provides one target description (see ISO/IEC 24752-4) or one or multiple separate user interface socket descriptions (see ISO/IEC 24752-2). Alternatively, the Web service can provide the target description and the socket description(s) in an “embedded” approach integrated with its Web service interface description. The web service description language (WSDL) defines suitable extension mechanisms for such integration. With this integrated approach, Web services do not need to provide a separate target description and separate socket descriptions. It is sufficient to integrate this information into their WSDL document. It is expected that this approach will help in the adoption of the URC technology for Web services and thus, make personalized and pluggable user interfaces widely available for Web services.

This part of ISO/IEC 24752 defines the syntax and semantics for embedding target description and socket descriptions in interface specifications of Web services so that there is a clear mapping between special elements in the WSDL document and elements of (implicit) target description and (implicit) socket descriptions.

This part of ISO/IEC 24752 lays the groundwork for an integration of the universal remote console framework within Web service environments. It gives rise to various URC-based architectures and implementations, including a middleware approach in which a user interface server provides access to web services, and Web service centric approach in which a Web service exposes a user interface socket via its WSDL-based interface.

Information technology — User interfaces — Universal remote console —

Part 6:
Web service integration

1 Scope

This part of ISO/IEC 24752 defines the syntax and semantics for embedding target description and socket descriptions in interface specifications of web services so that there is a clear mapping between special elements in the WSDL document and elements of (implicit) target description and (implicit) socket descriptions.

2 Conformance

A WSDL1 document conforms to this International Standard if it complies to the web services description language (WSDL) 1.1 specification and with the requirements and recommendations in Clause 6 and Clause 7.

A WSDL2 document conforms to this International Standard if it complies to the web services description language (WSDL) 2.0 specification and with the requirements and recommendations in Clause 6 and Clause 7.

NOTE Strict language conformance (i.e. no additional elements or attributes allowed) is not required because future versions of this part of ISO/IEC 24752 might add new elements, attributes, and values. Therefore, URC manufacturers are encouraged to implement their URCs so that unrecognized markup is ignored without failing.

A Web service conforms to this International Standard if it fulfils the requirements of a conforming target in ISO/IEC 24752-1, in all of the following ways.

- The Web service shall provide at least one service binding (as specified in the Web service's WSDL document) as Target-URC network link.
- The Web service shall have a target name, given as the target namespace of the Web service, as specified in 7.5.2.
- The Web service shall have exactly one target description which shall be embedded in its WSDL document and shall include references to external files containing the target resources (grouping sheets and resource sheets) conforming to at least one natural language, as specified in 7.5.
- The Web service shall provide a fetch mechanism for its target resources (grouping sheets, resource sheets, UIIDs) to be retrieved by URI, including support for MIME types.
- The Web service shall provide a target instance identifier through the 'getTargetInstanceId' operation in the "_target" partition, as specified in 7.5.11.2.
- The Web service shall support locator functions through a "_target" partition, as specified in 7.5.7.
- The Web service shall expose one or more sockets that, when considered together, cover the full functionality of the Web service as a target. For each of these sockets, a socket description shall be embedded in the Web service's WSDL document (as specified in 7.6).
- For each of the Web service's sockets, the socket shall have variables that include all of the dynamic data on the socket's state a user can perceive and/or manipulate and commands that include all of the socket's functions that can be called explicitly or implicitly by users and notifications that cover all exceptions that the Web service needs to inform the user about.

- The Web service shall provide one grouping resource for every socket through external grouping sheets.
- The Web service shall provide textual label resources through external resource sheets, in at least one natural language.
- The Web service shall provide dynamic atomic resources at runtime for those socket elements where no (static) atomic resources are available in the target resources, as specified in 7.6.21.5, 7.6.22.5, and 7.6.23.6.
- The Web service, if representing a session-full target, shall support an open session request from a URC, as specified in 7.6.15.
- The Web service, if representing a session-full target, shall support a close session event from a URC, as specified in 7.6.16.
- The Web service, if representing a session-full target, shall support a suspend session event from a URC, as specified in 7.6.17.
- The Web service, if representing a session-full target, shall support a resume session event from a URC, as specified in 7.6.18.
- The Web service, if representing a session-full target, shall send an abort session event in case of user session abortion, as specified in 7.6.14.
- The Web service shall track connection status information from the underlying network its operations are bound to.
- The Web service, if representing a session-full target, shall send a session forward event to the URC in case of session forwarding, as specified in 7.6.14.
- The Web service, if representing a session-full target, shall create and maintain a session between a socket and the URC after a successful open session request.

