

Integrity Lifecycle Manager Web Services Reference

This document provides reference information for Web services on the Integrity Lifecycle Manager server. Integrity Web services use the SOAP messaging protocol and are defined using the "wrapped" pattern style of the document/literal messaging style. In order to minimize memory requirements, Integrity Lifecycle Manager Web services use the SOAP Message Transmission Optimization Method (MTOM).

Integrity Lifecycle Manager web services support the following operations.

- Creating, viewing and manipulating items:
 - [Integrity 10.2 Web services](#)
 - [Integrity 10 Web services](#)
 - [Integrity 2009 Web services](#)
- Creating, viewing and manipulating members and projects:
 - [Integrity 2009 SP7 Web services](#)
 - [Integrity 2009 Web services](#)
- [Creating, viewing and manipulating documents \(Integrity 10 Web services\)](#)
- [Creating, viewing and manipulating test results \(Integrity 10 Web services\)](#)

For details of the differences between the 2009 Web services and the 2009 SP7 Web services, see the Integrity 2009 SP7 release notes.

Integrity Lifecycle Manager Web services also support using states and labels for members.

- [Setting states and labels for members \(Integrity 2009 SP7 Web services\)](#)

Note: This Web service also contains an operation for retrieving the URL to use when fetching the contents of a member.

Configuration management states allow members to evolve through various states defined by your administrator. The use of configuration management states is deprecated in Integrity Lifecycle Manager, but is still supported through Web services. PTC recommends using workflow to provide state transitions through the use of change packages.

In addition, there is a generic Integrity Lifecycle Manager Web service for operations that do not have a specific Web service.

- [Generic command that can be used for any Integrity operation \(Integrity 2009 Web services\)](#)

Generic Web services send response information using the same structure as the Integrity Lifecycle Manager API responses. For more information on API responses, see the *Integrity Lifecycle Manager Integrations Builder Guide*.

Using Project Configuration Path Information in Web Services

A development path is a branch of software development where changes are kept separate from the main development trunk. If you are using Web services to interact with projects or members on a development path, use the `getProjectConfigurationPath` Web service to request the development path to use in any method where project is specified as an input argument.

A build project is a static project based on a specific project checkpoint. If you are using Web services to interact with a build project, use the `getProjectConfigurationPath` Web service to request a specific version of a project and use it in any method where project is specified as an input argument.

Web Service Session Handling

- Web service session handling is provided automatically by the Integrity Lifecycle Manager server. The Web service session uses a concurrent license, which is tracked through FlexNet.
- The session that is created by the initial Web service request is reused for all subsequent requests to the server.
- You do not need to disconnect from Web services since sessions time out automatically based on the Web client timeout property defined on the server.

Availability of Web Services

- Web services are not accessible from an FSA server.
- Web services for creating, viewing and manipulating items are not available on a standalone configuration management server.
- Web services for creating, viewing and manipulating members and projects are not available on a standalone workflows and documents

server.

- The generic Web services is available on a standalone configuration management server or a standalone workflows and documents server.

Web Services Configuration Properties

In the database used by the Integrity Lifecycle Manager server, the following Web Services configuration properties affect the management of Web Services temporary files:

- `mksis.webservices.tmpfile.enable` controls whether or not the Web Services temporary file deletion thread is enabled. The default value is `true`, allowing the deletion of temporary files.
- `mksis.webservices.tmpfile.ageThreshold` determines how old (in minutes) the Web Services temporary files can be before being deleted. The default value is 1440 (or 24 hours), allowing the deletion of temporary Web Services files that are 24 hours old.
- `mksis.webservices.tmpfile.repeatDelay` determines the frequency (in minutes) for running the Web Services temporary file deletion thread. The default value is 120 (or 2 hours), allowing the deletion thread to run every 2 hours.

For detailed information on setting Integrity Lifecycle Manager server properties, see the *Integrity Lifecycle Manager Online Help*.

Important Copyright Information

Copyright © 2017 PTC Inc. All Rights Reserved.

This reference guide and the source code provided herewith are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of this documentation and the source code provided herewith. PTC hereby grants to the licensed software user the right to make copies of this documentation and to use the source code solely for the licensed software user's internal use and in accordance with the license agreement under which the applicable software is licensed. Any copy of this documentation shall include this PTC copyright notice. This documentation and the source code may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes. The information described herein is furnished for general information only and is subject to change without notice. No warranty is made with respect to source code provided herewith and PTC expressly disclaims any warranty other than any required by applicable law. PTC shall not be liable for any technical or editorial errors or omissions contained herein or in the source code provided herewith. For additional important information about PTC copyright, trademark, patent, and licensing, see the About Box, or copyright notice, of your PTC software.

PTC Inc. • 140 Kendrick Street • Needham, MA • 02494 • USA

Web Service: Integrity_10_2Service

Web Service: Integrity_10_2Service

Target Namespace: <http://webservice.mks.com/10/2/Integrity>

Port **Integrity_10_2Port** [Port type](#) [Source code](#)

Location: <http://IntegrityServer:9001/webservices/10/2/Integrity/>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. [createItem](#) [Detail](#) [Source code](#)
2. [editItem](#) [Detail](#) [Source code](#)
3. [fetchAttachments](#) [Detail](#) [Source code](#)
4. [getAttachmentDetails](#) [Detail](#) [Source code](#)
5. [getItem](#) [Detail](#) [Source code](#)
6. [getItemsByCustomQuery](#) [Detail](#) [Source code](#)
7. [getItemsByIDs](#) [Detail](#) [Source code](#)
8. [getItemsByNamedQuery](#) [Detail](#) [Source code](#)
9. [getProjects](#) [Detail](#) [Source code](#)
10. [removeAttachment](#) [Detail](#) [Source code](#)

Operations

Port type **Integrity_10_2** [Source code](#)

1. [createItem](#)

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_createItem (soap:body, use = literal) [Source code](#)

creatItem type [creatItem](#)

Output: Integrity_10_2_createItemResponse (soap:body, use = literal) [Source code](#)

creatItemResponse type [creatItemResponse](#)

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type [MKSEException](#)

- o class type [string](#)
- o id type [string](#)
- o implication type [string](#)

2. editItem

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_editItem (soap:body, use = literal) [Source code](#)

editItem type *editItem*

Output: Integrity_10_2_editItemResponse (soap:body, use = literal) [Source code](#)

editItemResponse type *editItemResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

3. fetchAttachments

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_fetchAttachments (soap:body, use = literal) [Source code](#)

fetchAttachments type *fetchAttachments*

Output: Integrity_10_2_fetchAttachmentsResponse (soap:body, use = literal) [Source code](#)

fetchAttachmentsResponse type *fetchAttachmentsResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

4. getAttachmentDetails

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_getAttachmentDetails (soap:body, use = literal) [Source code](#)

getAttachmentDetails type *getAttachmentDetails*

Output: Integrity_10_2_getAttachmentDetailsResponse (soap:body, use = literal) [Source code](#)

getAttachmentDetailsResponse type *getAttachmentDetailsResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

5. getItem

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_getItem (soap:body, use = literal) [Source code](#)

getItem type *getItem*

Output: Integrity_10_2_getItemResponse (soap:body, use = literal) [Source code](#)

getItemResponse type *getItemResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

6. getItemsByCustomQuery

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_getItemsByCustomQuery (soap:body, use = literal) [Source code](#)

getItemsByCustomQuery type *getItemsByCustomQuery*

Output: Integrity_10_2_getItemsByCustomQueryResponse (soap:body, use = literal) [Source code](#)

getItemsByCustomQueryResponse type *getItemsByCustomQueryResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

7. getItemsByIDs

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_getItemsByIDs (soap:body, use = literal) [Source code](#)

getItemsByIDs type *getItemsByIDs*

Output: Integrity_10_2_getItemsByIDsResponse (soap:body, use = literal) [Source code](#)

getItemsByIDsResponse type *getItemsByIDsResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

8. getItemsByNamedQuery

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_getItemsByNamedQuery (soap:body, use = literal) [Source code](#)

getItemsByNamedQuery type *getItemsByNamedQuery*

Output: Integrity_10_2_getItemsByNamedQueryResponse (soap:body, use = literal) [Source code](#)

getItemsByNamedQueryResponse type *getItemsByNamedQueryResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

9. getProjects

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_getProjects (soap:body, use = literal) [Source code](#)

getProjects type *getProjects*

Output: Integrity_10_2_getProjectsResponse (soap:body, use = literal) [Source code](#)

getProjectsResponse type *getProjectsResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

10. removeAttachment

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_2_removeAttachment (soap:body, use = literal) [Source code](#)

removeAttachment type *removeAttachment*

Output: Integrity_10_2_removeAttachmentResponse (soap:body, use = literal) [Source code](#)

removeAttachmentResponse type *removeAttachmentResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

WSDL source code

```
<?xml version="1.0"?>
<definitions name="Integrity_10_2Service" targetNamespace="http://webservice.mks.com/10/2/Integrity"
  xmlns:ns3="http://www.w3.org/2005/05/xmlmime"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://webservice.mks.com/10/2/Integrity/schema"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ns2="http://webservice.mks.com/10/2/Integrity/fault"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://webservice.mks.com/10/2/Integrity"
>

<types>
  <xsschema elementFormDefault="qualified"
    targetNamespace="http://webservice.mks.com/10/2/Integrity/schema" version="1.0">
    <xsi:import namespace="http://www.w3.org/2005/05/xmlmime" schemaLocation="xmlmime.xsd" />
    <xss:element name="AttachmentChecksum">
      <xss:complexType>
        <xss:sequence>
          <xss:element name="Field" type="xss:string" />
          <xss:element name="FileName" type="xss:string" />
          <xss:element name="FileSize" type="xss:integer" />
          <xss:element name="Checksum" type="xss:string" />
          <xss:element name="ChecksumAlgorithm" type="xss:string" />
        </xss:sequence>
        <xss:attribute name="status" type="xss:string" />
      </xss:complexType>
    </xss:element>
    <xss:element name="Fields">
      <xss:complexType>
        <xss:sequence>
          <xss:element maxOccurs="unbounded" minOccurs="0" name="Field">
            <xss:complexType>
              <xss:sequence />
              <xss:attribute name="Name" type="xss:string" use="required" />
              <xss:attribute name="Value" type="xss:string" />
            </xss:complexType>
          </xss:element>
        </xss:sequence>
      </xss:complexType>
    </xss:element>
    <xss:complexType name="ItemFieldRequest">
      <xss:complexContent>
        <xss:extension base="tns:ItemFieldBase">
          <xss:sequence>
            <xss:element minOccurs="0" name="user" type="tns:NullableString" />
          </xss:sequence>
        </xss:extension>
      </xss:complexContent>
    </xss:complexType>
  </xsschema>
</types>
```

```

</xs:complexContent>
</xs:complexType>
<xs:complexType name="ItemFieldResponse">
    <xs:complexContent>
        <xs:extension base="tns:ItemFieldBase">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
                            name="UserRecord" type="tns:UserRecord" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="UserRecord">
    <xs:sequence>
        <xs:element minOccurs="0" name="user" type="tns:NullableString" />
        <xs:element minOccurs="0" name="fullname" type="tns:NullableString" />
        <xs:element minOccurs="0" name="email" type="tns:NullableString" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ItemFieldBase">
    <xs:choice minOccurs="0">
        <xs:element name="shorttext" type="tns:NullableShortString" />
        <xs:element name="longtext" type="tns:NullableString" />
        <xs:element name="group" type="tns:NullableString" />
        <xs:element name="state" type="xs:string" />
        <xs:element name="type" type="xs:string" />
        <xs:element name="project" type="tns:NullableString" />
        <xs:element name="siProject" type="tns:NullableString" />
        <xs:element name="double" type="tns:NullableDouble" />
        <xs:element name="boolean" type="tns:NullableBoolean" />
        <xs:element name="integer" type="tns:NullableInteger" />
        <xs:element name="pick">
            <xs:complexType>
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0"
                                name="value" nillable="true" type="xs:string" />
                </xs:sequence>
                <xs:attribute name="null" type="xs:boolean" />
            </xs:complexType>
        </xs:element>
        <xs:element name="relationship">
            <xs:complexType>
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0"
                                name="value" nillable="true" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="date" type="tns:NullableString">
            <xs:annotation>

```

<xs:documentation>The value for the date field - that is, a date having no time component. The format may be specified in the DateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 If the format is not specified in the request, the default is "yyyy-MM-dd". Be careful to not send a date when a timestamp is required. For getItem and createItem, if the Integrity field is a timestamp but this field is used to send the information, the end user will get an error message that recommends the wrong formatting. This error message comes from the Integrity API and should be ignored since it is a symptom of the problem, not the root cause.**</xs:documentation>**

```

        </xs:annotation>
    </xs:element>
    <xs:element name="dateTime" type="tns:NullableString">
        <xs:annotation>
            <xs:documentation>The value for the timestamp field - that is, a date having a time component. The format may be specified in the DateFormat entity in the request, following the conventions outlined at this url:  

http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html  

    If the format is not specified in the request, the default is "yyyy-MM-dd HH:mm:ss". Be careful to not send a timestamp when a date is required. For getItem and createItem, if the Integrity field is a date but this field is used to send the information, the end user will get an error message that recommends the wrong formatting. This error message comes from the Integrity API and should be ignored since it is a symptom of the problem, not the root cause.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:choice>
    <xs:attribute name="Name" type="xs:string" use="required" />
</xs:complexType>
<xs:element name="Project">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" ref="tns:Fields" />
        </xs:sequence>
        <xs:attribute name="Name" type="xs:string" />
    <xs:complexType>
</xs:element>
<xs:element name="RelatedItem">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="Item" type="tns:Relationship" />
            <xs:element maxOccurs="unbounded" minOccurs="0" name="IBPL" type="xs:integer" />
        </xs:sequence>
            <xs:attribute name="FieldName" type="xs:string" use="required" />
        <xs:complexType>
</xs:element>
<xs:complexType name="GetItemType">
    <xs:complexContent>

```

```

<xs:extension base="tns:.GetItemCommon">
    <xs:sequence />
    <xs:attribute form="qualified" name="ItemId" type="xs:integer"
        use="required" />
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="GetItemCommon">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="InputField" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Credentials">
    <xs:sequence>
        <xs:element name="Username" type="xs:string" />
        <xs:element name="Password" type="xs:string" />
        <xs:element minOccurs="0" name="ImpersonatedUser" type="xs:string" />
        <xs:element minOccurs="0" name="DateFormat" type="xs:string" />
        <xs:element minOccurs="0" name="DateTimeFormat" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string" />
</xs:complexType>
<xs:complexType name="Item">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="ItemField" type="tns:ItemFieldResponse" />
                <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="Attachment" type="tns:AttachmentInfo" />
                <xs:element maxOccurs="unbounded" minOccurs="0"
                    ref="tns:RelatedItem" />
            </xs:sequence>
            <xs:attribute name="State" type="xs:string" />
            <xs:attribute name="Type" type="xs:string" />
            <xs:attribute form="qualified" name="ItemId" type="xs:integer" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="StatusResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string" />
</xs:complexType>

```

```

<xs:complexType name="NullableShortString">
    <xs:choice>
        <xs:element name="value" type="xs:string" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableString">
    <xs:choice>
        <xs:element name="value" type="xs:string" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableDouble">
    <xs:choice>
        <xs:element name="value" type="xs:double" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableBoolean">
    <xs:choice>
        <xs:element name="value" type="xs:boolean" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableInteger">
    <xs:choice>
        <xs:element name="value" type="xs:int" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="AttachmentInfo">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element name="Name" type="xs:string" />
                <xs:element name="Size" type="xs:int" />
                <xs:element name="Checksum" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Relationship">
    <xs:simpleContent>
        <xs:extension base="xs:integer">
            <xs:attribute name="flag" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="AttachmentRequest">

```

```

<xs:complexContent>
  <xs:extension base="tns:Credentials">
    <xs:sequence />
    <xs:attribute name="AttachmentName" type="xs:string" />
    <xs:attribute name="FieldName" type="xs:string" />
    <xs:attribute form="qualified" name="ItemId" type="xs:integer"
      use="required" />
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="AttachmentContents">
  <xs:complexContent>
    <xs:extension base="tns>StatusResponse">
      <xs:sequence>
        <xs:element minOccurs="0" name="Attachment"
          ns3:expectedContentTypes="application/octet-stream"
          type="xs:base64Binary" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectRequest">
  <xs:complexContent>
    <xs:extension base="tns:Credentials">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0"
          name="FieldList" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectList">
  <xs:complexContent>
    <xs:extension base="tns>StatusResponse">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0"
          ref="tns:Project" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateItemType">
  <xs:complexContent>
    <xs:extension base="tns>CreateEdit">
      <xs:sequence />
      <xs:attribute name="Type" type="xs:string" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateEdit">
  <xs:complexContent>

```

```

<xs:extension base="tns:Credentials">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0"
      name="ItemField" type="tns:ItemFieldRequest" />
    <xs:element maxOccurs="unbounded" minOccurs="0"
      name="Attachment" type="tns:AttachmentEdit" />
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="AttachmentEdit">
  <xs:sequence>
    <xs:element name="Name" type="xs:string" />
    <xs:element name="Field" type="xs:string" />
    <xs:element minOccurs="0" name="Description" type="xs:string" />
    <xs:element name="Attachment" ns3:expectedContentTypes="application/octet-
      stream" type="xs:base64Binary" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ItemResponse">
  <xs:complexContent>
    <xs:extension base="tns>StatusResponse">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0"
          ref="tns:AttachmentChecksum" />
      </xs:sequence>
      <xs:attribute form="qualified" name="ItemId" type="xs:integer"
        use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="EditItemType">
  <xs:complexContent>
    <xs:extension base="tns>CreateEdit">
      <xs:sequence />
      <xs:attribute form="qualified" name="ItemId" type="xs:integer" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AttachmentDetails">
  <xs:complexContent>
    <xs:extension base="tns:AttachmentInfo">
      <xs:sequence>
        <xs:element name="ContentType" type="xs:string" />
        <xs:element name="Description" type="xs:string" />
        <xs:element name="AddedBy" type="xs:string" />
        <xs:element name="AddedDate" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

<xs:complexType name="CustomQuery">
    <xs:complexContent>
        <xs:extension base="tns:.GetItemCommon">
            <xs:sequence>
                <xs:element name="QueryDefinition" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ItemList">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Item" type="tns:Item" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetItemsByIDsType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="InputField" type="xs:string" />
                <xs:element maxOccurs="unbounded" name="ItemId" type="xs:integer" />
            </xs:sequence>
            <xs:attribute name="Name" type="xs:string" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="NamedQuery">
    <xs:complexContent>
        <xs:extension base="tns:.GetItemCommon">
            <xs:sequence>
                <xs:element name="QueryName" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:attribute name="ItemId" type="xs:integer" />
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/10/2/Integrity/fault" version="1.0">
    <xs:element name="MKSEception" nillable="true" type="tns:MKSFault" />
    <xs:complexType name="MKSFault">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="class" type="xs:string" />
                <xs:attribute name="id" type="xs:string" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>

```

```

                <xs:attribute name="implication" type="xs:string" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>

<xs:schema targetNamespace="http://webservice.mks.com/10/2/Integrity" version="1.0">
    <xs:import namespace="http://webservice.mks.com/10/2/Integrity/schema" />
    <xs:element name="createItem" type="tns:createItem" />
    <xs:element name="createItemResponse" type="tns:createItemResponse" />
    <xs:element name="editItem" type="tns:editItem" />
    <xs:element name="editItemResponse" type="tns:editItemResponse" />
    <xs:element name="fetchAttachments" type="tns:fetchAttachments" />
    <xs:element name="fetchAttachmentsResponse" type="tns:fetchAttachmentsResponse" />
    <xs:element name="getAttachmentDetails" type="tns:getAttachmentDetails" />
    <xs:element name="getAttachmentDetailsResponse" type="tns:getAttachmentDetailsResponse" />
    <xs:element name="getItem" type="tns:getItem" />
    <xs:element name="getItemResponse" type="tns:getItemResponse" />
    <xs:element name="getItemsByCustomQuery" type="tns:getItemsByCustomQuery" />
    <xs:element name="getItemsByCustomQueryResponse"
    type="tns:getItemsByCustomQueryResponse" />
    <xs:element name="getItemsByIds" type="tns:getItemsByIds" />
    <xs:element name="getItemsByIdsResponse" type="tns:getItemsByIdsResponse" />
    <xs:element name="getItemsByNamedQuery" type="tns:getItemsByNamedQuery" />
    <xs:element name="getItemsByNamedQueryResponse"
    type="tns:getItemsByNamedQueryResponse" />
    <xs:element name="getProjects" type="tns:getProjects" />
    <xs:element name="getProjectsResponse" type="tns:getProjectsResponse" />
    <xs:element name="removeAttachment" type="tns:removeAttachment" />
    <xs:element name="removeAttachmentResponse" type="tns:removeAttachmentResponse" />
<xs:complexType name="getItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ItemType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:item" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchAttachments">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AttachmentRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchAttachmentsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AttachmentContents" />
    </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="getProjects">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ProjectRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjectsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ProjectList" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CreateItem" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createItemResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="editItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>EditItem" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="editItemResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getAttachmentDetails">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AttachmentRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getAttachmentDetailsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AttachmentDetails" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeAttachment">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AttachmentRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeAttachmentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AttachmentInfo" />
    </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="getItemsByCustomQuery">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CustomQuery" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByCustomQueryResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemList" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByIds">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetItemsByIdsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByIdsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemList" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByNamedQuery">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:NamedQuery" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByNamedQueryResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemList" />
    </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="Integrity_10_2_getAttachmentDetails">
    <part element="tns:getAttachmentDetails" name="getAttachmentDetails" />
</message>
<message name="Integrity_10_2_removeAttachment">
    <part element="tns:removeAttachment" name="removeAttachment" />
</message>
<message name="Integrity_10_2_getItem">
    <part element="tns:getItem" name="getItem" />
</message>
<message name="Integrity_10_2_editItem">
    <part element="tns:editItem" name="editItem" />
</message>
<message name="Integrity_10_2_createItem">
    <part element="tns:createItem" name="createItem" />
</message>
<message name="Integrity_10_2_getProjectsResponse">
    <part element="tns:getProjectsResponse" name="getProjectsResponse" />
</message>

```

```

<message name="MKSEception">
    <part element="ns2:MKSEception" name="MKSEception" />
</message>
<message name="Integrity_10_2_removeAttachmentResponse">
    <part element="tns:removeAttachmentResponse" name="removeAttachmentResponse" />
</message>
<message name="Integrity_10_2_getItemsByCustomQueryResponse">
    <part element="tns:getItemsByCustomQueryResponse" name="getItemsByCustomQueryResponse" />
</message>
<message name="Integrity_10_2_getProjects">
    <part element="tns:getProjects" name="getProjects" />
</message>
<message name="Integrity_10_2_fetchAttachments">
    <part element="tns:fetchAttachments" name="fetchAttachments" />
</message>
<message name="Integrity_10_2_getItemsByNamedQueryResponse">
    <part element="tns:getItemsByNamedQueryResponse" name="getItemsByNamedQueryResponse" />
</message>
<message name="Integrity_10_2_getItemsByNamedQuery">
    <part element="tns:getItemsByNamedQuery" name="getItemsByNamedQuery" />
</message>
<message name="Integrity_10_2_editItemResponse">
    <part element="tns:editItemResponse" name="editItemResponse" />
</message>
<message name="Integrity_10_2_getItemsByIDsResponse">
    <part element="tns:getItemsByIDsResponse" name="getItemsByIDsResponse" />
</message>
<message name="Integrity_10_2_getItemsByIDs">
    <part element="tns:getItemsByIDs" name="getItemsByIDs" />
</message>
<message name="Integrity_10_2_createItemResponse">
    <part element="tns:createItemResponse" name="createItemResponse" />
</message>
<message name="Integrity_10_2_getItemsByCustomQuery">
    <part element="tns:getItemsByCustomQuery" name="getItemsByCustomQuery" />
</message>
<message name="Integrity_10_2_getItemResponse">
    <part element="tns:getItemResponse" name="getItemResponse" />
</message>
<message name="Integrity_10_2_fetchAttachmentsResponse">
    <part element="tns:fetchAttachmentsResponse" name="fetchAttachmentsResponse" />
</message>
<message name="Integrity_10_2_getAttachmentDetailsResponse">
    <part element="tns:getAttachmentDetailsResponse" name="getAttachmentDetailsResponse" />
</message>
<portType name="Integrity_10_2">
    <operation name="createItem" parameterOrder="createItem">
        <input message="tns:Integrity_10_2_createItem" />
        <output message="tns:Integrity_10_2_createItemResponse" />
    
```

```

<fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="editItem" parameterOrder="editItem">
    <input message="tns:Integrity_10_2_editItem" />
    <output message="tns:Integrity_10_2_editItemResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="fetchAttachments" parameterOrder="fetchAttachments">
    <input message="tns:Integrity_10_2_fetchAttachments" />
    <output message="tns:Integrity_10_2_fetchAttachmentsResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getAttachmentDetails" parameterOrder="getAttachmentDetails">
    <input message="tns:Integrity_10_2_getAttachmentDetails" />
    <output message="tns:Integrity_10_2_getAttachmentDetailsResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getItem" parameterOrder="getItem">
    <input message="tns:Integrity_10_2_getItem" />
    <output message="tns:Integrity_10_2_getItemResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getItemsByCustomQuery" parameterOrder="getItemsByCustomQuery">
    <input message="tns:Integrity_10_2_getItemsByCustomQuery" />
    <output message="tns:Integrity_10_2_getItemsByCustomQueryResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getItemsByIDs" parameterOrder="getItemsByIDs">
    <input message="tns:Integrity_10_2_getItemsByIDs" />
    <output message="tns:Integrity_10_2_getItemsByIDsResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getItemsByNamedQuery" parameterOrder="getItemsByNamedQuery">
    <input message="tns:Integrity_10_2_getItemsByNamedQuery" />
    <output message="tns:Integrity_10_2_getItemsByNamedQueryResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getProjects" parameterOrder="getProjects">
    <input message="tns:Integrity_10_2_getProjects" />
    <output message="tns:Integrity_10_2_getProjectsResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="removeAttachment" parameterOrder="removeAttachment">
    <input message="tns:Integrity_10_2_removeAttachment" />
    <output message="tns:Integrity_10_2_removeAttachmentResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
</portType>

```

```
<binding name="Integrity_10_2Binding" type="tns:Integrity_10_2">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsp:PolicyReference URI="#wsmtom_policy" required="true"
        xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    />
    <operation name="createItem">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEException">
            <soap:fault name="MKSEException" use="literal" />
        </fault>
    </operation>
    <operation name="editItem">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEException">
            <soap:fault name="MKSEException" use="literal" />
        </fault>
    </operation>
    <operation name="fetchAttachments">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEException">
            <soap:fault name="MKSEException" use="literal" />
        </fault>
    </operation>
    <operation name="getAttachmentDetails">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
```

```

<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="getItem">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getItemsByCustomQuery">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getItemsByIds">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getItemsByNamedQuery">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">

```

```

        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getProjects">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="removeAttachment">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
</binding>
<service name="Integrity_10_2Service">
    <port binding="tns:Integrity_10_2Binding" name="Integrity_10_2Port">
        <soap:address location="http://IntegrityServer:9001/webservices/10/2/Integrity/" />
    </port>
</service>
<wsp:Policy wsu:id="wsmtom_policy"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
>
    <wsoma:OptimizedMimeSerialization
        xmlns:wsoma="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization"
    />
</wsp:Policy>
</definitions>

```

Web Service: Integrity_10Service

Web Service: Integrity_10Service

Target Namespace: <http://webservice.mks.com/10/Integrity>

Port **Integrity_10Port** *Port type* [Source code](#)

Location: <http://IntegrityServer:9001/webservices/10/Integrity/>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. [createItem](#) [Detail](#) [Source code](#)
2. [editItem](#) [Detail](#) [Source code](#)
3. [fetchAttachments](#) [Detail](#) [Source code](#)
4. [getAttachmentDetails](#) [Detail](#) [Source code](#)
5. [getItem](#) [Detail](#) [Source code](#)
6. [getItemsByCustomQuery](#) [Detail](#) [Source code](#)
7. [getItemsByIDs](#) [Detail](#) [Source code](#)
8. [getItemsByNamedQuery](#) [Detail](#) [Source code](#)
9. [getProjects](#) [Detail](#) [Source code](#)
10. [removeAttachment](#) [Detail](#) [Source code](#)

Operations

Port type **Integrity_10** [Source code](#)

1. [createItem](#)

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_createItem (soap:body, use = literal) [Source code](#)

createItem type [createItem](#)

Output: Integrity_10_createItemResponse (soap:body, use = literal) [Source code](#)

createItemResponse type [createItemResponse](#)

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type [MKSEException](#)

- o class type [string](#)
- o id type [string](#)
- o implication type [string](#)

2. editItem

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_editItem (soap:body, use = literal) [Source code](#)

editItem type *editItem*

Output: Integrity_10_editItemResponse (soap:body, use = literal) [Source code](#)

editItemResponse type *editItemResponse*

Fault: MKSEception (soap:fault, use = literal) [Source code](#)

MKSEception type *MKSEception*

- class type *string*
- id type *string*
- implication type *string*

3. fetchAttachments

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_fetchAttachments (soap:body, use = literal) [Source code](#)

fetchAttachments type *fetchAttachments*

Output: Integrity_10_fetchAttachmentsResponse (soap:body, use = literal) [Source code](#)

fetchAttachmentsResponse type *fetchAttachmentsResponse*

Fault: MKSEception (soap:fault, use = literal) [Source code](#)

MKSEception type *MKSEception*

- class type *string*
- id type *string*
- implication type *string*

4. getAttachmentDetails

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_getAttachmentDetails (soap:body, use = literal) [Source code](#)

getAttachmentDetails type *getAttachmentDetails*

Output: Integrity_10_getAttachmentDetailsResponse (soap:body, use = literal) [Source code](#)

getAttachmentDetailsResponse type *getAttachmentDetailsResponse*

Fault: MKSEception (soap:fault, use = literal) [Source code](#)

MKSEception type *MKSEception*

- class type *string*
- id type *string*
- implication type *string*

5. getItem

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_getItem (soap:body, use = literal) [Source code](#)

getItem type *getItem*

Output: Integrity_10_getItemResponse (soap:body, use = literal) [Source code](#)

getItemResponse type *getItemResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

6. getItemsByCustomQuery

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_getItemsByCustomQuery (soap:body, use = literal) [Source code](#)

getItemsByCustomQuery type *getItemsByCustomQuery*

Output: Integrity_10_getItemsByCustomQueryResponse (soap:body, use = literal) [Source code](#)

getItemsByCustomQueryResponse type *getItemsByCustomQueryResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

7. getItemsByIDs

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_getItemsByIDs (soap:body, use = literal) [Source code](#)

getItemsByIDs type *getItemsByIDs*

Output: Integrity_10_getItemsByIDsResponse (soap:body, use = literal) [Source code](#)

getItemsByIDsResponse type *getItemsByIDsResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

8. getItemsByNamedQuery

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_getItemsByNamedQuery (soap:body, use = literal) [Source code](#)

getItemsByNamedQuery type *getItemsByNamedQuery*

Output: Integrity_10_getItemsByNamedQueryResponse (soap:body, use = literal) [Source code](#)

getItemsByNamedQueryResponse type *getItemsByNamedQueryResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

9. getProjects

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_getProjects (soap:body, use = literal) [Source code](#)

getProjects type *getProjects*

Output: Integrity_10_getProjectsResponse (soap:body, use = literal) [Source code](#)

getProjectsResponse type *getProjectsResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

10. removeAttachment

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_10_removeAttachment (soap:body, use = literal) [Source code](#)

removeAttachment type *removeAttachment*

Output: Integrity_10_removeAttachmentResponse (soap:body, use = literal) [Source code](#)

removeAttachmentResponse type *removeAttachmentResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

WSDL source code

```
<?xml version="1.0"?>
<definitions name="Integrity_10Service" targetNamespace="http://webservice.mks.com/10/Integrity"
  xmlns:ns1="http://webservice.mks.com/10/Integrity/schema"
  xmlns:ns2="http://webservice.mks.com/10/Integrity/fault"
  xmlns:ns3="http://www.w3.org/2005/05/xmlmime"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://webservice.mks.com/10/Integrity"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
</definitions>
<types>
  <xs:schema elementFormDefault="qualified"
    targetNamespace="http://webservice.mks.com/10/Integrity/schema" version="1.0">
    <xs:import namespace="http://www.w3.org/2005/05/xmlmime" schemaLocation="xmlmime.xsd" />
    <xs:element name="AttachmentChecksum">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Field" type="xs:string" />
          <xs:element name="FileName" type="xs:string" />
          <xs:element name="FileSize" type="xs:integer" />
          <xs:element name="Checksum" type="xs:string" />
          <xs:element name="ChecksumAlgorithm" type="xs:string" />
        </xs:sequence>
        <xs:attribute name="status" type="xs:string" />
      </xs:complexType>
    </xs:element>
    <xs:element name="Fields">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="Field">
            <xs:complexType>
              <xs:sequence />
              <xs:attribute name="Name" type="xs:string" use="required" />
              <xs:attribute name="Value" type="xs:string" />
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="ItemFieldRequest">
      <xs:complexContent>
        <xs:extension base="tns:ItemFieldBase">
          <xs:sequence>
            <xs:element minOccurs="0" name="user" type="tns:NullableString" />
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:schema>
</types>
```

```

</xs:complexContent>
</xs:complexType>
<xs:complexType name="ItemFieldResponse">
    <xs:complexContent>
        <xs:extension base="tns:ItemFieldBase">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
                            name="UserRecord" type="tns:UserRecord" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="UserRecord">
    <xs:sequence>
        <xs:element minOccurs="0" name="user" type="tns:NullableString" />
        <xs:element minOccurs="0" name="fullname" type="tns:NullableString" />
        <xs:element minOccurs="0" name="email" type="tns:NullableString" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ItemFieldBase">
    <xs:choice minOccurs="0">
        <xs:element name="shorttext" type="tns:NullableShortString" />
        <xs:element name="relationship">
            <xs:complexType>
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0"
                                name="value" nillable="true" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="longtext" type="tns:NullableString" />
        <xs:element name="group" type="tns:NullableString" />
        <xs:element name="state" type="xs:string" />
        <xs:element name="type" type="xs:string" />
        <xs:element name="project" type="tns:NullableString" />
        <xs:element name="siProject" type="tns:NullableString" />
        <xs:element name="float" type="tns:NullableFloat" />
        <xs:element name="double" type="tns:NullableDouble" />
        <xs:element name="boolean" type="tns:NullableBoolean" />
        <xs:element name="integer" type="tns:NullableInteger" />
        <xs:element name="pick">
            <xs:complexType>
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0"
                                name="value" nillable="true" type="xs:string" />
                </xs:sequence>
                <xs:attribute name="null" type="xs:boolean" />
            </xs:complexType>
        </xs:element>
        <xs:element name="date" type="tns:NullableString" />
    </xs:choice>
</xs:complexType>

```

```

<xs:element name="dateTime" type="tns:NullableString" />
</xs:choice>
<xs:attribute name="Name" type="xs:string" use="required" />
</xs:complexType>
<xs:element name="Project">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" ref="tns:Fields" />
        </xs:sequence>
        <xs:attribute name="Name" type="xs:string" />
    </xs:complexType>
</xs:element>
<xs:element name="RelatedItem">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="Item"
type="tns:Relationship" />
            <xs:element maxOccurs="unbounded" minOccurs="0" name="IBPL"
type="xs:integer" />
        </xs:sequence>
        <xs:attribute name="FieldName" type="xs:string" use="required" />
    </xs:complexType>
</xs:element>
<xs:complexType name="GetItemType">
    <xs:complexContent>
        <xs:extension base="tns:.GetItemCommon">
            <xs:sequence />
            <xs:attribute form="qualified" name="ItemId" type="xs:integer"
use="required" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetItemCommon">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
name="InputField" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Credentials">
    <xs:sequence>
        <xs:element name="Username" type="xs:string" />
        <xs:element name="Password" type="xs:string" />
        <xs:element minOccurs="0" name="ImpersonatedUser" type="xs:string" />
        <xs:element minOccurs="0" name="DateFormat" type="xs:string" />
        <xs:element minOccurs="0" name="DateTimeFormat" type="xs:string" />
    </xs:sequence>

```

```

<xs:attribute name="transactionId" type="xs:string" />
</xs:complexType>
<xs:complexType name="Item">
    <xs:complexContent>
        <xs:extension base="tns:StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
                            name="ItemField" type="tns:ItemFieldResponse" />
                <xs:element maxOccurs="unbounded" minOccurs="0"
                            name="Attachment" type="tns:AttachmentInfo" />
                <xs:element maxOccurs="unbounded" minOccurs="0"
                            ref="tns:RelatedItem" />
            </xs:sequence>
            <xs:attribute name="State" type="xs:string" />
            <xs:attribute name="Type" type="xs:string" />
            <xs:attribute form="qualified" name="ItemId" type="xs:integer" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="StatusResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string" />
</xs:complexType>
<xs:complexType name="NullableShortString">
    <xs:choice>
        <xs:element name="value" type="xs:string" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableString">
    <xs:choice>
        <xs:element name="value" type="xs:string" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableFloat">
    <xs:choice>
        <xs:element name="value" type="xs:float" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableDouble">
    <xs:choice>
        <xs:element name="value" type="xs:double" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableBoolean">

```

```

<xs:choice>
    <xs:element name="value" type="xs:boolean" />
    <xs:element name="null" type="xs:boolean" />
</xs:choice>
</xs:complexType>
<xs:complexType name="NullableInteger">
    <xs:choice>
        <xs:element name="value" type="xs:int" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="AttachmentInfo">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element name="Name" type="xs:string" />
                <xs:element name="Size" type="xs:int" />
                <xs:element name="Checksum" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Relationship">
    <xs:simpleContent>
        <xs:extension base="xs:integer">
            <xs:attribute name="flag" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="AttachmentRequest">
    <xs:complexContent>
        <xs:extension base="tns>Credentials">
            <xs:sequence />
            <xs:attribute name="AttachmentName" type="xs:string" />
            <xs:attribute name="FieldName" type="xs:string" />
            <xs:attribute form="qualified" name="ItemId" type="xs:integer"
use="required" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AttachmentContents">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element minOccurs="0" name="Attachment"
ns3:expectedContentTypes="application/octet-stream"
type="xs:base64Binary" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

</xs:complexType>
<xs:complexType name="ProjectRequest">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="FieldList" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectList">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
                    ref="tns:Project" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateItemType">
    <xs:complexContent>
        <xs:extension base="tns>CreateEdit">
            <xs:sequence />
            <xs:attribute name="Type" type="xs:string" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateEdit">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="ItemField" type="tns:ItemFieldRequest" />
                <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="Attachment" type="tns:AttachmentEdit" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AttachmentEdit">
    <xs:sequence>
        <xs:element name="Name" type="xs:string" />
        <xs:element name="Field" type="xs:string" />
        <xs:element minOccurs="0" name="Description" type="xs:string" />
        <xs:element name="Attachment" ns3:expectedContentTypes="application/octet-
            stream" type="xs:base64Binary" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ItemResponse">

