GFD-R-P.187
OGSA-DMI WG

Michel Drescher, Fujitsu
Steven Newhouse, Microsoft
Mario Antonioletti, EPCC
August 1, 2011

OGSA-DMI Plain Web Service Rendering Specification 1.0

Status of This Document

This document specifies the normative renderings as plain Web Services for the two port types specified in the OGSA-DMI Functional Specification 1.0. Distribution is unlimited.

Copyright Notice

Trademark

Abstract

The Open Grid Services Architecture Data Movement Interface (OGSA-DMI) specification defines a standardized mechanism for moving data from a source to a destination. By abstracting this data transfer, the client complexity for moving data within a grid is greatly reduced. OGSA-DMI defines two port types for initiating, scheduling and managing data transfers from a given source to a specified destination. The source and destination are described through Data End Point References (DEPRs), a specialized form of a WS-Address.

The Open Grid Services Architecture Data Movement Interface Functional Specification 1.0 [ANTONIOLETTI] (hereafter called “the Functional Specification”) defines the abstract methods and attributes of the Data Transfer Factory (DTF) and the Data Transfer Instance (DTI) port types.

This document (hereafter called “the Rendering”) defines the normative WSDL interface for a plain (non-WSRF-based) Web Service rendering of the Functional Specification.

Contents

1. Introduction

This Rendering document defines a WSDL rendering for plain web services that purely follow the WS-I Basic Profile for designing and declaring Web Services. Hence this specification complements the Functional Specification: The Functional Specification cannot be implemented without selecting a concrete WSDL rendering. If implementers choose to implement the functional specification following the WS-I Basic Profile only, they MUST use this specification.

1.1 XML Elements, XML Types and OGSA-DMI Namespaces

The Functional Specification normatively defines information model related entities, for example "Data EPR"s. In the abstract context of the Functional Specification, these information model entities are rendered in a specific, normative data model expressed in XML Schema, for example as a `dmi:DataEPR`. For each information model entity, the Functional Specification[ANTONIOLETTI] defines complex XML Schema data types and XML Schema elements using the base namespace `http://schemas.ogf.org/dmi/2007/05/dmi`.

There are many different XML Schema and Web Service toolkits available that may aid developers to implement OGSA-DMI. However, all these toolkits follow different strategies and design patterns for the way they use XML Schema. It is for these different toolkits that this rendering specification defines some XML elements in this specification's namespace that have an XML type derived from the Functional Specification's namespace, effectively promoting and underpinning interoperability between different implementations of this rendering specification.

For example, the Functional Specification defines an information model entity [transfer requirements], which it renders into an XML complex type `dmi:TransferRequirementsType` and an XML Element `dmi:TransferRequirements` of that type. The Rendering, while normatively defining the structure of message exchanges, also defines an XML Element, `dmi-plain:TransferRequirements` which is of type `dmi:TransferRequirementsType` to ensure implementation interoperability.

2. Notational Conventions

The key words 'MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" are to be interpreted as described in RFC 2119 [BRADNER1].

When describing abstract data models, this specification uses the notational convention used by the XML Infoset [XML Infoset]. Specifically, abstract property names always appear in square brackets (e.g., [some property]).

When describing concrete XML schemas [XML Schema Part 1, Part 2], the Rendering specification uses the notational convention of WS-Security [WS-Security]. Specifically, each member of an element's [children] or [attributes] property is described using an XPath-like notation (e.g., /x:MyHeader/x:SomeProperty/@value1). The use of {any} indicates the presence of an element wildcard (<xs:any/>). The use of @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).

2.1 XML Namespaces

The Rendering uses a number of namespace prefixes throughout; they are listed in Table 1. Note that the choice of any namespace prefix is arbitrary and not semantically significant (see

[BRAY]).

Prefix	Namespace
s11	http://schemas.xmlsoap.org/soap/envelope
wsa	http://www.w3.org/2005/03/addressing
xs	http://www.w3.org/2001/XMLSchema
dmi	http://schemas.ogf.org/dmi/2007/05/dmi
dmi-plain	http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain
ws-sec	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd

Table 1: Namespaces and prefixes used in this document

3. Data Transfer Factory Port Type

This section normatively defines the message exchanges that standardize the means by which a client can communicate with a DTF as defined in the Functional Specification[ANTONIOLETTI].