- The Web service shall indicate to the URC the availability of socket elements at runtime (unavailable socket elements have an undefined value).
- The Web service shall synchronize the socket variables between the socket and the URCs that participate in a joint session with the socket (by means of the get-updates operation and the get operations of the variables).
- The Web service shall support command invocation requests from a URC (including handling of local parameters) and synchronization of command states (by means of the command operations).
- The Web service shall support propagation of notification states and, for custom-type notifications, embedded variables and commands, to the connected URCs, and acceptance of pertinent acknowledgments (by means of the get-updates operation and the check operations).
- The Web service shall synchronize actual indices of socket sets and elements (by means of the get-index operations).
- The Web service shall not rely on the URC doing the interpretation of socket element dependencies.
- Provide the following mechanisms with regard to user response timeouts:
  a) after a timeout extension, return to the state of the task the user had reached prior to the timeout;
  b) support the extend-timeout operation (see 7.6.23.4) for notifications that time out and let the client extend the timeout at least to five times the default timeout;
  c) note time out notifications in less than 10 s.

3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 24752-1, *Information technology — User interfaces — Universal remote console — Part 1: Framework*

ISO/IEC 24752-2:2013, *Information technology — User interfaces — Universal remote console — Part 2: User interface socket description*

ISO/IEC 24752-4:2013, *Information technology — User interfaces — Universal remote console — Part 4: Target description*

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 24752-1, ISO/IEC 24752-2 and ISO/IEC 24752-4 and the following apply:

4.1
fault item
named fault entity of a Web service operation, i.e. in WSDL1 a <message> referenced from a <fault> element of an <operation>, and in WSDL2 an <outfault> element of an <operation> referencing an interface fault

4.2
input item
named entity of input for a Web service operation, i.e. in WSDL1 a <part> of a <message> referenced from an <input> element of an <operation>; and in WSDL2 an <input> element of an <operation>

4.3
item element name
XML element name for an input or output item, i.e. in WSDL1 the value of the ‘element’ attribute on the pertaining <part> message; and in WSDL2 the value of the ‘element’ attribute on the <input> message

4.4
partition
Web service partition
named set of a Web service operations (“port type” in WSDL1, “interface” in WSDL2)

4.5
output item
named entity of output from a Web service operation, i.e. in WSDL1 a <part> of a <message> referenced from an <output> element of an <operation>, and in WSDL2 an <output> element of an <operation>

4.6
session-full socket
socket of a session-full target

4.7
session-less socket
socket of a session-less target

4.8
WSDL1 document
document that conforms to the Web Service Description Language (WSDL) 1.1 specification

4.9
WSDL2 document
document that conforms to the Web Service Description Language (WSDL) 2.0 specification

5 Relation to other standards

5.1 Relation to XML

This specification builds upon the extensible Markup Language (XML). Markup in XML is case sensitive.

Tag names, and attribute names and values are not localizable, i.e. they are identical for all international languages. However, the text content between tags can be language specific. As with all XML based languages, white space characters immediately surrounding tags are non-significant.

This specification makes use of the XML namespaces concept to enable the import of element and attribute names defined elsewhere.

Throughout this document, the following namespace prefixes and corresponding namespace identifiers are used for referencing namespaces. Authors are not bound to these prefixes, though their usage is recommended for better readability of public documents conforming to this International Standard.

- dc: The Dublin Core Metadata Element Set namespace ("http://purl.org/dc/elements/1.1/") (Element Set defined by ISO 15836);
- dcterms: The DCMI Metadata Terms namespace ("http://purl.org/dc/terms");
- td: The target description namespace ("http://openurc.org/ns/targetdesc-2");
- uis: The user interface socket description namespace ("http://openurc.org/ns/uisocketdesc-2");
- wsdl-urc: The namespace for extending a WSDL1 or WSDL2 document by embedding an implicit target description and implicit socket descriptions ("http://openurc.org/ns/wsdl-urc");
- xs: The XML Schema namespace ("http://www.w3.org/2001/XMLSchema");
- xsi: The XML Schema Instance namespace ("http://www.w3.org/2001/XMLSchema-instance").

6 Mapping Descriptions

6.1 General

A user interface socket (short "socket") consists of variables, commands, notifications, sets and type definitions. A Web service interface, as described by WSDL1, consists of port types, operations, messages, message parts, and type definitions. A Web service interface, as described by WSDL2, consists of interfaces, operations, messages, and type definitions.

A mapping description consists of the following parts: a mapping of a target and its properties to a Web service and its properties, a mapping of each of the target's sockets, its sets and its elements to one of the Web service's partition and its elements.

NOTE 1 A Web service partition is a functional unit of a Web service. In WSDL1, this is named a port type (element <portType>), in WSDL2 an interface (element <interface>).