```

```

<xs:complexContent>
  <xs:extension base="tns:StatusResponse">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0"
      ref="tns:AttachmentChecksum" />
    </xs:sequence>
    <xs:attribute form="qualified" name="ItemId" type="xs:integer"
    use="required" />
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="EditItemType">
  <xs:complexContent>
    <xs:extension base="tns:CreateEdit">
      <xs:sequence />
      <xs:attribute form="qualified" name="ItemId" type="xs:integer" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AttachmentDetails">
  <xs:complexContent>
    <xs:extension base="tns:AttachmentInfo">
      <xs:sequence>
        <xs:element name="ContentType" type="xs:string" />
        <xs:element name="Description" type="xs:string" />
        <xs:element name="AddedBy" type="xs:string" />
        <xs:element name="AddedDate" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="CustomQuery">
  <xs:complexContent>
    <xs:extension base="tns:GetItemCommon">
      <xs:sequence>
        <xs:element name="QueryDefinition" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ItemList">
  <xs:complexContent>
    <xs:extension base="tns:StatusResponse">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="Item"
        type="tns:Item" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetItemsByIdsType">

```

```

<xs:complexContent>
    <xs:extension base="tns:Credentials">
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0"
                name="InputField" type="xs:string" />
            <xs:element maxOccurs="unbounded" name="ItemId"
                type="xs:integer" />
        </xs:sequence>
        <xs:attribute name="Name" type="xs:string" />
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="NamedQuery">
    <xs:complexContent>
        <xs:extension base="tns:GetItemCommon">
            <xs:sequence>
                <xs:element name="QueryName" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:attribute name="ItemId" type="xs:integer" />
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/10/Integrity/fault" version="1.0">
    <xs:element name="MKSEception" nillable="true" type="tns:MKSFault" />
    <xs:complexType name="MKSFault">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="class" type="xs:string" />
                <xs:attribute name="id" type="xs:string" />
                <xs:attribute name="implication" type="xs:string" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/10/Integrity" version="1.0">
    <xs:import namespace="http://webservice.mks.com/10/Integrity/schema" />
    <xs:element name="createItem" type="tns:createItem" />
    <xs:element name="createItemResponse" type="tns:createItemResponse" />
    <xs:element name="editItem" type="tns:editItem" />
    <xs:element name="editItemResponse" type="tns:editItemResponse" />
    <xs:element name="fetchAttachments" type="tns:fetchAttachments" />
    <xs:element name="fetchAttachmentsResponse" type="tns:fetchAttachmentsResponse" />
    <xs:element name="getAttachmentDetails" type="tns:getAttachmentDetails" />
    <xs:element name="getAttachmentDetailsResponse" type="tns:getAttachmentDetailsResponse" />
    <xs:element name="getItem" type="tns:getItem" />
    <xs:element name="getItemResponse" type="tns:getItemResponse" />
    <xs:element name="getItemsByCustomQuery" type="tns:getItemsByCustomQuery" />
    <xs:element name="getItemsByCustomQueryResponse">

```

```

    type="tns:getItemsByCustomQueryResponse" />
<xs:element name="getItemsByIDs" type="tns:getItemsByIDs" />
<xs:element name="getItemsByIDsResponse" type="tns:getItemsByIDsResponse" />
<xs:element name="getItemsByNamedQuery" type="tns:getItemsByNamedQuery" />
<xs:element name="getItemsByNamedQueryResponse"
type="tns:getItemsByNamedQueryResponse" />
<xs:element name="getProjects" type="tns:getProjects" />
<xs:element name="getProjectsResponse" type="tns:getProjectsResponse" />
<xs:element name="removeAttachment" type="tns:removeAttachment" />
<xs:element name="removeAttachmentResponse" type="tns:removeAttachmentResponse" />
<xs:complexType name="getItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ItemType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:Item" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchAttachments">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AttachmentRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchAttachmentsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AttachmentContents" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjects">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ProjectRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjectsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ProjectList" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CreateItem" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createItemResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemResponse" />
    </xs:sequence>

```

```

</xs:complexType>
<xs:complexType name="editItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>EditItemType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="editItemResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getAttachmentDetails">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AttachmentRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getAttachmentDetailsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AttachmentDetails" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeAttachment">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AttachmentRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeAttachmentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AttachmentInfo" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByCustomQuery">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CustomQuery" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByCustomQueryResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemList" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByIDs">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetItemsByIDsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByIDsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemList" />
    </xs:sequence>

```

```

</xs:complexType>
<xs:complexType name="getItemsByNamedQuery">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:NamedQuery" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByNamedQueryResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemList" />
    </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="Integrity_10_getItemsByNamedQuery">
    <part element="tns:getItemsByNamedQuery" name="getItemsByNamedQuery" />
</message>
<message name="Integrity_10_getItemsByCustomQuery">
    <part element="tns:getItemsByCustomQuery" name="getItemsByCustomQuery" />
</message>
<message name="Integrity_10_createItem">
    <part element="tns:createItem" name="createItem" />
</message>
<message name="Integrity_10_editItemResponse">
    <part element="tns:editItemResponse" name="editItemResponse" />
</message>
<message name="Integrity_10_getItemsByIDsResponse">
    <part element="tns:getItemsByIDsResponse" name="getItemsByIDsResponse" />
</message>
<message name="Integrity_10_fetchAttachments">
    <part element="tns:fetchAttachments" name="fetchAttachments" />
</message>
<message name="Integrity_10_getItemsByIDs">
    <part element="tns:getItemsByIDs" name="getItemsByIDs" />
</message>
<message name="MKSEception">
    <part element="ns2:MKSEception" name="MKSEception" />
</message>
<message name="Integrity_10_getItemsByNamedQueryResponse">
    <part element="tns:getItemsByNamedQueryResponse" name="getItemsByNamedQueryResponse" />
</message>
<message name="Integrity_10_getProjectsResponse">
    <part element="tns:getProjectsResponse" name="getProjectsResponse" />
</message>
<message name="Integrity_10_getItem">
    <part element="tns:getItem" name="getItem" />
</message>
<message name="Integrity_10_getProjects">
    <part element="tns:getProjects" name="getProjects" />
</message>

```

```

<message name="Integrity_10_editItem">
    <part element="tns:editItem" name="editItem" />
</message>
<message name="Integrity_10_getAttachmentDetails">
    <part element="tns:getAttachmentDetails" name="getAttachmentDetails" />
</message>
<message name="Integrity_10_getItemResponse">
    <part element="tns:getItemResponse" name="getItemResponse" />
</message>
<message name="Integrity_10_getItemsByCustomQueryResponse">
    <part element="tns:getItemsByCustomQueryResponse" name="getItemsByCustomQueryResponse" />
</message>
<message name="Integrity_10_removeAttachmentResponse">
    <part element="tns:removeAttachmentResponse" name="removeAttachmentResponse" />
</message>
<message name="Integrity_10_removeAttachment">
    <part element="tns:removeAttachment" name="removeAttachment" />
</message>
<message name="Integrity_10_fetchAttachmentsResponse">
    <part element="tns:fetchAttachmentsResponse" name="fetchAttachmentsResponse" />
</message>
<message name="Integrity_10_createItemResponse">
    <part element="tns:createItemResponse" name="createItemResponse" />
</message>
<message name="Integrity_10_getAttachmentDetailsResponse">
    <part element="tns:getAttachmentDetailsResponse" name="getAttachmentDetailsResponse" />
</message>
<portType name="Integrity_10">
    <operation name="createItem" parameterOrder="createItem">
        <input message="tns:Integrity_10_createItem" />
        <output message="tns:Integrity_10_createItemResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="editItem" parameterOrder="editItem">
        <input message="tns:Integrity_10_editItem" />
        <output message="tns:Integrity_10_editItemResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="fetchAttachments" parameterOrder="fetchAttachments">
        <input message="tns:Integrity_10_fetchAttachments" />
        <output message="tns:Integrity_10_fetchAttachmentsResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="getAttachmentDetails" parameterOrder="getAttachmentDetails">
        <input message="tns:Integrity_10_getAttachmentDetails" />
        <output message="tns:Integrity_10_getAttachmentDetailsResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="getItem" parameterOrder="getItem">

```

```

<input message="tns:Integrity_10_getItem" />
<output message="tns:Integrity_10_getItemResponse" />
<fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getItemsByCustomQuery" parameterOrder="getItemsByCustomQuery">
    <input message="tns:Integrity_10_getItemsByCustomQuery" />
    <output message="tns:Integrity_10_getItemsByCustomQueryResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getItemsByIds" parameterOrder="getItemsByIds">
    <input message="tns:Integrity_10_getItemsByIds" />
    <output message="tns:Integrity_10_getItemsByIdsResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getItemsByNamedQuery" parameterOrder="getItemsByNamedQuery">
    <input message="tns:Integrity_10_getItemsByNamedQuery" />
    <output message="tns:Integrity_10_getItemsByNamedQueryResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getProjects" parameterOrder="getProjects">
    <input message="tns:Integrity_10_getProjects" />
    <output message="tns:Integrity_10_getProjectsResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="removeAttachment" parameterOrder="removeAttachment">
    <input message="tns:Integrity_10_removeAttachment" />
    <output message="tns:Integrity_10_removeAttachmentResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
</portType>
<binding name="Integrity_10Binding" type="tns:Integrity_10">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsp:PolicyReference URI="#wsmtom_policy" required="true"
        xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    />
    <operation name="createItem">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>
    <operation name="editItem">
        <soap:operation soapAction="" />

```

```
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEception">
    <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="fetchAttachments">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="getAttachmentDetails">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="getItem">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="getItemsByCustomQuery">
    <soap:operation soapAction="" />
    <input>
```

```

        <soap:body use="literal" />
</input>
<output>
        <soap:body use="literal" />
</output>
<fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="getItemsByIDs">
<soap:operation soapAction="" />
<input>
        <soap:body use="literal" />
</input>
<output>
        <soap:body use="literal" />
</output>
<fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="getItemsByNamedQuery">
<soap:operation soapAction="" />
<input>
        <soap:body use="literal" />
</input>
<output>
        <soap:body use="literal" />
</output>
<fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="getProjects">
<soap:operation soapAction="" />
<input>
        <soap:body use="literal" />
</input>
<output>
        <soap:body use="literal" />
</output>
<fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="removeAttachment">
<soap:operation soapAction="" />
<input>
        <soap:body use="literal" />

```

```
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEception">
    <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
</binding>
<service name="Integrity_10Service">
    <port binding="tns:Integrity_10Binding" name="Integrity_10Port">
        <soap:address location="http://IntegrityServer:9001/webservices/10/Integrity/" />
    </port>
</service>
<wsp:Policy wsu:id="wsmtom_policy"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
>
    <wsoma:OptimizedMimeSerialization
        xmlns:wsoma="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization"
    />
</wsp:Policy>
</definitions>
```

Web Service: Integrity_2009Service

Web Service: Integrity_2009Service

Target Namespace: <http://webservice.mks.com/2009/Integrity>

Port **Integrity_2009Port** [Port type](#) [Source code](#)

Location: <http://IntegrityServer:9001/webservices/2009/Integrity/>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. [createItem](#) [Detail](#) [Source code](#)
2. [editItem](#) [Detail](#) [Source code](#)
3. [fetchAttachments](#) [Detail](#) [Source code](#)
4. [getAttachmentDetails](#) [Detail](#) [Source code](#)
5. [getItem](#) [Detail](#) [Source code](#)
6. [getItemsByCustomQuery](#) [Detail](#) [Source code](#)
7. [getItemsByIDs](#) [Detail](#) [Source code](#)
8. [getItemsByNamedQuery](#) [Detail](#) [Source code](#)
9. [getProjects](#) [Detail](#) [Source code](#)
10. [removeAttachment](#) [Detail](#) [Source code](#)

Operations

Port type **Integrity_2009** [Source code](#)

1. [createItem](#)

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_createItem (soap:body, use = literal) [Source code](#)

creatItem type [creatItem](#)

Output: Integrity_2009_createItemResponse (soap:body, use = literal) [Source code](#)

creatItemResponse type [creatItemResponse](#)

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type [MKSEException](#)

- o class type [string](#)
- o id type [string](#)
- o implication type [string](#)

2. editItem

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_editItem (soap:body, use = literal) [Source code](#)

editItem type *editItem*

Output: Integrity_2009_editItemResponse (soap:body, use = literal) [Source code](#)

editItemResponse type *editItemResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

3. fetchAttachments

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_fetchAttachments (soap:body, use = literal) [Source code](#)

fetchAttachments type *fetchAttachments*

Output: Integrity_2009_fetchAttachmentsResponse (soap:body, use = literal) [Source code](#)

fetchAttachmentsResponse type *fetchAttachmentsResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

4. getAttachmentDetails

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_getAttachmentDetails (soap:body, use = literal) [Source code](#)

getAttachmentDetails type *getAttachmentDetails*

Output: Integrity_2009_getAttachmentDetailsResponse (soap:body, use = literal) [Source code](#)

getAttachmentDetailsResponse type *getAttachmentDetailsResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

5. getItem

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_getItem (soap:body, use = literal) [Source code](#)

getItem type *getItem*

Output: Integrity_2009_getItemResponse (soap:body, use = literal) [Source code](#)

getItemResponse type *getItemResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

6. getItemsByCustomQuery

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_getItemsByCustomQuery (soap:body, use = literal) [Source code](#)

getItemsByCustomQuery type *getItemsByCustomQuery*

Output: Integrity_2009_getItemsByCustomQueryResponse (soap:body, use = literal) [Source code](#)

getItemsByCustomQueryResponse type *getItemsByCustomQueryResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

7. getItemsByIDs

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_getItemsByIDs (soap:body, use = literal) [Source code](#)

getItemsByIDs type *getItemsByIDs*

Output: Integrity_2009_getItemsByIDsResponse (soap:body, use = literal) [Source code](#)

getItemsByIDsResponse type *getItemsByIDsResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- class type *string*
- id type *string*
- implication type *string*

8. getItemsByNamedQuery

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_getItemsByNamedQuery (soap:body, use = literal) [Source code](#)

getItemsByNamedQuery type *getItemsByNamedQuery*

Output: Integrity_2009_getItemsByNamedQueryResponse (soap:body, use = literal) [Source code](#)

getItemsByNamedQueryResponse type *getItemsByNamedQueryResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

9. getProjects

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_getProjects (soap:body, use = literal) [Source code](#)

getProjects type *getProjects*

Output: Integrity_2009_getProjectsResponse (soap:body, use = literal) [Source code](#)

getProjectsResponse type *getProjectsResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

10. removeAttachment

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Integrity_2009_removeAttachment (soap:body, use = literal) [Source code](#)

removeAttachment type *removeAttachment*

Output: Integrity_2009_removeAttachmentResponse (soap:body, use = literal) [Source code](#)

removeAttachmentResponse type *removeAttachmentResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

WSDL source code

```
<?xml version="1.0"?>
<definitions name="Integrity_2009Service" targetNamespace="http://webservice.mks.com/2009/Integrity"
  xmlns:ns1="http://webservice.mks.com/2009/Integrity/schema"
  xmlns:ns2="http://webservice.mks.com/2009/Integrity/fault"
  xmlns:ns3="http://www.w3.org/2005/05/xmlmime"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://webservice.mks.com/2009/Integrity"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

>

<types>
  <xs:schema elementFormDefault="qualified"
    targetNamespace="http://webservice.mks.com/2009/Integrity/schema" version="1.0">
    <xs:import namespace="http://www.w3.org/2005/05/xmlmime" schemaLocation="xmlmime.xsd" />
    <xs:element name="AttachmentChecksum">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Field" type="xs:string" />
          <xs:element name="FileName" type="xs:string" />
          <xs:element name="FileSize" type="xs:integer" />
          <xs:element name="Checksum" type="xs:string" />
          <xs:element name="ChecksumAlgorithm" type="xs:string" />
        </xs:sequence>
        <xs:attribute name="status" type="xs:string" />
      </xs:complexType>
    </xs:element>
    <xs:element name="Fields">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="Field">
            <xs:complexType>
              <xs:sequence />
              <xs:attribute name="Name" type="xs:string" use="required" />
              <xs:attribute name="Value" type="xs:string" />
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ItemField">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="shorttext" type="tns:NullableShortString"
          />
          <xs:element minOccurs="0" name="relationship">
            <xs:complexType>
```

```

<xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0"
        name="value" nullable="true" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element minOccurs="0" name="longtext" type="tns:NullableString" />
<xs:element minOccurs="0" name="user" type="tns:NullableString" />
<xs:element minOccurs="0" name="group" type="tns:NullableString" />
<xs:element minOccurs="0" name="state" type="xs:string" />
<xs:element minOccurs="0" name="type" type="xs:string" />
<xs:element minOccurs="0" name="project" type="tns:NullableString" />
<xs:element minOccurs="0" name="float" type="tns:NullableFloat" />
<xs:element minOccurs="0" name="double" type="tns:NullableDouble" />
<xs:element minOccurs="0" name="boolean" type="tns:NullableBoolean" />
<xs:element minOccurs="0" name="integer" type="tns:NullableInteger" />
<xs:element minOccurs="0" name="pick">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0"
                name="value" nullable="true" type="xs:string" />
        </xs:sequence>
        <xs:attribute name="null" type="xs:boolean" />
    </xs:complexType>
</xs:element>
<xs:element minOccurs="0" name="date" type="tns:NullableString" />
<xs:element minOccurs="0" name="dateTime" type="tns:NullableString" />
</xs:sequence>
<xs:attribute name="Name" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:element name="Project">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" ref="tns:Fields" />
        </xs:sequence>
        <xs:attribute name="Name" type="xs:string" />
    </xs:complexType>
</xs:element>
<xs:element name="RelatedItem">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="Item"
                type="tns:Relationship" />
            <xs:element maxOccurs="unbounded" minOccurs="0" name="IBPL"
                type="xs:integer" />
        </xs:sequence>
        <xs:attribute name="FieldName" type="xs:string" use="required" />
    </xs:complexType>
</xs:element>

```

```

</xs:element>
<xs:complexType name="GetItemType">
    <xs:complexContent>
        <xs:extension base="tns:.GetItemCommon">
            <xs:sequence />
            <xs:attribute form="qualified" name="ItemId" type="xs:integer"
use="required" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name=".GetItemCommon">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
name="InputField" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Credentials">
    <xs:sequence>
        <xs:element name="Username" type="xs:string" />
        <xs:element name="Password" type="xs:string" />
        <xs:element minOccurs="0" name="ImpersonatedUser" type="xs:string" />
        <xs:element minOccurs="0" name="DateFormat" type="xs:string" />
        <xs:element minOccurs="0" name="DateTimeFormat" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string" />
</xs:complexType>
<xs:complexType name="Item">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
ref="tns:ItemField" />
                <xs:element maxOccurs="unbounded" minOccurs="0"
name="Attachment" type="tns:AttachmentInfo" />
                <xs:element maxOccurs="unbounded" minOccurs="0"
ref="tns:RelatedItem" />
            </xs:sequence>
            <xs:attribute name="State" type="xs:string" />
            <xs:attribute name="Type" type="xs:string" />
            <xs:attribute form="qualified" name="ItemId" type="xs:integer" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="StatusResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:string" />

```

```

</xs:sequence>
<xs:attribute name="transactionId" type="xs:string" />
</xs:complexType>
<xs:complexType name="NullableShortString">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="null" type="xs:boolean" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="NullableString">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="null" type="xs:boolean" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="NullableFloat">
<xs:simpleContent>
<xs:extension base="xs:float">
<xs:attribute name="null" type="xs:boolean" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="NullableDouble">
<xs:simpleContent>
<xs:extension base="xs:double">
<xs:attribute name="null" type="xs:boolean" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="NullableBoolean">
<xs:simpleContent>
<xs:extension base="xs:boolean">
<xs:attribute name="null" type="xs:boolean" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="NullableInteger">
<xs:simpleContent>
<xs:extension base="xs:int">
<xs:attribute name="null" type="xs:boolean" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="AttachmentInfo">
<xs:complexContent>
<xs:extension base="tns>StatusResponse">
<xs:sequence>
<xs:element name="Name" type="xs:string" />

```

```

<xs:element name="Size" type="xs:int" />
<xs:element name="Checksum" type="xs:string" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Relationship">
<xs:simpleContent>
<xs:extension base="xs:integer">
<xs:attribute name="flag" type="xs:string" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="AttachmentRequest">
<xs:complexContent>
<xs:extension base="tns:Credentials">
<xs:sequence />
<xs:attribute name="AttachmentName" type="xs:string" />
<xs:attribute name="FieldName" type="xs:string" />
<xs:attribute form="qualified" name="ItemId" type="xs:integer"
use="required" />
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="AttachmentContents">
<xs:complexContent>
<xs:extension base="tns>StatusResponse">
<xs:sequence>
<xs:element minOccurs="0" name="Attachment"
ns3:expectedContentTypes="application/octet-stream"
type="xs:base64Binary" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectRequest">
<xs:complexContent>
<xs:extension base="tns:Credentials">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0"
name="FieldList" type="xs:string" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectList">
<xs:complexContent>
<xs:extension base="tns>StatusResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0"
ref="tns:Project" />

```

```

        </xs:sequence>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateItemType">
    <xs:complexContent>
        <xs:extension base="tns:CreateEdit">
            <xs:sequence />
            <xs:attribute name="Type" type="xs:string" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateEdit">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
ref="tns:ItemField" />
                <xs:element maxOccurs="unbounded" minOccurs="0"
name="Attachment" type="tns:AttachmentEdit" />
                <xs:element maxOccurs="unbounded" minOccurs="0"
ref="tns:RelatedItem" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AttachmentEdit">
    <xs:sequence>
        <xs:element name="Name" type="xs:string" />
        <xs:element name="Field" type="xs:string" />
        <xs:element minOccurs="0" name="Description" type="xs:string" />
        <xs:element name="Attachment" ns3:expectedContentTypes="application/octet-
stream" type="xs:base64Binary" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ItemResponse">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
ref="tns:AttachmentChecksum" />
            </xs:sequence>
            <xs:attribute form="qualified" name="ItemId" type="xs:integer"
use="required" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="EditItemType">
    <xs:complexContent>
        <xs:extension base="tns:CreateEdit">
            <xs:sequence />

```

```

        <xs:attribute form="qualified" name="ItemId" type="xs:integer" />
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="AttachmentDetails">
    <xs:complexContent>
        <xs:extension base="tns:AttachmentInfo">
            <xs:sequence>
                <xs:element name="ContentType" type="xs:string" />
                <xs:element name="Description" type="xs:string" />
                <xs:element name="AddedBy" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="CustomQuery">
    <xs:complexContent>
        <xs:extension base="tns:.GetItemCommon">
            <xs:sequence>
                <xs:element name="QueryDefinition" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="ItemList">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Item"
type="tns:Item" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="GetItemsByIdsType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
name="InputField" type="xs:string" />
                <xs:element maxOccurs="unbounded" name="ItemId"
type="xs:integer" />
            </xs:sequence>
            <xs:attribute name="Name" type="xs:string" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="NamedQuery">
    <xs:complexContent>
        <xs:extension base="tns:.GetItemCommon">
            <xs:sequence>

```

```

<xs:element name="QueryName" type="xs:string" />
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:attribute name="ItemId" type="xs:integer" />
</xs:schema>

<xs:schema targetNamespace="http://webservice.mks.com/2009/Integrity/fault" version="1.0">
    <xs:element name="MKSEception" nillable="true" type="tns:MKSFault" />
    <xs:complexType name="MKSFault">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="class" type="xs:string" />
                <xs:attribute name="id" type="xs:string" />
                <xs:attribute name="implication" type="xs:string" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>

<xs:schema targetNamespace="http://webservice.mks.com/2009/Integrity" version="1.0">
    <xs:import namespace="http://webservice.mks.com/2009/Integrity/schema" />
    <xs:element name="createItem" type="tns:createItem" />
    <xs:element name="createItemResponse" type="tns:createItemResponse" />
    <xs:element name="editItem" type="tns:editItem" />
    <xs:element name="editItemResponse" type="tns:editItemResponse" />
    <xs:element name="fetchAttachments" type="tns:fetchAttachments" />
    <xs:element name="fetchAttachmentsResponse" type="tns:fetchAttachmentsResponse" />
    <xs:element name="getAttachmentDetails" type="tns:getAttachmentDetails" />
    <xs:element name="getAttachmentDetailsResponse" type="tns:getAttachmentDetailsResponse" />
    <xs:element name="getItem" type="tns:getItem" />
    <xs:element name="getItemResponse" type="tns:getItemResponse" />
    <xs:element name="getItemsByCustomQuery" type="tns:getItemsByCustomQuery" />
    <xs:element name="getItemsByCustomQueryResponse"
    type="tns:getItemsByCustomQueryResponse" />
    <xs:element name="getItemsByIds" type="tns:getItemsByIds" />
    <xs:element name="getItemsByIdsResponse" type="tns:getItemsByIdsResponse" />
    <xs:element name="getItemsByNamedQuery" type="tns:getItemsByNamedQuery" />
    <xs:element name="getItemsByNamedQueryResponse"
    type="tns:getItemsByNamedQueryResponse" />
    <xs:element name="getProjects" type="tns:getProjects" />
    <xs:element name="getProjectsResponse" type="tns:getProjectsResponse" />
    <xs:element name="removeAttachment" type="tns:removeAttachment" />
    <xs:element name="removeAttachmentResponse" type="tns:removeAttachmentResponse" />
<xs:complexType name="getItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetItemType" />
    </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="getItemResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:item" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchAttachments">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AttachmentRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchAttachmentsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AttachmentContents" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjects">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ProjectRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjectsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ProjectList" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CreateItemType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createItemResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:itemResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="editItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>EditItemType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="editItemResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:itemResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getAttachmentDetails">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AttachmentRequest" />
    </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="getAttachmentDetailsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AttachmentDetails" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeAttachment">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AttachmentRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeAttachmentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AttachmentInfo" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByCustomQuery">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CustomQuery" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByCustomQueryResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemList" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByIds">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetItemsByIdsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByIdsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemList" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByNamedQuery">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:NamedQuery" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getItemsByNamedQueryResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ItemList" />
    </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="Integrity_2009_fetchAttachmentsResponse">
    <part element="tns:fetchAttachmentsResponse" name="fetchAttachmentsResponse" />
</message>

```

```
<message name="Integrity_2009_getItemsByCustomQueryResponse">
    <part element="tns:getItemsByCustomQueryResponse" name="getItemsByCustomQueryResponse" />
</message>
<message name="Integrity_2009_getAttachmentDetailsResponse">
    <part element="tns:getAttachmentDetailsResponse" name="getAttachmentDetailsResponse" />
</message>
<message name="Integrity_2009_removeAttachment">
    <part element="tns:removeAttachment" name="removeAttachment" />
</message>
<message name="Integrity_2009_removeAttachmentResponse">
    <part element="tns:removeAttachmentResponse" name="removeAttachmentResponse" />
</message>
<message name="Integrity_2009_getItemsByIDsResponse">
    <part element="tns:getItemsByIDsResponse" name="getItemsByIDsResponse" />
</message>
<message name="MKSEception">
    <part element="ns2:MKSEception" name="MKSEception" />
</message>
<message name="Integrity_2009_getItemsByIDs">
    <part element="tns:getItemsByIDs" name="getItemsByIDs" />
</message>
<message name="Integrity_2009_editItemResponse">
    <part element="tns:editItemResponse" name="editItemResponse" />
</message>
<message name="Integrity_2009_fetchAttachments">
    <part element="tns:fetchAttachments" name="fetchAttachments" />
</message>
<message name="Integrity_2009_getProjectsResponse">
    <part element="tns:getProjectsResponse" name="getProjectsResponse" />
</message>
<message name="Integrity_2009_getItemsByNamedQueryResponse">
    <part element="tns:getItemsByNamedQueryResponse" name="getItemsByNamedQueryResponse" />
</message>
<message name="Integrity_2009_createItem">
    <part element="tns:createItem" name="createItem" />
</message>
<message name="Integrity_2009_getItemResponse">
    <part element="tns:getItemResponse" name="getItemResponse" />
</message>
<message name="Integrity_2009_getAttachmentDetails">
    <part element="tns:getAttachmentDetails" name="getAttachmentDetails" />
</message>
<message name="Integrity_2009_getItemsByCustomQuery">
    <part element="tns:getItemsByCustomQuery" name="getItemsByCustomQuery" />
</message>
<message name="Integrity_2009_editItem">
    <part element="tns:editItem" name="editItem" />
</message>
<message name="Integrity_2009_getItem">
```

```

<part element="tns:getItem" name="getItem" />
</message>
<message name="Integrity_2009_getItemsByNamedQuery">
    <part element="tns:getItemsByNamedQuery" name="getItemsByNamedQuery" />
</message>
<message name="Integrity_2009_createItemResponse">
    <part element="tns:createItemResponse" name="createItemResponse" />
</message>
<message name="Integrity_2009_getProjects">
    <part element="tns:getProjects" name="getProjects" />
</message>
<portType name="Integrity_2009">
    <operation name="createItem" parameterOrder="createItem">
        <input message="tns:Integrity_2009_createItem" />
        <output message="tns:Integrity_2009_createItemResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="editItem" parameterOrder="editItem">
        <input message="tns:Integrity_2009_editItem" />
        <output message="tns:Integrity_2009_editItemResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="fetchAttachments" parameterOrder="fetchAttachments">
        <input message="tns:Integrity_2009_fetchAttachments" />
        <output message="tns:Integrity_2009_fetchAttachmentsResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="getAttachmentDetails" parameterOrder="getAttachmentDetails">
        <input message="tns:Integrity_2009_getAttachmentDetails" />
        <output message="tns:Integrity_2009_getAttachmentDetailsResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="getItem" parameterOrder="getItem">
        <input message="tns:Integrity_2009_getItem" />
        <output message="tns:Integrity_2009_getItemResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="getItemsByCustomQuery" parameterOrder="getItemsByCustomQuery">
        <input message="tns:Integrity_2009_getItemsByCustomQuery" />
        <output message="tns:Integrity_2009_getItemsByCustomQueryResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="getItemsByIds" parameterOrder="getItemsByIds">
        <input message="tns:Integrity_2009_getItemsByIds" />
        <output message="tns:Integrity_2009_getItemsByIdsResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="getItemsByNamedQuery" parameterOrder="getItemsByNamedQuery">

```

```

<input message="tns:Integrity_2009_getItemsByNamedQuery" />
<output message="tns:Integrity_2009_getItemsByNamedQueryResponse" />
<fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getProjects" parameterOrder="getProjects">
    <input message="tns:Integrity_2009_getProjects" />
    <output message="tns:Integrity_2009_getProjectsResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="removeAttachment" parameterOrder="removeAttachment">
    <input message="tns:Integrity_2009_removeAttachment" />
    <output message="tns:Integrity_2009_removeAttachmentResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
</portType>
<binding name="Integrity_2009Binding" type="tns:Integrity_2009">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsp:PolicyReference URI="#wsmtom_policy" required="true"
        xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    />
    <operation name="createItem">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEException">
            <soap:fault name="MKSEException" use="literal" />
        </fault>
    </operation>
    <operation name="editItem">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEException">
            <soap:fault name="MKSEException" use="literal" />
        </fault>
    </operation>
    <operation name="fetchAttachments">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>

```

```
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEception">
    <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="getAttachmentDetails">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="getItem">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="getItemsByCustomQuery">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="getItemsByIDs">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
```

```

        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="getItemsByNamedQuery">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="getProjects">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="removeAttachment">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
</binding>
<service name="Integrity_2009Service">
    <port binding="tns:Integrity_2009Binding" name="Integrity_2009Port">
        <soap:address location="http://IntegrityServer:9001/webservices/2009/Integrity/" />
    </port>
</service>
<wsp:Policy wsu:Id="wsmtom_policy">

```

```
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    >
        <wsoma:OptimizedMimeSerialization
            xmlns:wsoma="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization"
        />
    </wsp:Policy>
</definitions>
```

Web Service: Source_2009_7Service

Web Service: Source_2009_7Service

Target Namespace: <http://webservice.mks.com/2009/7/Source>

Port **Source_2009_7Port** Port type [Source code](#)

Location: <http://IntegrityServer:9001/webservices/2009/7/Source/>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. [addLabel](#) [Detail](#) [Source code](#)
2. [appendRevisionDescription](#) [Detail](#) [Source code](#)
3. [checkConnection](#) [Detail](#) [Source code](#)
4. [checkInMember](#) [Detail](#) [Source code](#)
5. [checkpointProject](#) [Detail](#) [Source code](#)
6. [createChangePackage](#) [Detail](#) [Source code](#)
7. [createMember](#) [Detail](#) [Source code](#)
8. [createSubproject](#) [Detail](#) [Source code](#)
9. [discardChangePackage](#) [Detail](#) [Source code](#)
10. [dropMember](#) [Detail](#) [Source code](#)
11. [fetchMember](#) [Detail](#) [Source code](#)
12. [getMemberHistory](#) [Detail](#) [Source code](#)
13. [getMemberInfo](#) [Detail](#) [Source code](#)
14. [getMemberList](#) [Detail](#) [Source code](#)
15. [getProjectConfigurationPath](#) [Detail](#) [Source code](#)
16. [getProjectInfo](#) [Detail](#) [Source code](#)
17. [getRevisionInfo](#) [Detail](#) [Source code](#)
18. [getSubprojectList](#) [Detail](#) [Source code](#)
19. [lockMember](#) [Detail](#) [Source code](#)
20. [projectLabelExists](#) [Detail](#) [Source code](#)
21. [setSubprojectRevision](#) [Detail](#) [Source code](#)
22. [shareSubproject](#) [Detail](#) [Source code](#)
23. [submitChangePackage](#) [Detail](#) [Source code](#)
24. [unlockMember](#) [Detail](#) [Source code](#)
25. [updateRevision](#) [Detail](#) [Source code](#)

Operations

Port type **Source_2009_7** [Source code](#)

1. addLabel

[Source code](#)

Description: Uses the "addlabel" MKS command. Adds a label to a particular revision for a project member.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_addLabel (soap:body, use = literal) [Source code](#)

addLabel type *addLabel*

Output: Source_2009_7_addLabelResponse (soap:body, use = literal) [Source code](#)

addLabelResponse type *addLabelResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

o class type *string*

The class of exception

o id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

o implication type *string*

The implication from the exception

2. appendRevisionDescription

[Source code](#)

Description: Uses the "appendrevdesc" MKS command. Adds a description to a revision of a member of a project.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_appendRevisionDescription (soap:body, use = literal) [Source code](#)

appendRevisionDescription type *appendRevisionDescription*

Output: Source_2009_7_appendRevisionDescriptionResponse (soap:body, use = literal) [Source code](#)

appendRevisionDescriptionResponse type *appendRevisionDescriptionResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

o class type *string*

The class of exception

o id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

o implication type *string*

The implication from the exception

3. checkConnection

[Source code](#)

Description: Uses the "connect" MKS command. Validates credentials and connection information. This method is not required to initiate a session. Session management is handled automatically by the web services.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_checkConnection (soap:body, use = literal) [Source code](#)

checkConnection type *checkConnection*

Output: Source_2009_7_checkConnectionResponse (soap:body, use = literal) [Source code](#)

checkConnectionResponse type *checkConnectionResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*

The class of exception

- id type *String*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- implication type *string*

The implication from the exception

4. checkInMember

[Source code](#)

Description: Uses the "projectci" MKS command. Checks in changes to a member of a project.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_checkInMember (soap:body, use = literal) [Source code](#)

checkInMember type *checkInMember*

Output: Source_2009_7_checkInMemberResponse (soap:body, use = literal) [Source code](#)

checkInMemberResponse type *checkInMemberResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*

The class of exception

- id type *String*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- implication type *string*

The implication from the exception

5. checkpointProject

[Source code](#)

Description: Uses the "checkpoint" MKS command. Creates a new project revision and associates all current member revisions with the checkpoint.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_checkpointProject (soap:body, use = literal) [Source code](#)

checkpointProject type *checkpointProject*

Output: Source_2009_7_checkpointProjectResponse (soap:body, use = literal) [Source code](#)

checkpointProjectResponse type *checkpointProjectResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class **type string**
The class of exception
- o id **type string**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication **type string**
The implication from the exception

6. createChangePackage

[Source code](#)

Description: Uses the "createcp" MKS command. Creates a change package attached to the indicated Integrity item.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_createChangePackage (soap:body, use = literal) [Source code](#)

createChangePackage type *createChangePackage*

Output: Source_2009_7_createChangePackageResponse (soap:body, use = literal) [Source code](#)

createChangePackageResponse type *createChangePackageResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class **type string**
The class of exception
- o id **type string**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication **type string**
The implication from the exception

7. createMember

[Source code](#)

Description: Uses the "projectadd" MKS command. Adds a member to a project.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_createMember (soap:body, use = literal) [Source code](#)

createMember type *createMember*

Output: Source_2009_7_createMemberResponse (soap:body, use = literal) [Source code](#)

createMemberResponse type *createMemberResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class **type string**
The class of exception
- o id **type string**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication **type string**
The implication from the exception

8. createSubproject

[Source code](#)

Description: Uses the "createsubproject" MKS command. Creates a new subproject as a child of the indicated project or subproject.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_createSubproject (soap:body, use = literal) [Source code](#)

createSubproject type *createSubproject*

Output: Source_2009_7_createSubprojectResponse (soap:body, use = literal) [Source code](#)

createSubprojectResponse type *createSubprojectResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

o class type *string*

The class of exception

o id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

o implication type *string*

The implication from the exception

9. discardChangePackage

[Source code](#)

Description: Uses the "discardcp" MKS command. Discards a change package attached to the indicated Integrity item.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_discardChangePackage (soap:body, use = literal) [Source code](#)

discardChangePackage type *discardChangePackage*

Output: Source_2009_7_discardChangePackageResponse (soap:body, use = literal) [Source code](#)

discardChangePackageResponse type *discardChangePackageResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

o class type *string*

The class of exception

o id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

o implication type *string*

The implication from the exception

10. dropMember

[Source code](#)

Description: Uses the "drop" MKS command. Removes the member from the project.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_dropMember (soap:body, use = literal) [Source code](#)

dropMember type *dropMember*

Output: Source_2009_7_dropMemberResponse (soap:body, use = literal) [Source code](#)

dropMemberResponse type *dropMemberResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

- class **type string**
The class of exception
- id **type String**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type string**
The implication from the exception

11. fetchMember[Source code](#)**Description:** Uses the "projectco" MKS command. Returns the contents of a member of a project.**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.**Input:** Source_2009_7_fetchMember (soap:body, use = literal) [Source code](#)**fetchMember** type *fetchMember***Output:** Source_2009_7_fetchMemberResponse (soap:body, use = literal) [Source code](#)**fetchMemberResponse** type *fetchMemberResponse***Fault:** MKSEException (soap:fault, use = literal) [Source code](#)**MKSEException type MKSEException**

- class **type string**
The class of exception
- id **type String**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type string**
The implication from the exception

12. getMemberHistory[Source code](#)**Description:** Uses the "rlog" MKS command. Returns metadata about the revisions.**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.**Input:** Source_2009_7_getMemberHistory (soap:body, use = literal) [Source code](#)**getMemberHistory** type *getMemberHistory***Output:** Source_2009_7_getMemberHistoryResponse (soap:body, use = literal) [Source code](#)**getMemberHistoryResponse** type *getMemberHistoryResponse***Fault:** MKSEException (soap:fault, use = literal) [Source code](#)**MKSEException type MKSEException**

- class **type string**
The class of exception
- id **type String**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type string**
The implication from the exception

13. getMemberInfo

[Source code](#)

Description: Uses the "memberinfo" MKS command. Returns metadata about the project member.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_getMemberInfo (soap:body, use = literal) [Source code](#)

getMemberInfo type *getMemberInfo*

Output: Source_2009_7_getMemberInfoResponse (soap:body, use = literal) [Source code](#)

getMemberInfoResponse type *getMemberInfoResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class *type string*
The class of exception
- o id *type string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication *type string*
The implication from the exception

14. getMemberList

[Source code](#)

Description: Uses the "viewproject" MKS command. Returns the list of project members not including the subprojects.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_getMemberList (soap:body, use = literal) [Source code](#)

getMemberList type *getMemberList*

Output: Source_2009_7_getMemberListResponse (soap:body, use = literal) [Source code](#)

getMemberListResponse type *getMemberListResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class *type string*
The class of exception
- o id *type string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication *type string*
The implication from the exception

15. getProjectConfigurationPath

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_getProjectConfigurationPath (soap:body, use = literal) [Source code](#)

getProjectConfigurationPath type *getProjectConfigurationPath*

Output: Source_2009_7_getProjectConfigurationPathResponse (soap:body, use = literal) [Source code](#)

getProjectConfigurationPathResponse type *getProjectConfigurationPathResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

- The original fault generated from MKS Integrity. The element value is the full description of the error.
- class **type string**
The class of exception
 - id **type String**
The unique MKS identifier for this error, generally a 6-digit positive integer.
 - implication **type string**
The implication from the exception

16. **getProjectInfo**

[Source code](#)

Description: Uses the "viewproject" MKS command. Returns project metadata.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_getProjectInfo (soap:body, use = literal) [Source code](#)

getProjectInfo type getProjectInfo

Output: Source_2009_7_getProjectInfoResponse (soap:body, use = literal) [Source code](#)

getProjectInfoResponse type getProjectInfoResponse

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

- The original fault generated from MKS Integrity. The element value is the full description of the error.
- class **type string**
The class of exception
 - id **type String**
The unique MKS identifier for this error, generally a 6-digit positive integer.
 - implication **type string**
The implication from the exception

17. **getRevisionInfo**

[Source code](#)

Description: Uses the "rlog" MKS command. Returns metadata about the project member, if a specific revision is requested it returns metadata for the requested revision otherwise metadata is returned for the member revision.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_getRevisionInfo (soap:body, use = literal) [Source code](#)

getRevisionInfo type getRevisionInfo

Output: Source_2009_7_getRevisionInfoResponse (soap:body, use = literal) [Source code](#)

getRevisionInfoResponse type getRevisionInfoResponse

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

- The original fault generated from MKS Integrity. The element value is the full description of the error.
- class **type string**
The class of exception
 - id **type String**
The unique MKS identifier for this error, generally a 6-digit positive integer.
 - implication **type string**
The implication from the exception

18. getSubprojectList

[Source code](#)

Description: Uses the "viewproject" MKS command. Returns the list of subprojects not including the members.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_getSubprojectList (soap:body, use = literal) [Source code](#)

getSubprojectList type *getSubprojectList*

Output: Source_2009_7_getSubprojectListResponse (soap:body, use = literal) [Source code](#)

getSubprojectListResponse type *getSubprojectListResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class **type string**
The class of exception
- id **type String**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type string**
The implication from the exception

19. lockMember

[Source code](#)

Description: Uses the "lock" MKS command. Creates an exclusive lock on the member.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_lockMember (soap:body, use = literal) [Source code](#)

lockMember type *lockMember*

Output: Source_2009_7_lockMemberResponse (soap:body, use = literal) [Source code](#)

lockMemberResponse type *lockMemberResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class **type string**
The class of exception
- id **type String**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type string**
The implication from the exception

20. projectLabelExists

[Source code](#)

Description: Uses the "viewprojecthistory" MKS command. Determines if a label has been applied to any revision in the project's history.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_projectLabelExists (soap:body, use = literal) [Source code](#)

projectLabelExists type *projectLabelExists*

Output: Source_2009_7_projectLabelExistsResponse (soap:body, use = literal) [Source code](#)

projectLabelExistsResponse type *projectLabelExistsResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class **type string**

The class of exception

- o id **type string**

The unique MKS identifier for this error, generally a 6-digit positive integer.