3.1 GetDataTransferInstance

A DTF implementation MUST support the message exchange defined in this section. This allows the requestor to set up a data transfer with a specified set of parameters.

The format of the request message MUST follow:

```
<dmi-plain:GetDataTransferInstanceRequestMessage>
  <dmi-plain:SourceDEPR>
    wsa:EndpointReferenceType
  </dmi-plain:SourceDEPR>
  <dmi-plain:SinkDEPR>
    wsa:EndpointReferenceType
  </dmi-plain:SinkDEPR>
  <dmi-plain:TransferRequirements>
    dmi:TransferRequirementsType
  </dmi-plain:TransferRequirements>
</dmi-plain:GetDataTransferInstanceRequestMessage>
```

The wsa:Action for the request MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransferFactory/GetDataTransferInstanceRequest
```

Where:

/dmi-plain:GetDataTransferInstanceRequestMessage
defines the XML format of the GetDataTransferInstance request as described in the Functional Specification section 4.2.1.

/dmi-plain:GetDataTransferInstanceRequestMessage/dmi-plain:SourceDEPR
/dmi-plain:GetDataTransferInstanceRequestMessage/dmi-plain:SinkDEPR
describe the Data EPRs for the data at the source location and sink location, respectively, as described in the Functional Specification section 4.2.1.2.

/dmi-plain:GetDataTransferInstanceRequestMessage/dmi-plain:TransferRequirements
describes additional transfer requirements as defined in the Functional Specification

section 4.2.3.

The format of a response to such request MUST follow:

```
<dmi-plain:GetDataTransferInstanceResponseMessage>
  <dmi-plain:ServiceInstance>
    wsa:EndpointReferenceType
  </dmi-plain:ServiceInstance>
</dmi-plain:GetDataTransferInstanceResponseMessage>
```

The wsa:Action for the response MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erFactory/GetDataTransferInstanceResponse
```

Where:

/dmi-plain:GetDataTransferInstanceResponseMessage

defines the XML format of the response to a GetDataTransferInstance request as described in the Functional Specification section 4.2.1.

/dmi-plain:GetDataTransferInstanceResponseMessage/dmi-plain:ServiceInstance

contains the Endpoint Reference to the created DTI, as defined in [ANTONIOLETTI] section 4.2.1.1.

3.1.1 Example SOAP encoding

The following is a non-normative example of a GetDataTransferInstance request message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erFactory/GetDataTransferInstanceRequest
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:GetDataTransferInstanceRequestMessage>
      <dmi-plain:SourceDEPR>
        <wsa:Address>
          http://www.ogf.org/ogsa/2007/08/addressing/none
        </wsa:Address>
        ...
        <wsa:Metadata>
          <dmi:DataLocations>
            <dmi:Data
              DataUrl="ftp://ftp.siteA.com/source/example.zip"
              ProtocolUri="http://www.ogf.org/ogsa-
dmi/2006/03/im/protocol/ftp">
              <dmi:Credentials>
                <ws-sec:UsernameToken>
                  <Username>foo</Username>
                  <Password>bar</Password>
                </ws-sec:UsernameToken>
              </dmi:Credentials>
            </dmi:Data>
          </dmi:DataLocations>
          ...
        </wsa:Metadata>
        ...
      </dmi-plain:SourceDEPR>
      <dmi-plain:SinkDEPR>
        <wsa:Address>
          http://www.ogf.org/ogsa/2007/08/addressing/none
        </wsa:Address>
        ...
        <wsa:Metadata>
          <dmi:DataLocations>
            <dmi:Data
              DataUrl="ftp://ftp.siteB.com/sink/example.zip"
              ProtocolUri="http://www.ogf.org/ogsa-
dmi/2006/03/im/protocol/ftp">
              <dmi:Credentials>
                <ws-sec:UsernameToken>
                  <Username>baz</Username>
                  <Password>boo</Password>
                </ws-sec:UsernameToken>
              </dmi:Credentials>
            </dmi:Data>
          </dmi:DataLocations>
          ...
        </wsa:Metadata>
      </dmi-plain:SinkDEPR>
      <dmi-plain:TransferRequirements />
    </dmi-plain:GetDataTransferInstanceRequestMessage>
  </s11:Body>
</s11:Envelope>
```

The following is a non-normative example of a GetDataTransferInstance response message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erFactory/GetDataTransferInstanceResponse
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:GetDataTransferInstanceResponseMessage>
      <dmi-plain:ServiceInstance>
        <wsa:Address> ... </wsa:Address>
        ...
      </dmi-plain:ServiceInstance>
    </dmi-plain:GetDataTransferInstanceResponseMessage>
  </s11:Body>
</s11:Envelope>
```