NOTE 2 This section specifies general (semantic) requirements for mapping descriptions. The following section specifies a concrete syntax for mapping descriptions.

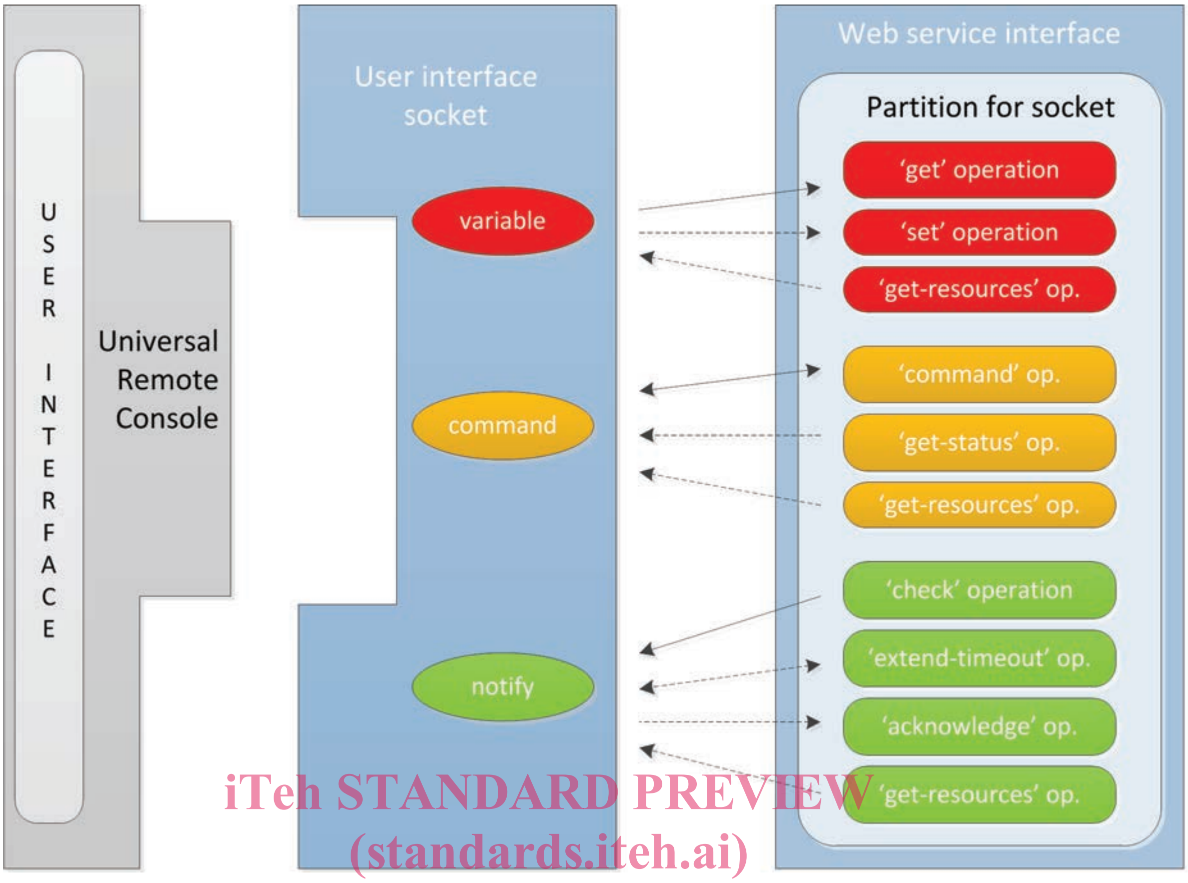

Figure 1 — Schematic mapping of a socket variable, a socket command, and a socket notify element to get, set, get-resources, command, get-status, check, extend-timeout and acknowledge operations provided by a partition of a Web service interface. The arrows indicate flow of socket content. Dashed arrows denote optional operations. Note that the Universal Remote Console (depicted on the left in gray) is included to provide contextual information for the socket, but is out of scope for this International Standard. (See ISO/IEC 24752-1 for more information on the Universal Remote Console.)

NOTE 3 In the following subsections, mappings are introduced for targets, sockets, variables, commands, notifications and type definitions. However, mappings for sets are implicitly contained in mappings for variables, commands and notifications (reflecting the structure of the socket).

6.2 Mapping a target to a Web service

The mapping description shall map exactly one target to exactly one Web service. The target shall be specified by its name (URI), and the Web service by its target namespace.