- o implication **type string**

The implication from the exception

21. setSubprojectRevision

[Source code](#)

Description: Uses the "configuresubproject" MKS command. Sets the revision number for a subproject.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_setSubprojectRevision (soap:body, use = literal) [Source code](#)

setSubprojectRevision type *setSubprojectRevision*

Output: Source_2009_7_setSubprojectRevisionResponse (soap:body, use = literal) [Source code](#)

setSubprojectRevisionResponse type *setSubprojectRevisionResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class **type string**

The class of exception

- o id **type string**

The unique MKS identifier for this error, generally a 6-digit positive integer.

- o implication **type string**

The implication from the exception

22. shareSubproject

[Source code](#)

Description: Uses the "sharesubproject" MKS command. Shares the subproject as a build subproject as of a specific revision.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_shareSubproject (soap:body, use = literal) [Source code](#)

shareSubproject type *shareSubproject*

Output: Source_2009_7_shareSubprojectResponse (soap:body, use = literal) [Source code](#)

shareSubprojectResponse type *shareSubprojectResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class **type string**

The class of exception

- o id **type string**

The unique MKS identifier for this error, generally a 6-digit positive integer.
o implication type *string*
The implication from the exception

23. submitChangePackage

[Source code](#)

Description: Uses the "submitcp" MKS command. Closes the change package. All entries must be pending on the server (via the checkInMember operation) prior to executing this command.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_submitChangePackage (soap:body, use = literal) [Source code](#)

submitChangePackage type submitChangePackage

Output: Source_2009_7_submitChangePackageResponse (soap:body, use = literal) [Source code](#)

submitChangePackageResponse type submitChangePackageResponse

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

24. unlockMember

[Source code](#)

Description: Uses the "unlock" MKS command. Removes the lock on the specified member.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_unlockMember (soap:body, use = literal) [Source code](#)

unlockMember type unlockMember

Output: Source_2009_7_unlockMemberResponse (soap:body, use = literal) [Source code](#)

unlockMemberResponse type unlockMemberResponse

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

25. updateRevision

[Source code](#)

Description: Uses the "updaterevision" MKS command. Sets the member revision to the specified revision.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_7_updateRevision (soap:body, use = literal) [Source code](#)

updateRevision type updateRevision

Output: Source_2009_7_updateRevisionResponse (soap:body, use = literal) [Source code](#)

updateRevisionResponse type updateRevisionResponse

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type MKSException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type string
The class of exception
- o id type string
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type string
The implication from the exception

WSDL source code

```
<?xml version="1.0"?>
<definitions name="Source_2009_7Service" targetNamespace="http://webservice.mks.com/2009/7/Source"
    xmlns:tns="http://webservice.mks.com/2009/7/Source"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ns1="http://webservice.mks.com/2009/7/Source/schema"
    xmlns:ns2="http://webservice.mks.com/2009/7/Source/fault"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
>
<types>
    <xs:schema elementFormDefault="qualified" targetNamespace="http://webservice.mks.com/2009/7/Source/schema" version="1.0">
        <xs:element name="Fields">
            <xs:complexType>
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="Field">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="Value" type="xs:string" />
                            </xs:sequence>
                            <xs:attribute name="Name" type="xs:string" use="required" />
                            <xs:attribute name="Type" type="xs:string" use="required">
                                <xs:annotation>
                                    <xs:documentation>The type of data stored in this field: int, string, boolean, datetime, or float. Note that due to a limitation in the product, if a field is not set in the response then the web service is not told what type of data would be returned if there were data. In that case, the type is set to "string".</xs:documentation>
                                </xs:annotation>
                            </xs:attribute>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:schema>
</types>
```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="CreateMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:UpdateMemberRequest">
            <xs:sequence>
                <xs:element minOccurs="0" name="OnExistingArchive">
                    <xs:annotation>
                        <xs:documentation>What to do if the archive already exists. Default is to cancel the operation.</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="cancel">
                                <xs:annotation>
                                    <xs:documentation>Cancel the operation. This is the default.</xs:documentation>
                                </xs:annotation>
                            </xs:enumeration>
                            <xs:enumeration value="share">
                                <xs:annotation>
                                    <xs:documentation>Reuse the archive if it has been dropped from the project, or share it if it is currently being used in other development paths.</xs:documentation>
                                </xs:annotation>
                            </xs:enumeration>
                            <xs:enumeration value="new">
                                <xs:annotation>
                                    <xs:documentation>Create a new archive instead of trying to reuse or share the old one.</xs:documentation>
                                </xs:annotation>
                            </xs:enumeration>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element minOccurs="0" name="Format">
                    <xs:annotation>
                        <xs:documentation>File format - binary or text. If unspecified the system will determine the format based on line length criteria specified by the administrator.</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="text">
                                <xs:annotation>
                                    <xs:documentation>Text format, which uses delta storage in the repository.</xs:documentation>
                                </xs:annotation>
                            </xs:enumeration>
                            <xs:enumeration value="binary">
                                <xs:annotation>
                                    <xs:documentation>Binary format, which records the entire file each time it is updated in the repository.</xs:documentation>
                                </xs:annotation>
                            </xs:enumeration>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="UpdateMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="File" ns1:expectedContentTypes="application/octet-stream" type="xs:base64Binary"
                    xmlns:ns1="http://www.w3.org/2005/05/xmlmime"
                />
                <xs:element minOccurs="0" name="Description" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The revision description for the newly created revision.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="ChangePackageID" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The change package for this operation. Note this operation will fail if a change package is not
                            specified and the policy for the project is to require change packages.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="MemberOperation">
    <xs:complexContent>
        <xs:extension base="tns:ProjectOperation">
            <xs:sequence>
                <xs:element name="Member" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The member name. If the member is in a subdirectory of the project, then preface the member
                            like this: /subdirectory1/subdirectory2/membername</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="UpdateMemberOperation">
    <xs:complexContent>
        <xs:extension base="tns:UpdateProjectOperation">
            <xs:sequence>
                <xs:element name="Member" type="tns:NonEmptyString">
                    <xs:annotation>

```

```

        <xs:documentation>The member name. If the member is in a subdirectory of the project, then preface the member like this: /subdirectory1/subdirectory2/membername</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="UpdateSubprojectOperation">
    <xs:complexContent>
        <xs:extension base="tns:UpdateProjectOperation">
            <xs:sequence>
                <xs:element name="Subproject" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The name of the subproject.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="UpdateProjectOperation">
    <xs:complexContent>
        <xs:extension base="tns:ProjectOperation">
            <xs:sequence>
                <xs:element minOccurs="0" name="ChangePackageID" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The change package for the operation. If unspecified, then the system will attempt to bypass the change package but if the server policy requires change packages then the command will return an error.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectOperation">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element name="Project" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The project or subproject referenced by the service.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Credentials">

```

```

<xs:sequence>
    <xs:element name="Username" type="tns:NonEmptyString" />
    <xs:element name="Password" type="xs:string" />
    <xs:element minOccurs="0" name="ImpersonatedUser" type="tns:NonEmptyString" />
</xs:sequence>
<xs:attribute name="transactionId" type="xs:string">
    <xs:annotation>
        <xs:documentation>An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.</xs:documentation>
    </xs:annotation>
</xs:attribute>
</xs:complexType>
<xs:complexType name="MemberResponse">
    <xs:complexContent>
        <xs:extension base="tns:ProjectResponse">
            <xs:sequence>
                <xs:element name="Member" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The member name.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Revision" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The affected revision.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectResponse">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element name="Project" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The project or subproject that is the subject of this operation.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="StatusResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string" />
</xs:complexType>

```

```

<xs:complexType name="AddLabelRequest">
    <xs:complexContent>
        <xs:extension base="tns:RevisionOperation">
            <xs:sequence>
                <xs:element name="Label" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>Label to be applied to the indicated revision</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="MoveLabel" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>If true then the label will be moved if it has already been applied to the member. Default is to throw an exception if the label has already been specified.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AddLabelResponseType">
    <xs:complexContent>
        <xs:extension base="tns:MemberResponse">
            <xs:sequence>
                <xs:element name="Label" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AppendRevisionDescriptionRequest">
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="Description" type="tns:NonEmptyString" />
                <xs:element minOccurs="0" name="Revision" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The member revision against which the description will be applied. If empty, assumes "headrev".</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AppendRevisionDescriptionResponseType">
    <xs:complexContent>
        <xs:extension base="tns:MemberResponse">
            <xs:sequence>
                <xs:element name="Description" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

</xs:complexContent>
</xs:complexType>
<xs:complexType name="CheckInMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:UpdateMemberRequest">
            <xs:sequence>
                <xs:element minOccurs="0" name="CheckInUnchanged" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>Check in the file even if it has not been changed from the last revision.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="ConvertArchiveOnGovernor" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>Convert the archive to a store-by-reference format if the file exceeds the size governor.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateSubprojectResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element name="Project" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The display name of the root project or subproject in which the subproject is created. The value is the same regardless of whether one subproject was created, or multiple subprojects.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="CanonicalParent" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The canonical name of the project or subproject which is the immediate parent of the subproject that was created. If only one subproject was created, then this is the canonical name of the project specified in the initial request.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Name" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The display name of the top-most subproject created by this command. For example, if the requested subproject was /sub1/sub2/sub3/project.pj, then this value is the complete display name of the project sub3.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="CanonicalName" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The canonical path of the top-most subproject created by this command.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="UpdateRevisionOperation">
    <xs:complexContent>
        <xs:extension base="tns:UpdateMemberOperation">
            <xs:sequence>
                <xs:element minOccurs="0" name="Revision" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The requested member revision. If unspecified, then the member revision is assumed.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="RevisionOperation">
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element minOccurs="0" name="Revision" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The requested member revision. If unspecified, then the member revision is assumed.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="LockMemberResponseType">
    <xs:complexContent>
        <xs:extension base="tns:MemberResponse">
            <xs:sequence>
                <xs:element name="WasBranched" type="xs:boolean" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="FetchMemberResponseType">
    <xs:complexContent>
        <xs:extension base="tns:MemberResponse">
            <xs:sequence>
                <xs:element name="File" ns2:expectedContentTypes="application/octet-stream" type="xs:base64Binary"
                           xmlns:ns2="http://www.w3.org/2005/05/xmlmime"
                />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

</xs:complexContent>
</xs:complexType>
<xs:complexType name="GetSubprojectListResponseType">
    <xs:complexContent>
        <xs:extension base="tns:StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Project" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The list of subprojects.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetMemberListResponseType">
    <xs:complexContent>
        <xs:extension base="tns:StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Member" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The list of members.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetMemberInfoRequest">
    <xs:annotation>
        <xs:documentation>If the list of fields is not specified in the Details element then the following fields are returned: MemberRevision, Author, CheckInDate and Locker.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="Details" type="tns:MemberInfoDetails" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="InfoRequestDetails">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="Field" type="tns:ProjectInfoEnumeration" />
    </xs:sequence>
    <xs:attribute ref="tns:DateTimeFormat" use="optional" />
</xs:complexType>
<xs:simpleType name="ProjectInfoEnumeration">
    <xs:restriction base="xs:string">

```

```

<xs:enumeration value="Attributes" />
<xs:enumeration value="ChangePackageId" />
<xs:enumeration value="CreationChangePackageId" />
<xs:enumeration value="Frozen" />
<xs:enumeration value="HasNewRevision" />
<xs:enumeration value="Labels" />
<xs:enumeration value="Locker" />
<xs:enumeration value="LockedViaChangePackage" />
<xs:enumeration value="MemberArchive" />
<xs:enumeration value="MemberDescription" />
<xs:enumeration value="MemberRevision" />
<xs:enumeration value="MemberTimeStamp" />
<xs:enumeration value="MemberName" />
<xs:enumeration value="NewRevision" />
<xs:enumeration value="PendingChangePackageId" />
<xs:enumeration value="SharedArchive" />
<xs:enumeration value="State" />
<xs:enumeration value="SymbolicLink" />
<xs:enumeration value="Type" />
</xs:restriction>
</xs:simpleType>
<xs:complexType name="MemberInfoDetails">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="Field" type="tns:MemberInfoEnumeration" />
</xs:sequence>
<xs:attribute ref="tns:DateTimeFormat" use="optional" />
</xs:complexType>
<xs:simpleType name="MemberInfoEnumeration">
<xs:restriction base="xs:string">
<xs:enumeration value="Attributes" />
<xs:enumeration value="Author" />
<xs:enumeration value="ChangePackageID" />
<xs:enumeration value="ChangePackageSummary" />
<xs:enumeration value="CheckInDate" />
<xs:enumeration value="Description" />
<xs:enumeration value="Frozen" />
<xs:enumeration value="Labels" />
<xs:enumeration value="Locker" />
<xs:enumeration value="LockTimestamp" />
<xs:enumeration value="LockSandbox" />
<xs:enumeration value="LockHost" />
<xs:enumeration value="MemberName" />
<xs:enumeration value="MemberRevision" />
<xs:enumeration value="Merge" />
<xs:enumeration value="ProjectName" />
<xs:enumeration value="RevisionRule" />

```

```

<xs:enumeration value="State" />
</xs:restriction>
</xs:simpleType>
<xs:complexType name="GetMemberInfoResponseType">
    <xs:complexContent>
        <xs:extension base="tns:StatusResponse">
            <xs:sequence>
                <xs:element ref="tns:Fields">
                    <xs:annotation>
                        <xs:documentation>The list of field attributes for the requested object.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetRevisionInfoRequest">
    <xs:annotation>
        <xs:documentation>If the list of fields is not specified in the Details element then the following fields are returned: Revision, Author, CheckInDate, IsLocked, IsLockedExclusively, Locker and LockedViaChangePackage.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="Details" type="tns:RevisionInfoDetails" />
                <xs:element minOccurs="0" name="Revision" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>Must be a single value (eg., 1.3.1.4)</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="RevisionInfoDetails">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="Field" type="tns:RevisionInfoEnumeration" />
    </xs:sequence>
    <xs:attribute ref="tns:DateTimeFormat" use="optional" />
</xs:complexType>
<xs:simpleType name="RevisionInfoEnumeration">
    <xs:restriction base="xs:string">
        <xs:enumeration value="ArchiveDescription" />
        <xs:enumeration value="ArchiveName" />
        <xs:enumeration value="Attributes" />
        <xs:enumeration value="Author" />
        <xs:enumeration value="Binary" />
        <xs:enumeration value="BranchCount" />
        <xs:enumeration value="BranchID" />
    </xs:restriction>
</xs:simpleType>

```

```

<xs:enumeration value="ChangePackageID" />
<xs:enumeration value="ChangePackageSummary" />
<xs:enumeration value="CheckInDate" />
<xs:enumeration value="Description" />
<xs:enumeration value="ExclusiveLockMandatory" />
<xs:enumeration value="Frozen" />
<xs:enumeration value="HasLabels" />
<xs:enumeration value="HeadRevision" />
<xs:enumeration value="IsLocked" />
<xs:enumeration value="IsLockedExclusively" />
<xs:enumeration value="IsReference" />
<xs:enumeration value="Labels" />
<xs:enumeration value="LinesAdded" />
<xs:enumeration value="LinesDeleted" />
<xs:enumeration value="LockedViaChangePackage" />
<xs:enumeration value="Locker" />
<xs:enumeration value="MemberName" />
<xs:enumeration value="MemberNameRelativeToProject" />
<xs:enumeration value="MemberRevision" />
<xs:enumeration value="MemberRevisionNotOnTip" />
<xs:enumeration value="MemberRule" />
<xs:enumeration value="ProjectName" />
<xs:enumeration value="ProjectReferences" />
<xs:enumeration value="ProjectType" />
<xs:enumeration value="Revision" />
<xs:enumeration value="RevisionCount" />
<xs:enumeration value="SharedArchive" />
<xs:enumeration value="State" />
<xs:enumeration value="StorageFormat" />
<xs:enumeration value="TipRevisionOnBranch" />
<xs:enumeration value="TrunkTipRevision" />
</xs:restriction>
</xs:simpleType>
<xs:complexType name="GetRevisionInfoResponseType">
<xs:complexContent>
<xs:extension base="tns>StatusResponse">
<xs:sequence>
<xs:element name="ChangePackageID" type="xs:string" />
<xs:element ref="tns:Fields">
<xs:annotation>
<xs:documentation>The list of field attributes for the requested object.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>

```

```

</xs:complexType>
<xs:complexType name="GetMemberHistoryRequest">
    <xs:annotation>
        <xs:documentation>If the list of fields is not specified in the Details element then the following fields are returned: Revision, Author, CheckInDate, IsLocked, IsLockedExclusively, Locker and LockedViaChangePackage.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="Details" type="tns:RevisionInfoDetails" />
                <xs:element name="RevisionSelection" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>Can be either all, or range:low-high, or branchrange:low-high, or daterange:low-high | :past: <nb>:years|months|days|hours, or branch:name, or labeled[:name], or labellike:regexp, or locked[:me]:anyone|username], or locktype[:exclusive]:nonexclusive|any], or state:name, or author[:me]:username], or pending, or anyspecial, or branchboundaries.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Filter" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>Filters that can be applied to the set of revisions returned from the Revision option. For more details please refer to the MKS documentation. Valid syntax includes the following phrases: attribute:name[=value], changed[:working]:sync|newer|size|missing|newmem|all], frozen, label[:name], locked[:name], locktype[:exclusive]:nonexclusive|any], state[:name], format[:text]:binary], workingbranch, anylabel[:name], deferred[:add]:addfromarchive|checkin|drop|import|move|rename|updaterevision|:all], unresolvedmerges, memberonbranch, pending[:add]:addfromarchive|drop|import|movememberfrom|:movememberto|renamefrom|renameto|update|updaterevision|:all], workInProgress, sparsecontents, rule[:defined]:invalid|memberrevdiffers], archiveshared, caseinsensitivefile:expression, outofscope. Note that the command line interface allows you to specify a file. This is not allowed in a non-sandbox scenario such as web services and is explicitly prevented from happening.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetMemberHistoryResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="RevisionInfo" type="tns:GetRevisionInfoResponseType" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateChangePackageRequest">
    <xs:annotation>
        <xs:documentation>Creates a change package associated with the indicated Integrity item.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>

```

```

<xs:element name="ItemID" type="tns:PositiveInteger" />
<xs:element minOccurs="0" name="Summary" type="xs:string" />
<xs:element minOccurs="0" name="Description" type="xs:string" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateChangePackageResponseType">
<xs:complexContent>
<xs:extension base="tns>StatusResponse">
<xs:sequence>
<xs:element name="ChangePackageID" type="xs:string">
<xs:annotation>
<xs:documentation>The new change package ID.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ChangePackageRequestType">
<xs:complexContent>
<xs:extension base="tns:Credentials">
<xs:sequence>
<xs:element name="ChangePackageID" type="tns:NonEmptyString" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="GetProjectInfoType">
<xs:annotation>
<xs:documentation>If the field list is not specified in the Details element then the following fields are returned: MemberName, MemberTimestamp and MemberRevision. Note that all of those fields, containing project and project path are automatically included in the output schema.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="tns:ProjectOperation">
<xs:sequence>
<xs:element name="Details" type="tns:InfoRequestDetails" />
</xs:sequence>
<xs:attribute name="recurse" type="xs:boolean">
<xs:annotation>
<xs:documentation>If true, then this method recurses into subprojects. Default is false.</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="GetProjectInfoResponseType">

```

```

<xs:complexContent>
  <xs:extension base="tns>StatusResponse">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="MemberDetails">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Project" type="xs:string" />
            <xs:element name="Path" type="xs:string" />
            <xs:element name="MemberName" type="xs:string" />
            <xs:element minOccurs="0" name="MemberRevision" type="xs:string">
              <xs:annotation>
                <xs:documentation>The member revision for this project member.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element minOccurs="0" name="MemberTimeStamp" type="xs:string">
              <xs:annotation>
                <xs:documentation>The timestamp for the member revision in the format specified by the DateOfBirth attribute in the request.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element ref="tns:Fields">
              <xs:annotation>
                <xs:documentation>The list of field attributes for the member.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
<xs:complexType name="GetProjectConfigurationPathType">
  <xs:annotation>
    <xs:documentation>Returns the unique name for the project. The name can be used to specify the exact context in which a member or subproject can be found.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="tns:ProjectOperation">
      <xs:choice>
        <xs:element minOccurs="0" name="DevelopmentPath" type="tns:NonEmptyString">
          <xs:annotation>
            <xs:documentation>The development path in which the archive is located.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="ProjectRevision" type="tns:NonEmptyString">
          <xs:annotation>
            <xs:documentation>The project revision in which the archive can be found.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:element>
    </xs:choice>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="GetProjectConfigurationPathResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element name="ProjectConfigurationPath" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="LockRecord">
    <xs:annotation>
        <xs:documentation>Revision lock details.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Locker" type="xs:string">
            <xs:annotation>
                <xs:documentation>The user name for the user holding the lock.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="LockerFullName" type="xs:string">
            <xs:annotation>
                <xs:documentation>The full name for the user holding the lock, if available.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Exclusive" type="xs:boolean">
            <xs:annotation>
                <xs:documentation>True if the lock is exclusive, false if non-exclusive.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Revision" type="xs:string">
            <xs:annotation>
                <xs:documentation>Revision against which the lock record was obtained.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="DateTime" type="xs:string">
            <xs:annotation>
                <xs:documentation>Timestamp of the lock record, in the format specified by the DateFormat in the request. See  

http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html for full details. If not specified in the request the default format  

is "yyyy-MM-dd HH:mm:ss Z" which would give you for example, "2010-09-25 14:39:22 -0500"</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="ChangePackageId" type="xs:string" />
        <xs:element minOccurs="0" name="DevelopmentPath" type="xs:string">
            <xs:annotation>

```

```

<xs:documentation>The development path in which the lock was obtained. Unspecified implies headrev.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="Sandbox" type="xs:string">
    <xs:annotation>
        <xs:documentation>The sandbox name for the user holding the lock. Unspecified implies it is a project lock.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="CheckpointProjectType">
    <xs:complexContent>
        <xs:extension base="tns:ProjectOperation">
            <xs:sequence>
                <xs:element minOccurs="0" name="Description" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The optional checkpoint description.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="Label" type="tns:NonEmptyString" use="optional">
                <xs:annotation>
                    <xs:documentation>The optional label for the checkpoint. Note all colons : are replaced with underscores _</xs:documentation>
                </xs:annotation>
            </xs:attribute>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectLabelExistsType">
    <xs:complexContent>
        <xs:extension base="tns:ProjectOperation">
            <xs:sequence>
                <xs:element name="Label" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The label to search for among all the checkpoints of the project. Note all colons : in the requested label are replaced with underscores _ prior to searching for the label.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="SubprojectRevisionOperation">
    <xs:complexContent>
        <xs:extension base="tns:UpdateSubprojectOperation">
            <xs:sequence>
                <xs:element name="Revision" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The revision identifier for the build subproject. This may be a literal revision number (e.g., 1.6), a

```

```

        "symbolic revision id" (e.g., :head, :trunktip) or a label (since a label uniquely identifies a revision within an
        archive).</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ShareSubprojectOperation">
    <xs:complexContent>
        <xs:extension base="tns:UpdateSubprojectOperation">
            <xs:sequence>
                <xs:element minOccurs="1" name="SharedSubproject" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The subproject to be shared.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Revision" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The revision identifier for the build subproject. This may be a literal revision number (e.g., 1.6), a
                            "symbolic revision id" (e.g., :head, :trunktip) or a label (since a label uniquely identifies a revision within an archive).
                            If not specified, then the subproject is shared as regular subproject, not a build subproject.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:attribute name="DateTimeFormat">
    <xs:annotation>
        <xs:documentation>The format of timestamps in the output. See http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html for full
            details. The default format is "yyyy-MM-dd HH:mm:ss Z" which would give you for example, "2010-09-25 14:39:22 -0500"</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="tns:NonEmptyString" />
    </xs:simpleType>
</xs:attribute>
<xs:simpleType name="NonEmptyString">
    <xs:restriction base="xs:string">
        <xs:minLength value="1" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PositiveInteger">
    <xs:restriction base="xs:integer">
        <xs:minInclusive value="1" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/2009/7/Source/fault" version="1.0">
    <xs:element name="MKSEception" nillable="true" type="tns:MKSFault" />

```

```

<xs:complexType name="MKSFault">
    <xs:annotation>
        <xs:documentation>The original fault generated from MKS Integrity. The element value is the full description of the error.</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="class" type="xs:string">
                <xs:annotation>
                    <xs:documentation>The class of exception</xs:documentation>
                </xs:annotation>
            </xs:attribute>
            <xs:attribute name="id" type="xs:string">
                <xs:annotation>
                    <xs:documentation>The unique MKS identifier for this error, generally a 6-digit positive integer.</xs:documentation>
                </xs:annotation>
            </xs:attribute>
            <xs:attribute name="implication" type="xs:string">
                <xs:annotation>
                    <xs:documentation>The implication from the exception</xs:documentation>
                </xs:annotation>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/2009/7/Source" version="1.0">
    <xs:import namespace="http://webservice.mks.com/2009/7/Source/schema" />
    <xs:element name="addLabel" type="tns:addLabel" />
    <xs:element name="addLabelResponse" type="tns:addLabelResponse" />
    <xs:element name="createMember" type="tns:createMember" />
    <xs:element name="createMemberResponse" type="tns:createMemberResponse" />
    <xs:element name="appendRevisionDescription" type="tns:appendRevisionDescription" />
    <xs:element name="appendRevisionDescriptionResponse" type="tns:appendRevisionDescriptionResponse" />
    <xs:element name="checkInMember" type="tns:checkInMember" />
    <xs:element name="checkInMemberResponse" type="tns:checkInMemberResponse" />
    <xs:element name="checkpointProject" type="tns:checkpointProject" />
    <xs:element name="checkpointProjectResponse" type="tns:checkpointProjectResponse" />
    <xs:element name="createSubproject" type="tns:createSubproject" />
    <xs:element name="createSubprojectResponse" type="tns:createSubprojectResponse" />
    <xs:element name="dropMember" type="tns:dropMember" />
    <xs:element name="dropMemberResponse" type="tns:dropMemberResponse" />
    <xs:element name="fetchMember" type="tns:fetchMember" />
    <xs:element name="fetchMemberResponse" type="tns:fetchMemberResponse" />
    <xs:element name="getMemberInfo" type="tns:getMemberInfo" />
    <xs:element name="getMemberInfoResponse" type="tns:getMemberInfoResponse" />
    <xs:element name="getRevisionInfo" type="tns:getRevisionInfo" />
    <xs:element name="getRevisionInfoResponse" type="tns:getRevisionInfoResponse" />

```

```

<xs:element name="getMemberHistory" type="tns:getMemberHistory" />
<xs:element name="getMemberHistoryResponse" type="tns:getMemberHistoryResponse" />
<xs:element name="getMemberList" type="tns:getMemberList" />
<xs:element name="getMemberListResponse" type="tns:getMemberListResponse" />
<xs:element name="getProjectInfo" type="tns:getProjectInfo" />
<xs:element name="getProjectInfoResponse" type="tns:getProjectInfoResponse" />
<xs:element name="getProjectConfigurationPath" type="tns:getProjectConfigurationPath" />
<xs:element name="getProjectConfigurationPathResponse" type="tns:getProjectConfigurationPathResponse" />
<xs:element name="getSubprojectList" type="tns:getSubprojectList" />
<xs:element name="getSubprojectListResponse" type="tns:getSubprojectListResponse" />
<xs:element name="lockMember" type="tns:lockMember" />
<xs:element name="lockMemberResponse" type="tns:lockMemberResponse" />
<xs:element name="projectLabelExists" type="tns:projectLabelExists" />
<xs:element name="projectLabelExistsResponse" type="tns:projectLabelExistsResponse" />
<xs:element name="setSubprojectRevision" type="tns:setSubprojectRevision" />
<xs:element name="setSubprojectRevisionResponse" type="tns:setSubprojectRevisionResponse" />
<xs:element name="shareSubproject" type="tns:shareSubproject" />
<xs:element name="shareSubprojectResponse" type="tns:shareSubprojectResponse" />
<xs:element name="unlockMember" type="tns:unlockMember" />
<xs:element name="unlockMemberResponse" type="tns:unlockMemberResponse" />
<xs:element name="updateRevision" type="tns:updateRevision" />
<xs:element name="updateRevisionResponse" type="tns:updateRevisionResponse" />
<xs:element name="checkConnection" type="tns:checkConnection" />
<xs:element name="checkConnectionResponse" type="tns:checkConnectionResponse" />
<xs:element name="createChangePackage" type="tns:createChangePackage" />
<xs:element name="createChangePackageResponse" type="tns:createChangePackageResponse" />
<xs:element name="discardChangePackage" type="tns:discardChangePackage" />
<xs:element name="discardChangePackageResponse" type="tns:discardChangePackageResponse" />
<xs:element name="submitChangePackage" type="tns:submitChangePackage" />
<xs:element name="submitChangePackageResponse" type="tns:submitChangePackageResponse" />
<xs:complexType name="createMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CreateMemberRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:MemberResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="addLabel">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AddLabelRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="addLabelResponse">

```

```
<xs:sequence>
    <xs:element minOccurs="0" name="return" type="ns1:AddLabelResponseType" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="appendRevisionDescription">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AppendRevisionDescriptionRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="appendRevisionDescriptionResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AppendRevisionDescriptionResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkInMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CheckInMemberRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkInMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AppendRevisionDescriptionResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createSubproject">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:UpdateSubprojectOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createSubprojectResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>CreateSubprojectResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="lockMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:UpdateMemberOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="lockMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:LockMemberResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:RevisionOperation" />
    </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="fetchMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:FetchMemberResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="unlockMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:MemberOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="unlockMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:MemberResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getSubprojectList">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ProjectOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getSubprojectListResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetSubprojectListResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMemberList">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ProjectOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMemberListResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetMemberListResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMemberInfo">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetMemberInfoRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMemberInfoResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetMemberInfoResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getRevisionInfo">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetRevisionInfoRequest" />
    </xs:sequence>
</xs:complexType>
```

```
</xs:complexType>
<xs:complexType name="getRevisionInfoResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetRevisionInfoResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMemberHistory">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetMemberHistoryRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMemberHistoryResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetMemberHistoryResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjectInfo">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetProjectInfoType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjectInfoResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetProjectInfoResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjectConfigurationPath">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetProjectConfigurationPathType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjectConfigurationPathResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetProjectConfigurationPathResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkpointProject">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CheckpointProjectType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkpointProjectResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="projectLabelExists">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ProjectLabelExistsType" />
```

```
</xs:sequence>
</xs:complexType>
<xs:complexType name="projectLabelExistsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="dropMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:UpdateMemberOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="dropMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="setSubprojectRevision">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:SubprojectRevisionOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="setSubprojectRevisionResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="updateRevision">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:UpdateRevisionOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="updateRevisionResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="shareSubproject">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ShareSubprojectOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="shareSubprojectResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkConnection">
    <xs:sequence>
```

```

<xs:element minOccurs="0" name="arg0" type="ns1:Credentials" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="checkConnectionResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createChangePackage">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>CreateChangePackageRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createChangePackageResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>CreateChangePackageResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="discardChangePackage">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ChangePackageRequestType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="discardChangePackageResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="submitChangePackage">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ChangePackageRequestType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="submitChangePackageResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="Source_2009_7_setSubprojectRevision">
    <part element="tns:setSubprojectRevision" name="setSubprojectRevision" />
</message>
<message name="Source_2009_7_projectLabelExistsResponse">
    <part element="tns:projectLabelExistsResponse" name="projectLabelExistsResponse" />
</message>
<message name="Source_2009_7_getMemberInfo">
    <part element="tns:getMemberInfo" name="getMemberInfo" />
</message>

```

```
<message name="Source_2009_7_checkInMember">
    <part element="tns:checkInMember" name="checkInMember" />
</message>
<message name="Source_2009_7_discardChangePackage">
    <part element="tns:discardChangePackage" name="discardChangePackage" />
</message>
<message name="Source_2009_7_getProjectConfigurationPath">
    <part element="tns:getProjectConfigurationPath" name="getProjectConfigurationPath" />
</message>
<message name="Source_2009_7_createMemberResponse">
    <part element="tns:createMemberResponse" name="createMemberResponse" />
</message>
<message name="MKSException">
    <part element="ns2:MKSException" name="MKSException" />
</message>
<message name="Source_2009_7_addLabel">
    <part element="tns:addLabel" name="addLabel" />
</message>
<message name="Source_2009_7_unlockMemberResponse">
    <part element="tns:unlockMemberResponse" name="unlockMemberResponse" />
</message>
<message name="Source_2009_7_checkpointProject">
    <part element="tns:checkpointProject" name="checkpointProject" />
</message>
<message name="Source_2009_7_getRevisionInfoResponse">
    <part element="tns:getRevisionInfoResponse" name="getRevisionInfoResponse" />
</message>
<message name="Source_2009_7_getSubprojectList">
    <part element="tns:getSubprojectList" name="getSubprojectList" />
</message>
<message name="Source_2009_7_fetchMember">
    <part element="tns:fetchMember" name="fetchMember" />
</message>
<message name="Source_2009_7_checkConnection">
    <part element="tns:checkConnection" name="checkConnection" />
</message>
<message name="Source_2009_7_createSubproject">
    <part element="tns:createSubproject" name="createSubproject" />
</message>
<message name="Source_2009_7_dropMember">
    <part element="tns:dropMember" name="dropMember" />
</message>
<message name="Source_2009_7_createMember">
    <part element="tns:createMember" name="createMember" />
</message>
<message name="Source_2009_7_getMemberInfoResponse">
    <part element="tns:getMemberInfoResponse" name="getMemberInfoResponse" />
</message>
```

```
<message name="Source_2009_7_getMemberListResponse">
    <part element="tns:getMemberListResponse" name="getMemberListResponse" />
</message>
<message name="Source_2009_7_getProjectConfigurationPathResponse">
    <part element="tns:getProjectConfigurationPathResponse" name="getProjectConfigurationPathResponse" />
</message>
<message name="Source_2009_7_getProjectInfoResponse">
    <part element="tns:getProjectInfoResponse" name="getProjectInfoResponse" />
</message>
<message name="Source_2009_7_submitChangePackageResponse">
    <part element="tns:submitChangePackageResponse" name="submitChangePackageResponse" />
</message>
<message name="Source_2009_7_dropMemberResponse">
    <part element="tns:dropMemberResponse" name="dropMemberResponse" />
</message>
<message name="Source_2009_7_getMemberHistoryResponse">
    <part element="tns:getMemberHistoryResponse" name="getMemberHistoryResponse" />
</message>
<message name="Source_2009_7_updateRevision">
    <part element="tns:updateRevision" name="updateRevision" />
</message>
<message name="Source_2009_7_setSubprojectRevisionResponse">
    <part element="tns:setSubprojectRevisionResponse" name="setSubprojectRevisionResponse" />
</message>
<message name="Source_2009_7_getMemberHistory">
    <part element="tns:getMemberHistory" name="getMemberHistory" />
</message>
<message name="Source_2009_7_fetchMemberResponse">
    <part element="tns:fetchMemberResponse" name="fetchMemberResponse" />
</message>
<message name="Source_2009_7_checkpointProjectResponse">
    <part element="tns:checkpointProjectResponse" name="checkpointProjectResponse" />
</message>
<message name="Source_2009_7_updateRevisionResponse">
    <part element="tns:updateRevisionResponse" name="updateRevisionResponse" />
</message>
<message name="Source_2009_7_createChangePackageResponse">
    <part element="tns:createChangePackageResponse" name="createChangePackageResponse" />
</message>
<message name="Source_2009_7_checkConnectionResponse">
    <part element="tns:checkConnectionResponse" name="checkConnectionResponse" />
</message>
<message name="Source_2009_7_submitChangePackage">
    <part element="tns:submitChangePackage" name="submitChangePackage" />
</message>
<message name="Source_2009_7_appendRevisionDescriptionResponse">
    <part element="tns:appendRevisionDescriptionResponse" name="appendRevisionDescriptionResponse" />
</message>
```