3.1.2 Faults

If a DTF does not respond with a GetDataTransferInstanceResponseMessage, then it MUST send a fault. The Functional Specification defines which faults a Web Service Endpoint MAY send and for which circumstances. The following sub sections define the message format for each permissible fault.

3.1.2.1 UnsatisfiableRequestOptionsFault

If the transfer requirements given in a GetDataTransferInstance request message are invalid or impossible to fulfill, then a DTF MUST send this fault.

The format of the UnsatisfiableRequestOptionsFault MUST follow:

```
<dmi-plain:UnsatisfiableRequestOptionsFault>
  <dmi-plain:Message> xs:string </dmi-plain:Message>
  <dmi-plain:Timestamp> xs:dateTime </dmi-plain:Timestamp>
  {any namespace=##other}*
</dmi-plain:UnsatisfiableRequestOptionsFault>
```

The wsa:Action for the fault MUST contain the URI:
`http://www.w3.org/2005/08/addressing/soap/fault`

Where:

/dmi-plain:UnsatisfiableRequestOptionsFault
defines the message structure for the UnsatisfiableRequestOptionsFault as described in [ANTONIOLETTI] section 4.3.1.

/dmi-plain:UnsatisfiableRequestOptionsFault/dmi-plain:Message
is a descriptive, user-readable text describing the exact fault conditions. Its type is `xs:string`.

/dmi-plain:UnsatisfiableRequestOptionsFault/dmi-plain:Timestamp
describes the point in time when the fault condition has been detected (as opposed to when the message is generated, or sent).

/dmi-plain:UnsatisfiableRequestOptionsFault/{any}
denotes an extensibility point where implementations MAY add additional structured information to the fault.

3.1.2.1.1 Example SOAP encoding

The following is a non-normative example of a UnsatisfiableRequestOptions fault message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://www.w3.org/2005/08/addressing/soap/fault
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
```

```
    <s11:Fault>
      <faultcode>s11:Server</faultcode>
      <faultstring>’EndNoLaterThan’ lies in the past!</faultstring>
      <detail>
        <dmi-plain:UnsatisfiableRequestOptionsFault>
          <dmi-plain:Message>
            ’EndNoLaterThan’ lies in the past!
          </dmi-plain:Message>
          <dmi-plain:Timestamp>
            2008-10-12T09:44:20.137+01:00
          </dmi-plain:Timestamp>
        </dmi-plain:UnsatisfiableRequestOptionsFault>
        ...
      </detail>
    </s11:Fault>
  </s11:Body>
</s11:Envelope>
```

3.1.2.2 NoSourceSinkProtocolMatchFault

If a DTF cannot resolve, or determine, a suitable underlying data transfer protocol when instantiating a DTI then it MUST send this fault.

The format of the NoSourceSinkProtocolMatchFault MUST follow:

```
<dmi-plain:NoSourceSinkProtocolMatchFault>
  <dmi-plain:Message> xs:string </dmi-plain:Message>
  <dmi-plain:Timestamp> xs:dateTime </dmi-plain:Timestamp>
  {any namespace=##other}*
</dmi-plain:NoSourceSinkProtocolMatchFault>
```

The wsa:Action for the fault MUST contain the URI:
`http://www.w3.org/2005/08/addressing/soap/fault`

Where:

/dmi-plain:NoSourceSinkProtocolMatchFault
defines the message structure for the NoSourceSinkProtocolMatchFault as described in [ANTONIOLETTI] section 4.3.2.

/dmi-plain:NoSourceSinkProtocolMatchFault/dmi-plain:Message
is a descriptive, user-readable text describing the exact fault conditions. Its type is `xs:string`.

/dmi-plain:NoSourceSinkProtocolMatchFault/dmi-plain:Timestamp
describes the point in time when the fault condition has been detected (as opposed to when the message is generated, or sent).

/dmi-plain:NoSourceSinkProtocolMatchFault/{any}
denotes an extensibility point where implementations MAY add additional structured information to the fault.

3.1.2.2.1 Example SOAP encoding

The following is a non-normative example of a NoSourceSinkProtocolMatchFault fault message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://www.w3.org/2005/08/addressing/soap/fault
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <s11:Fault>
      <faultcode>s11:Server</faultcode>
      <faultstring>No suitable protocol found.</faultstring>
      <detail>
        <dmi-plain:NoSourceSinkProtocolMatchFault>
          <dmi-plain:Message>
            No suitable protocol found.
          </dmi-plain:Message>
          <dmi-plain:Timestamp>
            2008-10-12T09:44:20.137+01:00
          </dmi-plain:Timestamp>
        </dmi-plain:NoSourceSinkProtocolMatchFault>
        ...
      </detail>
    </s11:Fault>
  </s11:Body>
</s11:Envelope>
```

3.1.2.3 NoDataLocationsSpecifiedInEprFault

If a DTF cannot resolve or does not find dmi:Data elements within either a SourceDEPR or a SinkDEPR, then it MUST send this fault.

The format of the NoDataLocationsSpecifiedInEprFault MUST follow:

```
<dmi-plain:NoDataLocationsSpecifiedInEprFault>
  <dmi-plain:Message> xs:string </dmi-plain:Message>
  <dmi-plain:Timestamp> xs:dateTime </dmi-plain:Timestamp>
  {any namespace=##other}*
</dmi-plain:NoDataLocationsSpecifiedInEprFault>
```

The wsa:Action for the fault MUST contain the URI:
`http://www.w3.org/2005/08/addressing/soap/fault`

Where:

/dmi-plain:NoDataLocationsSpecifiedInEprFault
defines the message structure for the NoDataLocationsSpecifiedInEprFault as described in [ANTONIOLETTI] section 4.3.3.

/dmi-plain:NoDataLocationsSpecifiedInEprFault/dmi-plain:Message
is a descriptive, user-readable text describing the exact fault conditions. Its type is `xs:string`.

/dmi-plain:NoDataLocationsSpecifiedInEprFault/dmi-plain:Timestamp
describes the point in time when the fault condition has been detected (as opposed to

when the message is generated, or sent).

/dmi-plain:NoDataLocationsSpecifiedInEprFault/{any}
denotes an extensibility point where implementations MAY add additional structured information to the fault.

3.1.2.3.1 Example SOAP encoding

The following is a non-normative example of a NoDataLocationsSpecifiedInEprFault fault message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://www.w3.org/2005/08/addressing/soap/fault
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <s11:Fault>
      <faultcode>s11:Server</faultcode>
      <faultstring>No data locations found in source DEPR</faultstring>
      <detail>
        <dmi-plain:NoDataLocationsSpecifiedInEprFault>
          <dmi-plain:Message>
            No data locations found in source DEPR
          </dmi-plain:Message>
          <dmi-plain:Timestamp>
            2008-10-12T09:44:20.137+01:00
          </dmi-plain:Timestamp>
        </dmi-plain:NoDataLocationsSpecifiedInEprFault>
        ...
      </detail>
    </s11:Fault>
  </s11:Body>
</s11:Envelope>
```

3.1.2.4 CustomFault

If a DTF encounters any other fault condition that is not captured in the faults defined above then it MUST send a CustomFault.

The format of the CustomFault MUST follow:

```
<dmi-plain:CustomFault>
  <dmi-plain:Message> xs:string </dmi-plain:Message>
  <dmi-plain:Timestamp> xs:dateTime </dmi-plain:Timestamp>
  {any namespace=##other}*
</dmi-plain:CustomFault>
```

The wsa:Action for the fault MUST contain the URI:

```
      http://www.w3.org/2005/08/addressing/soap/fault
```

Where:

/dmi-plain:CustomFault

defines the message structure for the CustomFault as described in [ANTONIOLETTI] section 4.3.4.

/dmi-plain:CustomFault/dmi-plain:Message

is a descriptive, user-readable text describing the exact fault conditions. Its type is `xs:string`.

/dmi-plain:CustomFault/dmi-plain:Timestamp

describes the point in time when the fault condition has been detected (as opposed to when the message is generated, or sent).

/dmi-plain:CustomFault/{any}

denotes an extensibility point where implementations MAY add additional structured information to the fault.

3.1.2.4.1 Example SOAP encoding

The following is a non-normative example of a CustomFault fault message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://www.w3.org/2005/08/addressing/soap/fault
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <s11:Fault>
      <faultcode>s11:Server</faultcode>
      <faultstring>
        Instantiation error for transfer protocol
      </faultstring>
      <detail>
        <dmi-plain:CustomFault>
          <dmi-plain:Message>
            Instantiation error for transfer protocol
          </dmi-plain:Message>
          <dmi-plain:Timestamp>
            2008-10-12T09:44:20.137+01:00
          </dmi-plain:Timestamp>
        </dmi-plain:CustomFault>
        ...
      </detail>
    </s11:Fault>
  </s11:Body>
</s11:Envelope>
```

3.2 GetFactoryAttributesDocument

A DTF implementation MUST support the message exchange defined in this section. This allows the requestor to retrieve the attribute values of the addressed DTF.

The format of the request message MUST be:

```
<dmi-plain:GetFactoryAttributesDocumentRequestMessage />
```

The wsa:Action for the request MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransferFactory/GetFactoryAttributesDocumentRequest
```

Where:

/dmi-plain:GetFactoryAttributesDocumentRequestMessage

defines the message structure for the GetFactoryAttributesDocument request message as described in [ANTONIOLETTI] section 4.2.2. The message MUST be empty.

The format of a response to such request MUST follow:

```
<dmi-plain:GetFactoryAttributesDocumentResponseMessage>
  <dmi-plain:FactoryAttributes>
    <dmi-plain:SupportedProtocol/>+
    {any namespace=##other}*
  </dmi-plain:FactoryAttributes>
</dmi-plain:GetFactoryAttributesDocumentResponseMessage>
```

The wsa:Action for the response MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransferFactory/GetFactoryAttributesDocumentResponse
```

Where:

/dmi-plain:GetFactoryAttributesDocumentResponseMessage

defines the message structure of the response to a GetFactoryAttributesDocument request message as described in [ANTONIOLETTI] section 4.2.2.

/dmi-plain:GetFactoryAttributesDocumentResponseMessage/dmi-plain:FactoryAttributes

defines the structure of the wrapper XML element containing all DTF attributes, as described in [ANTONIOLETTI] section 4.2.2.1.

/dmi-plain:GetFactoryAttributesDocumentResponseMessage/dmi-plain:FactoryAttributes/dmi:SupportedProtocol

describes a data transfer protocol the DTF supports, as described in [ANTONIOLETTI] section 4.1.1. For each supported protocol there MUST be exactly one dmi:SupportedProtocol element present in the response message.

/dmi-plain:GetFactoryAttributesDocumentResponseMessage/{any}

denotes an extensibility point where implementations MAY add additional structured content describing other DTF attributes.

Implementations of this operation MUST NOT send DMI-defined faults.

3.2.1 Example SOAP encoding

The following is a non-normative example of a GetFactoryAttributesDocument request message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erFactory/GetFactoryAttributesDocumentRequest
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:GetFactoryAttributesDocumentRequestMessage/>
  </s11:Body>
</s11:Envelope>
```

The following is a non-normative example of a GetFactoryAttributesDocument response message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erFactory/GetFactoryAttributesDocumentResponse
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:GetFactoryAttributesDocumentResponseMessage>
      <dmi-plain:FactoryAttributes>
        <dmi-plain:SupportedProtocol
          name="http://www.ogf.org/ogsa-dmi/2006/03/im/protocol/ftp">
          <dmi-plain:UndoStrategy
            name="http://www.ogf.org/ogsa-dmi/2006/03/im/retry/none"/>
          </dmi-plain:SupportedProtocol>
        ...
      </dmi-plain:FactoryAttributes>
    </dmi-plain:GetFactoryAttributesDocumentResponseMessage>
  </s11:Body>
</s11:Envelope>
```

4. Data Transfer Instance Port Type

This section normatively defines the message exchanges that standardize the means by which a client can communicate with a DTI as defined in the Functional Specification.