6.3 Mapping a socket to a Web service partition

A mapping description shall map the sockets of a mapped target to the partitions of a Web service that is mapped to the target (as specified in 6.2). Each socket included in the mapping shall be mapped to a single partition. The sockets shall be specified by their names (URIs), and the Web service partitions by their names.

6.4 Mapping a socket variable

6.4.1 General

A socket variable shall be mapped to the following Web service operations:

- a get operation,
- a set operation (only for writeable variables), and
- a get-resources operation (optional).

6.4.2 The get operation

A mapping description for a socket variable shall specify a get operation of a Web service with no input item and a single output item. At runtime, the get operation shall provide the current value of the socket variable through the output item. The get operation shall not change the state of the Web service. The types of the socket variable and the output item of the get operation shall be compatible. Time interval and expiration time for polling should be defined for each mapping mechanism.

6.4.3 The set operation

Unless the write dependency of the socket variable is always false (i.e. the variable is read-only), the mapping description shall also specify a set operation of a Web service with a single input item. The types of the socket variable and the input item of the set operation shall be compatible. At runtime, the set operation provides a way to modify the Web service's state. The input item shall reflect the requested new value for the variable. The output item shall carry the actual value of the variable after the operation has finished (i.e. the newly assigned value if the operation succeeded, or the old value if the operation failed).

6.4.4 The get-resources operation

The mapping description for a socket variable may specify a get-resources operation, if the Web service wants to provide dynamic atomic resources for the socket variable. If present, the get-resources operation shall have no input item and one output item with the element <wsdl-urc:resItems>, carrying a (possibly empty) set of dynamic atomic resource descriptions, each represented by an element <wsdl-urc:aResDesc> with the following subelements:

- either <wsdl-urc:content> (of type xs:string) or <wsdl-urc:contentAt> (of type xs:anyURI)
- <dc:type> (optional) - with any of the following string values: "Collection", "Dataset", "Event", "Image", "Interactive Resource", "Moving Image", "Physical Object", "Service", "Software", "Sound", "Still Image", "Text"
- <dc:format> (optional) - a valid MIME type
- <wsdl-urc:valRef> (optional) - of type xs:string
- <wsdl-urc:opRef> (optional) - with any of the following URIs: "http://openurc.org/ns/res#up", "http://openurc.org/ns/res#down"
- <wsdl-urc:role> - with any of the following URIs: "http://openurc.org/ns/res#label", "http://openurc.org/ns/res#help", "http://openurc.org/ns/res#accesskey", "http://openurc.org/ns/res#keyword", "http://openurc.org/ns/res#location"
- <wsdl-urc:forLang> (optional) - of type xs:language

These subelements correspond to the equally named properties of an atomic resource description in a resource sheet (see ISO/IEC 24752-5). However, as a restriction, the <wsdl-urc:content> element shall only contain text content, and shall not contain any subelements (such as <title>, <span> or <value>).

EXAMPLE The socket variable “deviceStatus” is connected to the following dynamic atomic resources: (1) A text label for the current status value, and (2) an icon for the current status value. The current internal value of the “status” variable is “2” (which means standby mode, but this is not known before runtime). The output item of the get-resources operation would then assume the following XML fragment, provided that the default namespace is WSDL-URC (“http://openurc.org/ns/wsdl-urc”), and the namespace prefix “dc” is mapped to “http://purl.org/dc/elements/1.1/”:

```
<resItems>
  <aResDesc>
    <content>standby</content>
    <dc:type>Text</dc:type>
    <dc:format>text</dc:format>
    <valRef>2</valRef>
    <role>http://openurc.org/ns/res#label</role>
    <forLang>en</forLang>
  </aResDesc>
  <aResDesc>
    <contentAt>http://example.com/deviceXY/standby.png</contentAt>
    <dc:type>Image</dc:type>
    <dc:format>image/png</dc:format>
    <valRef>2</valRef>
    <role>http://openurc.org/ns/res#label</role>
  </aResDesc>
</resItems>
```

NOTE By calling the get-resources operation, a client can retrieve a new dynamic resource (label, help text, access key, or location) for the socket variable or for any of its values. Note that this simple mechanism is restricted in that it does not allow for retrieving dynamic resources for types and type values rather than for socket elements and their values. Also, it does not allow for structuring textual labels, or integrating current values of socket elements.

6.5 Mapping a socket command

6.5.1 General

A socket command shall be mapped to the following Web service operations:

- a command operation,
- a get-status operation (only for commands of type other than ‘uis:voidCommand’), and
- a get-resources operation (optional).

6.5.2 The command operation

A mapping description for a command shall specify a single command operation of a Web service. At runtime, the command operation is executed when the command is activated.