```
<message name="Source_2009_7_shareSubproject">
    <part_element="tns:shareSubproject" name="shareSubproject" />
</message>
<message name="Source_2009_7_addLabelResponse">
    <part_element="tns:addLabelResponse" name="addLabelResponse" />
</message>
<message name="Source_2009_7_getSubprojectListResponse">
    <part_element="tns:getSubprojectListResponse" name="getSubprojectListResponse" />
</message>
<message name="Source_2009_7_createChangePackage">
    <part_element="tns:createChangePackage" name="createChangePackage" />
</message>
<message name="Source_2009_7_projectLabelExists">
    <part_element="tns:projectLabelExists" name="projectLabelExists" />
</message>
<message name="Source_2009_7_createSubprojectResponse">
    <part_element="tns:createSubprojectResponse" name="createSubprojectResponse" />
</message>
<message name="Source_2009_7_unlockMember">
    <part_element="tns:unlockMember" name="unlockMember" />
</message>
<message name="Source_2009_7_appendRevisionDescription">
    <part_element="tns:appendRevisionDescription" name="appendRevisionDescription" />
</message>
<message name="Source_2009_7_getRevisionInfo">
    <part_element="tns:getRevisionInfo" name="getRevisionInfo" />
</message>
<message name="Source_2009_7_lockMemberResponse">
    <part_element="tns:lockMemberResponse" name="lockMemberResponse" />
</message>
<message name="Source_2009_7_discardChangePackageResponse">
    <part_element="tns:discardChangePackageResponse" name="discardChangePackageResponse" />
</message>
<message name="Source_2009_7_getProjectInfo">
    <part_element="tns:getProjectInfo" name="getProjectInfo" />
</message>
<message name="Source_2009_7_lockMember">
    <part_element="tns:lockMember" name="lockMember" />
</message>
<message name="Source_2009_7_shareSubprojectResponse">
    <part_element="tns:shareSubprojectResponse" name="shareSubprojectResponse" />
</message>
<message name="Source_2009_7_checkInMemberResponse">
    <part_element="tns:checkInMemberResponse" name="checkInMemberResponse" />
</message>
<message name="Source_2009_7_getMemberList">
    <part_element="tns:getMemberList" name="getMemberList" />
</message>
```

```

<portType name="Source_2009_7">
    <operation name="addLabel" parameterOrder="addLabel">
        <documentation>Uses the "addlabel" MKS command. Adds a label to a particular revision for a project member.</documentation>
        <input message="tns:Source_2009_7_addLabel" />
        <output message="tns:Source_2009_7_addLabelResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="createMember" parameterOrder="createMember">
        <documentation>Uses the "projectadd" MKS command. Adds a member to a project.</documentation>
        <input message="tns:Source_2009_7_createMember" />
        <output message="tns:Source_2009_7_createMemberResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="appendRevisionDescription" parameterOrder="appendRevisionDescription">
        <documentation>Uses the "appendrevdesc" MKS command. Adds a description to a revision of a member of a project.</documentation>
        <input message="tns:Source_2009_7_appendRevisionDescription" />
        <output message="tns:Source_2009_7_appendRevisionDescriptionResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="checkInMember" parameterOrder="checkInMember">
        <documentation>Uses the "projectci" MKS command. Checks in changes to a member of a project.</documentation>
        <input message="tns:Source_2009_7_checkInMember" />
        <output message="tns:Source_2009_7_checkInMemberResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="checkpointProject" parameterOrder="checkpointProject">
        <documentation>Uses the "checkpoint" MKS command. Creates a new project revision and associates all current member revisions with the checkpoint.</documentation>
        <input message="tns:Source_2009_7_checkpointProject" />
        <output message="tns:Source_2009_7_checkpointProjectResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="createSubproject" parameterOrder="createSubproject">
        <documentation>Uses the "createsubproject" MKS command. Creates a new subproject as a child of the indicated project or subproject.</documentation>
        <input message="tns:Source_2009_7_createSubproject" />
        <output message="tns:Source_2009_7_createSubprojectResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="dropMember" parameterOrder="dropMember">
        <documentation>Uses the "drop" MKS command. Removes the member from the project.</documentation>
        <input message="tns:Source_2009_7_dropMember" />
        <output message="tns:Source_2009_7_dropMemberResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="fetchMember" parameterOrder="fetchMember">
        <documentation>Uses the "projectco" MKS command. Returns the contents of a member of a project.</documentation>
        <input message="tns:Source_2009_7_fetchMember" />

```

```

<output message="tns:Source_2009_7_fetchMemberResponse" />
<fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getMemberInfo" parameterOrder="getMemberInfo">
    <documentation>Uses the "memberinfo" MKS command. Returns metadata about the project member.</documentation>
    <input message="tns:Source_2009_7_getMemberInfo" />
    <output message="tns:Source_2009_7_getMemberInfoResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getRevisionInfo" parameterOrder="getRevisionInfo">
    <documentation>Uses the "rlog" MKS command. Returns metadata about the project member, if a specific revision is requested it returns metadata for the requested revision otherwise metadata is returned for the member revision.</documentation>
    <input message="tns:Source_2009_7_getRevisionInfo" />
    <output message="tns:Source_2009_7_getRevisionInfoResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getMemberHistory" parameterOrder="getMemberHistory">
    <documentation>Uses the "rlg" MKS command. Returns metadata about the revisions.</documentation>
    <input message="tns:Source_2009_7_getMemberHistory" />
    <output message="tns:Source_2009_7_getMemberHistoryResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getMemberList" parameterOrder="getMemberList">
    <documentation>Uses the "viewproject" MKS command. Returns the list of project members not including the subprojects.</documentation>
    <input message="tns:Source_2009_7_getMemberList" />
    <output message="tns:Source_2009_7_getMemberListResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getProjectInfo" parameterOrder="getProjectInfo">
    <documentation>Uses the "viewproject" MKS command. Returns project metadata.</documentation>
    <input message="tns:Source_2009_7_getProjectInfo" />
    <output message="tns:Source_2009_7_getProjectInfoResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getProjectConfigurationPath" parameterOrder="getProjectConfigurationPath">
    <input message="tns:Source_2009_7_getProjectConfigurationPath" />
    <output message="tns:Source_2009_7_getProjectConfigurationPathResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="getSubprojectList" parameterOrder="getSubprojectList">
    <documentation>Uses the "viewproject" MKS command. Returns the list of subprojects not including the members.</documentation>
    <input message="tns:Source_2009_7_getSubprojectList" />
    <output message="tns:Source_2009_7_getSubprojectListResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="lockMember" parameterOrder="lockMember">
    <documentation>Uses the "lock" MKS command. Creates an exclusive lock on the member.</documentation>

```

```

<input message="tns:Source_2009_7_lockMember" />
<output message="tns:Source_2009_7_lockMemberResponse" />
<fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="projectLabelExists" parameterOrder="projectLabelExists">
    <documentation>Uses the "viewprojecthistory" MKS command. Determines if a label has been applied to any revision in the project's history.</documentation>
    <input message="tns:Source_2009_7_projectLabelExists" />
    <output message="tns:Source_2009_7_projectLabelExistsResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="setSubprojectRevision" parameterOrder="setSubprojectRevision">
    <documentation>Uses the "configuresubproject" MKS command. Sets the revision number for a subproject.</documentation>
    <input message="tns:Source_2009_7_setSubprojectRevision" />
    <output message="tns:Source_2009_7_setSubprojectRevisionResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="shareSubproject" parameterOrder="shareSubproject">
    <documentation>Uses the "sharesubproject" MKS command. Shares the subproject as a build subproject as of a specific revision.</documentation>
    <input message="tns:Source_2009_7_shareSubproject" />
    <output message="tns:Source_2009_7_shareSubprojectResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="unlockMember" parameterOrder="unlockMember">
    <documentation>Uses the "unlock" MKS command. Removes the lock on the specified member.</documentation>
    <input message="tns:Source_2009_7_unlockMember" />
    <output message="tns:Source_2009_7_unlockMemberResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="updateRevision" parameterOrder="updateRevision">
    <documentation>Uses the "updaterevision" MKS command. Sets the member revision to the specified revision.</documentation>
    <input message="tns:Source_2009_7_updateRevision" />
    <output message="tns:Source_2009_7_updateRevisionResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="checkConnection" parameterOrder="checkConnection">
    <documentation>Uses the "connect" MKS command. Validates credentials and connection information. This method is not required to initiate a session. Session management is handled automatically by the web services.</documentation>
    <input message="tns:Source_2009_7_checkConnection" />
    <output message="tns:Source_2009_7_checkConnectionResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="createChangePackage" parameterOrder="createChangePackage">
    <documentation>Uses the "createcp" MKS command. Creates a change package attached to the indicated Integrity item.</documentation>
    <input message="tns:Source_2009_7_createChangePackage" />
    <output message="tns:Source_2009_7_createChangePackageResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>

```

```

</operation>
<operation name="discardChangePackage" parameterOrder="discardChangePackage">
    <documentation>Uses the "discardcp" MKS command. Discards a change package attached to the indicated Integrity item.</documentation>
    <input message="tns:Source_2009_7_discardChangePackage" />
    <output message="tns:Source_2009_7_discardChangePackageResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="submitChangePackage" parameterOrder="submitChangePackage">
    <documentation>Uses the "submitcp" MKS command. Closes the change package. All entries must be pending on the server (via the checkInMember operation) prior to executing this command.</documentation>
    <input message="tns:Source_2009_7_submitChangePackage" />
    <output message="tns:Source_2009_7_submitChangePackageResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
</portType>
<binding name="Source_2009_7Binding" type="tns:Source_2009_7">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="createMember">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>
    <operation name="addLabel">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>
    <operation name="appendRevisionDescription">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>

```

```
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="checkInMember">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="checkpointProject">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="createSubproject">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="dropMember">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
```

```
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="fetchMember">
<soap:operation soapAction="" />
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="getMemberInfo">
<soap:operation soapAction="" />
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="getRevisionInfo">
<soap:operation soapAction="" />
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="getMemberHistory">
<soap:operation soapAction="" />
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
```

```
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="getMemberList">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getProjectInfo">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getProjectConfigurationPath">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getSubprojectList">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
```

```
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="lockMember">
<soap:operation soapAction="" />
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="projectLabelExists">
<soap:operation soapAction="" />
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="setSubprojectRevision">
<soap:operation soapAction="" />
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="shareSubproject">
<soap:operation soapAction="" />
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
```

```
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="unlockMember">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="updateRevision">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="checkConnection">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="createChangePackage">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
```

```
</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="discardChangePackage">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="submitChangePackage">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
</binding>
<service name="Source_2009_7Service">
    <port binding="tns:Source_2009_7Binding" name="Source_2009_7Port">
        <soap:address location="http://IntegrityServer:9001/webservices/2009/7/Source/" />
    </port>
</service>
</definitions>
```

Web Service: Source_2009Service

Web Service: Source_2009Service

Target Namespace: <http://webservice.mks.com/2009/Source>

Port **Source_2009Port** *Port type* [Source code](#)

Location: <http://IntegrityServer:9001/webservices/2009/Source/>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. [addLabel](#) [Detail](#) [Source code](#)
2. [addMember](#) [Detail](#) [Source code](#)
3. [addRevisionDescription](#) [Detail](#) [Source code](#)
4. [checkInMember](#) [Detail](#) [Source code](#)
5. [checkpointProject](#) [Detail](#) [Source code](#)
6. [createSubproject](#) [Detail](#) [Source code](#)
7. [dropMember](#) [Detail](#) [Source code](#)
8. [fetchRevision](#) [Detail](#) [Source code](#)
9. [getMemberInfo](#) [Detail](#) [Source code](#)
10. [getMembers](#) [Detail](#) [Source code](#)
11. [getProjectDetails](#) [Detail](#) [Source code](#)
12. [getSubprojects](#) [Detail](#) [Source code](#)
13. [lockMember](#) [Detail](#) [Source code](#)
14. [projectLabelExists](#) [Detail](#) [Source code](#)
15. [setSubprojectRevision](#) [Detail](#) [Source code](#)
16. [shareSubproject](#) [Detail](#) [Source code](#)
17. [unlockRevision](#) [Detail](#) [Source code](#)
18. [updateRevision](#) [Detail](#) [Source code](#)

Operations

Port type **Source_2009** [Source code](#)

1. addLabel

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_addLabel (soap:body, use = literal) [Source code](#)

addLabel type *addLabel*

Output: Source_2009_addLabelResponse (soap:body, use = literal) [Source code](#)

addLabelResponse type *addLabelResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*

The class of exception

- o id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- o implication type *string*

The implication from the exception

2. addMember

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_addMember (soap:body, use = literal) [Source code](#)

addMember type *addMember*

Output: Source_2009_addMemberResponse (soap:body, use = literal) [Source code](#)

addMemberResponse type *addMemberResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*

The class of exception

- o id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- o implication type *string*

The implication from the exception

3. addRevisionDescription

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_addRevisionDescription (soap:body, use = literal) [Source code](#)

addRevisionDescription type *addRevisionDescription*

Output: Source_2009_addRevisionDescriptionResponse (soap:body, use = literal) [Source code](#)

addRevisionDescriptionResponse type *addRevisionDescriptionResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

4. checkInMember

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_checkInMember (soap:body, use = literal) [Source code](#)

checkInMember type *checkInMember*

Output: Source_2009_checkInMemberResponse (soap:body, use = literal) [Source code](#)

checkInMemberResponse type *checkInMemberResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

5. checkpointProject

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_checkpointProject (soap:body, use = literal) [Source code](#)

checkpointProject type *checkpointProject*

Output: Source_2009_checkpointProjectResponse (soap:body, use = literal) [Source code](#)

checkpointProjectResponse type *checkpointProjectResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

6. createSubproject

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_createSubproject (soap:body, use = literal) [Source code](#)

createSubproject type *createSubproject*

Output: Source_2009_createSubprojectResponse (soap:body, use = literal) [Source code](#)

createSubprojectResponse type *createSubprojectResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

7. dropMember

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_dropMember (soap:body, use = literal) [Source code](#)

dropMember type *dropMember*

Output: Source_2009_dropMemberResponse (soap:body, use = literal) [Source code](#)

dropMemberResponse type *dropMemberResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

8. fetchRevision

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_fetchRevision (soap:body, use = literal) [Source code](#)

fetchRevision type *fetchRevision*

Output: Source_2009_fetchRevisionResponse (soap:body, use = literal) [Source code](#)

fetchRevisionResponse type *fetchRevisionResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

9. getMemberInfo

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_getMemberInfo (soap:body, use = literal) [Source code](#)

getMemberInfo type getMemberInfo

Output: Source_2009_getMemberInfoResponse (soap:body, use = literal) [Source code](#)

getMemberInfoResponse type getMemberInfoResponse

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

10. getMembers

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_getMembers (soap:body, use = literal) [Source code](#)

getMembers type getMembers

Output: Source_2009_getMembersResponse (soap:body, use = literal) [Source code](#)

getMembersResponse type getMembersResponse

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

11. `getProjectDetails`

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: `Source_2009_getProjectDetails` (soap:body, use = literal) [Source code](#)

getProjectDetails type `getProjectDetails`

Output: `Source_2009_getProjectDetailsResponse` (soap:body, use = literal) [Source code](#)

getProjectDetailsResponse type `getProjectDetailsResponse`

Fault: `MKSException` (soap:fault, use = literal) [Source code](#)

MKSException type `MKSException`

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type `string`
The class of exception
- o id type `string`
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type `string`
The implication from the exception

12. `getSubprojects`

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: `Source_2009_getSubprojects` (soap:body, use = literal) [Source code](#)

getSubprojects type `getSubprojects`

Output: `Source_2009_getSubprojectsResponse` (soap:body, use = literal) [Source code](#)

getSubprojectsResponse type `getSubprojectsResponse`

Fault: `MKSException` (soap:fault, use = literal) [Source code](#)

MKSException type `MKSException`

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type `string`
The class of exception
- o id type `string`
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type `string`
The implication from the exception

13. `lockMember`

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: `Source_2009_lockMember` (soap:body, use = literal) [Source code](#)

lockMember type `lockMember`

Output: `Source_2009_lockMemberResponse` (soap:body, use = literal) [Source code](#)

lockMemberResponse type `lockMemberResponse`

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class **type string**
The class of exception
- id **type string**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type string**
The implication from the exception

14. projectLabelExists

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_projectLabelExists (soap:body, use = literal) [Source code](#)

projectLabelExists type projectLabelExists

Output: Source_2009_projectLabelExistsResponse (soap:body, use = literal) [Source code](#)

projectLabelExistsResponse type projectLabelExistsResponse

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class **type string**
The class of exception
- id **type string**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type string**
The implication from the exception

15. setSubprojectRevision

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_setSubprojectRevision (soap:body, use = literal) [Source code](#)

setSubprojectRevision type setSubprojectRevision

Output: Source_2009_setSubprojectRevisionResponse (soap:body, use = literal) [Source code](#)

setSubprojectRevisionResponse type setSubprojectRevisionResponse

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class **type string**
The class of exception
- id **type string**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type string**
The implication from the exception

16. shareSubproject

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_shareSubproject (soap:body, use = literal) [Source code](#)

shareSubproject type *shareSubproject*

Output: Source_2009_shareSubprojectResponse (soap:body, use = literal) [Source code](#)

shareSubprojectResponse type *shareSubprojectResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

17. unlockRevision

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_unlockRevision (soap:body, use = literal) [Source code](#)

unlockRevision type *unlockRevision*

Output: Source_2009_unlockRevisionResponse (soap:body, use = literal) [Source code](#)

unlockRevisionResponse type *unlockRevisionResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

18. updateRevision

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Source_2009_updateRevision (soap:body, use = literal) [Source code](#)

updateRevision type *updateRevision*

Output: Source_2009_updateRevisionResponse (soap:body, use = literal) [Source code](#)

updateRevisionResponse type *updateRevisionResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

WSDL source code

```
<?xml version="1.0"?>
<definitions name="Source_2009Service" targetNamespace="http://webservice.mks.com/2009/Source"
    xmlns:ns2="http://webservice.mks.com/2009/Source/fault"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://webservice.mks.com/2009/Source"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:ns1="http://webservice.mks.com/2009/Source/schema"
>
<types>
    <xs:schema elementFormDefault="qualified"
        targetNamespace="http://webservice.mks.com/2009/Source/schema" version="1.0">
        <xs:element name="Fields">
            <xs:complexType>
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="Field">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="Value" type="xs:string" />
                            </xs:sequence>
                            <xs:attribute name="Name" type="xs:string" use="required" />
                            <xs:attribute name="Type" type="xs:string" use="required" />
                            <xs:annotation>
                                <xs:documentation>The type of data stored in this field: int, string, boolean, datetime, or float. Note that due to a limitation in the product, if a field is not set in the response then the web service is not told what type of data would be returned if there were data. In that case, the type is set to "string".</xs:documentation>
                            <xs:annotation>
                                <xs:documentation>The type of data stored in this field: int, string, boolean, datetime, or float. Note that due to a limitation in the product, if a field is not set in the response then the web service is not told what type of data would be returned if there were data. In that case, the type is set to "string".</xs:documentation>
                            </xs:annotation>
                            </xs:attribute>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:schema>
</types>
```

```

</xs:complexType>
</xs:element>
<xs:complexType name="AddMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:UpdateMemberRequest">
            <xs:sequence>
                <xs:element minOccurs="0" name="ArchiveDescription" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>Note this is the description for the overall archive, not for the initial revision of the archive.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="UpdateMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="File" ns1:expectedContentTypes="application/octet-stream" type="xs:base64Binary" xmlns:ns1="http://www.w3.org/2005/05/xmlmime">
                    />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="MemberOperation">
    <xs:complexContent>
        <xs:extension base="tns:ProjectOperation">
            <xs:sequence>
                <xs:element name="Member" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The member name. If the member is in a subdirectory of the project, then preface the member like this:  
/subdirectory1/subdirectory2/membername</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectOperation">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element name="Project" type="xs:string">

```

```

        <xs:annotation>
            <xs:documentation>The project or subproject
                referenced by the service.</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="Credentials">
    <xs:sequence>
        <xs:element name="Username" type="xs:string" />
        <xs:element name="Password" type="xs:string" />
        <xs:element minOccurs="0" name="ImpersonatedUser" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string">
        <xs:annotation>
            <xs:documentation>An optional external identifier that can be passed in to
                Integrity. It is passed back verbatim with the
                response.</xs:documentation>
        </xs:annotation>
    </xs:attribute>
</xs:complexType>

<xs:complexType name="MemberResponse">
    <xs:complexContent>
        <xs:extension base="tns:ProjectResponse">
            <xs:sequence>
                <xs:element name="Member" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The member
                            name.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Revision" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The affected
                            revision.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="ProjectResponse">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element name="Project" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The project or subproject that is
                            the subject of this operation.</xs:documentation>
                    </xs:annotation>

```

```

        </xs:annotation>
    </xs:element>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="StatusResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:string" />
    <xs:sequence>
        <xs:attribute name="transactionId" type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="AddLabelRequest">
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="Label" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>Label to be applied to the indicated revision</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Revision" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>If revision is not specified, then headrev is assumed, on the mainline development path.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AddLabelResponseType">
    <xs:complexContent>
        <xs:extension base="tns:MemberResponse">
            <xs:sequence>
                <xs:element name="Label" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AddRevisionDescriptionRequest">
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="Description" type="xs:string" />
                <xs:element minOccurs="0" name="Revision" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The member revision against which the description will be applied. If empty, assumes

```

```

        "headrev".</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="AddRevisionDescriptionResponseType">
    <xs:complexContent>
        <xs:extension base="tns:MemberResponse">
            <xs:sequence>
                <xs:element name="Description" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CheckInMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:UpdateMemberRequest">
            <xs:sequence>
                <xs:element minOccurs="0" name="Description" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The revision description for the newly created revision.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Label" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>A label to be applied against the newly created revision. Note that all colons : are replaced by underscores _</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CheckInMemberResponseType">
    <xs:complexContent>
        <xs:extension base="tns:AddRevisionDescriptionResponseType">
            <xs:sequence>
                <xs:element minOccurs="0" name="Label" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateSubprojectRequest">
    <xs:annotation>
        <xs:documentation>Creates one or more subprojects of the requested project. The subproject is specified as /subprojectname/project.pj. Every additional slash character "/" is interpreted as another subproject to be created. Thus if the subproject is specified as /sub1/sub2/sub3/project.pj then 3 subprojects are

```

created. Note also that this command does not respond with the list of subprojects created by this command - it only lists the top-most subproject, such as sub3 in the above example.</xs:documentation>

```
</xs:annotation>
<xs:complexContent>
  <xs:extension base="tns:Credentials">
    <xs:sequence>
      <xs:element name="Project" type="xs:string">
        <xs:annotation>
          <xs:documentation>The project or subproject in which the subproject is to be created</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Subproject" type="xs:string">
        <xs:annotation>
          <xs:documentation>The name of the subproject. A subproject is created for each sub-element Xn in the directory structure /X1/X2/project.pj</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateSubprojectResponseType">
  <xs:complexContent>
    <xs:extension base="tns>StatusResponse">
      <xs:sequence>
        <xs:element name="Project" type="xs:string">
          <xs:annotation>
            <xs:documentation>The display name of the root project or subproject in which the subproject is created. The value is the same regardless of whether one subproject was created, or multiple subprojects.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="CanonicalParent" type="xs:string">
          <xs:annotation>
            <xs:documentation>The canonical name of the project or subproject which is the immediate parent of the subproject that was created. If only one subproject was created, then this is the canonical name of the project specified in the initial request.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Name" type="xs:string">
          <xs:annotation>
            <xs:documentation>The display name of the top-most subproject created by this command. For example, if the requested subproject was /sub1/sub2/sub3/project.pj, then this value is the complete display name of the project sub3.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

</xs:element>
<xs:element name="CanonicalName" type="xs:string">
    <xs:annotation>
        <xs:documentation>The canonical path of the top-most
        subproject created by this
        command.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="RevisionOperation">
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element minOccurs="0" name="Revision" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The requested member revision. If
                        unspecified, then "headrev" is
                        assumed.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="LockMemberResponseType">
        <xs:complexContent>
            <xs:extension base="tns:MemberResponse">
                <xs:sequence>
                    <xs:element name="WasBranched" type="xs:boolean" />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="FetchRevisionResponseType">
        <xs:complexContent>
            <xs:extension base="tns:MemberResponse">
                <xs:sequence>
                    <xs:element name="File"
                    ns2:expectedContentTypes="application/octet-stream"
                    type="xs:base64Binary"
                    xmlns:ns2="http://www.w3.org/2005/05/xmlmime"
                    />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="UnlockRevisionRequest">
        <xs:complexContent>
            <xs:extension base="tns:RevisionOperation">

```

```

<xs:sequence>
    <xs:element minOccurs="0" name="BreakLock" type="xs:boolean">
        <xs:annotation>
            <xs:documentation>If true, then the lock is completely dropped. If false, the lock is downgraded.</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="GetSubprojectsResponseType">
    <xs:complexContent>
        <xs:extension base="tns:StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Project" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The list of subprojects.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetMembersResponseType">
    <xs:complexContent>
        <xs:extension base="tns:StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Member" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The list of members.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetMemberInfoRequest">
    <xs:annotation>
        <xs:documentation>If the list of fields is not specified in the Details element then the following fields are returned: memberrevision, author, date and locker.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="Details" type="tns:InfoRequestDetails" />

```

```

        </xs:sequence>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="InfoRequestDetails">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="Field"
            type="xs:string" />
    </xs:sequence>
    <xs:attribute ref="tns:DateTimeFormat" use="optional" />
</xs:complexType>
<xs:complexType name="GetMemberInfoResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element ref="tns:Fields">
                    <xs:annotation>
                        <xs:documentation>The list of field attributes for the
                            requested object.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetProjectDetailsType">
    <xs:annotation>
        <xs:documentation>If the field list is not specified in the Details element then the
            following fields are returned: name, locker, membertimestamp and memberrev.
            Note the member's name, path and containing project are automatically included
            in the output schema.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns:ProjectOperation">
            <xs:sequence>
                <xs:element name="Details" type="tns:InfoRequestDetails" />
            </xs:sequence>
            <xs:attribute name="recurse" type="xs:boolean">
                <xs:annotation>
                    <xs:documentation>If true, then this method recurses into
                        subprojects. Default is false.</xs:documentation>
                </xs:annotation>
            </xs:attribute>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetProjectDetailsResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0"
                    name="MemberDetails">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Name" type="xs:string" />
    <xs:element name="Project" type="xs:string" />
    <xs:element name="Path" type="xs:string" />
    <xs:element ref="tns:Fields">
      <xs:annotation>
        <xs:documentation>The list of field
        attributes for the
        member.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="CheckpointProjectType">
  <xs:complexContent>
    <xs:extension base="tns:ProjectOperation">
      <xs:sequence>
        <xs:element minOccurs="0" name="Description" type="xs:string">
          <xs:annotation>
            <xs:documentation>The optional checkpoint
            description.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="Label" type="xs:string" use="optional">
        <xs:annotation>
          <xs:documentation>The optional label for the checkpoint.
          Note all colons : are replaced with underscores
          _</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectLabelExistsType">
  <xs:complexContent>
    <xs:extension base="tns:ProjectOperation">
      <xs:sequence>
        <xs:element name="Label" type="xs:string">
          <xs:annotation>
            <xs:documentation>The label to search for among all
            the checkpoints of the project. Note all colons : in the
            requested label are replaced with underscores _ prior
            to searching for the label.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:sequence>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="SubprojectRevisionOperation">
    <xs:complexContent>
        <xs:extension base="tns:ProjectOperation">
            <xs:sequence>
                <xs:element name="Subproject" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The subproject name.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Revision" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The revision identifier for the build subproject. This may be a literal revision number (e.g., 1.6), a "symbolic revision id" (e.g., :head, :trunktip) or a label (since a label uniquely identifies a revision within an archive).</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    <xs:complexContent>
</xs:complexType>
<xs:complexType name="ShareSubprojectOperation">
    <xs:complexContent>
        <xs:extension base="tns:ProjectOperation">
            <xs:sequence>
                <xs:element minOccurs="1" name="SharedSubproject" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The subproject to be shared.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="1" name="Subproject" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The name of the shared subproject in its newly shared location.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Revision" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The revision identifier for the build subproject. This may be a literal revision number (e.g., 1.6), a "symbolic revision id" (e.g., :head, :trunktip) or a label (since a label uniquely identifies a revision within an archive). If not specified, then the subproject is shared as regular subproject, not a build subproject.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    <xs:complexContent>
</xs:complexType>

```

```

        </xs:annotation>
    </xs:element>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:attribute name="DateTimeFormat">
    <xs:annotation>
        <xs:documentation>The format of timestamps in the output. See
http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html for full details.
        The default format is "yyyy-MM-dd HH:mm:ss Z" which would give you for
        example, "2009-09-25 14:39:22 -0500"</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string" />
    </xs:simpleType>
</xs:attribute>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/2009/Source/fault" version="1.0">
    <xs:element name="MKSException" nillable="true" type="tns:MKSFault" />
    <xs:complexType name="MKSFault">
        <xs:annotation>
            <xs:documentation>The original fault generated from MKS Integrity. The element
            value is the full description of the error.</xs:documentation>
        </xs:annotation>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="class" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The class of
                        exception</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:attribute name="id" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The unique MKS identifier for this error,
                        generally a 6-digit positive integer.</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:attribute name="implication" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The implication from the
                        exception</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/2009/Source" version="1.0">
    <xs:import namespace="http://webservice.mks.com/2009/Source/schema" />

```

```

<xs:element name="addLabel" type="tns:addLabel" />
<xs:element name="addLabelResponse" type="tns:addLabelResponse" />
<xs:element name="addMember" type="tns:addMember" />
<xs:element name="addMemberResponse" type="tns:addMemberResponse" />
<xs:element name="addRevisionDescription" type="tns:addRevisionDescription" />
<xs:element name="addRevisionDescriptionResponse"
type="tns:addRevisionDescriptionResponse" />
<xs:element name="checkInMember" type="tns:checkInMember" />
<xs:element name="checkInMemberResponse" type="tns:checkInMemberResponse" />
<xs:element name="checkpointProject" type="tns:checkpointProject" />
<xs:element name="checkpointProjectResponse" type="tns:checkpointProjectResponse" />
<xs:element name="createSubproject" type="tns:createSubproject" />
<xs:element name="createSubprojectResponse" type="tns:createSubprojectResponse" />
<xs:element name="dropMember" type="tns:dropMember" />
<xs:element name="dropMemberResponse" type="tns:dropMemberResponse" />
<xs:element name="fetchRevision" type="tns:fetchRevision" />
<xs:element name="fetchRevisionResponse" type="tns:fetchRevisionResponse" />
<xs:element name="getMemberInfo" type="tns:getMemberInfo" />
<xs:element name="getMemberInfoResponse" type="tns:getMemberInfoResponse" />
<xs:element name="getMembers" type="tns:getMembers" />
<xs:element name="getMembersResponse" type="tns:getMembersResponse" />
<xs:element name="getProjectDetails" type="tns:getProjectDetails" />
<xs:element name="getProjectDetailsResponse" type="tns:getProjectDetailsResponse" />
<xs:element name="getSubprojects" type="tns:getSubprojects" />
<xs:element name="getSubprojectsResponse" type="tns:getSubprojectsResponse" />
<xs:element name="lockMember" type="tns:lockMember" />
<xs:element name="lockMemberResponse" type="tns:lockMemberResponse" />
<xs:element name="projectLabelExists" type="tns:projectLabelExists" />
<xs:element name="projectLabelExistsResponse" type="tns:projectLabelExistsResponse" />
<xs:element name="setSubprojectRevision" type="tns:setSubprojectRevision" />
<xs:element name="setSubprojectRevisionResponse" type="tns:setSubprojectRevisionResponse"
/>
<xs:element name="shareSubproject" type="tns:shareSubproject" />
<xs:element name="shareSubprojectResponse" type="tns:shareSubprojectResponse" />
<xs:element name="unlockRevision" type="tns:unlockRevision" />
<xs:element name="unlockRevisionResponse" type="tns:unlockRevisionResponse" />
<xs:element name="updateRevision" type="tns:updateRevision" />
<xs:element name="updateRevisionResponse" type="tns:updateRevisionResponse" />
<xs:complexType name="addMember">
  <xs:sequence>
    <xs:element minOccurs="0" name="arg0" type="ns1:AddMemberRequest" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="addMemberResponse">
  <xs:sequence>
    <xs:element minOccurs="0" name="return" type="ns1:MemberResponse" />
  </xs:sequence>
</xs:complexType>

```

```

</xs:complexType>
<xs:complexType name="addLabel">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:AddLabelRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="addLabelResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:AddLabelResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="addRevisionDescription">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0"
            type="ns1:AddRevisionDescriptionRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="addRevisionDescriptionResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return"
            type="ns1:AddRevisionDescriptionResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkInMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CheckInMemberRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkInMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return"
            type="ns1:CheckInMemberResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createSubproject">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>CreateSubprojectRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createSubprojectResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return"
            type="ns1>CreateSubprojectResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="lockMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:RevisionOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="lockMemberResponse">

```

```

<xs:sequence>
    <xs:element minOccurs="0" name="return" type="ns1:LockMemberResponseType"
    />
</xs:sequence>
</xs:complexType>
<xs:complexType name="fetchRevision">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:RevisionOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchRevisionResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:FetchRevisionResponseType"
        />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="unlockRevision">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:UnlockRevisionRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="unlockRevisionResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:MemberResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getSubprojects">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ProjectOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getSubprojectsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return"
        type="ns1:GetSubprojectsResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMembers">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ProjectOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMembersResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetMembersResponseType"
        />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMemberInfo">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetMemberInfoRequest" />
    </xs:sequence>

```

```

</xs:sequence>
</xs:complexType>
<xs:complexType name="getMemberInfoResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return"
                    type="ns1:GetMemberInfoResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjectDetails">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetProjectDetailsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getProjectDetailsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return"
                    type="ns1:GetProjectDetailsResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkpointProject">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CheckpointProjectType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkpointProjectResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="projectLabelExists">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ProjectLabelExistsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="projectLabelExistsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="dropMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:MemberOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="dropMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="setSubprojectRevision">

```

```

<xs:sequence>
    <xs:element minOccurs="0" name="arg0" type="ns1:SubprojectRevisionOperation" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="setSubprojectRevisionResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="updateRevision">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:RevisionOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="updateRevisionResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="shareSubproject">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ShareSubprojectOperation" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="shareSubprojectResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="Source_2009_setSubprojectRevision">
    <part element="tns:setSubprojectRevision" name="setSubprojectRevision" />
</message>
<message name="Source_2009_fetchRevisionResponse">
    <part element="tns:fetchRevisionResponse" name="fetchRevisionResponse" />
</message>
<message name="Source_2009_updateRevisionResponse">
    <part element="tns:updateRevisionResponse" name="updateRevisionResponse" />
</message>
<message name="Source_2009_getSubprojectsResponse">
    <part element="tns:getSubprojectsResponse" name="getSubprojectsResponse" />
</message>
<message name="Source_2009_projectLabelExistsResponse">
    <part element="tns:projectLabelExistsResponse" name="projectLabelExistsResponse" />
</message>
<message name="Source_2009_fetchRevision">
    <part element="tns:fetchRevision" name="fetchRevision" />
</message>

```

```

<message name="Source_2009_addMember">
    <part element="tns:addMember" name="addMember" />
</message>
<message name="Source_2009_unlockRevision">
    <part element="tns:unlockRevision" name="unlockRevision" />
</message>
<message name="Source_2009_addRevisionDescription">
    <part element="tns:addRevisionDescription" name="addRevisionDescription" />
</message>
<message name="Source_2009_checkpointProject">
    <part element="tns:checkpointProject" name="checkpointProject" />
</message>
<message name="Source_2009_getMemberInfo">
    <part element="tns:getMemberInfo" name="getMemberInfo" />
</message>
<message name="Source_2009_setSubprojectRevisionResponse">
    <part element="tns:setSubprojectRevisionResponse" name="setSubprojectRevisionResponse" />
</message>
<message name="Source_2009_unlockRevisionResponse">
    <part element="tns:unlockRevisionResponse" name="unlockRevisionResponse" />
</message>
<message name="Source_2009_lockMember">
    <part element="tns:lockMember" name="lockMember" />
</message>
<message name="Source_2009_addRevisionDescriptionResponse">
    <part element="tns:addRevisionDescriptionResponse" name="addRevisionDescriptionResponse" />
</message>
<message name="Source_2009_dropMember">
    <part element="tns:dropMember" name="dropMember" />
</message>
<message name="Source_2009_getMemberInfoResponse">
    <part element="tns:getMemberInfoResponse" name="getMemberInfoResponse" />
</message>
<message name="Source_2009_projectLabelExists">
    <part element="tns:projectLabelExists" name="projectLabelExists" />
</message>
<message name="Source_2009_dropMemberResponse">
    <part element="tns:dropMemberResponse" name="dropMemberResponse" />
</message>
<message name="Source_2009_createSubprojectResponse">
    <part element="tns:createSubprojectResponse" name="createSubprojectResponse" />
</message>
<message name="Source_2009_createSubproject">
    <part element="tns:createSubproject" name="createSubproject" />
</message>
<message name="Source_2009_getSubprojects">
    <part element="tns:getSubprojects" name="getSubprojects" />
</message>
<message name="Source_2009_getProjectDetailsResponse">

```

```

<part element="tns:getProjectDetailsResponse" name="getProjectDetailsResponse" />
</message>
<message name="Source_2009_shareSubprojectResponse">
    <part element="tns:shareSubprojectResponse" name="shareSubprojectResponse" />
</message>
<message name="Source_2009_getProjectDetails">
    <part element="tns:getProjectDetails" name="getProjectDetails" />
</message>
<message name="Source_2009_getMembersResponse">
    <part element="tns:getMembersResponse" name="getMembersResponse" />
</message>
<message name="Source_2009_checkInMember">
    <part element="tns:checkInMember" name="checkInMember" />
</message>
<message name="Source_2009_checkInMemberResponse">
    <part element="tns:checkInMemberResponse" name="checkInMemberResponse" />
</message>
<message name="Source_2009_checkpointProjectResponse">
    <part element="tns:checkpointProjectResponse" name="checkpointProjectResponse" />
</message>
<message name="Source_2009_getMembers">
    <part element="tns:getMembers" name="getMembers" />
</message>
<message name="Source_2009_updateRevision">
    <part element="tns:updateRevision" name="updateRevision" />
</message>
<message name="Source_2009_lockMemberResponse">
    <part element="tns:lockMemberResponse" name="lockMemberResponse" />
</message>
<message name="Source_2009_addLabel">
    <part element="tns:addLabel" name="addLabel" />
</message>
<message name="Source_2009_addLabelResponse">
    <part element="tns:addLabelResponse" name="addLabelResponse" />
</message>
<message name="MKSEException">
    <part element="ns2:MKSEException" name="MKSEException" />
</message>
<message name="Source_2009_addMemberResponse">
    <part element="tns:addMemberResponse" name="addMemberResponse" />
</message>
<message name="Source_2009_shareSubproject">
    <part element="tns:shareSubproject" name="shareSubproject" />
</message>
<portType name="Source_2009">
    <operation name="addLabel" parameterOrder="addLabel">
        <input message="tns:Source_2009_addLabel" />
        <output message="tns:Source_2009_addLabelResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    
```