6.5.3 The get-status operation

A mapping description for a command may specify a single get-status operation of a Web service, if the command is executed synchronously, i.e. if it has a command type other than uis:voidCommand. At runtime, the URC can call the get-status operation to receive information on the command’s status. Possible status values are: “initial”, “rejected”, “inProgress”, “done”, “succeeded”, and “failed” (see ISO/IEC 24752-2).

6.5.4 The get-resources operation

The mapping description for a command may specify a get-resources operation, if the Web service wants to provide dynamic atomic resources for the socket command. If present, the get-resources operation shall have the same input and output items as for the get-resources operation for a socket variable (see 6.4.4).

6.6 Mapping a socket notification

6.6.1 General

A socket notification shall be mapped to the following Web service operations:

- a check operation,
- an extend-timeout operation (optional),
- an acknowledge operation (only for notifications of type other than “show”), and
- a get-resources operation (optional).

6.6.2 The check operation

A mapping description for a socket notification shall specify a check operation. At runtime, the check operation is frequently called (“polled”) for capturing the status of the notification.

6.6.3 The extend-timeout operation

A mapping description for a socket notification shall specify an extend-timeout operation if the notification has a ‘timeout’ attribute (see ISO/IEC 24752-2). Otherwise, it may specify an extend-timeout operation. At runtime, the extend-timeout operation can be called by the client to request an extension of the notification timeout.

6.6.4 The acknowledge operation

A mapping description for a socket notification shall specify an acknowledge operation of a Web service, if the notification has a type other than “show”. Upon user acknowledgment, the acknowledge operation is called.

6.6.5 The get-resources operation

The mapping description for a socket notification may specify a get-resources operation, if the Web service wants to provide dynamic atomic resources for the socket notification. If present, the get-resources operation shall have the same input and output items as for the get-resources operation for a socket variable (see 6.4.4).

6.7 Mapping a socket-internal type definition to a Web service’s type definition

A socket-internal type definition shall be mapped to a type definition in a Web service’s interface description.

NOTE 1 A socket-internal type definition is expressed by a <simpleType> or <complexType> element in the <xs:schema> part of a socket description (see ISO/IEC 24752-2).

NOTE 2 In a WSDL (both WSDL1 and WSDL2) document, a type definition is expressed by a <simpleType> or <complexType> element, contained in an <xs:schema> element in the <types> part.

7 Embedding target description and socket descriptions in a WSDL document

7.1 General

This section specifies syntax and conventions for providing mapping descriptions (see 6) inside a WSDL document, hereby providing an implicit target description and implicit socket descriptions. This approach is applicable to WSDL1 and WSDL2 documents, and is based on the generic mapping rules (as outlined in 6), naming conventions and some additional markup inside the <documentation> element in a WSDL file.

NOTE 1 In some cases, the syntax differs between WSDL1 and WSDL2. In these cases both versions are specified.

NOTE 2 When using the embedded approach, the structure of a target and its sockets is publicly available with the WSDL document. This means that the availability of the socket description cannot be made dependent on a user's opening a session (with proper credentials).

7.2 Restriction on target namespaces of internal schema part

A WSDL document shall define types in the same namespace as the target namespace of the WSDL document, unless types are imported from an external namespace via the <xs:import> directive.

EXAMPLE The target namespace of the type definition part is the same as the target namespace of the overall WSDL document (WSDL1). WSDL2 format accordingly. Ellipses ("…") indicate omissions.

```
<definitions targetNamespace="http://openurc.org/TPL/basic-thermostat-1.0/" ...>
  …
  <types>
     <xs:schema targetNamespace="http://openurc.org/TPL/basic-thermostat-1.0/">
       ...
     </xs:schema>
  </types>
  ...
</definitions>
```

7.3 Restriction on identifiers within target and socket descriptions

The values of 'id' attributes on XML elements in a target description or socket description that is implicitly mapped to a WSDL1 or WSDL2 document shall be restricted as follows:

— They shall not begin with an underscore character ('_').

— They shall not contain a dot character ('.').

— They shall not contain a hyphen character ('-').

NOTE Element names starting with underscore are reserved for internal use (e.g. the timeToComplete field for a timed command). Dots are not allowed since they are used in 'name' attributes inside WSDL documents as delimiters between path components for socket elements. Hyphens are not allowed since they are used as delimiters between notifications and their contained variables and commands.

7.4 'name' attribute values

A WSDL document contains 'name' attributes on various elements to identify an implicit target description and an implicit user interface socket description, and their components (see the following sections).

The values of these 'name' attributes shall be unique within the WSDL document, across all types of elements.

NOTE This is stricter than WSDL that requires uniqueness of 'name' attribute values within the same type of elements only.