```

</operation>
<operation name="addMember" parameterOrder="addMember">
    <input message="tns:Source_2009_addMember" />
    <output message="tns:Source_2009_addMemberResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="addRevisionDescription" parameterOrder="addRevisionDescription">
    <input message="tns:Source_2009_addRevisionDescription" />
    <output message="tns:Source_2009_addRevisionDescriptionResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="checkInMember" parameterOrder="checkInMember">
    <input message="tns:Source_2009_checkInMember" />
    <output message="tns:Source_2009_checkInMemberResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="checkpointProject" parameterOrder="checkpointProject">
    <input message="tns:Source_2009_checkpointProject" />
    <output message="tns:Source_2009_checkpointProjectResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="createSubproject" parameterOrder="createSubproject">
    <input message="tns:Source_2009_createSubproject" />
    <output message="tns:Source_2009_createSubprojectResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="dropMember" parameterOrder="dropMember">
    <input message="tns:Source_2009_dropMember" />
    <output message="tns:Source_2009_dropMemberResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="fetchRevision" parameterOrder="fetchRevision">
    <input message="tns:Source_2009_fetchRevision" />
    <output message="tns:Source_2009_fetchRevisionResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getMemberInfo" parameterOrder="getMemberInfo">
    <input message="tns:Source_2009_getMemberInfo" />
    <output message="tns:Source_2009_getMemberInfoResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getMembers" parameterOrder="getMembers">
    <input message="tns:Source_2009_getMembers" />
    <output message="tns:Source_2009_getMembersResponse" />
    <fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getProjectDetails" parameterOrder="getProjectDetails">
    <input message="tns:Source_2009_getProjectDetails" />

```

```

<output message="tns:Source_2009_getProjectDetailsResponse" />
<fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="getSubprojects" parameterOrder="getSubprojects">
<input message="tns:Source_2009_getSubprojects" />
<output message="tns:Source_2009_getSubprojectsResponse" />
<fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="lockMember" parameterOrder="lockMember">
<input message="tns:Source_2009_lockMember" />
<output message="tns:Source_2009_lockMemberResponse" />
<fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="projectLabelExists" parameterOrder="projectLabelExists">
<input message="tns:Source_2009_projectLabelExists" />
<output message="tns:Source_2009_projectLabelExistsResponse" />
<fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="setSubprojectRevision" parameterOrder="setSubprojectRevision">
<input message="tns:Source_2009_setSubprojectRevision" />
<output message="tns:Source_2009_setSubprojectRevisionResponse" />
<fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="shareSubproject" parameterOrder="shareSubproject">
<input message="tns:Source_2009_shareSubproject" />
<output message="tns:Source_2009_shareSubprojectResponse" />
<fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="unlockRevision" parameterOrder="unlockRevision">
<input message="tns:Source_2009_unlockRevision" />
<output message="tns:Source_2009_unlockRevisionResponse" />
<fault message="tns:MKSEException" name="MKSEException" />
</operation>
<operation name="updateRevision" parameterOrder="updateRevision">
<input message="tns:Source_2009_updateRevision" />
<output message="tns:Source_2009_updateRevisionResponse" />
<fault message="tns:MKSEException" name="MKSEException" />
</operation>
</portType>
<binding name="Source_2009Binding" type="tns:Source_2009">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="addLabel">
<soap:operation soapAction="" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />

```

```

</output>
<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="addMember">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="addRevisionDescription">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="checkInMember">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="checkpointProject">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>

```

```

<fault name="MKSEException">
    <soap:fault name="MKSEException" use="literal" />
</fault>
</operation>
<operation name="createSubproject">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="dropMember">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="fetchRevision">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getMemberInfo">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">

```

```

        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getMembers">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getProjectDetails">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="getSubprojects">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="lockMember">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>

```

```
</fault>
</operation>
<operation name="projectLabelExists">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSException">
        <soap:fault name="MKSException" use="literal" />
    </fault>
</operation>
<operation name="setSubprojectRevision">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSException">
        <soap:fault name="MKSException" use="literal" />
    </fault>
</operation>
<operation name="shareSubproject">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSException">
        <soap:fault name="MKSException" use="literal" />
    </fault>
</operation>
<operation name="unlockRevision">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSException">
        <soap:fault name="MKSException" use="literal" />
    </fault>
```

```
</operation>
<operation name="updateRevision">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
</binding>
<service name="Source_2009Service">
    <port binding="tns:Source_2009Binding" name="Source_2009Port">
        <soap:address location="http://IntegrityServer:9001/webservices/2009/Source/" />
    </port>
</service>
</definitions>
```

Web Service: Segment_10Service

Web Service: Segment_10Service

Target Namespace: <http://webservice.mks.com/10/Segment>

Port **Segment_10Port** [Port type](#) [Source code](#)

Location: <http://IntegrityServer:9001/webservices/10/Segment/>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. [baseline](#) [Detail](#) [Source code](#)
2. [branchSegment](#) [Detail](#) [Source code](#)
3. [changeSegmentProject](#) [Detail](#) [Source code](#)
4. [copyContent](#) [Detail](#) [Source code](#)
5. [createContent](#) [Detail](#) [Source code](#)
6. [createSegment](#) [Detail](#) [Source code](#)
7. [importContent](#) [Detail](#) [Source code](#)
8. [importSegment](#) [Detail](#) [Source code](#)
9. [insertSegment](#) [Detail](#) [Source code](#)
10. [moveContent](#) [Detail](#) [Source code](#)
11. [propagateTraces](#) [Detail](#) [Source code](#)
12. [referenceMode](#) [Detail](#) [Source code](#)
13. [removeBaseline](#) [Detail](#) [Source code](#)
14. [removeContent](#) [Detail](#) [Source code](#)
15. [toggleIncludeReference](#) [Detail](#) [Source code](#)
16. [viewSegment](#) [Detail](#) [Source code](#)

Operations

Port type **Segment_10** [Source code](#)

1. **baseline**

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: baseline

Input: Segment_10_baseline (soap:body, use = literal) [Source code](#)

baseline type **baseline**

- arg0 - optional; type *BaselineType* - extension of type *RemoveBaselineType* - extension of type *Credentials*

■ Username type *string*

■ Password type *string*

■ ImpersonatedUser - optional; type *string*

Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.

- **DateFormat - optional; type *string***
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- **DateTimeFormat - optional; type *string***
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- **transactionId type *string***
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- **ID - unbounded; type *ItemIdReference* - type *integer* with restriction**
The IDs of the segments affected by the baseline operation.
- **Label type *string***
The label for the baseline.
- **SubSegment - optional; type *boolean***
When a selection is a node that references a subsegment, if this flag is true then the application will operate on the referenced subsegment, not the owning segment. Default is false, meaning the segment that is affected is the owning segment.
- **MoveLabel - optional; type *boolean***
If the label already exists on the item, then it is moved to the new time.
- **AsOf - optional; type *AsOf***

Output: Segment_10_baselineResponse (soap:body, use = literal) [Source code](#).

- baselineResponse** type *baselineResponse*
- **return - optional; type *BaselineResponseType* - extension of type *StatusResponse***
 - **return - optional; type *string***
 - **transactionId type *string***
The transaction ID that was passed in with the original request.
 - **Segment - unbounded; type *ContentDispositionType* - extension of type *ContentType***
Details about contents that were copied, moved or baselined.
 - **ItemId type *ItemIdReference* - type *integer* with restriction**
The ID of the item. This is the primary item of the record.
 - **ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }**
 - **ParentID - optional; type *ParentID***
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - **SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction**
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - **SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction**
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - **SegmentId - optional; type *ItemIdReference* - type *integer* with restriction**
If the content is a reference to a subsegment, this field holds the ID of the subsegment.
 - **OriginalContent type *ContentType***
 - **ItemId type *ItemIdReference* - type *integer* with restriction**
The ID of the item. This is the primary item of the record.
 - **ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }**
 - **ParentID - optional; type *ParentID***
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - **SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction**
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - **SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction**
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.

restriction

The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.

- SegmentId - optional; type *ItemIdReference* - type *integer* with restriction

If the content is a reference to a subsegment, this field holds the ID of the subsegment.

Details about the original content. For baseline operations this is the item that was included in the selection list.

The content that was affected by the baseline operation. The primary content is the affected segment. The original content is the selection item, which could be a node inside the segment or a node referencing the subsegment in the case of SubSegment=true.

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

2. branchSegment

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: BranchSegment

Input: Segment_10_branchSegment (soap:body, use = literal) [Source code](#)

branchSegment type *branchSegment*

- arg0 - optional; type *BranchSegmentType* - extension of type *Credentials*

- Username type *string*
- Password type *string*

- ImpersonatedUser - optional; type *string*

Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.

- DateFormat - optional; type *string*

The formatting syntax for date fields without a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:

<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- DateTimeFormat - optional; type *string*

The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:

<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- transactionId type *string*

An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.

- ID - unbounded; type *ItemIdReference* - type *integer* with restriction

The IDs for the segments to be branched.

- project type *string*

The project into which the branched document is to be placed.

- RecurseInclude - optional; type *boolean*

Branch all included subsegments too. Default is false.

- SubSegment - optional; type *boolean*

Branch the referenced subsegment, not the owning segment. Default is false, meaning if the ID is a node then the document that is branched is the owner of the node.

- Swap - optional; type *boolean*

If true, branch the referenced subsegment and swap in the newly created branch.

- Default is false.
- BranchOperation - optional; type *boolean*
If true or unspecified then the resulting document is linked to its parent in the parent's branch table and the document has a "branched from" annotation. If false the system treats the resulting document as if it is developed from scratch.
 - ParentID - optional; type *ParentID*
After branching, the parent of the newly branched segment. If unspecified then a stand-alone segment is created. Note this option conflicts with the Swap option.
 - InsertLocation - optional; type *InsertLocation*
If ParentID is specified, the optional position of the newly branched segment under its parent. If unspecified then "last" is assumed.
 - AsOf - optional; type *AsOf*

Output: Segment_10_branchSegmentResponse (soap:body, use = literal) [Source code](#)

branchSegmentResponse type *branchSegmentResponse*

- return - optional; type *BranchSegmentResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.
- Results - unbounded; type *BranchResult*
 - Source type *ContentType*
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.
 - The original segment in the request list.
- Target type *ContentType*
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.

the subsegment.
The resulting branched segment.
For each segment in the request list, the resulting content information for the newly branched segment.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class *type string*
The class of exception
- o id *type string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication *type string*
The implication from the exception

3. changeSegmentProject

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: ChangeSegmentProject

Input: Segment_10_changeSegmentProject (soap:body, use = literal) [Source code](#)

changeSegmentProject type *changeSegmentProject*

- arg0 - optional; type *ChangeSegmentProjectType* - extension of type *Credentials*
 - Username *type string*
 - Password *type string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- o transactionId *type string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- ID - unbounded; type *ItemIdReference* - type *integer* with restriction
The IDs for the segments to be reallocated.
- project *type string*
The project into which the segment should be placed.
- RecurseInclude - optional; type *boolean*
Default true. If true then included subsegments are included in the result.
- RecurseReference - optional; type *boolean*
Default false. If true then referenced subsegments are included in the result.
- SubSegment - optional; type *boolean*
If true, and one or more of the CopyTo or CopyFrom IDs is a node that references a subsegment, then the propagation will use the ID of the referenced subsegment, not the node. Default is false, that is, all IDs must be segment root IDs.

Output: Segment_10_changeSegmentProjectResponse (soap:body, use = literal) [Source code](#)

changeSegmentProjectResponse type *changeSegmentProjectResponse*

- return - optional; type *ChangeSegmentProjectResponseType* - extension of type *StatusResponse*
List of modifications.
 - return - optional; type *string*
- o transactionId *type string*
The transaction ID that was passed in with the original request.

- Project type *string*
The project to which the segments have been allocated.
- ID - unbounded; type *ItemIdReference* - type *integer* with restriction
The list of segment IDs allocated to the requested project. The list order is not guaranteed and may vary from one invocation to the next.

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- The original fault generated from MKS Integrity. The element value is the full description of the error.
- class type *string*
The class of exception
 - id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
 - implication type *string*
The implication from the exception

4. copyContent

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: CopyContent

Input: Segment_10_copyContent (soap:body, use = literal) [Source code](#)

copyContent type *copyContent*

- arg0 - optional; type *CopyContentType* - extension of type *MoveContentType* - extension of type *RemoveContentType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- ID - unbounded; type *ItemIdReference* - type *integer* with restriction
The IDs of the nodes to be removed.
- Recurse - optional; type *boolean*
If true then the system removes all sub-content underneath the selection(s). Default is false.
- ParentID type *ParentID*
The parent of the newly copied content.
- InsertLocation - optional; type *InsertLocation*
The optional position of the newly copied content under its parent. If unspecified then "last" is assumed.
- BranchOperation - optional; type *boolean*
If true or unspecified then the new nodes are designated as branches of the originals.
- CopyTrace - optional; type *boolean*
If true then all traces going into or out of the selection are also copied. Default is false.
- ReferenceMode - optional; type *ReferenceMode* - type *string* with restriction - enum { 'reuse', 'share' }
The reference mode for all regular content in the new copy. If unspecified, the default is determined from the Default Reference Mode attribute on the Type of the parentID segment.

- AsOf - optional; type *AsOf*

Output: Segment_10_copyContentResponse (soap:body, use = literal) [Source code](#)

copyContentResponse type *copyContentResponse*

- return - optional; type *CopyMoveContentResponseType* - extension of type *StatusResponse*

Information about the contents that were copied or moved.

- return - optional; type *string*

- transactionId type *string*

The transaction ID that was passed in with the original request.

- Result - unbounded; type *ContentDispositionType* - extension of type *ContentType*

Details about contents that were copied, moved or baselined.

- ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.

- ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }

- ParentID - optional; type *ParentID*

The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.

- SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction

The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.

- SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction

The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.

- SegmentId - optional; type *ItemIdReference* - type *integer* with restriction

If the content is a reference to a subsegment, this field holds the ID of the subsegment.

- OriginalContent type *ContentType*

- ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.

- ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }

- ParentID - optional; type *ParentID*

The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.

- SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction

The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.

- SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction

The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.

- SegmentId - optional; type *ItemIdReference* - type *integer* with restriction

If the content is a reference to a subsegment, this field holds the ID of the subsegment.

Details about the original content. For baseline operations this is the item that was included in the selection list.

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*

The class of exception

- id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- implication type *string*

The implication from the exception

5. createContent

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: CreateContent

Input: Segment_10_createContent (soap:body, use = literal) [Source code](#)

createContent type *createContent*

- arg0 - optional; type *CreateContentType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- type type *string*
The document content type
- ParentID - optional; type *ParentID*
The proposed owner this this item. This parameter is required for *CreateContent* and optional for the Import commands.
- InsertLocation - optional; type *InsertLocation*
- ItemField - optional, unbounded; type *ItemField*
- Attachment - optional, unbounded; type *AttachmentEdit*
 - Name type *string*
 - Field type *string*
 - Description - optional; type *string*
 - Attachment - optional; type *base64Binary*
The attachment. Ignored and may be omitted if the requested disposition is remove.
- RelatedItem - optional, unbounded; type *CreateEditRelatedItem* - extension of type *RelatedItemType*
 - Item - unbounded; type *Relationship*
For each related item, its ID and relationship flags.
 - flag type *string*
 - IBPL - unbounded; type *ItemIdReference* - type *integer* with restriction
 - FieldName type *string*
The name of the relationship field

Output: Segment_10_createContentResponse (soap:body, use = literal) [Source code](#)

createContentResponse type *createContentResponse*

- return - optional; type *CreateResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.
- AttachmentChecksum - optional, unbounded; type *AttachmentChecksum*
- Content type *ContentType*
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included' }

- SubSegment' }
- ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
- SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
- SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
- SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

- The original fault generated from MKS Integrity. The element value is the full description of the error.
- class type *string*
The class of exception
 - id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
 - implication type *string*
The implication from the exception

6. createSegment

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: CreateSegment

Input: Segment_10_createSegment (soap:body, use = literal) [Source code](#)

createSegment type *createSegment*

- arg0 - optional; type *CreateSegmentType* - extension of type *CreateContentType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
 - type type *string*
The document content type
 - ParentID - optional; type *ParentID*
The proposed owner this this item. This parameter is required for CreateContent and optional for the Import commands.
 - InsertLocation - optional; type *InsertLocation*
 - ItemField - optional, unbounded; type *ItemField*
 - Attachment - optional, unbounded; type *AttachmentEdit*
 - Name type *string*
 - Field type *string*

Description - optional; type *string*

- Attachment - optional; type *base64Binary*
The attachment. Ignored and may be omitted if the requested disposition is remove.
- RelatedItem - optional, unbounded; type *CreateEditRelatedItem* - extension of type *RelatedItemType*
 - Item - unbounded; type *Relationship*
For each related item, its ID and relationship flags.
 - flag type *string*
 - IBPL - unbounded; type *ItemIdReference* - type *integer* with restriction
 - FieldName type *string*
The name of the relationship field
- InsertMode - optional; type *InsertMode* - type *string* with restriction - enum { 'reference', 'include' }
The insert mode for the newly created segment, if a parent is specified. If insert mode is not specified, then reference is assumed.

Output: Segment_10_createSegmentResponse (soap:body, use = literal) [Source code](#)

createSegmentResponse type *createSegmentResponse*

- return - optional; type *CreateResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.
- AttachmentChecksum - optional, unbounded; type *AttachmentChecksum*
- Content type *ContentType*
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

7. importContent

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: ImportContent

Input: Segment_10_importContent (soap:body, use = literal) [Source code](#)

importContent type *importContent*

- arg0 - optional; type *ImportContentType* - extension of type *CreateContentType* -

extension of type *Credentials*

- Username type *string*

- Password type *string*

- ImpersonatedUser - optional; type *string*

Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.

- DateFormat - optional; type *string*

The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:

<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- DateTimeFormat - optional; type *string*

The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:

<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- transactionId type *string*

An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.

- type type *string*

The document content type

- ParentID - optional; type *ParentID*

The proposed owner this this item. This parameter is required for *CreateContent* and optional for the *Import* commands.

- InsertLocation - optional; type *InsertLocation*

- ItemField - optional, unbounded; type *ItemField*

- Attachment - optional, unbounded; type *AttachmentEdit*

- Name type *string*

- Field type *string*

- Description - optional; type *string*

- Attachment - optional; type *base64Binary*

The attachment. Ignored and may be omitted if the requested disposition is remove.

- RelatedItem - optional, unbounded; type *CreateEditRelatedItem* - extension of type *RelatedItemType*

- Item - unbounded; type *Relationship*

For each related item, its ID and relationship flags.

- flag type *string*

- IBPL - unbounded; type *ItemIdReference* - type *integer* with restriction

- FieldName type *string*

The name of the relationship field

- SubInsertID type *SubInsertID*

- SegmentType - optional; type *string*

The type of the segment under which the content is to be inserted. This is required only if ParentId is not supplied.

Output: Segment_10_importContentResponse (soap:body, use = literal) [Source code](#)

importContentResponse type *importContentResponse*

- return - optional; type *CreateResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*

- transactionId type *string*

The transaction ID that was passed in with the original request.

- AttachmentChecksum - optional, unbounded; type *AttachmentChecksum*

- Content type *ContentType*

- ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.

- ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }

- ParentID - optional; type *ParentID*

The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.

- SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction

The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.

- **SiblingAfterId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
- **SegmentId** - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

8. importSegment

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: ImportSegment

Input: Segment_10_importSegment (soap:body, use = literal) [Source code](#)

importSegment type *importSegment*

- **arg0** - optional; type *ImportSegmentType* - extension of type *CreateSegmentType* - extension of type *CreateContentType* - extension of type *Credentials*

▪ Username type *string*

▪ Password type *string*

▪ ImpersonatedUser - optional; type *string*

Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.

▪ DateFormat - optional; type *string*

The formatting syntax for date fields without a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:

<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

▪ DateTimeFormat - optional; type *string*

The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:

<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- transactionId type *string*

An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.

- type type *string*

The document content type

- ParentID - optional; type *ParentID*

The proposed owner this this item. This parameter is required for CreateContent and optional for the Import commands.

- InsertLocation - optional; type *InsertLocation*

- ItemField - optional, unbounded; type *ItemField*

- Attachment - optional, unbounded; type *AttachmentEdit*

 - Name type *string*

 - Field type *string*

 - Description - optional; type *string*

 - Attachment - optional; type *base64Binary*

The attachment. Ignored and may be omitted if the requested disposition is remove.

- RelatedItem - optional, unbounded; type *CreateEditRelatedItem* - extension of type *RelatedItemType*

- Item - unbounded; type *Relationship*
For each related item, its ID and relationship flags.
 - flag type *string*
- IBPL - unbounded; type *ItemIdReference* - type *integer* with restriction
- FieldName type *String*
The name of the relationship field
- InsertMode - optional; type *InsertMode* - type *string* with restriction - enum { 'reference', 'include' }
The insert mode for the newly created segment, if a parent is specified. If insert mode is not specified, then reference is assumed.
- SubInsertID type *SubInsertID*

Output: Segment_10_importSegmentResponse (soap:body, use = literal) [Source code](#)

- importSegmentResponse** type *importSegmentResponse*
- return - optional; type *CreateResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.
 - AttachmentChecksum - optional, unbounded; type *AttachmentChecksum*
 - Content type *ContentType*
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

- MKSException** type *MKSException*
- The original fault generated from MKS Integrity. The element value is the full description of the error.
- class type *string*
The class of exception
 - id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
 - implication type *string*
The implication from the exception

9. insertSegment

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: insertSegment

Input: Segment_10_insertSegment (soap:body, use = literal) [Source code](#)

- insertSegment** type *insertSegment*
- arg0 - optional; type *InsertSegmentType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username

- must be a user to whom impersonation privileges have been granted.
- **DateFormat** - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - **DateTimeFormat** - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - **transactionId** type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
 - **ID** - unbounded; type *ItemIdReference* - type *integer* with restriction
The IDs of the nodes to be removed.
 - **ParentID** type *ParentID*
The parent of the newly copied content.
 - **InsertLocation** - optional; type *InsertLocation*
The optional position of the newly copied content under its parent. If unspecified then "last" is assumed.
 - **InsertMode** - optional; type *InsertMode* - type *string* with restriction - enum { 'reference', 'include' }
The insert mode for the segment. If unspecified, then reference is assumed.

Output: Segment_10_insertSegmentResponse (soap:body, use = literal) [Source code](#)

- insertSegmentResponse** type *insertSegmentResponse*
- **return** - optional; type *InsertSegmentResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
 - **transactionId** type *string*
The transaction ID that was passed in with the original request.
 - **InsertedContent** - unbounded; type *ContentType*
 - **ItemId** type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - **ModelDesignation** type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - **ParentID** - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - **SiblingBeforeId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - **SiblingAfterId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - **SegmentId** - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.
- The content that was added to the parent.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

- MKSException** type *MKSException*
- The original fault generated from MKS Integrity. The element value is the full description of the error.
- **class** type *string*
The class of exception
 - **id** type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
 - **implication** type *string*
The implication from the exception

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: MoveContent

Input: Segment_10_moveContent (soap:body, use = literal) [Source code](#)

moveContent type *moveContent*

- arg0 - optional; type *MoveContentType* - extension of type *RemoveContentType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- ID - unbounded; type *ItemIdReference* - type *integer* with restriction
The IDs of the nodes to be removed.
- Recurse - optional; type *boolean*
If true then the system removes all sub-content underneath the selection(s). Default is false.
- ParentID type *ParentID*
The parent of the newly copied content.
- InsertLocation - optional; type *InsertLocation*
The optional position of the newly copied content under its parent. If unspecified then "last" is assumed.

Output: Segment_10_moveContentResponse (soap:body, use = literal) [Source code](#)

moveContentResponse type *moveContentResponse*

- return - optional; type *CopyMoveContentResponseType* - extension of type *StatusResponse*
Information about the contents that were copied or moved.
 - return - optional; type *string*
- transactionId type *string*
The transaction ID that was passed in with the original request.
- Result - unbounded; type *ContentDispositionType* - extension of type *ContentType*
Details about contents that were copied, moved or baselined.
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the

- subsegment.
- OriginalContent **type** *ContentType*
 - ItemId **type** *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation **type** *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.
- Details about the original content. For baseline operations this is the item that was included in the selection list.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException **type** *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class **type** *string*
The class of exception
- id **type** *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type** *string*
The implication from the exception

11. propagateTraces

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: PropagateTraces

Input: Segment_10_propagateTraces (soap:body, use = literal) [Source code](#)

propagateTraces **type** *propagateTraces*

- arg0 - optional; type *PropagateTracesType* - extension of type *Credentials*
 - Username **type** *string*
 - Password **type** *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId **type** *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.

- **CopyFromStart** type *ItemIdReference* - type *integer* with restriction
The ID of the starting document from which the traces will be copied. CopyFromStart and CopyFromEnd are the two documents having traces between them that you wish to copy.
- **CopyFromEnd** type *ItemIdReference* - type *integer* with restriction
The ID of the ending document from which the traces will be copied. CopyFromStart and CopyFromEnd are the two documents having traces between them that you wish to copy.
- **CopyToStart** type *ItemIdReference* - type *integer* with restriction
The ID of the starting document to which the traces will be copied. CopyToStart and CopyToEnd are the documents to which you wish to copy the traces.
- **CopyToEnd** type *ItemIdReference* - type *integer* with restriction
The ID of the ending document to which the traces will be copied. CopyToStart and CopyToEnd are the documents to which you wish to copy the traces.
- **SubSegment** - optional; type *boolean*
If true, and one or more of the CopyTo or CopyFrom IDs is a node that references a subsegment, then the propagation will use the ID of the referenced subsegment, not the node. Default is false, that is, all IDs must be segment root IDs.
- **Scope** - optional; type *PropagateScope*
The scope of propagation. If you want to recurse into subdocuments, you must indicate whether to recurse into included subdocuments, or referenced, or both. If not recursive, you have the option of specifying which particular traces you wish to propagate.
 - **TraceFields**
The list of trace relationships to propagate. If none is specified then all trace fields are propagated.
 - **FieldName** - optional, unbounded; type *string*

Output: Segment_10_propagateTracesResponse (soap:body, use = literal) [Source code](#)

propagateTracesResponse type *propagateTracesResponseType*

- **return** - optional; type *PropagateTracesResponseType* - extension of type *StatusResponse*
Information about the traces that were propagated.
 - **return** - optional; type *string*
- **transactionId** type *string*
The transaction ID that was passed in with the original request.
- **CopyToStart** type *PropagationResult* - extension of type *ContentType*
 - **ItemId** type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - **ModelDesignation** type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - **ParentID** - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - **SiblingBeforeId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - **SiblingAfterId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - **SegmentId** - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.
 - **NumberOfTraces** type *int*
The number of traces created on this target segment.
 - **NumberSuspect** type *int*
The number of traces created on this target segment that are suspect.
- Information about the traces created on the "start" document.
- **CopyToEnd** type *PropagationResult* - extension of type *ContentType*
 - **ItemId** type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - **ModelDesignation** type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - **ParentID** - optional; type *ParentID*

The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.

- **SiblingBeforeId** - optional; type *ItemIdReference* - type *integer* with restriction

The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.

- **SiblingAfterId** - optional; type *ItemIdReference* - type *integer* with restriction

The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.

- **SegmentId** - optional; type *ItemIdReference* - type *integer* with restriction

If the content is a reference to a subsegment, this field holds the ID of the subsegment.

- **NumberOfTraces** type *int*

The number of traces created on this target segment.

- **NumberSuspect** type *int*

The number of traces created on this target segment that are suspect.

Information about the traces created on the "end" document.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*

The class of exception

- id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- implication type *string*

The implication from the exception

12. referenceMode

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: referenceMode

Input: Segment_10_referenceMode (soap:body, use = literal) [Source code](#)

referenceMode type *referenceMode*

- **arg0** - optional; type *ReferenceModeType* - extension of type *Credentials*

- Username type *string*

- Password type *string*

- ImpersonatedUser - optional; type *string*

Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.

- DateFormat - optional; type *string*

The formatting syntax for date fields without a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- DateTimeFormat - optional; type *string*

The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- transactionId type *string*

An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.

- ID - unbounded; type *ItemIdReference* - type *integer* with restriction

- ReferenceMode type *ReferenceMode* - type *string* with restriction - enum { 'reuse', 'share' }

- ShareID - optional; type *ItemIdReference* - type *integer* with restriction

For share reference mode, the ID of the item from which the selected item will share its content. If unspecified, then the default is to share from the node from which the current node was originally branched.

Output: Segment_10_referenceModeResponse (soap:body, use = literal) [Source code](#)

referenceModeResponse type *referenceModeResponse*

- return - optional; type *ReferenceModeResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.
 - ID - unbounded; type *ContentType*
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.

The content that was affected by the reference mode operation.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

13. removeBaseline

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: removeBaseline

Input: Segment_10_removeBaseline (soap:body, use = literal) [Source code](#)

removeBaseline type *removeBaseline*

- arg0 - optional; type *RemoveBaselineType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.

- ID - unbounded; type *ItemIdReference* - type *integer* with restriction
The IDs of the segments affected by the baseline operation.
- Label type *string*
The label for the baseline.
- SubSegment - optional; type *boolean*
When a selection is a node that references a subsegment, if this flag is true then the application will operate on the referenced subsegment, not the owning segment.
Default is false, meaning the segment that is affected is the owning segment.

Output: Segment_10_removeBaselineResponse (soap:body, use = literal) [Source code](#)

removeBaselineResponse type *removeBaselineResponse*

- return - optional; type *BaselineResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
- transactionId type *string*
The transaction ID that was passed in with the original request.
- Segment - unbounded; type *ContentDispositionType* - extension of type *ContentType*
Details about contents that were copied, moved or baselined.
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.
 - OriginalContent type *ContentType*
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.

Details about the original content. For baseline operations this is the item that was included in the selection list.

The content that was affected by the baseline operation. The primary content is the affected segment. The original content is the selection item, which could be a node inside the segment or a node referencing the subsegment in the case of SubSegment=true.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSEException type MKSEException

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class **type string**
The class of exception
- id **type string**
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication **type string**
The implication from the exception

14. removeContent[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: RemoveContent

Input: Segment_10_removeContent (soap:body, use = literal) [Source code](#)

removeContent type removeContent

- arg0 - optional; type **RemoveContentType** - extension of type **Credentials**
 - Username **type string**
 - Password **type string**
 - ImpersonatedUser - optional; type **string**
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type **string**
The formatting syntax for date fields without a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type **string**
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId **type string**
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- ID - unbounded; type **ItemIdReference** - type **integer** with restriction
The IDs of the nodes to be removed.
- Recurse - optional; type **boolean**
If true then the system removes all sub-content underneath the selection(s). Default is false.

Output: Segment_10_removeContentResponse (soap:body, use = literal) [Source code](#)

removeContentResponse type removeContentResponse

- return - optional; type **RemoveContentResponseType** - extension of type **StatusResponse**
 - return - optional; type **string**
 - transactionId **type string**
The transaction ID that was passed in with the original request.
 - RemovedContent - unbounded; type **ContentType**
 - ItemId type **ItemIdReference** - type **integer** with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type **CreatedContentModelType** - type **string** with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type **ParentID**
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - SiblingBeforeId - optional; type **ItemIdReference** - type **integer** with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - SiblingAfterId - optional; type **ItemIdReference** - type **integer** with restriction

The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.

- SegmentId - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.

The content that was removed from the segment. Note that this list will not include the sub-contents that are removed from the document if recurse is requested. It lists only the top-level contents that are removed.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

15. toggleIncludeReference

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: toggleIncludeReference

Input: Segment_10_toggleIncludeReference (soap:body, use = literal) [Source code](#)

toggleIncludeReference type *toggleIncludeReference*

- arg0 - optional; type *ToggleIncludeReferenceType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- ID - unbounded; type *ItemIdReference* - type *integer* with restriction

Output: Segment_10_toggleIncludeReferenceResponse (soap:body, use = literal) [Source code](#)

toggleIncludeReferenceResponse type *toggleIncludeReferenceResponse*

- return - optional; type *ToggleIncludeReferenceResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
- transactionId type *string*
The transaction ID that was passed in with the original request.
- ID - unbounded; type *ContentDispositionType* - extension of type *ContentType*
Details about contents that were copied, moved or baselined.
 - ItemId type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - ModelDesignation type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - ParentID - optional; type *ParentID*

The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.

- **SiblingBeforeId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
- **SiblingAfterId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
- **SegmentId** - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.
- **OriginalContent** type *ContentType*
 - **ItemId** type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - **ModelDesignation** type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - **ParentID** - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - **SiblingBeforeId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - **SiblingAfterId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - **SegmentId** - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.

Details about the original content. For baseline operations this is the item that was included in the selection list.

The current state of the subsegment reference, along with its original state.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- **class** type *string*
The class of exception
- **id** type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- **implication** type *string*
The implication from the exception

16. viewSegment

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: ViewSegment

Input: Segment_10_viewSegment (soap:body, use = literal) [Source code](#)

viewSegment type *viewSegment*

- **arg0** - optional; type *ViewSegmentType* - extension of type *Credentials*
 - **Username** type *string*
 - **Password** type *string*
 - **ImpersonatedUser** - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.

- **DateFormat** - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- **DateTimeFormat** - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- **transactionId** type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- **Filter** - optional; type *string*
Filter rule to be applied to the document. Only items passing the filter will be included in the output.
- **RecurseInclude** - optional; type *boolean*
Default true. If true then included subsegments are included in the result.
- **SubstituteParameters** - optional; type *boolean*
Default is false. If true, then the values of parameters embedded in the fields are substituted in the field return value.
- **InputField** - optional, unbounded; type *string*
- **AsOf** - optional; type *AsOf*
- **ItemId** type *ItemIdReference* - type *integer* with restriction

Output: Segment_10_viewSegmentResponse (soap:body, use = literal) [Source code](#)

viewSegmentResponse type *viewSegmentResponse*

- **return** - optional; type *ViewSegmentResponseType* - extension of type *StatusResponse*
 - **return** - optional; type *string*
 - **transactionId** type *string*
The transaction ID that was passed in with the original request.
 - **DocumentElement** - unbounded; type *DocumentElementType*
 - **Content** type *ContentType*
 - **ItemId** type *ItemIdReference* - type *integer* with restriction
The ID of the item. This is the primary item of the record.
 - **ModelDesignation** type *CreatedContentModelType* - type *string* with restriction - enum { 'Content', 'Segment', 'SubSegment Reference', 'Included SubSegment' }
 - **ParentID** - optional; type *ParentID*
The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.
 - **SiblingBeforeId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.
 - **SiblingAfterId** - optional; type *ItemIdReference* - type *integer* with restriction
The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.
 - **SegmentId** - optional; type *ItemIdReference* - type *integer* with restriction
If the content is a reference to a subsegment, this field holds the ID of the subsegment.
 - **GetItemField** - optional, unbounded; type *GetItemField*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- **class** type *string*
The class of exception
- **id** type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- **implication** type *string*

WSDL source code

```

<?xml version="1.0"?>
<definitions name="Segment_10Service" targetNamespace="http://webservice.mks.com/10/Segment"
    xmlns:ns3="http://www.w3.org/2005/05/xmlmime"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://webservice.mks.com/10/Segment"
    xmlns:ns1="http://webservice.mks.com/10/Segment/schema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:ns2="http://webservice.mks.com/10/Segment/fault"
>

<types>
    <xs:schema elementFormDefault="qualified" targetNamespace="http://webservice.mks.com/10/Segment/schema"
        version="1.0">
        <xs:import namespace="http://www.w3.org/2005/05/xmlmime" schemaLocation="xmlmime.xsd" />
        <xs:simpleType name="InsertMode">
            <xs:annotation>
                <xs:documentation>Insertion mode for subsegment creation.</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string">
                <xs:enumeration value="reference" />
                <xs:enumeration value="include" />
            </xs:restriction>
        </xs:simpleType>
        <xs:element name="InsertLocation">
            <xs:annotation>
                <xs:documentation>Insertion location for content and subsegment creation. If unspecified then "last" is assumed. Allowed options are 1. Insert location number, where 0 is the first element in the list, 1 is the second, etc.; 2. "first" indicating the first element in the list; 3. "last" indicating the last element in the list; 4. "before:id" where id is an existing node in the segment; and 5. "after:id" where id is an existing node in the segment.</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string" />
            </xs:simpleType>
        </xs:element>
        <xs:simpleType name="ItemIdReference">
            <xs:annotation>
                <xs:documentation>A reference to an item in the repository.</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:integer" />
        </xs:simpleType>
        <xs:element name="ParentID" type="tns:ItemIdReference">
            <xs:annotation>
                <xs:documentation>The ID of the segment or node under which the new item will be placed. Read the notes to determine the scenarios for which a value is required.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:complexType name="Credentials">
            <xs:sequence>
                <xs:element name="Username" type="xs:string" />

```

```

<xs:element name="Password" type="xs:string" />
<xs:element minOccurs="0" name="ImpersonatedUser" type="xs:string">
    <xs:annotation>
        <xs:documentation>Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="DateFormat" type="xs:string">
    <xs:annotation>
        <xs:documentation>The formatting syntax for date fields without a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:  

http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="DateTimeFormat" type="xs:string">
    <xs:annotation>
        <xs:documentation>The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:  

http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
<xs:attribute name="transactionId" type="xs:string">
    <xs:annotation>
        <xs:documentation>An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.</xs:documentation>
    </xs:annotation>
</xs:attribute>
</xs:complexType>
<xs:complexType name="StatusResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string">
        <xs:annotation>
            <xs:documentation>The transaction ID that was passed in with the original request.</xs:documentation>
        </xs:annotation>
    </xs:attribute>
</xs:complexType>
<xs:element name="AsOf">
    <xs:complexType>
        <xs:choice>
            <xs:element name="DateTime" type="xs:string">
                <xs:annotation>
                    <xs:documentation>The date and time as of which the request is being made. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:  

http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html If the format is not specified in the request, the default is "yyyy-MM-dd HH:mm:ss".</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Label" type="xs:string">
                <xs:annotation>

```

```

<xs:documentation>The item label as of which the request is being made.  

Note the label must be attached to the actual item being requested -  

content inside documents are not labelled individually when the document is  

labelled.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="ItemField" type="tns:CompleteItemFieldType" />
<xs:complexType name="CompleteItemFieldType">
<xs:complexContent>
<xs:extension base="tns:SimpleItemFieldType">
<xs:choice minOccurs="0">
<xs:element name="relationship">
<xs:annotation>
<xs:documentation>Each entry in the relationship is a reference to  

the item followed by optional relationship flag identifiers. For  

example: 4032x or 4033x? or 4044</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0"  

name="value" nillable="true" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="float" type="tns:NullableFloat" />
<xs:element name="pick">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0"  

name="value" nillable="true" type="xs:string" />
</xs:sequence>
<xs:attribute name="null" type="xs:boolean" />
</xs:complexType>
</xs:element>
<xs:element name="siProject" type="tns:NullableString">
<xs:annotation>
<xs:documentation>The configuration management project being  

referenced by this field</xs:documentation>
</xs:annotation>
</xs:element>
</xs:choice>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="SimpleItemFieldType">
<xs:choice minOccurs="0">
<xs:element name="longtext" type="tns:NullableString">
<xs:annotation>
<xs:documentation>Content to be used when setting this field. Note that the  

content is recorded literally - that is, HTML tags are not interpreted if you use  

this element. If you need to embed HTML tags in a rich longtext field, use the  

richcontent element.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="shorttext" type="tns:NullableShortString" />

```

```

<xs:element name="richcontent" type="tns:NullableString">
    <xs:annotation>
        <xs:documentation>HTML content to be used when setting this field. Use CDATA blocks to escape the angle braces. Please refer to the Integrity User Guide for a list of HTML tags supported by Integrity.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="user" type="tns:NullableString" />
<xs:element name="multiuser">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="user" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="group" type="tns:NullableString" />
<xs:element name="multigroup">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="group" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="state" type="xs:string" />
<xs:element name="type" type="xs:string">
    <xs:annotation>
        <xs:documentation>The type of item. Note this field setting is used only in method responses. Do not use it for create or edit operations. For CreateItem there is an explicit Type attribute.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="project" type="tns:NullableString">
    <xs:annotation>
        <xs:documentation>The project in which the item is found</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="double" type="tns:NullableDouble" />
<xs:element name="boolean" type="tns:NullableBoolean" />
<xs:element name="integer" type="tns:NullableInteger" />
<xs:element name="date" type="tns:NullableString">
    <xs:annotation>
        <xs:documentation>The value for the date field - that is, a date having no time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url: http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html If the format is not specified in the request, the default is "yyyy-MM-dd". Be careful to not send a date when a timestamp is required! For GetItem and CreateItem, if the Integrity field is a timestamp but this field is used to send the information, the end user will get an error message that recommends the wrong formatting. This error message comes from the Integrity API and should be ignored since it is a symptom of the problem, not the root cause.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="dateTime" type="tns:NullableString">
    <xs:annotation>
        <xs:documentation>The value for the timestamp field - that is, a date having a
```

time component. The format may be specified in the `dateTimeFormat` entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html> If the format is not specified in the request, the default is "yyyy-MM-dd HH:mm:ss". Be careful to not send a timestamp when a date is required! For `GetItem` and `CreateItem`, if the `Integrity` field is a date but this field is used to send the information, the end user will get an error message that recommends the wrong formatting. This error message comes from the `Integrity` API and should be ignored since it is a symptom of the problem, not the root cause.</xs:documentation>

```
</xs:annotation>
</xs:element>
</xs:choice>
<xs:attribute name="Name" type="xs:string" use="required">
    <xs:annotation>
        <xs:documentation>The field name</xs:documentation>
    </xs:annotation>
</xs:attribute>
</xs:complexType>
<xs:complexType name="NullableShortString">
    <xs:choice>
        <xs:element name="value" type="xs:string" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableString">
    <xs:choice>
        <xs:element name="value" type="xs:string" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableFloat">
    <xs:choice>
        <xs:element name="value" type="xs:float" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableDouble">
    <xs:choice>
        <xs:element name="value" type="xs:double" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableBoolean">
    <xs:choice>
        <xs:element name="value" type="xs:boolean" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableInteger">
    <xs:choice>
        <xs:element name="value" type="xs:int" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="AttachmentEdit">
    <xs:sequence>
        <xs:element name="Name" type="xs:string" />
```

```

<xs:element name="Field" type="xs:string" />
<xs:element minOccurs="0" name="Description" type="xs:string" />
<xs:element minOccurs="0" name="Attachment" ns3:expectedContentTypes="application/octet-stream" type="xs:base64Binary">
    <xs:annotation>
        <xs:documentation>The attachment. Ignored and may be omitted if the requested disposition is remove.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="Relationship">
    <xs:annotation>
        <xs:documentation>For each related item, its ID and relationship flags.</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="xs:integer">
            <xs:attribute name="flag" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="RelatedItemType">
    <xs:choice>
        <xs:element maxOccurs="unbounded" name="Item" type="tns:Relationship" />
        <xs:element maxOccurs="unbounded" name="IBPL" type="tns:itemIdReference" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="CreateEditRelatedItem">
    <xs:complexContent>
        <xs:extension base="tns:RelatedItemType">
            <xs:attribute name="FieldName" type="xs:string" use="required">
                <xs:annotation>
                    <xs:documentation>The name of the relationship field</xs:documentation>
                </xs:annotation>
            </xs:attribute>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateContentType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element minOccurs="1" name="type" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The document content type</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" ref="tns:ParentID">
                    <xs:annotation>
                        <xs:documentation>The proposed owner this this item. This parameter is required for CreateContent and optional for the Import commands.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" ref="tns:InsertLocation" />
                <xs:element maxOccurs="unbounded" minOccurs="0" ref="tns:ItemField" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

<xs:element maxOccurs="unbounded" minOccurs="0" name="Attachment"
    type="tns:AttachmentEdit" />
<xs:element maxOccurs="unbounded" minOccurs="0" name="RelatedItem"
    type="tns>CreateEditRelatedItem" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateSegmentType">
    <xs:complexContent>
        <xs:extension base="tns>CreateContentType">
            <xs:sequence>
                <xs:element minOccurs="0" name="InsertMode" type="tns:InsertMode">
                    <xs:annotation>
                        <xs:documentation>The insert mode for the newly created segment,
                            if a parent is specified. If insert mode is not specified, then
                            reference is assumed.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="AttachmentChecksum">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="1" name="Field" type="xs:string" />
            <xs:element minOccurs="1" name="FileName" type="xs:string" />
            <xs:element minOccurs="1" name="FileSize" type="xs:integer" />
            <xs:element minOccurs="1" name="DateAdded" type="xs:string" />
            <xs:element minOccurs="1" name="Checksum" type="xs:string" />
            <xs:element minOccurs="1" name="ChecksumAlgorithm" type="xs:string">
                <xs:annotation>
                    <xs:documentation>Currently hard-coded to MD5</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="status" type="xs:string">
            <xs:annotation>
                <xs:documentation>Value is either Added if the attachment had not previously
                    existed on the item, or Updated if the attachment was
                    changed.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
    </xs:complexType>
</xs:element>
<xs:simpleType name="CreatedContentModelType">
    <xs:annotation>
        <xs:documentation>Model designation of created content.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:enumeration value="Content">
            <xs:annotation>
                <xs:documentation>Regular document content.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>

```

```

<xs:enumeration value="Segment">
    <xs:annotation>
        <xs:documentation>A standalone document.</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="SubSegment Reference">
    <xs:annotation>
        <xs:documentation>The segment you just created is being referenced from a parent document.</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="Included SubSegment">
    <xs:annotation>
        <xs:documentation>The segment you just created is being included into a parent document.</xs:documentation>
    </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
<xs:complexType name="ContentType">
    <xs:sequence>
        <xs:element name="ItemId" type="tns:ItemIdReference">
            <xs:annotation>
                <xs:documentation>The ID of the item. This is the primary item of the record.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ModelDesignation" type="tns:CreatedContentModelType" />
        <xs:element minOccurs="0" ref="tns:ParentID">
            <xs:annotation>
                <xs:documentation>The ID of the item containing the primary item. If the item is a standalone segment then this field is not specified.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="SiblingBeforeId" type="tns:ItemIdReference">
            <xs:annotation>
                <xs:documentation>The ID of the item just preceding the primary item. If the item is a standalone segment or it is the first sibling under its parent then this field is not specified.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="SiblingAfterId" type="tns:ItemIdReference">
            <xs:annotation>
                <xs:documentation>The ID of the item just after the primary item. If the item is a standalone segment or it is the last sibling under its parent then this field is not specified.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="SegmentId" type="tns:ItemIdReference">
            <xs:annotation>
                <xs:documentation>If the content is a reference to a subsegment, this field holds the ID of the subsegment.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CreateResponseType">

```

```

<xs:complexContent>
  <xs:extension base="tns>StatusResponse">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0"
      ref="tns:AttachmentChecksum" />
      <xs:element name="Content" type="tns:ContentType" />
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ViewSegmentType">
  <xs:complexContent>
    <xs:extension base="tns:Credentials">
      <xs:sequence>
        <xs:element minOccurs="0" name="Filter" type="xs:string">
          <xs:annotation>
            <xs:documentation>Filter rule to be applied to the document. Only
            items passing the filter will be included in the
            output.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="RecurseInclude" type="xs:boolean">
          <xs:annotation>
            <xs:documentation>Default true. If true then included subsegments
            are included in the result.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="SubstituteParameters" type="xs:boolean">
          <xs:annotation>
            <xs:documentation>Default is false. If true, then the values of
            parameters embedded in the fields are substituted in the field return
            value.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="InputField"
        type="xs:string" />
        <xs:element minOccurs="0" ref="tns:AsOf" />
      </xs:sequence>
      <xs:attribute form="qualified" name="ItemId" type="tns:ItemIdReference" use="required" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="DocumentElementType">
  <xs:sequence>
    <xs:element name="Content" type="tns:ContentType" />
    <xs:element maxOccurs="unbounded" minOccurs="0" ref="tns:GetItemField" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ViewSegmentResponseType">
  <xs:complexContent>
    <xs:extension base="tns>StatusResponse">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="1" name="DocumentElement"
        type="tns:DocumentElementType" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

</xs:complexType>
<xs:element name="GetItemField" type="tns:FormItemType" />
<xs:complexType name="FormItemType">
<xs:complexContent>
<xs:extension base="tns:SimpleItemFieldType">
<xs:choice minOccurs="0">
<xs:element name="Relationship" type="tns:RelatedItemType" />
<xs:element name="Attachment" type="tns:AttachmentInfo" />
<xs:element name="SIProject" type="tns:SIProjectType" />
<xs:element name="List" type="tns:ListType" />
<xs:element name="PickList" type="tns:PickListType" />
<xs:element name="Phase" type="tns:PhaseType" />
</xs:choice>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="PickListType">
<xs:annotation>
<xs:documentation>A pick list reported from Integrity.</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element name="MultiValued" type="xs:boolean" />
<xs:element maxOccurs="unbounded" minOccurs="0" name="Value" type="tns:PickListValue" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="PickListValue">
<xs:annotation>
<xs:documentation>A pick list value.</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element name="Label" type="xs:string" />
<xs:element name="Position" type="xs:int" />
<xs:element name="Value" type="xs:int" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="PhaseType">
<xs:annotation>
<xs:documentation>A phase value.</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element name="Label" type="xs:string" />
<xs:element name="Position" type="xs:int" />
<xs:element name="Value" type="tns:PhaseValueList" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="PhaseValueList">
<xs:annotation>
<xs:documentation>The list of states for a given phase.</xs:documentation>
</xs:annotation>
<xs:sequence>
<xs:element maxOccurs="unbounded" name="State" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="ListType">

```

```

<xs:annotation>
    <xs:documentation>A list reported from Integrity. The field type may be User or Group.</xs:documentation>
</xs:annotation>
<xs:sequence>
    <xs:element maxOccurs="unbounded" name="Element" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="SIProjectType">
    <xs:sequence>
        <xs:element name="Server" type="xs:string">
            <xs:annotation>
                <xs:documentation>The hostname and port for the configuration management server, in the format <hostname>:<port>.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Name" type="xs:string">
            <xs:annotation>
                <xs:documentation>The name of the configuration management project.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="Revision" type="xs:string">
            <xs:annotation>
                <xs:documentation>The revision number of the configuration management project. If not specified, then the tip of the tree is assumed.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="DevelopmentPath" type="xs:string">
            <xs:annotation>
                <xs:documentation>The name of the development path for the configuration management project. If not specified, then headrev is assumed.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="null" type="xs:boolean" />
</xs:complexType>
<xs:complexType name="AttachmentInfo">
    <xs:annotation>
        <xs:documentation>Basic information about the attachment.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element name="Name" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The attachment name</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Size" type="xs:int">
                    <xs:annotation>
                        <xs:documentation>The size of the attachment, in bytes</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Checksum" type="xs:string">

```

```

<xs:annotation>
    <xs:documentation>The checksum value for the attachment.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="SubInsertID">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="SubInsertMode" type="tns:InsertMode">
                <xs:annotation>
                    <xs:documentation>Indicates how to insert subsegments that are included in the selection list.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="ID" type="tns:ItemIdReference">
                <xs:annotation>
                    <xs:documentation>The IDs of the segments or nodes to be inserted under the new item.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="ImportSegmentType">
    <xs:complexContent>
        <xs:extension base="tns>CreateSegmentType">
            <xs:sequence>
                <xs:element ref="tns:SubInsertID" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ImportContentType">
    <xs:complexContent>
        <xs:extension base="tns>CreateContentType">
            <xs:sequence>
                <xs:element ref="tns:SubInsertID" />
                <xs:element minOccurs="0" name="SegmentType" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The type of the segment under which the content is to be inserted. This is required only if ParentId is not supplied.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="BranchSegmentType">
    <xs:complexContent>
        <xs:extension base="tns>Credentials">
            <xs:sequence>

```

```

<xs:element maxOccurs="unbounded" name="ID" type="tns:ItemIDReference">
    <xs:annotation>
        <xs:documentation>The IDs for the segments to be branched.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="project" type="xs:string">
    <xs:annotation>
        <xs:documentation>The project into which the branched document is to be placed.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="RecurseInclude" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>Branch all included subsegments too. Default is false.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="SubSegment" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>Branch the referenced subsegment, not the owning segment. Default is false, meaning if the ID is a node then the document that is branched is the owner of the node.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="Swap" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>If true, branch the referenced subsegment and swap in the newly created branch. Default is false.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="BranchOperation" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>If true or unspecified then the resulting document is linked to its parent in the parent's branch table and the document has a "branched from" annotation. If false the system treats the resulting document as if it is developed from scratch.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" ref="tns:ParentID">
    <xs:annotation>
        <xs:documentation>After branching, the parent of the newly branched segment. If unspecified then a stand-alone segment is created. Note this option conflicts with the Swap option.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" ref="tns:InsertLocation">
    <xs:annotation>
        <xs:documentation>If ParentID is specified, the optional position of the newly branched segment under its parent. If unspecified then "last" is assumed.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" ref="tns:AsOf" />
</xs:sequence>

```

```

</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="BranchResult">
<xs:sequence>
<xs:element name="Source" type="tns:ContentType">
<xs:annotation>
<xs:documentation>The original segment in the request list.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Target" type="tns:ContentType">
<xs:annotation>
<xs:documentation>The resulting branched segment.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="BranchSegmentResponseType">
<xs:complexContent>
<xs:extension base="tns>StatusResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" name="Results" type="tns:BranchResult">
<xs:annotation>
<xs:documentation>For each segment in the request list, the resulting content information for the newly branched segment.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="PropagateTracesType">
<xs:complexContent>
<xs:extension base="tns:Credentials">
<xs:sequence>
<xs:element name="CopyFromStart" type="tns:ItemIdReference">
<xs:annotation>
<xs:documentation>The ID of the starting document from which the traces will be copied. CopyFromStart and CopyFromEnd are the two documents having traces between them that you wish to copy.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="CopyFromEnd" type="tns:ItemIdReference">
<xs:annotation>
<xs:documentation>The ID of the ending document from which the traces will be copied. CopyFromStart and CopyFromEnd are the two documents having traces between them that you wish to copy.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="CopyToStart" type="tns:ItemIdReference">
<xs:annotation>
<xs:documentation>The ID of the starting document to which the traces will be copied. CopyToStart and CopyToEnd are the documents to which you wish to copy the traces.</xs:documentation>
</xs:annotation>

```

```

        </xs:annotation>
    </xs:element>
    <xs:element name="CopyToEnd" type="tns:ItemIdReference">
        <xs:annotation>
            <xs:documentation>The ID of the ending document to which the traces will be copied. CopyToStart and CopyToEnd are the documents to which you wish to copy the traces.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="SubSegment" type="xs:boolean">
        <xs:annotation>
            <xs:documentation>If true, and one or more of the CopyTo or CopyFrom IDs is a node that references a subsegment, then the propagation will use the ID of the referenced subsegment, not the node. Default is false, that is, all IDs must be segment root IDs.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="Scope" type="tns:PropagateScope">
        <xs:annotation>
            <xs:documentation>The scope of propagation. If you want to recurse into subdocuments, you must indicate whether to recurse into included subdocuments, or referenced, or both. If not recursive, you have the option of specifying which particular traces you wish to propagate.</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="PropagateScope">
    <xs:sequence>
        <xs:element name="TraceFields">
            <xs:annotation>
                <xs:documentation>The list of trace relationships to propagate. If none is specified then all trace fields are propagated.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="FieldName" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PropagationResult">
    <xs:complexContent>
        <xs:extension base="tns:ContentType">
            <xs:sequence>
                <xs:element name="NumberOfTraces" type="xs:int">
                    <xs:annotation>
                        <xs:documentation>The number of traces created on this target segment.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="NumberSuspect" type="xs:int">

```

```

<xs:annotation>
    <xs:documentation>The number of traces created on this target
    segment that are suspect.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="PropagateTracesResponseType">
    <xs:annotation>
        <xs:documentation>Information about the traces that were propagated.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns:StatusResponseType">
            <xs:sequence>
                <xs:element name="CopyToStart" type="tns:PropagationResult">
                    <xs:annotation>
                        <xs:documentation>Information about the traces created on the
                        "start" document.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="CopyToEnd" type="tns:PropagationResult">
                    <xs:annotation>
                        <xs:documentation>Information about the traces created on the
                        "end" document.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ChangeSegmentProjectType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="ID" type="tns:ItemIdReference">
                    <xs:annotation>
                        <xs:documentation>The IDs for the segments to be
                        reallocated.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="project" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The project into which the segment should be
                        placed.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="RecurseInclude" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>Default true. If true then included subsegments
                        are included in the result.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="RecurseReference" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>Default false. If true then referenced

```

```

    subsegments are included in the result.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="SubSegment" type="xs:boolean">
<xs:annotation>
<xs:documentation>If true, and one or more of the CopyTo or
CopyFrom IDs is a node that references a subsegment, then the
propagation will use the ID of the referenced subsegment, not the
node. Default is false, that is, all IDs must be segment root
IDs.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ChangeSegmentProjectResponseType">
<xs:annotation>
<xs:documentation>List of modifications.</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="tns>StatusResponse">
<xs:sequence>
<xs:element name="Project" type="xs:string">
<xs:annotation>
<xs:documentation>The project to which the segments have been
allocated.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" name="ID" type="tns:ItemIdReference">
<xs:annotation>
<xs:documentation>The list of segment IDs allocated to the
requested project. The list order is not guaranteed and may vary
from one invocation to the next.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:simpleType name="ReferenceMode">
<xs:restriction base="xs:string">
<xs:enumeration value="reuse" />
<xs:enumeration value="share" />
</xs:restriction>
</xs:simpleType>
<xs:complexType name="CopyContentType">
<xs:complexContent>
<xs:extension base="tns:MoveContentType">
<xs:sequence>
<xs:element minOccurs="0" name="BranchOperation" type="xs:boolean">
<xs:annotation>
<xs:documentation>If true or unspecified then the new nodes are
designated as branches of the originals.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="CopyTrace" type="xs:boolean">

```

```

<xs:annotation>
    <xs:documentation>If true then all traces going into or out of the
selection are also copied. Default is false.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="ReferenceMode" type="tns:ReferenceMode">
    <xs:annotation>
        <xs:documentation>The reference mode for all regular content in the
new copy. If unspecified, the default is determined from the Default
Reference Mode attribute on the Type of the parentID
segment.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" ref="tns:AsOf" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ContentDispositionType">
    <xs:annotation>
        <xs:documentation>Details about contents that were copied, moved or
baselined.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns:ContentType">
            <xs:sequence>
                <xs:element name="OriginalContent" type="tns:ContentType">
                    <xs:annotation>
                        <xs:documentation>Details about the original content. For baseline
operations this is the item that was included in the selection
list.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CopyMoveContentResponseType">
    <xs:annotation>
        <xs:documentation>Information about the contents that were copied or
moved.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="Result"
type="tns:ContentDispositionType" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="MoveContentType">
    <xs:complexContent>
        <xs:extension base="tns:RemoveContentType">
            <xs:sequence>
                <xs:element ref="tns:ParentID">
                    <xs:annotation>

```

```

                <xs:documentation>The parent of the newly copied
                content.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" ref="tns:InsertLocation">
            <xs:annotation>
                <xs:documentation>The optional position of the newly copied content
                under its parent. If unspecified then "last" is
                assumed.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="RemoveContentType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="ID" type="tns:ItemIdReference">
                    <xs:annotation>
                        <xs:documentation>The IDs of the nodes to be
                        removed.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Recurse" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>If true then the system removes all sub-content
                        underneath the selection(s). Default is false.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="RemoveContentResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="RemovedContent"
                type="tns:ContentType">
                    <xs:annotation>
                        <xs:documentation>The content that was removed from the
                        segment. Note that this list will not include the sub-contents that are
                        removed from the document if recurse is requested. It lists only the
                        top-level contents that are removed.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="InsertSegmentType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="ID" type="tns:ItemIdReference">
                    <xs:annotation>

```

```

        <xs:documentation>The IDs of the nodes to be
        removed.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element ref="tns:ParentID">
    <xs:annotation>
        <xs:documentation>The parent of the newly copied
        content.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" ref="tns:InsertLocation">
    <xs:annotation>
        <xs:documentation>The optional position of the newly copied content
        under its parent. If unspecified then "last" is
        assumed.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="InsertMode" type="tns:InsertMode">
    <xs:annotation>
        <xs:documentation>The insert mode for the segment. If unspecified,
        then reference is assumed.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="InsertSegmentResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="InsertedContent"
                type="tns:ContentType">
                    <xs:annotation>
                        <xs:documentation>The content that was added to the
                        parent.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="BaselineType">
    <xs:complexContent>
        <xs:extension base="tns:RemoveBaselineType">
            <xs:sequence>
                <xs:element minOccurs="0" name="MoveLabel" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>If the label already exists on the item, then it is
                        moved to the new time.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" ref="tns:AsOf" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

<xs:complexType name="RemoveBaselineType">
  <xs:complexContent>
    <xs:extension base="tns:Credentials">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="ID" type="tns:ItemIdReference">
          <xs:annotation>
            <xs:documentation>The IDs of the segments affected by the baseline operation.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Label" type="xs:string">
          <xs:annotation>
            <xs:documentation>The label for the baseline.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="SubSegment" type="xs:boolean">
          <xs:annotation>
            <xs:documentation>When a selection is a node that references a subsegment, if this flag is true then the application will operate on the referenced subsegment, not the owning segment. Default is false, meaning the segment that is affected is the owning segment.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="BaselineResponseType">
  <xs:complexContent>
    <xs:extension base="tns>StatusResponse">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Segment" type="tns:ContentDispositionType">
          <xs:annotation>
            <xs:documentation>The content that was affected by the baseline operation. The primary content is the affected segment. The original content is the selection item, which could be a node inside the segment or a node referencing the subsegment in the case of SubSegment=true.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ReferenceModeType">
  <xs:complexContent>
    <xs:extension base="tns:Credentials">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="ID" type="tns:ItemIdReference" />
        <xs:element name="ReferenceMode" type="tns:ReferenceMode" />
        <xs:element minOccurs="0" name="ShareID" type="tns:ItemIdReference">
          <xs:annotation>
            <xs:documentation>For share reference mode, the ID of the item from which the selected item will share its content. If unspecified, then the default is to share from the node from which the current node was originally branched.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ReferenceModeResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="ID" type="tns:ContentType">
                    <xs:annotation>
                        <xs:documentation>The content that was affected by the reference mode operation.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ToggleIncludeReferenceType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="ID" type="tns:ItemIdReference" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ToggleIncludeReferenceResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="ID" type="tns:ContentDispositionType">
                    <xs:annotation>
                        <xs:documentation>The current state of the subsegment reference, along with its original state.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/10/Segment/fault" version="1.0">
    <xs:element name="MKSException" nillable="true" type="tns:MKSFault" />
    <xs:complexType name="MKSFault">
        <xs:annotation>
            <xs:documentation>The original fault generated from MKS Integrity. The element value is the full description of the error.</xs:documentation>
        </xs:annotation>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="class" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The class of exception</xs:documentation>

```

```

        </xs:annotation>
    </xs:attribute>
    <xs:attribute name="id" type="xs:string">
        <xs:annotation>
            <xs:documentation>The unique MKS identifier for this error, generally a 6-digit positive integer.</xs:documentation>
        </xs:annotation>
    </xs:attribute>
    <xs:attribute name="implication" type="xs:string">
        <xs:annotation>
            <xs:documentation>The implication from the exception</xs:documentation>
        </xs:annotation>
    </xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/10/Segment" version="1.0">
    <xs:import namespace="http://webservice.mks.com/10/Segment/schema" />
    <xs:complexType name="createSegmentRequest">
        <xs:sequence>
            <xs:element minOccurs="0" name="arg0" type="ns1:CreateSegmentType" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="createSegmentResponse">
        <xs:sequence>
            <xs:element minOccurs="0" name="return" type="ns1:CreateResponseType" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="createContentRequest">
        <xs:sequence>
            <xs:element minOccurs="0" name="arg0" type="ns1:CreateContentType" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="createContentResponse">
        <xs:sequence>
            <xs:element minOccurs="0" name="return" type="ns1:CreateResponseType" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="viewSegmentRequest">
        <xs:sequence>
            <xs:element minOccurs="0" name="arg0" type="ns1:ViewSegmentType" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="viewSegmentResponse">
        <xs:sequence>
            <xs:element minOccurs="0" name="return" type="ns1:ViewSegmentResponseType" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="importSegmentRequest">
        <xs:sequence>
            <xs:element minOccurs="0" name="arg0" type="ns1:ImportSegmentType" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="importSegmentResponse">

```

```

<xs:sequence>
    <xs:element minOccurs="0" name="return" type="ns1>CreateResponseType" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="importContentRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ImportContentType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="importContentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>CreateResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="branchSegmentRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:BranchSegmentType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="branchSegmentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:BranchSegmentResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="propagateTracesRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:PropagateTracesType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="propagateTracesResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:PropagateTracesResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="changeSegmentProjectRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ChangeSegmentProjectType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="changeSegmentProjectResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ChangeSegmentProjectResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="copyContentRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:CopyContentType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="copyContentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:CopyMoveContentResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="moveContentRequest">

```

```

<xs:sequence>
    <xs:element minOccurs="0" name="arg0" type="ns1:MoveContentType" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="moveContentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:CopyMoveContentResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeContentRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:RemoveContentType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeContentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:RemoveContentResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="insertSegmentRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:InsertSegmentType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="insertSegmentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:InsertSegmentResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="baselineRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:BaselineType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="baselineResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:BaselineResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeBaselineRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:RemoveBaselineType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="removeBaselineResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:BaselineResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="referenceModeRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ReferenceModeType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="referenceModeResponse">

```

```

<xs:sequence>
    <xs:element minOccurs="0" name="return" type="ns1:ReferenceModeResponseType" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="toggleIncludeReferenceRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ToggleIncludeReferenceType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="toggleIncludeReferenceResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:ToggleIncludeReferenceResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:element name="createSegment" type="tns:createSegmentRequest" />
<xs:element name="createSegmentResponse" type="tns:createSegmentResponse" />
<xs:element name="createContent" type="tns:createContentRequest" />
<xs:element name="createContentResponse" type="tns:createContentResponse" />
<xs:element name="viewSegment" type="tns:viewSegmentRequest" />
<xs:element name="viewSegmentResponse" type="tns:viewSegmentResponse" />
<xs:element name="importSegment" type="tns:importSegmentRequest" />
<xs:element name="importSegmentResponse" type="tns:importSegmentResponse" />
<xs:element name="importContent" type="tns:importContentRequest" />
<xs:element name="importContentResponse" type="tns:importContentResponse" />
<xs:element name="branchSegment" type="tns:branchSegmentRequest" />
<xs:element name="branchSegmentResponse" type="tns:branchSegmentResponse" />
<xs:element name="propagateTraces" type="tns:propagateTracesRequest" />
<xs:element name="propagateTracesResponse" type="tns:propagateTracesResponse" />
<xs:element name="changeSegmentProject" type="tns:changeSegmentProjectRequest" />
<xs:element name="changeSegmentProjectResponse" type="tns:changeSegmentProjectResponse" />
<xs:element name="copyContent" type="tns:copyContentRequest" />
<xs:element name="copyContentResponse" type="tns:copyContentResponse" />
<xs:element name="moveContent" type="tns:moveContentRequest" />
<xs:element name="moveContentResponse" type="tns:moveContentResponse" />
<xs:element name="removeContent" type="tns:removeContentRequest" />
<xs:element name="removeContentResponse" type="tns:removeContentResponse" />
<xs:element name="insertSegment" type="tns:insertSegmentRequest" />
<xs:element name="insertSegmentResponse" type="tns:insertSegmentResponse" />
<xs:element name="baseline" type="tns:baselineRequest" />
<xs:element name="baselineResponse" type="tns:baselineResponse" />
<xs:element name="removeBaseline" type="tns:removeBaselineRequest" />
<xs:element name="removeBaselineResponse" type="tns:removeBaselineResponse" />
<xs:element name="referenceMode" type="tns:referenceModeRequest" />
<xs:element name="referenceModeResponse" type="tns:referenceModeResponse" />
<xs:element name="toggleIncludeReference" type="tns:toggleIncludeReferenceRequest" />
<xs:element name="toggleIncludeReferenceResponse" type="tns:toggleIncludeReferenceResponse" />
</xs:schema>
</types>
<message name="Segment_10_importContentResponse">
    <part element="tns:importContentResponse" name="importContentResponse" />
</message>
<message name="Segment_10_baseline">
    <part element="tns:baseline" name="baseline" />

```

```

</message>
<message name="Segment_10_propagateTracesResponse">
    <part element="tns:propagateTracesResponse" name="propagateTracesResponse" />
</message>
<message name="Segment_10_viewSegment">
    <part element="tns:viewSegment" name="viewSegment" />
</message>
<message name="Segment_10_toggleIncludeReferenceResponse">
    <part element="tns:toggleIncludeReferenceResponse" name="toggleIncludeReferenceResponse" />
</message>
<message name="Segment_10_branchSegment">
    <part element="tns:branchSegment" name="branchSegment" />
</message>
<message name="Segment_10_createContentResponse">
    <part element="tns:createContentResponse" name="createContentResponse" />
</message>
<message name="Segment_10_changeSegmentProjectResponse">
    <part element="tns:changeSegmentProjectResponse" name="changeSegmentProjectResponse" />
</message>
<message name="Segment_10_removeBaseline">
    <part element="tns:removeBaseline" name="removeBaseline" />
</message>
<message name="Segment_10_importSegmentResponse">
    <part element="tns:importSegmentResponse" name="importSegmentResponse" />
</message>
<message name="Segment_10_copyContentResponse">
    <part element="tns:copyContentResponse" name="copyContentResponse" />
</message>
<message name="Segment_10_removeContent">
    <part element="tns:removeContent" name="removeContent" />
</message>
<message name="Segment_10_propagateTraces">
    <part element="tns:propagateTraces" name="propagateTraces" />
</message>
<message name="Segment_10_moveContent">
    <part element="tns:moveContent" name="moveContent" />
</message>
<message name="Segment_10_insertSegmentResponse">
    <part element="tns:insertSegmentResponse" name="insertSegmentResponse" />
</message>
<message name="Segment_10_branchSegmentResponse">
    <part element="tns:branchSegmentResponse" name="branchSegmentResponse" />
</message>
<message name="Segment_10_insertSegment">
    <part element="tns:insertSegment" name="insertSegment" />
</message>
<message name="Segment_10_toggleIncludeReference">
    <part element="tns:toggleIncludeReference" name="toggleIncludeReference" />
</message>
<message name="Segment_10_viewSegmentResponse">
    <part element="tns:viewSegmentResponse" name="viewSegmentResponse" />
</message>
<message name="Segment_10_changeSegmentProject">
    <part element="tns:changeSegmentProject" name="changeSegmentProject" />

```

```

</message>
<message name="Segment_10_removeContentResponse">
    <part element="tns:removeContentResponse" name="removeContentResponse" />
</message>
<message name="MKSEception">
    <part element="ns2:MKSEception" name="MKSEception" />
</message>
<message name="Segment_10_createSegmentResponse">
    <part element="tns:createSegmentResponse" name="createSegmentResponse" />
</message>
<message name="Segment_10_createSegment">
    <part element="tns:createSegment" name="createSegment" />
</message>
<message name="Segment_10_removeBaselineResponse">
    <part element="tns:removeBaselineResponse" name="removeBaselineResponse" />
</message>
<message name="Segment_10_createContent">
    <part element="tns:createContent" name="createContent" />
</message>
<message name="Segment_10_moveContentResponse">
    <part element="tns:moveContentResponse" name="moveContentResponse" />
</message>
<message name="Segment_10_importSegment">
    <part element="tns:importSegment" name="importSegment" />
</message>
<message name="Segment_10_referenceModeResponse">
    <part element="tns:referenceModeResponse" name="referenceModeResponse" />
</message>
<message name="Segment_10_referenceMode">
    <part element="tns:referenceMode" name="referenceMode" />
</message>
<message name="Segment_10_copyContent">
    <part element="tns:copyContent" name="copyContent" />
</message>
<message name="Segment_10_baselineResponse">
    <part element="tns:baselineResponse" name="baselineResponse" />
</message>
<message name="Segment_10_importContent">
    <part element="tns:importContent" name="importContent" />
</message>
<portType name="Segment_10">
    <operation name="createSegment" parameterOrder="createSegment">
        <input message="tns:Segment_10_createSegment" />
        <output message="tns:Segment_10_createSegmentResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="createContent" parameterOrder="createContent">
        <input message="tns:Segment_10_createContent" />
        <output message="tns:Segment_10_createContentResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="viewSegment" parameterOrder="viewSegment">
        <input message="tns:Segment_10_viewSegment" />
        <output message="tns:Segment_10_viewSegmentResponse" />
    </operation>

```

```

<fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="importSegment" parameterOrder="importSegment">
    <input message="tns:Segment_10_importSegment" />
    <output message="tns:Segment_10_importSegmentResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="importContent" parameterOrder="importContent">
    <input message="tns:Segment_10_importContent" />
    <output message="tns:Segment_10_importContentResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="branchSegment" parameterOrder="branchSegment">
    <input message="tns:Segment_10_branchSegment" />
    <output message="tns:Segment_10_branchSegmentResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="propagateTraces" parameterOrder="propagateTraces">
    <input message="tns:Segment_10_propagateTraces" />
    <output message="tns:Segment_10_propagateTracesResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="changeSegmentProject" parameterOrder="changeSegmentProject">
    <input message="tns:Segment_10_changeSegmentProject" />
    <output message="tns:Segment_10_changeSegmentProjectResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="copyContent" parameterOrder="copyContent">
    <input message="tns:Segment_10_copyContent" />
    <output message="tns:Segment_10_copyContentResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="moveContent" parameterOrder="moveContent">
    <input message="tns:Segment_10_moveContent" />
    <output message="tns:Segment_10_moveContentResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="removeContent" parameterOrder="removeContent">
    <input message="tns:Segment_10_removeContent" />
    <output message="tns:Segment_10_removeContentResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="insertSegment" parameterOrder="insertSegment">
    <input message="tns:Segment_10_insertSegment" />
    <output message="tns:Segment_10_insertSegmentResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="baseline" parameterOrder="baseline">
    <input message="tns:Segment_10_baseline" />
    <output message="tns:Segment_10_baselineResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="removeBaseline" parameterOrder="removeBaseline">

```

```

<input message="tns:Segment_10_removeBaseline" />
<output message="tns:Segment_10_removeBaselineResponse" />
<fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="referenceMode" parameterOrder="referenceMode">
<input message="tns:Segment_10_referenceMode" />
<output message="tns:Segment_10_referenceModeResponse" />
<fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="toggleIncludeReference" parameterOrder="toggleIncludeReference">
<input message="tns:Segment_10_toggleIncludeReference" />
<output message="tns:Segment_10_toggleIncludeReferenceResponse" />
<fault message="tns:MKSEception" name="MKSEception" />
</operation>
</portType>
<binding name="Segment_10Binding" type="tns:Segment_10">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<wsp:PolicyReference URI="#wsmtom_policy" required="true"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
/>
<operation name="createSegment">
<soap:operation soapAction="CreateSegment" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
<fault name="MKSEception">
<soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="createContent">
<soap:operation soapAction="CreateContent" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
<fault name="MKSEception">
<soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="viewSegment">
<soap:operation soapAction="ViewSegment" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
<fault name="MKSEception">
<soap:fault name="MKSEception" use="literal" />
</fault>

```

```

        </fault>
    </operation>
<operation name="importSegment">
    <soap:operation soapAction="ImportSegment" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="importContent">
    <soap:operation soapAction="ImportContent" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="branchSegment">
    <soap:operation soapAction="BranchSegment" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="propagateTraces">
    <soap:operation soapAction="PropagateTraces" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEException">
        <soap:fault name="MKSEException" use="literal" />
    </fault>
</operation>
<operation name="changeSegmentProject">
    <soap:operation soapAction="ChangeSegmentProject" />
    <input>
        <soap:body use="literal" />

```

```

</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEception">
    <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="copyContent">
    <soap:operation soapAction="CopyContent" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="moveContent">
    <soap:operation soapAction="MoveContent" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="removeContent">
    <soap:operation soapAction="RemoveContent" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="insertSegment">
    <soap:operation soapAction="insertSegment" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>

```

```

        </fault>
    </operation>
<operation name="baseline">
    <soap:operation soapAction="baseline" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="removeBaseline">
    <soap:operation soapAction="removeBaseline" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="referenceMode">
    <soap:operation soapAction="referenceMode" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="toggleIncludeReference">
    <soap:operation soapAction="toggleIncludeReference" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
</binding>
<service name="Segment_10Service">
    <port binding="tns:Segment_10Binding" name="Segment_10Port">
        <soap:address location="http://IntegrityServer:9001/webservices/10/Segment/" />
    </port>

```

```
</service>
<wsp:Policy wsu:id="wsmtom_policy"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <wsoma:OptimizedMimeSerialization
        xmlns:wsoma="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization"
        />
</wsp:Policy>
</definitions>
```

Web Service: Test_10Service

Web Service: Test_10Service

Target Namespace: <http://webservice.mks.com/10/Test>

Port **Test_10Port** [Port type](#) [Source code](#)

Location: <http://IntegrityServer:9001/webservices/10/Test/>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. [createResults](#) [Detail](#) [Source code](#)
2. [createStepResults](#) [Detail](#) [Source code](#)
3. [deleteResults](#) [Detail](#) [Source code](#)
4. [deleteStepResults](#) [Detail](#) [Source code](#)
5. [editResults](#) [Detail](#) [Source code](#)
6. [editStepResults](#) [Detail](#) [Source code](#)
7. [fetchAttachment](#) [Detail](#) [Source code](#)
8. [getResultsDetails](#) [Detail](#) [Source code](#)
9. [getResults](#) [Detail](#) [Source code](#)
10. [getStepResults](#) [Detail](#) [Source code](#)
11. [getTestCases](#) [Detail](#) [Source code](#)
12. [setResults](#) [Detail](#) [Source code](#)
13. [viewUntested](#) [Detail](#) [Source code](#)

Operations

Port type **Test_10** [Source code](#)

1. [createResults](#)

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_createResults (soap:body, use = literal) [Source code](#)

createResults type [createResults](#)

- arg0 - optional; type [CreateResultsType](#) - extension of type [GenericCreateResultsType](#) - extension of type [Credentials](#)
 - Username type [string](#)
 - Password type [string](#)
 - ImpersonatedUser - optional; type [string](#)
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type [string](#)
The formatting syntax for date fields without a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type [string](#)

The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the `dateTimeFormat` entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- `transactionId` type `string`
 An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- `SessionID` type `ItemIdReference` - type `integer` with restriction
 The test session for this result.
- `Verdict` type `string`
 Overall test result verdict. Could be Passed, Failed, Skipped or a custom verdict configured on the system.
- `Overwrite` - optional; type `boolean`
 Default true. If a result already exists, overwrite it with the new values. This flag is ignored for edit operations.
- `Note` - optional; type `string`
 Add an annotation to the result.
- `TestCaseID` - unbounded; type `ItemIdReference` - type `integer` with restriction
 The test case(s) for this result. The same information is attached to each test case in this list.
- `RelatedItemID` - optional, unbounded; type `ItemIdReference` - type `integer` with restriction
 Add a related item to the result.
- `Attachment` - optional, unbounded; type `Attachment`
 Add an annotation to the result.
 - Name type `string`
 The attachment name.
 - Description - optional; type `string`
 A description of the attachment.
 - Contents - optional; type `base64Binary`
 The attachment contents.

Output: Test_10_createResultsResponse (soap:body, use = literal) [Source code](#)

createResultsResponse type `createResultsResponse`

- `return` - optional; type `StatusResponse`
 - `return` - optional; type `string`
- `transactionId` type `string`
 The transaction ID that was passed in with the original request.

Fault: MKSEception (soap:fault, use = literal) [Source code](#)

MKSEception type `MKSEception`

The original fault generated from MKS Integrity. The element value is the full description of the error.

- `class` type `string`
 The class of exception
- `id` type `string`
 The unique MKS identifier for this error, generally a 6-digit positive integer.
- `implication` type `string`
 The implication from the exception

2. createStepResults

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_createStepResults (soap:body, use = literal) [Source code](#)

createStepResults type `createStepResults`

- `arg0` - optional; type `CreateStepResultsType` - extension of type `GenericCreateResultsType` - extension of type `Credentials`
 - Username type `string`
 - Password type `string`
 - ImpersonatedUser - optional; type `string`
 Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.

- **DateFormat** - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the `dateTimeFormat` entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- **DateTimeFormat** - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the `dateTimeFormat` entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- **transactionId** type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- **SessionID** type *ItemIdReference* - type *integer* with restriction
The test session for this result.
- **Verdict** type *string*
Overall test result verdict. Could be Passed, Failed, Skipped or a custom verdict configured on the system.
- **Overwrite** - optional; type *boolean*
Default true. If a result already exists, overwrite it with the new values. This flag is ignored for edit operations.
- **Note** - optional; type *String*
Add an annotation to the result.
- **TestCaseID** type *ItemIdReference* - type *integer* with restriction
The test case for this result.
- **TestStepID** - unbounded; type *ItemIdReference* - type *integer* with restriction
The test step(s) to which the verdict is to be assigned.

Output: Test_10_createStepResultsResponse (soap:body, use = literal) [Source code](#)

createStepResultsResponse type *createStepResultsResponse*

- **return** - optional; type *StatusResponse*
 - **return** - optional; type *string*
 - **transactionId** type *string*
The transaction ID that was passed in with the original request.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- **class** type *string*
The class of exception
- **id** type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- **implication** type *string*
The implication from the exception

3. **deleteResults**

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_deleteResults (soap:body, use = literal) [Source code](#)

deleteResults type *deleteResults*

- **arg0** - optional; type *DeleteResultsType* - extension of type *Credentials*
 - **Username** type *string*
 - **Password** type *string*
 - **ImpersonatedUser** - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - **DateFormat** - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the `dateTimeFormat` entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

DateTimeFormat - optional; type *string*

The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- transactionId type *string*

An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.

- TestCases type *TestCaseList*

- SessionID type *ItemIdReference* - type *integer* with restriction
The test session.

- TestCaseID - unbounded; type *ItemIdReference* - type *integer* with restriction

One or more test cases.

Output: Test_10_deleteResultsResponse (soap:body, use = literal) [Source code](#)

deleteResultsResponse type *deleteResultsResponse*

- return - optional; type *StatusResponse*

- return - optional; type *string*

- transactionId type *string*

The transaction ID that was passed in with the original request.

Fault: MKSEception (soap:fault, use = literal) [Source code](#)

MKSEception type *MKSEception*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*

The class of exception

- id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- implication type *string*

The implication from the exception

4. deleteStepResults

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_deleteStepResults (soap:body, use = literal) [Source code](#)

deleteStepResults type *deleteStepResults*

- arg0 - optional; type *DeleteStepResultsType* - extension of type *Credentials*

- Username type *string*

- Password type *string*

- ImpersonatedUser - optional; type *string*

Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.

- DateFormat - optional; type *string*

The formatting syntax for date fields without a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- DateTimeFormat - optional; type *string*

The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- transactionId type *string*

An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.

- SessionID type *ItemIdReference* - type *integer* with restriction
The test session.

- TestCaseID type *ItemIdReference* - type *integer* with restriction
Test affected test case.

- TestStepID - unbounded; type *ItemIdReference* - type *integer* with restriction

One or more test steps.

Output: Test_10_deleteStepResultsResponse (soap:body, use = literal) [Source code](#)

deleteStepResultsResponse type *deleteStepResultsResponse*

- return - optional; type *StatusResponse*
 - return - optional; type *string*
- transactionId type *string*
The transaction ID that was passed in with the original request.

Fault: MKSEexception (soap:fault, use = literal) [Source code](#)

MKSEexception type *MKSEexception*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

5. editResults

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_editResults (soap:body, use = literal) [Source code](#)

editResults type *editResults*

- arg0 - optional; type *EditResultsType* - extension of type *CreateResultsType* - extension of type *GenericCreateResultsType* - extension of type *Credentials*
 - Username type *string*
 - Password type *String*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- SessionID type *ItemIdReference* - type *integer* with restriction
The test session for this result.
- Verdict type *String*
Overall test result verdict. Could be Passed, Failed, Skipped or a custom verdict configured on the system.
- Overwrite - optional; type *boolean*
Default true. If a result already exists, overwrite it with the new values. This flag is ignored for edit operations.
- Note - optional; type *string*
Add an annotation to the result.
- TestCaseID - unbounded; type *ItemIdReference* - type *integer* with restriction
The test case(s) for this result. The same information is attached to each test case in this list.
- RelatedItemID - optional, unbounded; type *ItemIdReference* - type *integer* with restriction
Add a related item to the result.
- Attachment - optional, unbounded; type *Attachment*

Add an annotation to the result.

- Name type *string*
The attachment name.
- Description - optional; type *string*
A description of the attachment.
- Contents - optional; type *base64Binary*
The attachment contents.
- RemoveRelatedItemID - optional, unbounded; type *ItemIdReference* - type *integer* with restriction
Remove a related item from the result.
- RemoveAttachment - optional, unbounded; type *string*
Remove the attachment with the indicated name from the result.
- CreateIfNotExists - optional; type *boolean*
Default true, meaning a result is created if one does not exist. If false and the result does not exist, then a new one is not created.

Output: Test_10_editResultsResponse (soap:body, use = literal) [Source code](#)

editResultsResponse type *editResultsResponse*

- return - optional; type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

6. editStepResults

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_editStepResults (soap:body, use = literal) [Source code](#)

editStepResults type *editStepResults*

- arg0 - optional; type *EditStepResultsType* - extension of type *CreateStepResultsType* - extension of type *GenericCreateResultsType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- SessionID type *ItemIdReference* - type *integer* with restriction
The test session for this result.
- Verdict type *string*

- Overall test result verdict. Could be Passed, Failed, Skipped or a custom verdict configured on the system.
- Overwrite - optional; type *boolean*
Default true. If a result already exists, overwrite it with the new values. This flag is ignored for edit operations.
 - Note - optional; type *String*
Add an annotation to the result.
 - TestCaseID type *ItemIdReference* - type *integer* with restriction
The test case for this result.
 - TestStepID - unbounded; type *ItemIdReference* - type *integer* with restriction
The test step(s) to which the verdict is to be assigned.
 - CreateIfNotExists - optional; type *boolean*
Default true, meaning a result is created if one does not exist. If true and the result does not yet exist, then a new one is not created.

Output: Test_10_editStepResultsResponse (soap:body, use = literal) [Source code](#)

editStepResultsResponse type *editStepResultsResponse*

- return - optional; type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

7. fetchAttachment

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_fetchAttachment (soap:body, use = literal) [Source code](#)

fetchAttachment type *fetchAttachment*

- arg0 - optional; type *FetchAttachmentType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- Name type *string*
The attachment name.
- ResultID type *string*
Get attachment for this particular result, in the format .

Output: Test_10_fetchAttachmentResponse (soap:body, use = literal) [Source code](#)

fetchAttachmentResponse type *fetchAttachmentResponse*

- return - optional; type *FetchAttachmentResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.
 - Attachment - optional; type *base64Binary*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

8. getResultDetails

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_getResultDetails (soap:body, use = literal) [Source code](#)

getResultDetails type *getResultDetails*

- arg0 - optional; type *GetResultDetailsType* - extension of type *GetStepResultsType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- TestCases type *TestCaseList*
 - SessionID type *ItemIdReference* - type *integer* with restriction
The test session.
 - TestCaseID - unbounded; type *ItemIdReference* - type *integer* with restriction
One or more test cases.
- IncludeStepResults - optional; type *boolean*
Default is true. If true, then the results for test steps are included in the output.

Output: Test_10_getResultDetailsResponse (soap:body, use = literal) [Source code](#)

getResultSetDetailsResponse type *getResultSetDetailsResponse*

- return - optional; type *GetResultSetDetailsResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.
 - Result - optional, unbounded; type *ResultDetails* - extension of type *ResultInfo*

- **ID** type *string*
The test result ID in the form ::
- **SessionID** type *ItemIdReference* - type *integer* with restriction
The test session during which this result was collected.
- **TestCaseID** type *ItemIdReference* - type *integer* with restriction
The test case ID.
- **SharesTestCaseID** - optional; type *ItemIdReference* - type *integer* with restriction
If the result was entered for a test case that shares content with this test case, and the results are shared to this test case, then the shared test case ID is that of the test case for which the result was entered.
- **ModifiedDate** type *string*
Time stamp for the last modification, in the format specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html> If the format is not specified in the request, the default is "yyyy-MM-dd HH:mm:ss".
- **ModifiedBy** type *string*
The login ID of the user who entered this result.
- **Verdict** - optional; type *string*
The verdict for this result. Could be Passed, Failed, Skipped or a custom verdict configured on the system. May also be missing if no verdict has yet to be set on the test case.
- **VerdictType** - optional; type *VerdictType* - type *string* with restriction - enum { 'Pass', 'Fail', 'Other' }
The class of verdict for this result. Not specified if there is no verdict.
- **Note** - optional; type *string*
Annotation entered by the tester regarding this result.
- **HasStepResult** type *boolean*
Does this result include results for individual test steps?
- **HasAttachment** type *boolean*
Does this result have an associated attachment?
- **HasRelatedItem** type *boolean*
Has this result been associated with another item in the repository?
- **Attachment** - optional, unbounded; type *string*
The attachment name for each file attached to this result.
- **RelatedItemID** - optional, unbounded; type *ItemIdReference* - type *integer* with restriction
The ID of each item related to this result.
- **Step** - optional, unbounded; type *StepResult* - extension of type *StepVerdict*
The verdicts for test steps associated with this test result. If the test case has one or more test steps but some step verdicts were not entered for this result, then only those verdicts that were entered are included in the output. Thus it is possible to have a test case that has multiple test steps yet none of the steps is included in the result for a particular session.
 - **ID** type *ItemIdReference* - type *integer* with restriction
The test step item ID.
 - **Verdict** type *string*
The verdict for this step result. Could be Passed, Failed, Skipped or a custom verdict configured on the system.
 - **Note** - optional; type *string*
Annotation entered by the tester regarding this test step result.
 - **TestStepID** type *string*
The test step ID in the format ::.

Fault: MKSEception (soap:fault, use = literal) [Source code](#)

MKSEception type *MKSEception*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- **class** type *string*
The class of exception
- **id** type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- **implication** type *string*
The implication from the exception

9. getResults

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_getResults (soap:body, use = literal) [Source code](#)

getResults type *getResults*

- arg0 - optional; type *GetResultsType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- Selections type *ResultsSelectionType*
 - TestCase type *ItemIdReference* - type *integer* with restriction
Get results for a single test case, including results for test cases in group documents that share content with this test case.
 - Session type *SessionResultsSelectionType*
Get results for a single session. Note that you can't request multiple arbitrary combinations of test sessions and test cases. You may request multiple test cases for a single session.
 - ID type *ItemIdReference* - type *integer* with restriction
Get results for this test session.
If neither of these options is specified, then results for all test cases and all test suites are included in the output.
 - SuiteID type *ItemIdReference* - type *integer* with restriction
Include results only from this test suite.
 - TestCaseID - unbounded; type *ItemIdReference* - type *integer* with restriction
Include results only from these test cases.
 - RelatedToItem type *ItemIdReference* - type *integer* with restriction
Get results that are related to this item.
- OnlyLastResult - optional; type *boolean*
Include only the latest results. Default is to include all historical results.
- Verdict - optional; type *string*
Include only verdicts having the specified value. Default is to include all verdicts.

Output: Test_10_getResultsResponse (soap:body, use = literal) [Source code](#)

getResultsResponse type *getResultsResponse*

- return - optional; type *GetResultsResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
- transactionId type *string*
The transaction ID that was passed in with the original request.
- Result - optional, unbounded; type *ResultInfo*
 - ID type *string*
The test result ID in the form : ..
 - SessionID type *ItemIdReference* - type *integer* with restriction
The test session during which this result was collected.
 - TestCaseID type *ItemIdReference* - type *integer* with restriction
The test case ID.
 - SharesTestCaseID - optional; type *ItemIdReference* - type *integer* with restriction

If the result was entered for a test case that shares content with this test case, and the results are shared to this test case, then the shared test case ID is that of the test case for which the result was entered.

- **ModifiedDate** type *string*
Time stamp for the last modification, in the format specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html> If the format is not specified in the request, the default is "yyyy-MM-dd HH:mm:ss".
- **ModifiedBy** type *string*
The login ID of the user who entered this result.
- **Verdict - optional;** type *string*
The verdict for this result. Could be Passed, Failed, Skipped or a custom verdict configured on the system. May also be missing if no verdict has yet to be set on the test case.
- **VerdictType - optional;** type *VerdictType* - type *string* with restriction - enum { 'Pass', 'Fail', 'Other' }
The class of verdict for this result. Not specified if there is no verdict.
- **Note - optional;** type *string*
Annotation entered by the tester regarding this result.
- **HasStepResult** type *boolean*
Does this result include results for individual test steps?
- **HasAttachment** type *boolean*
Does this result have an associated attachment?
- **HasRelatedItem** type *boolean*
Has this result been associated with another item in the repository?

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- **class** type *string*
The class of exception
- **id** type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- **implication** type *string*
The implication from the exception

10. getStepResults

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_getStepResults (soap:body, use = literal) [Source code](#)

getStepResults type *getStepResults*

- **arg0 - optional;** type *GetStepResultsType* - extension of type *Credentials*
 - **Username** type *string*
 - **Password** type *string*
 - **ImpersonatedUser - optional;** type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - **DateFormat - optional;** type *string*
The formatting syntax for date fields without a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - **DateTimeFormat - optional;** type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the *dateTimeFormat* entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
- **transactionId** type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- **TestCases** type *TestCaseList*

- SessionID type *ItemIdReference* - type *integer* with restriction
The test session.
- TestCaseID - unbounded; type *ItemIdReference* - type *integer* with restriction
One or more test cases.

Output: Test_10_getStepResultsResponse (soap:body, use = literal) [Source code](#)

getStepResultsResponse type *getStepResultsResponse*

- return - optional; type *GetStepResultsResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
The transaction ID that was passed in with the original request.
 - Result - optional, unbounded; type *FullStepResult* - extension of type *StepResult* - extension of type *StepVerdict*
 - ID type *ItemIdReference* - type *integer* with restriction
The test step item ID.
 - Verdict type *string*
The verdict for this step result. Could be Passed, Failed, Skipped or a custom verdict configured on the system.
 - Note - optional; type *string*
Annotation entered by the tester regarding this test step result.
 - TestStepID type *string*
The test step ID in the format ::.
 - SessionID type *ItemIdReference* - type *integer* with restriction
The test session for this step result.
 - TestCaseID type *ItemIdReference* - type *integer* with restriction
The test case for this step result.

Fault: MKSEception (soap:fault, use = literal) [Source code](#)

MKSEception type *MKSEception*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

11. getTestCases

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_getTestCases (soap:body, use = literal) [Source code](#)

getTestCases type *getTestCases*

- arg0 - optional; type *GetTestCasesType* - extension of type *Credentials*
Note that this command only allows information to be requested for one session at a time.
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>

- transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- SessionID type *ItemIdReference* - type *integer* with restriction
- InputField - optional, unbounded; type *string*
Comma-separated list of fields to include in the output.
- SubstituteParameters - optional; type *boolean*
Default is false. If true, then the values of parameters embedded in the fields are substituted in the field return value.
- Filter - optional; type *string*
A condition that filters the results. For example, (field[ID] = 610) reports only on item 610.

Output: Test_10_getTestCasesResponse (soap:body, use = literal) [Source code](#)

- getTestCasesResponse** type *getTestCasesResponse*
- return - optional; type *GetTestCasesResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*
 - transactionId type *string*
The transaction ID that was passed in with the original request.
 - Item - optional, unbounded; type *ItemType*
 - GetItemField - optional, unbounded; type *GetItemField*
 - ItemId type *ItemIdReference* - type *integer* with restriction

Fault: MKSEception (soap:fault, use = literal) [Source code](#)

- MKSEception** type *MKSEception*
- The original fault generated from MKS Integrity. The element value is the full description of the error.
- class type *string*
The class of exception
 - id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
 - implication type *string*
The implication from the exception

12. setResults

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_setResults (soap:body, use = literal) [Source code](#)

- setResults** type *setResults*
- arg0 - optional; type *SetResultsType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateTimeFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
 - SessionID type *ItemIdReference* - type *integer* with restriction
The session for which these results have been gathered.
 - ActionDefinitionFile type *base64Binary*

The file containing the instructions.

- Attachment - optional, unbounded; type *SetResultsAttachment*
The list of attachments referenced from the action definition file.
 - Name type *string*
The name of the file as referenced inside the action definition file. The file can be referenced from more than one AddAttachment entity. Note that the AddAttachment entity must not specify a directory offset to the file.
 - Contents type *base64Binary*
The contents of the attachment.

Output: Test_10_setResultsResponse (soap:body, use = literal) [Source code](#)

setResultsResponse type *setResultsResponse*

- return - optional; type *StatusResponse*
 - return - optional; type *string*
- transactionId type *string*
The transaction ID that was passed in with the original request.

Fault: MKSEception (soap:fault, use = literal) [Source code](#)

MKSEception type *MKSEception*
The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*
The class of exception
- id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- implication type *string*
The implication from the exception

13. viewUntested

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Test_10_viewUntested (soap:body, use = literal) [Source code](#)

viewUntested type *viewUntested*

- arg0 - optional; type *ViewUntestedType* - extension of type *Credentials*
 - Username type *string*
 - Password type *string*
 - ImpersonatedUser - optional; type *string*
Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.
 - DateFormat - optional; type *string*
The formatting syntax for date fields without a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - DateTimeFormat - optional; type *string*
The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:
<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>
 - transactionId type *string*
An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.
- Selection type *UntestedSelection*
 - SessionID type *ItemIdReference* - type *integer* with restriction
The ID of a test session. When this option is selected the command returns all test cases related to the session via the Tests relationship that were not tested in that session.
 - UntestedByObjective
 - TestObjectiveID type *ItemIdReference* - type *integer* with restriction
Any item (typically a test objective) for which the built-in Tests field is visible. The test cases listed in the output will include only those related via the Tests field (and/or by walking down the document hierarchy) will be included in the output.

- SessionID - optional, unbounded; type *ItemIdReference* - type *integer* with restriction

Test sessions. If no session IDs are listed in the request document then all test cases related via the Tests relationship to the objective are included in the output. If at least one session is included in the request document then only those test cases that were not tested in any of the indicated sessions are included in the output.

- InputField - optional, unbounded; type *string*

Comma-separated list of fields to include in the output. If none is specified then a minimal set of fields is included in the output.

Output: Test_10_viewUntestedResponse (soap:body, use = literal) [Source code](#)

viewUntestedResponse type *viewUntestedResponse*

- return - optional; type *GetTestCasesResponseType* - extension of type *StatusResponse*
 - return - optional; type *string*

- transactionId type *string*

The transaction ID that was passed in with the original request.

- Item - optional, unbounded; type *ItemType*
 - GetItemField - optional, unbounded; type *GetItemField*

- ItemId type *ItemIdReference* - type *integer* with restriction

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- class type *string*

The class of exception

- id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- implication type *string*

The implication from the exception

WSDL source code

```
<?xml version="1.0"?>
<definitions name="Test_10Service" targetNamespace="http://webservice.mks.com/10/Test"
    xmlns:tns="http://webservice.mks.com/10/Test"
    xmlns:ns3="http://www.w3.org/2005/05/xmlmime"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:ns1="http://webservice.mks.com/10/Test/schema"
    xmlns:ns2="http://webservice.mks.com/10/Test/fault"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

    <types>
        <xs:schema elementFormDefault="qualified" targetNamespace="http://webservice.mks.com/10/Test/schema" version="1.0">
            <xs:import namespace="http://www.w3.org/2005/05/xmlmime" schemaLocation="xmlmime.xsd" />
            <xs:simpleType name="ItemIdReference">
                <xs:annotation>
                    <xs:documentation>A reference to an item in the repository.</xs:documentation>
                </xs:annotation>
                <xs:restriction base="xs:integer" />
            </xs:simpleType>
            <xs:complexType name="Credentials">
                <xs:sequence>
                    <xs:element name="Username" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:schema>
    </types>
    <message name="viewUntestedRequest">
        <part name="parameters" type="tns:viewUntestedParameters"/>
    </message>
    <message name="viewUntestedResponse">
        <part name="parameters" type="tns:viewUntestedResponse"/>
    </message>
    <message name="fault">
        <part name="parameters" type="tns:MKSEException"/>
    </message>
    <portType name="Test_10PortType">
        <operation name="viewUntested">
            <input message="viewUntestedRequest"/>
            <output message="viewUntestedResponse"/>
        </operation>
    </portType>
    <binding name="Test_10Binding" type="tns:Test_10PortType">
        <operation name="viewUntested">
            <input message="viewUntestedRequest"/>
            <output message="viewUntestedResponse"/>
        </operation>
    </binding>
    <service name="Test_10Service">
        <port name="Test_10Port" binding="Test_10Binding"/>
    </service>
</definitions>
```

```

<xs:element name="Password" type="xs:string" />
<xs:element minOccurs="0" name="ImpersonatedUser" type="xs:string">
    <xs:annotation>
        <xs:documentation>Specify an impersonated user if you wish to do this operation on behalf of a particular user, with that person's permissions. In this case the username must be a user to whom impersonation privileges have been granted.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="DateFormat" type="xs:string">
    <xs:annotation>
        <xs:documentation>The formatting syntax for date fields without a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:  

http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="DateTimeFormat" type="xs:string">
    <xs:annotation>
        <xs:documentation>The formatting syntax for timestamp fields - that is, dates having a time component. The format may be specified in the dateFormat entity in the request, following the conventions outlined at this url:  

http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
<xs:attribute name="transactionId" type="xs:string">
    <xs:annotation>
        <xs:documentation>An optional external identifier that can be passed in to Integrity. It is passed back verbatim with the response.</xs:documentation>
    </xs:annotation>
</xs:attribute>
</xs:complexType>
<xs:complexType name="StatusResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string">
        <xs:annotation>
            <xs:documentation>The transaction ID that was passed in with the original request.</xs:documentation>
        </xs:annotation>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="SimpleItemFieldType">
    <xs:choice minOccurs="0">
        <xs:element name="longtext" type="tns:NullableString">
            <xs:annotation>
                <xs:documentation>Content to be used when setting this field. Note that the content is recorded literally - that is, HTML tags are not interpreted if you use this element. If you need to embed HTML tags in a rich longtext field, use the richcontent element.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="shorttext" type="tns:NullableShortString" />
        <xs:element name="richcontent" type="tns:NullableString">
            <xs:annotation>
                <xs:documentation>HTML content to be used when setting this field. Use CDATA blocks to escape the angle braces. Please refer to the Integrity User Guide for a</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:choice>
</xs:complexType>

```

list of HTML tags supported by Integrity.<xs:documentation>

```

</xs:annotation>
</xs:element>
<xs:element name="user" type="tns:NullableString" />
<xs:element name="multiuser">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="user"
type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="group" type="tns:NullableString" />
<xs:element name="multigroup">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="group"
type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="state" type="xs:string" />
<xs:element name="type" type="xs:string">
    <xs:annotation>
        <xs:documentation>The type of item. Note this field setting is used only in
method responses. Do not use it for create or edit operations. For createltem
there is an explicit Type attribute.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="project" type="tns:NullableString">
    <xs:annotation>
        <xs:documentation>The project in which the item is found</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="double" type="tns:NullableDouble" />
<xs:element name="boolean" type="tns:NullableBoolean" />
<xs:element name="integer" type="tns:NullableInteger" />
<xs:element name="date" type="tns:NullableString">
    <xs:annotation>
        <xs:documentation>The value for the date field - that is, a date having no time
component. The format may be specified in the dateTimeFormat entity in the
request, following the conventions outlined at this url:
http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html If the format
is not specified in the request, the default is "yyyy-MM-dd". Be careful to not
send a date when a timestamp is required! For GetItem and Createltem, if the
Integrity field is a timestamp but this field is used to send the information, the
end user will get an error message that recommends the wrong formatting. This
error message comes from the Integrity API and should be ignored since it is a
symptom of the problem, not the root cause.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="dateTime" type="tns:NullableString">
    <xs:annotation>
        <xs:documentation>The value for the timestamp field - that is, a date having a
time component. The format may be specified in the dateTimeFormat entity in the
request, following the conventions outlined at this url:
http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html If the format
is not specified in the request, the default is "yyyy-MM-dd HH:mm:ss". Be careful
to not send a timestamp when a date is required! For GetItem and Createltem, if

```

the Integrity field is a date but this field is used to send the information, the end user will get an error message that recommends the wrong formatting. This error message comes from the Integrity API and should be ignored since it is a symptom of the problem, not the root cause.<xs:documentation>

```
</xs:annotation>
</xs:element>
</xs:choice>
<xs:attribute name="Name" type="xs:string" use="required">
    <xs:annotation>
        <xs:documentation>The field name</xs:documentation>
    </xs:annotation>
</xs:attribute>
</xs:complexType>
<xs:complexType name="NullableShortString">
    <xs:choice>
        <xs:element name="value" type="xs:string" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableString">
    <xs:choice>
        <xs:element name="value" type="xs:string" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableFloat">
    <xs:choice>
        <xs:element name="value" type="xs:float" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableDouble">
    <xs:choice>
        <xs:element name="value" type="xs:double" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableBoolean">
    <xs:choice>
        <xs:element name="value" type="xs:boolean" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="NullableInteger">
    <xs:choice>
        <xs:element name="value" type="xs:int" />
        <xs:element name="null" type="xs:boolean" />
    </xs:choice>
</xs:complexType>
<xs:complexType name="Relationship">
    <xs:annotation>
        <xs:documentation>For each related item, its ID and relationship flags.</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="xs:integer">
```

```

            <xs:attribute name="flag" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="RelatedItemType">
    <xs:choice>
        <xs:element maxOccurs="unbounded" name="Item" type="tns:Relationship" />
        <xs:element maxOccurs="unbounded" name="IBPL" type="tns:ItemIdReference" />
    </xs:choice>
</xs:complexType>
<xs:element name="GetItemField" type="tns:GetItemFieldType" />
<xs:complexType name="GetItemFieldType">
    <xs:complexContent>
        <xs:extension base="tns:SimpleItemFieldType">
            <xs:choice minOccurs="0">
                <xs:element name="Relationship" type="tns:RelatedItemType" />
                <xs:element name="Attachment" type="tns:AttachmentInfo" />
                <xs:element name="SIProject" type="tns:SIProjectType" />
                <xs:element name="List" type="tns>ListType" />
                <xs:element name="PickList" type="tns:PickListType" />
                <xs:element name="Phase" type="tns:PhaseType" />
            </xs:choice>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="PickListType">
    <xs:annotation>
        <xs:documentation>A pick list reported from Integrity.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="MultiValued" type="xs:boolean" />
        <xs:element maxOccurs="unbounded" minOccurs="0" name="Value" type="tns:PickListValue" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PickListValue">
    <xs:annotation>
        <xs:documentation>A pick list value.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Label" type="xs:string" />
        <xs:element name="Position" type="xs:int" />
        <xs:element name="Value" type="xs:int" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PhaseType">
    <xs:annotation>
        <xs:documentation>A phase value.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Label" type="xs:string" />
        <xs:element name="Position" type="xs:int" />
        <xs:element name="Value" type="tns:PhaseValueList" />
    </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="PhaseValueList">
    <xs:annotation>
        <xs:documentation>The list of states for a given phase.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="State" type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ListType">
    <xs:annotation>
        <xs:documentation>A list reported from Integrity. The field type may be User or Group.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Element" type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="SIProjectType">
    <xs:sequence>
        <xs:element name="Server" type="xs:string">
            <xs:annotation>
                <xs:documentation>The hostname and port for the configuration management server, in the format <hostname>:<port>.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Name" type="xs:string">
            <xs:annotation>
                <xs:documentation>The name of the configuration management project.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="Revision" type="xs:string">
            <xs:annotation>
                <xs:documentation>The revision number of the configuration management project. If not specified, then the tip of the tree is assumed.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="DevelopmentPath" type="xs:string">
            <xs:annotation>
                <xs:documentation>The name of the development path for the configuration management project. If not specified, then headrev is assumed.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="null" type="xs:boolean" />
</xs:complexType>
<xs:complexType name="AttachmentInfo">
    <xs:annotation>
        <xs:documentation>Basic information about the attachment.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element name="Name" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The attachment name</xs:documentation>

```

```

        </xs:annotation>
    </xs:element>
    <xs:element name="Size" type="xs:int">
        <xs:annotation>
            <xs:documentation>The size of the attachment, in bytes</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Checksum" type="xs:string">
        <xs:annotation>
            <xs:documentation>The checksum value for the attachment.</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="GetTestCasesType">
    <xs:annotation>
        <xs:documentation>Note that this command only allows information to be requested for one session at a time.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element name="SessionID" type="tns:ItemIdReference" />
                <xs:element maxOccurs="unbounded" minOccurs="0" name="InputField" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>Comma-separated list of fields to include in the output.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="SubstituteParameters" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>Default is false. If true, then the values of parameters embedded in the fields are substituted in the field return value.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Filter" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>A condition that filters the results. For example, (field[ID] = 610) reports only on item 610.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ItemType">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" ref="tns:GetItemField" />
    </xs:sequence>
    <xs:attribute form="qualified" name="ItemId" type="tns:ItemIdReference" />
</xs:complexType>

```

```

<xs:complexType name="GetTestCasesResponseType">
  <xs:complexContent>
    <xs:extension base="tns>StatusResponse">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="Item"
                    type="tns:ItemType" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SessionResultsSelectionType">
  <xs:annotation>
    <xs:documentation>Get results for a single session. Note that you can't request multiple arbitrary combinations of test sessions and test cases. You may request multiple test cases for a single session.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ID" type="tns:ItemIdReference">
      <xs:annotation>
        <xs:documentation>Get results for this test session.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:choice minOccurs="0">
      <xs:annotation>
        <xs:documentation>If neither of these options is specified, then results for all test cases and all test suites are included in the output.</xs:documentation>
      </xs:annotation>
      <xs:element name="SuiteID" type="tns:ItemIdReference">
        <xs:annotation>
          <xs:documentation>Include results only from this test suite.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element maxOccurs="unbounded" name="TestCaseID" type="tns:ItemIdReference">
        <xs:annotation>
          <xs:documentation>Include results only from these test cases.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ResultsSelectionType">
  <xs:choice>
    <xs:element name="TestCase" type="tns:ItemIdReference">
      <xs:annotation>
        <xs:documentation>Get results for a single test case, including results for test cases in group documents that share content with this test case.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Session" type="tns:SessionResultsSelectionType" />
    <xs:element name="RelatedToltem" type="tns:ItemIdReference">
      <xs:annotation>
        <xs:documentation>Get results that are related to this item.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:choice>
</xs:complexType>

```

```

</xs:choice>
</xs:complexType>
<xs:complexType name="GetResultsType">
  <xs:complexContent>
    <xs:extension base="tns:Credentials">
      <xs:sequence>
        <xs:element name="Selections" type="tns:ResultsSelectionType" />
        <xs:element minOccurs="0" name="OnlyLastResult" type="xs:boolean">
          <xs:annotation>
            <xs:documentation>Include only the latest results. Default is to include all historical results.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="Verdict" type="xs:string">
          <xs:annotation>
            <xs:documentation>Include only verdicts having the specified value. Default is to include all verdicts.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetResultsResponseType">
  <xs:complexContent>
    <xs:extension base="tns>StatusResponse">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="Result" type="tns:ResultInfo" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ResultInfo">
  <xs:sequence>
    <xs:element name="ID" type="xs:string">
      <xs:annotation>
        <xs:documentation>The test result ID in the form <sessionID>:<testCaseID>.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SessionID" type="tns:ItemIdReference">
      <xs:annotation>
        <xs:documentation>The test session during which this result was collected.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TestCaseID" type="tns:ItemIdReference">
      <xs:annotation>
        <xs:documentation>The test case ID.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="SharesTestCaseID" type="tns:ItemIdReference">
      <xs:annotation>
        <xs:documentation>If the result was entered for a test case that shares content with this test case, and the results are shared to this test case, then the shared test case ID is that of the test case for which the result was</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

        entered.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="ModifiedDate" type="xs:string">
    <xs:annotation>
        <xs:documentation>Time stamp for the last modification, in the format specified in
        the dateFormat entity in the request, following the conventions outlined at this
        url: http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html If the
        format is not specified in the request, the default is "yyyy-MM-dd
        HH:mm:ss".</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="ModifiedBy" type="xs:string">
    <xs:annotation>
        <xs:documentation>The login ID of the user who entered this
        result.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="Verdict" type="xs:string">
    <xs:annotation>
        <xs:documentation>The verdict for this result. Could be Passed, Failed, Skipped
        or a custom verdict configured on the system. May also be missing if no verdict
        has yet to be set on the test case.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="VerdictType" type="tns:VerdictType">
    <xs:annotation>
        <xs:documentation>The class of verdict for this result. Not specified if there is no
        verdict.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="Note" type="xs:string">
    <xs:annotation>
        <xs:documentation>Annotation entered by the tester regarding this
        result.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="HasStepResult" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>Does this result include results for individual test
        steps?</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="HasAttachment" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>Does this result have an associated
        attachment?</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="HasRelatedItem" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>Has this result been associated with another item in the
        repository?</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
<xssimpleType name="VerdictType">

```

```

<xs:annotation>
    <xs:documentation>There are only three possible types of verdict.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
    <xs:enumeration value="Pass" />
    <xs:enumeration value="Fail" />
    <xs:enumeration value="Other" />
</xs:restriction>
</xs:simpleType>
<xs:complexType name="GetResultDetailsType">
    <xs:complexContent>
        <xs:extension base="tns:GetStepResultsType">
            <xs:sequence>
                <xs:element minOccurs="0" name="IncludeStepResults" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>Default is true. If true, then the results for test steps are included in the output.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetResultDetailsResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Result" type="tns:ResultDetails" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ResultDetails">
    <xs:complexContent>
        <xs:extension base="tns:ResultInfo">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Attachment" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The attachment name for each file attached to this result.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="RelatedItemID" type="tns:ItemIDReference">
                    <xs:annotation>
                        <xs:documentation>The ID of each item related to this result.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Step" type="tns:StepResult">
                    <xs:annotation>
                        <xs:documentation>The verdicts for test steps associated with this test result. If the test case has one or more test steps but some step verdicts were not entered for this result, then only those verdicts that were entered are included in the output. Thus it is

```

possible to have a test case that has multiple test steps yet none of the steps is included in the result for a particular session.</xs:documentation>

```

        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="StepVerdict">
  <xs:sequence>
    <xs:element name="ID" type="tns:ItemIdReference">
      <xs:annotation>
        <xs:documentation>The test step item ID.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Verdict" type="xs:string">
      <xs:annotation>
        <xs:documentation>The verdict for this step result. Could be Passed, Failed, Skipped or a custom verdict configured on the system.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="Note" type="xs:string">
      <xs:annotation>
        <xs:documentation>Annotation entered by the tester regarding this test step result.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="StepResult">
  <xs:complexContent>
    <xs:extension base="tns:StepVerdict">
      <xs:sequence>
        <xs:element name="TestStepID" type="xs:string">
          <xs:annotation>
            <xs:documentation>The test step ID in the format <sessionID>:<testCaseID>:<testStepID>.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="TestCaseList">
  <xs:sequence>
    <xs:element name="SessionID" type="tns:ItemIdReference">
      <xs:annotation>
        <xs:documentation>The test session.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="unbounded" name="TestCaseID" type="tns:ItemIdReference">
      <xs:annotation>
        <xs:documentation>One or more test cases.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>

```

```

</xs:complexType>
<xs:complexType name="GetStepResultsType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element name="TestCases" type="tns:TestCaseList" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetStepResultsResponseType">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="Result" type="tns:FullStepResult" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="FullStepResult">
    <xs:complexContent>
        <xs:extension base="tns:StepResult">
            <xs:sequence>
                <xs:element name="SessionID" type="tns:ItemIdReference">
                    <xs:annotation>
                        <xs:documentation>The test session for this step result.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="TestCaseID" type="tns:ItemIdReference">
                    <xs:annotation>
                        <xs:documentation>The test case for this step result.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="FetchAttachmentType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element name="Name" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The attachment name.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="ResultID" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>Get attachment for this particular result, in the format <sessionID><testCaseID>.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="FetchAttachmentResponseType">
        <xs:complexContent>
            <xs:extension base="tns:StatusResponseType">
                <xs:sequence>
                    <xs:element minOccurs="0" name="Attachment"
                        ns3:expectedContentTypes="application/octet-stream" type="xs:base64Binary" />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="GenericCreateResultsType">
        <xs:complexContent>
            <xs:extension base="tns:Credentials">
                <xs:sequence>
                    <xs:element name="SessionID" type="tns:ItemIdReference">
                        <xs:annotation>
                            <xs:documentation>The test session for this
                                result.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="Verdict" type="xs:string">
                        <xs:annotation>
                            <xs:documentation>Overall test result verdict. Could be Passed,
                                Failed, Skipped or a custom verdict configured on the
                                system.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element minOccurs="0" name="Overwrite" type="xs:boolean">
                        <xs:annotation>
                            <xs:documentation>Default true. If a result already exists, overwrite
                                it with the new values. This flag is ignored for edit
                                operations.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element minOccurs="0" name="Note" type="xs:string">
                        <xs:annotation>
                            <xs:documentation>Add an annotation to the
                                result.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="CreateResultsType">
        <xs:complexContent>
            <xs:extension base="tns:GenericCreateResultsType">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" name="TestCaseID"
                        type="tns:ItemIdReference">
                        <xs:annotation>
                            <xs:documentation>The test case(s) for this result. The same
                                information is attached to each test case in this
                                list.</xs:documentation>
                        </xs:annotation>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

```

```

<xs:element maxOccurs="unbounded" minOccurs="0" name="RelatedItemID"
type="tns:ItemIdReference">
    <xs:annotation>
        <xs:documentation>Add a related item to the
result.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" minOccurs="0" name="Attachment"
type="tns:Attachment">
    <xs:annotation>
        <xs:documentation>Add an annotation to the
result.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Attachment">
    <xs:sequence>
        <xs:element name="Name" type="xs:string">
            <xs:annotation>
                <xs:documentation>The attachment name.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="Description" type="xs:string">
            <xs:annotation>
                <xs:documentation>A description of the attachment.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="Contents" ns3:expectedContentTypes="application/octet-
stream" type="xs:base64Binary">
            <xs:annotation>
                <xs:documentation>The attachment contents.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CreateStepResultsType">
    <xs:complexContent>
        <xs:extension base="tns:GenericCreateResultsType">
            <xs:sequence>
                <xs:element name="TestCaseID" type="tns:ItemIdReference">
                    <xs:annotation>
                        <xs:documentation>The test case for this
result.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element maxOccurs="unbounded" name="TestStepID"
type="tns:ItemIdReference">
                    <xs:annotation>
                        <xs:documentation>The test step(s) to which the verdict is to be
assigned.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>

```

```

</xs:complexContent>
</xs:complexType>
<xs:complexType name="EditResultsType">
<xs:complexContent>
<xs:extension base="tns>CreateResultsType">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="RemoveRelatedItemID" type="tns:itemIdReference">
<xs:annotation>
<xs:documentation>Remove a related item from the result.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" minOccurs="0" name="RemoveAttachment" type="xs:string">
<xs:annotation>
<xs:documentation>Remove the attachment with the indicated name from the result.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="CreateIfNotExists" type="xs:boolean">
<xs:annotation>
<xs:documentation>Default true, meaning a result is created if one does not exist. If false and the result does not exist, then a new one is not created.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="EditStepResultsType">
<xs:complexContent>
<xs:extension base="tns>CreateStepResultsType">
<xs:sequence>
<xs:element minOccurs="0" name="CreateIfNotExists" type="xs:boolean">
<xs:annotation>
<xs:documentation>Default true, meaning a result is created if one does not exist. If true and the result does not yet exist, then a new one is not created.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="SetResultsType">
<xs:complexContent>
<xs:extension base="tns>Credentials">
<xs:sequence>
<xs:element name="SessionID" type="tns:itemIdReference">
<xs:annotation>
<xs:documentation>The session for which these results have been gathered.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="ActionDefinitionFile" ns3:expectedContentTypes="application/octet-stream" type="xs:base64Binary">

```

```

<xs:annotation>
    <xs:documentation>The file containing the
    instructions.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" minOccurs="0" name="Attachment"
type="tns:SetResultsAttachment">
    <xs:annotation>
        <xs:documentation>The list of attachments referenced from the
        action definition file.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="SetResultsAttachment">
    <xs:sequence>
        <xs:element name="Name" type="xs:string">
            <xs:annotation>
                <xs:documentation>The name of the file as referenced inside the action definition
                file. The file can be referenced from more than one AddAttachment entity. Note
                that the AddAttachment entity must not specify a directory offset to the
                file.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Contents" ns3:expectedContentTypes="application/octet-stream"
type="xs:base64Binary">
            <xs:annotation>
                <xs:documentation>The contents of the attachment.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="DeleteResultsType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element name="TestCases" type="tns:TestCaseList" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="DeleteStepResultsType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element name="SessionID" type="tns:ItemIdReference">
                    <xs:annotation>
                        <xs:documentation>The test session.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="TestCaseID" type="tns:ItemIdReference">
                    <xs:annotation>
                        <xs:documentation>Test affected test case.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

<xs:element maxOccurs="unbounded" name="TestStepID"
type="tns:ItemIdReference">
    <xs:annotation>
        <xs:documentation>One or more test steps.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="UntestedSelection">
    <xs:choice>
        <xs:element name="SessionID" type="tns:ItemIdReference">
            <xs:annotation>
                <xs:documentation>The ID of a test session. When this option is selected the command returns all test cases related to the session via the Tests relationship that were not tested in that session.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="UntestedByObjective">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="TestObjectiveID" type="tns:ItemIdReference">
                        <xs:annotation>
                            <xs:documentation>Any item (typically a test objective) for which the built-in Tests field is visible. The test cases listed in the output will include only those related via the Tests field (and/or by walking down the document hierarchy) will be included in the output.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="SessionID"
type="tns:ItemIdReference">
                        <xs:annotation>
                            <xs:documentation>Test sessions. If no session IDs are listed in the request document then all test cases related via the Tests relationship to the objective are included in the output. If at least one session is included in the request document then only those test cases that were not tested in any of the indicated sessions are included in the output.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:choice>
</xs:complexType>
<xs:complexType name="ViewUntestedType">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element name="Selection" type="tns:UntestedSelection" />
                <xs:element maxOccurs="unbounded" minOccurs="0" name="InputField"
type="xs:string">
                    <xs:annotation>
                        <xs:documentation>Comma-separated list of fields to include in the output. If none is specified then a minimal set of fields is included in the output.</xs:documentation>
                    </xs:annotation>

```

```

        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/10/Test/fault" version="1.0">
<xs:element name="MKSEception" nillable="true" type="tns:MKSFault" />
<xs:complexType name="MKSFault">
<xs:annotation>
<xs:documentation>The original fault generated from MKS Integrity. The element value is the full description of the error.</xs:documentation>
</xs:annotation>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="class" type="xs:string">
<xs:annotation>
<xs:documentation>The class of exception</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="id" type="xs:string">
<xs:annotation>
<xs:documentation>The unique MKS identifier for this error, generally a 6-digit positive integer.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="implication" type="xs:string">
<xs:annotation>
<xs:documentation>The implication from the exception</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/10/Test" version="1.0">
<xs:import namespace="http://webservice.mks.com/10/Test/schema" />
<xs:complexType name="getTestCasesRequest">
<xs:sequence>
<xs:element minOccurs="0" name="arg0" type="ns1:GetTestCasesType" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="getTestCasesResponse">
<xs:sequence>
<xs:element minOccurs="0" name="return" type="ns1:GetTestCasesResponseType" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="getResultsRequest">
<xs:sequence>
<xs:element minOccurs="0" name="arg0" type="ns1:GetResultsType" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="getResultsResponse">
<xs:sequence>
<xs:element minOccurs="0" name="return" type="ns1:GetResultsResponseType" />

```

```

        </xs:sequence>
    </xs:complexType>
<xs:complexType name="getResultDetailsRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetResultDetailsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getResultDetailsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetResultDetailsResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getStepResultsRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:GetStepResultsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getStepResultsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetStepResultsResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchAttachmentRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:FetchAttachmentType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchAttachmentResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:FetchAttachmentResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createResultsRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>CreateResultsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createResultsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createStepResultsRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>CreateStepResultsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createStepResultsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="editResultsRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>EditResultsType" />
    </xs:sequence>

```

```

        </xs:sequence>
    </xs:complexType>
<xs:complexType name="editResultsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="editStepResultsRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>EditStepResultsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="editStepResultsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="setResultsRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:SetResultsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="setResultsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="deleteResultsRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>DeleteResultsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="deleteResultsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="deleteStepResultsRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1>DeleteStepResultsType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="deleteStepResultsResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1>StatusResponse" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="viewUntestedRequest">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:ViewUntestedType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="viewUntestedResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:GetTestCasesResponseType" />
    </xs:sequence>

```

```

        </xs:sequence>
    </xs:complexType>
    <xs:element name="getTestCases" type="tns:getTestCasesRequest" />
    <xs:element name="getTestCasesResponse" type="tns:getTestCasesResponse" />
    <xs:element name="getResults" type="tns:getResultsRequest" />
    <xs:element name="getResultsResponse" type="tns:getResultsResponse" />
    <xs:element name="getResultDetails" type="tns:getResultDetailsRequest" />
    <xs:element name="getResultDetailsResponse" type="tns:getResultDetailsResponse" />
    <xs:element name="getStepResults" type="tns:getStepResultsRequest" />
    <xs:element name="getStepResultsResponse" type="tns:getStepResultsResponse" />
    <xs:element name="fetchAttachment" type="tns:fetchAttachmentRequest" />
    <xs:element name="fetchAttachmentResponse" type="tns:fetchAttachmentResponse" />
    <xs:element name="createResults" type="tns:createResultsRequest" />
    <xs:element name="createResultsResponse" type="tns:createResultsResponse" />
    <xs:element name="createStepResults" type="tns:createStepResultsRequest" />
    <xs:element name="createStepResultsResponse" type="tns:createStepResultsResponse" />
    <xs:element name="editResults" type="tns:editResultsRequest" />
    <xs:element name="editResultsResponse" type="tns:editResultsResponse" />
    <xs:element name="editStepResults" type="tns:editStepResultsRequest" />
    <xs:element name="editStepResultsResponse" type="tns:editStepResultsResponse" />
    <xs:element name="setResults" type="tns:setResultsRequest" />
    <xs:element name="setResultsResponse" type="tns:setResultsResponse" />
    <xs:element name="deleteResults" type="tns:deleteResultsRequest" />
    <xs:element name="deleteResultsResponse" type="tns:deleteResultsResponse" />
    <xs:element name="deleteStepResults" type="tns:deleteStepResultsRequest" />
    <xs:element name="deleteStepResultsResponse" type="tns:deleteStepResultsResponse" />
    <xs:element name="viewUntested" type="tns:viewUntestedRequest" />
    <xs:element name="viewUntestedResponse" type="tns:viewUntestedResponse" />
</xs:schema>
</types>
<message name="Test_10_getTestCasesResponse">
    <part element="tns:getTestCasesResponse" name="getTestCasesResponse" />
</message>
<message name="Test_10_getResults">
    <part element="tns:getResults" name="getResults" />
</message>
<message name="Test_10_fetchAttachment">
    <part element="tns:fetchAttachment" name="fetchAttachment" />
</message>
<message name="Test_10_editResults">
    <part element="tns:editResults" name="editResults" />
</message>
<message name="Test_10_getResultsResponse">
    <part element="tns:getResultsResponse" name="getResultsResponse" />
</message>
<message name="Test_10_getResultDetails">
    <part element="tns:getResultDetails" name="getResultDetails" />
</message>
<message name="Test_10_deleteStepResults">
    <part element="tns:deleteStepResults" name="deleteStepResults" />
</message>
<message name="Test_10_editResultsResponse">
    <part element="tns:editResultsResponse" name="editResultsResponse" />

```

```
</message>
<message name="Test_10_getResultDetailsResponse">
    <part element="tns:getResultDetailsResponse" name="getResultDetailsResponse" />
</message>
<message name="Test_10_fetchAttachmentResponse">
    <part element="tns:fetchAttachmentResponse" name="fetchAttachmentResponse" />
</message>
<message name="Test_10_setResults">
    <part element="tns:setResults" name="setResults" />
</message>
<message name="Test_10_deleteStepResultsResponse">
    <part element="tns:deleteStepResultsResponse" name="deleteStepResultsResponse" />
</message>
<message name="Test_10_deleteResults">
    <part element="tns:deleteResults" name="deleteResults" />
</message>
<message name="Test_10_viewUntestedResponse">
    <part element="tns:viewUntestedResponse" name="viewUntestedResponse" />
</message>
<message name="Test_10_createStepResults">
    <part element="tns:createStepResults" name="createStepResults" />
</message>
<message name="Test_10_deleteResultsResponse">
    <part element="tns:deleteResultsResponse" name="deleteResultsResponse" />
</message>
<message name="MKSEception">
    <part element="ns2:MKSEception" name="MKSEception" />
</message>
<message name="Test_10_createResultsResponse">
    <part element="tns:createResultsResponse" name="createResultsResponse" />
</message>
<message name="Test_10_getStepResultsResponse">
    <part element="tns:getStepResultsResponse" name="getStepResultsResponse" />
</message>
<message name="Test_10_getTestCases">
    <part element="tns:getTestCases" name="getTestCases" />
</message>
<message name="Test_10_editStepResultsResponse">
    <part element="tns:editStepResultsResponse" name="editStepResultsResponse" />
</message>
<message name="Test_10_setResultsResponse">
    <part element="tns:setResultsResponse" name="setResultsResponse" />
</message>
<message name="Test_10_createResults">
    <part element="tns:createResults" name="createResults" />
</message>
<message name="Test_10_createStepResultsResponse">
    <part element="tns:createStepResultsResponse" name="createStepResultsResponse" />
</message>
<message name="Test_10_editStepResults">
    <part element="tns:editStepResults" name="editStepResults" />
</message>
<message name="Test_10_viewUntested">
    <part element="tns:viewUntested" name="viewUntested" />
```

```

</message>
<message name="Test_10_getStepResults">
    <part element="tns:getStepResults" name="getStepResults" />
</message>
<portType name="Test_10">
    <operation name="getTestCases" parameterOrder="getTestCases">
        <input message="tns:Test_10_getTestCases" />
        <output message="tns:Test_10_getTestCasesResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="getResults" parameterOrder="getResults">
        <input message="tns:Test_10_getResults" />
        <output message="tns:Test_10_getResultsResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="getResultDetails" parameterOrder="getResultDetails">
        <input message="tns:Test_10_getResultDetails" />
        <output message="tns:Test_10_getResultDetailsResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="getStepResults" parameterOrder="getStepResults">
        <input message="tns:Test_10_getStepResults" />
        <output message="tns:Test_10_getStepResultsResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="fetchAttachment" parameterOrder="fetchAttachment">
        <input message="tns:Test_10_fetchAttachment" />
        <output message="tns:Test_10_fetchAttachmentResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="createResults" parameterOrder="createResults">
        <input message="tns:Test_10_createResults" />
        <output message="tns:Test_10_createResultsResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="createStepResults" parameterOrder="createStepResults">
        <input message="tns:Test_10_createStepResults" />
        <output message="tns:Test_10_createStepResultsResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="editResults" parameterOrder="editResults">
        <input message="tns:Test_10_editResults" />
        <output message="tns:Test_10_editResultsResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="editStepResults" parameterOrder="editStepResults">
        <input message="tns:Test_10_editStepResults" />
        <output message="tns:Test_10_editStepResultsResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
    <operation name="setResults" parameterOrder="setResults">
        <input message="tns:Test_10_setResults" />
        <output message="tns:Test_10_setResultsResponse" />
    </operation>

```

```

<fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="deleteResults" parameterOrder="deleteResults">
    <input message="tns:Test_10_deleteResults" />
    <output message="tns:Test_10_deleteResultsResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="deleteStepResults" parameterOrder="deleteStepResults">
    <input message="tns:Test_10_deleteStepResults" />
    <output message="tns:Test_10_deleteStepResultsResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
<operation name="viewUntested" parameterOrder="viewUntested">
    <input message="tns:Test_10_viewUntested" />
    <output message="tns:Test_10_viewUntestedResponse" />
    <fault message="tns:MKSEception" name="MKSEception" />
</operation>
</portType>
<binding name="Test_10Binding" type="tns:Test_10">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsp:PolicyReference URI="#wsmtom_policy" required="true"
        xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    />
    <operation name="getTestCases">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>
    <operation name="getResults">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>
    <operation name="getResultDetails">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>

```

```

</output>
<fault name="MKSEception">
    <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="getStepResults">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="fetchAttachment">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="createResults">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="createStepResults">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="editResults">

```

```

<soap:operation soapAction="" />
<input>
    <soap:body use="literal" />
</input>
<output>
    <soap:body use="literal" />
</output>
<fault name="MKSEception">
    <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="editStepResults">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="setResults">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="deleteResults">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
<operation name="deleteStepResults">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>

```

```

</output>
<fault name="MKSEception">
    <soap:fault name="MKSEception" use="literal" />
</fault>
</operation>
<operation name="viewUntested">
    <soap:operation soapAction="" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="MKSEception">
        <soap:fault name="MKSEception" use="literal" />
    </fault>
</operation>
</binding>
<service name="Test_10Service">
    <port binding="tns:Test_10Binding" name="Test_10Port">
        <soap:address location="http://IntegrityServer:9001/webservices/10/Test/" />
    </port>
</service>
<wsp:Policy wsu:id="wsmtom_policy">
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
>
    <wsoma:OptimizedMimeSerialization
        xmlns:wsoma="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization"
    />
</wsp:Policy>
</definitions>

```

Web Service: SourceDeprecated_2009_7Service

Web Service: SourceDeprecated_2009_7Service

Target Namespace: <http://webservice.mks.com/2009/7/SourceDeprecated>

Port **SourceDeprecated_2009_7Port** *Port type* [Source code](#)

Location: <http://IntegrityServer:9001/webservices/2009/7/SourceDeprecated/>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. **checkInMember** [Detail](#) [Source code](#)
2. **createMember** [Detail](#) [Source code](#)
3. **fetchMemberURL** [Detail](#) [Source code](#)
4. **setState** [Detail](#) [Source code](#)

Operations

Port type **SourceDeprecated_2009_7** [Source code](#)

1. **checkInMember**

[Source code](#)

Description: Uses the "projectci" MKS command. Checks in changes to a member of a project and optionally sets the state and label.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: SourceDeprecated_2009_7_checkInMember (soap:body, use = literal) [Source code](#)

checkInMember type [checkInMember](#)

Output: SourceDeprecated_2009_7_checkInMemberResponse (soap:body, use = literal) [Source code](#)

checkInMemberResponse type [checkInMemberResponse](#)

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type [MKSEException](#)

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o **class** type [string](#)
The class of exception
- o **id** type [string](#)
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o **implication** type [string](#)
The implication from the exception

2. createMember

[Source code](#)

Description: Uses the "projectadd" MKS command. Adds a member to a project and sets state and label.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: SourceDeprecated_2009_7_createMember (soap:body, use = literal) [Source code](#)

createMember type *createMember*

Output: SourceDeprecated_2009_7_createMemberResponse (soap:body, use = literal) [Source code](#)

createMemberResponse type *createMemberResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*

The class of exception

- o id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- o implication type *string*

The implication from the exception

3. fetchMemberURL

[Source code](#)

Description: Returns a URL that can be used to fetch the contents for the specified member.

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: SourceDeprecated_2009_7_fetchMemberURL (soap:body, use = literal) [Source code](#)

fetchMemberURL type *fetchMemberURL*

Output: SourceDeprecated_2009_7_fetchMemberURLResponse (soap:body, use = literal) [Source code](#)

fetchMemberURLResponse type *fetchMemberURLResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*

The class of exception

- o id type *string*

The unique MKS identifier for this error, generally a 6-digit positive integer.

- o implication type *string*

The implication from the exception

4. setState

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated

message.

Input: SourceDeprecated_2009_7_setState (soap:body, use = literal) [Source code](#)

setState type *setState*

Output: SourceDeprecated_2009_7_setStateResponse (soap:body, use = literal) [Source code](#)

setStateResponse type *setStateResponse*

Fault: MKSException (soap:fault, use = literal) [Source code](#)

MKSException type *MKSException*

The original fault generated from MKS Integrity. The element value is the full description of the error.

- o class type *string*
The class of exception
- o id type *string*
The unique MKS identifier for this error, generally a 6-digit positive integer.
- o implication type *string*
The implication from the exception

WSDL source code

```
<?xml version="1.0"?>
<definitions name="SourceDeprecated_2009_7Service"
targetNamespace="http://webservice.mks.com/2009/7/SourceDeprecated"
  xmlns:tns="http://webservice.mks.com/2009/7/SourceDeprecated"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://webservice.mks.com/2009/7/SourceDeprecated/schema"
  xmlns:ns2="http://webservice.mks.com/2009/7/SourceDeprecated/fault"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

>

<types>
  <xs:schema elementFormDefault="qualified"
    targetNamespace="http://webservice.mks.com/2009/7/SourceDeprecated/schema" version="1.0">
    <xs:complexType name="fetchMemberURLResponseType">
      <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
          <xs:sequence>
            <xs:element name="URL" type="xs:string">
              <xs:annotation>
                <xs:documentation>URL to retrieve the member contents.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="fetchMemberURLRequest">
      <xs:complexContent>
        <xs:extension base="tns:RevisionOperation">
```

```

<xs:sequence>
  <xs:element minOccurs="0" name="ClientIP"
  type="tns:NonEmptyString">
    <xs:annotation>
      <xs:documentation>The IP of the requesting
      client.</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="StateResponseType">
  <xs:complexContent>
    <xs:extension base="tns:MemberResponse">
      <xs:sequence>
        <xs:element name="State" type="xs:string">
          <xs:annotation>
            <xs:documentation>The new state for the
            member.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SetStateOperation">
  <xs:complexContent>
    <xs:extension base="tns:RevisionOperation">
      <xs:sequence>
        <xs:element name="State" type="tns:NonEmptyString">
          <xs:annotation>
            <xs:documentation>The requested revision
            state.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="LabelResponseType">
  <xs:complexContent>
    <xs:extension base="tns:MemberResponse">
      <xs:sequence>
        <xs:element name="Label" type="xs:string">
          <xs:annotation>
            <xs:documentation>The newly added label for the
            member.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

</xs:complexContent>
</xs:complexType>
<xs:complexType name="AddLabelOperation">
    <xs:complexContent>
        <xs:extension base="tns:RevisionOperation">
            <xs:sequence>
                <xs:element name="Label" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The label to be added.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="RevisionOperation">
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element minOccurs="0" name="Revision" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The requested revision. If unspecified, then member revision is assumed for the indicated development path. If no development path is specified, then the main line of development is assumed.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AugmentedCreateMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:CreateMemberRequest">
            <xs:sequence>
                <xs:element minOccurs="0" name="State" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The initial state to which the revision should be set.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Label" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The member label to be applied to the initial revision.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AugmentedCheckInMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:CheckInMemberRequest">
            <xs:sequence>
                <xs:element minOccurs="0" name="State"
type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The initial state to which the
revision should be set.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="Label"
type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The member label to be applied to
the initial revision.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:UpdateMemberRequest">
            <xs:sequence>
                <xs:element minOccurs="0" name="OnExistingArchive">
                    <xs:annotation>
                        <xs:documentation>What to do if the archive already
exists. Default is to cancel the
operation.</xs:documentation>
                    </xs:annotation>
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="cancel">
                            <xs:annotation>
                                <xs:documentation>Cancel the
operation. This is the
default.</xs:documentation>
                            </xs:annotation>
                        </xs:enumeration>
                        <xs:enumeration value="share">
                            <xs:annotation>
                                <xs:documentation>Reuse the
archive if it has been dropped from
the project, or share it if it is
currently being used in other
development
paths.</xs:documentation>
                            </xs:annotation>

```

```

</xs:enumeration>
<xs:enumeration value="new">
    <xs:annotation>
        <xs:documentation>Create a new archive instead of trying to reuse or share the old one.</xs:documentation>
    </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element minOccurs="0" name="Format">
    <xs:annotation>
        <xs:documentation>File format - binary or text. If unspecified the system will determine the format based on line length criteria specified by the administrator.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="text">
                <xs:annotation>
                    <xs:documentation>Text format, which uses delta storage in the repository.</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="binary">
                <xs:annotation>
                    <xs:documentation>Binary format, which records the entire file each time it is updated in the repository.</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="UpdateMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:MemberOperation">
            <xs:sequence>
                <xs:element name="File"
                    ns1:expectedContentTypes="application/octet-stream"
                    type="xs:base64Binary"
                    xmlns:ns1="http://www.w3.org/2005/05/xmlmime"
                />
                <xs:element minOccurs="0" name="Description">

```

```

        type="tns:NonEmptyString">
        <xs:annotation>
            <xs:documentation>The revision description for the
            newly created revision.</xs:documentation>
        </xs:annotation>
    </xs:element>
<xs:element minOccurs="0" name="ChangePackageID"
type="tns:NonEmptyString">
    <xs:annotation>
        <xs:documentation>The change package for this
        operation. Note this operation will fail if a change
        package is not specified and the policy for the project
        is to require change packages.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="CheckInMemberRequest">
    <xs:complexContent>
        <xs:extension base="tns:UpdateMemberRequest">
            <xs:sequence>
                <xs:element minOccurs="0" name="CheckInUnchanged"
type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>Check in the file even if it has not
                        been changed from the last
                        revision.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="ConvertArchiveOnGovernor"
type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>Convert the archive to a store-by-
                        reference format if the file exceeds the size
                        governor.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="MemberOperation">
    <xs:complexContent>
        <xs:extension base="tns:ProjectOperation">
            <xs:sequence>
                <xs:element name="Member" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The member name. If the member
                        is in a subdirectory of the project, then preface the
                        member like this:
                        /subdirectory1/subdirectory2/membername</xs:documentation>

```

```

        </xs:annotation>
    </xs:element>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectOperation">
    <xs:complexContent>
        <xs:extension base="tns:Credentials">
            <xs:sequence>
                <xs:element name="Project" type="tns:NonEmptyString">
                    <xs:annotation>
                        <xs:documentation>The project or subproject  
referenced by the service.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Credentials">
    <xs:sequence>
        <xs:element name="Username" type="tns:NonEmptyString" />
        <xs:element name="Password" type="xs:string" />
        <xs:element minOccurs="0" name="ImpersonatedUser" type="tns:NonEmptyString" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string">
        <xs:annotation>
            <xs:documentation>An optional external identifier that can be passed in to  
Integrity. It is passed back verbatim with the  
response.</xs:documentation>
        </xs:annotation>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="AugmentedCheckInMemberResponseType">
    <xs:complexContent>
        <xs:extension base="tns:MemberResponse">
            <xs:sequence>
                <xs:element name="StateChanged" type="xs:boolean" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AugmentedCreateMemberResponseType">
    <xs:complexContent>
        <xs:extension base="tns:MemberResponse">
            <xs:sequence>
                <xs:element name="StateChanged" type="xs:boolean" />
                <xs:element name="LabelAdded" type="xs:boolean" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="AppendRevisionDescriptionResponseType">
    <xs:complexContent>
        <xs:extension base="tns:MemberResponse">
            <xs:sequence>
                <xs:element name="Description" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="MemberResponse">
    <xs:complexContent>
        <xs:extension base="tns:ProjectResponse">
            <xs:sequence>
                <xs:element name="Member" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The member name.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Revision" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The affected revision.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ProjectResponse">
    <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
            <xs:sequence>
                <xs:element name="Project" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The project or subproject that is the subject of this operation.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="StatusResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="transactionId" type="xs:string" />

```

```

</xs:complexType>
<xs:simpleType name="NonEmptyString">
    <xs:restriction base="xs:string">
        <xs:minLength value="1" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/2009/7/SourceDeprecated/fault" version="1.0">
    <xs:element name="MKSEception" nillable="true" type="tns:MKSFault" />
    <xs:complexType name="MKSFault">
        <xs:annotation>
            <xs:documentation>The original fault generated from MKS Integrity. The element value is the full description of the error.</xs:documentation>
        </xs:annotation>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="class" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The class of exception</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:attribute name="id" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The unique MKS identifier for this error, generally a 6-digit positive integer.</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:attribute name="implication" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>The implication from the exception</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/2009/7/SourceDeprecated" version="1.0">
    <xs:import namespace="http://webservice.mks.com/2009/7/SourceDeprecated/schema" />
    <xs:element name="setState" type="tns:setState" />
    <xs:element name="setStateResponse" type="tns:setStateResponse" />
    <xs:element name="createMember" type="tns:createMember" />
    <xs:element name="createMemberResponse" type="tns:createMemberResponse" />
    <xs:element name="checkInMember" type="tns:checkInMember" />
    <xs:element name="checkInMemberResponse" type="tns:checkInMemberResponse" />
    <xs:element name="fetchMemberURL" type="tns:fetchMemberURL" />
    <xs:element name="fetchMemberURLResponse" type="tns:fetchMemberURLResponse" />
    <xs:complexType name="setState">

```

```

<xs:sequence>
    <xs:element minOccurs="0" name="arg0" type="ns1:SetStateOperation" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="setStateResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="ns1:StateResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0"
            type="ns1:AugmentedCreateMemberRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="createMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return"
            type="ns1:AugmentedCreateMemberResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkInMember">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0"
            type="ns1:AugmentedCheckInMemberRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="checkInMemberResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return"
            type="ns1:AugmentedCheckInMemberResponseType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchMemberURL">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="ns1:fetchMemberURLRequest" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="fetchMemberURLResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return"
            type="ns1:fetchMemberURLResponseType" />
    </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
<message name="MKSEception">
    <part element="ns2:MKSEception" name="MKSEception" />
</message>
<message name="SourceDeprecated_2009_7_fetchMemberURLResponse">
    <part element="tns:fetchMemberURLResponse" name="fetchMemberURLResponse" />

```

```

</message>
<message name="SourceDeprecated_2009_7_setState">
    <part element="tns:setState" name="setState" />
</message>
<message name="SourceDeprecated_2009_7_createMember">
    <part element="tns:createMember" name="createMember" />
</message>
<message name="SourceDeprecated_2009_7_checkInMemberResponse">
    <part element="tns:checkInMemberResponse" name="checkInMemberResponse" />
</message>
<message name="SourceDeprecated_2009_7_createMemberResponse">
    <part element="tns:createMemberResponse" name="createMemberResponse" />
</message>
<message name="SourceDeprecated_2009_7_checkInMember">
    <part element="tns:checkInMember" name="checkInMember" />
</message>
<message name="SourceDeprecated_2009_7_fetchMemberURL">
    <part element="tns:fetchMemberURL" name="fetchMemberURL" />
</message>
<message name="SourceDeprecated_2009_7_setStateResponse">
    <part element="tns:setStateResponse" name="setStateResponse" />
</message>
<portType name="SourceDeprecated_2009_7">
    <operation name="setState" parameterOrder="setState">
        <input message="tns:SourceDeprecated_2009_7_setState" />
        <output message="tns:SourceDeprecated_2009_7_setStateResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="createMember" parameterOrder="createMember">
        <documentation>Uses the "projectadd" MKS command. Adds a member to a project and sets state and label.</documentation>
        <input message="tns:SourceDeprecated_2009_7_createMember" />
        <output message="tns:SourceDeprecated_2009_7_createMemberResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="checkInMember" parameterOrder="checkInMember">
        <documentation>Uses the "projectci" MKS command. Checks in changes to a member of a project and optionally sets the state and label.</documentation>
        <input message="tns:SourceDeprecated_2009_7_checkInMember" />
        <output message="tns:SourceDeprecated_2009_7_checkInMemberResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>
    <operation name="fetchMemberURL" parameterOrder="fetchMemberURL">
        <documentation>Returns a URL that can be used to fetch the contents for the specified member.</documentation>
        <input message="tns:SourceDeprecated_2009_7_fetchMemberURL" />
        <output message="tns:SourceDeprecated_2009_7_fetchMemberURLResponse" />
        <fault message="tns:MKSEException" name="MKSEException" />
    </operation>

```

```
</portType>
<binding name="SourceDeprecated_2009_7Binding" type="tns:SourceDeprecated_2009_7">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="setState">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>
    <operation name="createMember">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>
    <operation name="checkInMember">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>
    <operation name="fetchMemberURL">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>

```

```
</fault>
</operation>
</binding>
<service name="SourceDeprecated_2009_7Service">
    <port binding="tns:SourceDeprecated_2009_7Binding" name="SourceDeprecated_2009_7Port">
        <soap:address location="http://IntegrityServer:9001/webservices/2009/7/SourceDeprecated/" />
    </port>
</service>
</definitions>
```

Web Service: Generic_2009Service

Web Service: Generic_2009Service

Target Namespace: <http://webservice.mks.com/2009/Generic>

Port **Generic_2009Port** [Port type](#) [Source code](#)

Location: <http://IntegrityServer:9001/webservices/2009/Generic/>

Protocol: SOAP

Default style: document

Transport protocol: SOAP over HTTP

Operations:

1. **command** [Detail](#) [Source code](#)

Operations

Port type **Generic_2009** [Source code](#)

1. **command**

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

Input: Generic_2009_command (soap:body, use = literal) [Source code](#)

command type *command*

Output: Generic_2009_commandResponse (soap:body, use = literal) [Source code](#)

commandResponse type *commandResponse*

Fault: MKSEException (soap:fault, use = literal) [Source code](#)

MKSEException type *MKSEException*

- class type *string*
- id type *string*
- implication type *string*

WSDL source code

```
<?xml version="1.0"?>
<definitions name="Generic_2009Service" targetNamespace="http://webservice.mks.com/2009/Generic"
    xmlns:ns1="http://webservice.mks.com/2009/Generic/schema"
    xmlns:ns2="http://webservice.mks.com/2009/Generic/fault"
    xmlns:tns="http://webservice.mks.com/2009/Generic"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
```

```

xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

>
<types>
  <xs:schema elementFormDefault="qualified"
  targetNamespace="http://webservice.mks.com/2009/Generic/schema" version="1.0">
    <xs:complexType name="GenericRequest">
      <xs:complexContent>
        <xs:extension base="tns:Credentials">
          <xs:sequence>
            <xs:element name="Application" type="tns:application" />
            <xs:element name="Command" type="xs:string" />
            <xs:element maxOccurs="unbounded" minOccurs="0"
            name="Option" type="tns:Option" />
            <xs:element maxOccurs="unbounded" minOccurs="0"
            name="Selection" type="xs:string" />
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="Credentials">
      <xs:sequence>
        <xs:element name="Username" type="xs:string" />
        <xs:element name="Password" type="xs:string" />
        <xs:element minOccurs="0" name="ImpersonatedUser" type="xs:string" />
        <xs:element minOccurs="0" name="DateTimeFormat" type="tns:DateTimeFormat"
        />
      </xs:sequence>
      <xs:attribute name="TransactionId" type="xs:string" />
    </xs:complexType>
    <xs:complexType name="Option">
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="Name" type="xs:string" />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="DateTimeFormat">
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="Country" type="xs:string" />
          <xs:attribute name="Language" type="xs:string" />
          <xs:attribute name="Variant" type="xs:string" />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="GenericResponse">
      <xs:complexContent>
        <xs:extension base="tns>StatusResponse">
          <xs:sequence>
            <xs:element name="Application" type="tns:application" />

```

```

<xs:element name="Command" type="xs:string" />
<xs:element minOccurs="0" name="Result" type="tns:Result" />
<xs:element maxOccurs="unbounded" minOccurs="0"
name="WorkItems" type="tns:Item" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="StatusResponse">
<xs:sequence>
<xs:element minOccurs="0" name="return" type="xs:string" />
</xs:sequence>
<xs:attribute name="TransactionId" type="xs:string" />
</xs:complexType>
<xs:complexType name="Result">
<xs:sequence>
<xs:element name="Message" type="xs:string" />
<xs:element name="PrimaryValue" type="tns:Item" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="Item">
<xs:sequence>
<xs:element name="Id" type="xs:string" />
<xs:element name="ModelType" type="xs:string" />
<xs:element minOccurs="0" name="Context" type="xs:string" />
<xs:element minOccurs="0" name="DisplayId" type="xs:string" />
<xs:element maxOccurs="unbounded" minOccurs="0" name="Field"
type="tns:Field" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="Field">
<xs:sequence>
<xs:element name="Name" type="xs:string" />
<xs:element name="Value" type="tns:Value" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="Value">
<xs:sequence>
<xs:element minOccurs="0" name="int" type="xs:integer" />
<xs:element minOccurs="0" name="long" type="xs:long" />
<xs:element minOccurs="0" name="string" type="xs:string" />
<xs:element minOccurs="0" name="boolean" type="xs:boolean" />
<xs:element minOccurs="0" name="float" type="xs:float" />
<xs:element minOccurs="0" name="double" type="xs:double" />
<xs:element minOccurs="0" name="datetime" type="xs:anySimpleType" />
<xs:element minOccurs="0" name="Item" type="tns:Item" />
<xs:element maxOccurs="unbounded" minOccurs="0" name="ItemList"
type="tns:Item" />
<xs:element minOccurs="0" name="ValueList" type="tns:ValueList" />

```

```

</xs:sequence>
</xs:complexType>
<xs:complexType name="ValueList">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="int"
            type="xs:integer" />
        <xs:element maxOccurs="unbounded" minOccurs="0" name="string"
            nillable="true" type="xs:string" />
        <xs:element maxOccurs="unbounded" minOccurs="0" name="boolean"
            type="xs:boolean" />
        <xs:element maxOccurs="unbounded" minOccurs="0" name="float" type="xs:float"
            />
        <xs:element maxOccurs="unbounded" minOccurs="0" name="datetime"
            nillable="true" type="xs:anySimpleType" />
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="application">
    <xs:restriction base="xs:string">
        <xs:enumeration value="tm" />
        <xs:enumeration value="sd" />
        <xs:enumeration value="rq" />
        <xs:enumeration value="aa" />
        <xs:enumeration value="integrity" />
        <xs:enumeration value="si" />
        <xs:enumeration value="im" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/2009/Generic" version="1.0">
    <xs:import namespace="http://webservice.mks.com/2009/Generic/schema" />
    <xs:element name="command" type="tns:command" />
    <xs:element name="commandResponse" type="tns:commandResponse" />
    <xs:complexType name="command">
        <xs:sequence>
            <xs:element minOccurs="0" name="arg0" type="ns1:GenericRequest" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="commandResponse">
        <xs:sequence>
            <xs:element minOccurs="0" name="return" type="ns1:GenericResponse" />
        </xs:sequence>
    </xs:complexType>
</xs:schema>
<xs:schema targetNamespace="http://webservice.mks.com/2009/Generic/fault" version="1.0">
    <xs:element name="MKSException" nillable="true" type="tns:MKSFault" />
    <xs:complexType name="MKSFault">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="class" type="xs:string" />
                <xs:attribute name="id" type="xs:string" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>

```

```

                <xs:attribute name="implication" type="xs:string" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
</types>
<message name="MKSEception">
    <part element="ns2:MKSEception" name="MKSEception" />
</message>
<message name="Generic_2009_command">
    <part element="tns:command" name="command" />
</message>
<message name="Generic_2009_commandResponse">
    <part element="tns:commandResponse" name="commandResponse" />
</message>
<portType name="Generic_2009">
    <operation name="command" parameterOrder="command">
        <input message="tns:Generic_2009_command" />
        <output message="tns:Generic_2009_commandResponse" />
        <fault message="tns:MKSEception" name="MKSEception" />
    </operation>
</portType>
<binding name="Generic_2009Binding" type="tns:Generic_2009">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="command">
        <soap:operation soapAction="" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
        <fault name="MKSEception">
            <soap:fault name="MKSEception" use="literal" />
        </fault>
    </operation>
</binding>
<service name="Generic_2009Service">
    <port binding="tns:Generic_2009Binding" name="Generic_2009Port">
        <soap:address location="http://IntegrityServer:9001/webservices/2009/Generic/" />
    </port>
</service>
</definitions>

```