The following sections describe the message exchange expected for normal DTI usage, and briefly list the faults that an implementation MAY send instead of the normal response message. Each fault is then described at the end of this section in detail, as many of the operations defined for a DTI may send the same faults (with or without identical contents).

4.1 Start

A DTI implementation MUST support the message exchange defined in this section. This allows the requestor to manually start the DTI.

The format of the request message MUST be:

```
<dmi-plain:StartRequestMessage />
```

The wsa:Action for the request MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/StartRequest
```

Where:

/dmi-plain:StartRequestMessage

defines the message structure for the request message for the Start operation as described in [ANTONIOLETTI] section 5.2.1. The message MUST be empty.

The format of a response to such request MUST be:

```
<dmi-plain:StartResponseMessage />
```

The wsa:Action for the response MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/StartResponse
```

Where:

/dmi-plain:StartResponseMessage

defines the message structure for the response message for the Start operation as described in [ANTONIOLETTI] section 5.2.1.The message MUST be empty.

4.1.1 Example SOAP encoding

The following is a non-normative example of a StartRequestMessage request message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/StartRequest
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:StartRequestMessage/>
  </s11:Body>
</s11:Envelope>
```

The following is a non-normative example of a StartResponseMessage response message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/StartResponse
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:StartResponseMessage/>
  </s11:Body>
</s11:Envelope>
```

4.1.2 Faults

If a DTI does not respond with the StartResponseMessage then it MUST send a fault. Depending on the fault condition the DTI MAY send any of the following faults:

- IncorrectStateFault
- FailedStateTransitionFault
- TransferProtocolNotInstantiatableFault
- RequestedStateNotSupportedFault
- CustomFault

4.2 Stop

A DTI implementation MUST support the message exchange defined in this section. This allows the requestor to manually stop the DTI.

The format of the request message MUST be:

```
<dmi-plain:StopRequestMessage />
```

The wsa:Action for the request MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransferInstance/StopRequest
```

Where:

/dmi-plain:StopRequestMessage

defines the message structure for the request message for the Stop operation as described in [ANTONIOLETTI] section 5.2.2. The message MUST be empty.

The format of a response to such request MUST be:

```
<dmi-plain:StopResponseMessage />
```

The wsa:Action for the response MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransferInstance/StopResponse
```

Where:

/dmi-plain:StopResponseMessage

defines the message structure for the response message for the Stop operation as described in [ANTONIOLETTI] section 5.2.2. The message MUST be empty.

4.2.1 Example SOAP encoding

The following is a non-normative example of a StopRequestMessage request message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/StopRequest
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:StopRequestMessage/>
  </s11:Body>
</s11:Envelope>
```

The following is a non-normative example of a StopResponseMessage response message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/StopResponse
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:StopResponseMessage/>
  </s11:Body>
</s11:Envelope>
```

4.2.2 Faults

If a DTI does not respond with the StopResponseMessage then it MUST send a fault. Depending on the fault condition the DTI MAY send any of the following faults:

- FailedStateTransitionFault
- CustomFault

4.3 Resume

A DTI implementation MUST support the message exchange defined in this section. This allows the requestor to manually resume the DTI.

The format of the request message MUST be:

```
<dmi-plain:ResumeRequestMessage />
```

The wsa:Action for the request MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/ResumeRequest
```

Where:

/dmi-plain:ResumeRequestMessage

defines the message structure for the request message for the Resume operation as described in [ANTONIOLETTI] section 5.2.3. The message MUST be empty.

The format of a response to such request MUST be:

```
<dmi-plain:ResumeResponseMessage />
```

The wsa:Action for the response MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/ResumeResponse
```

Where:

/dmi-plain:ResumeResponseMessage

defines the message structure for the response message for the Resume operation as described in [ANTONIOLETTI] section 5.2.3. The message MUST be empty.

4.3.1 Example SOAP encoding

The following is a non-normative example of a ResumeRequestMessage request message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/ResumeRequest
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:ResumeRequestMessage/>
  </s11:Body>
</s11:Envelope>
```

The following is a non-normative example of a ResumeResponseMessage response message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/ResumeResponse
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:ResumeResponseMessage/>
  </s11:Body>
</s11:Envelope>
```

4.3.2 Faults

If a DTI does not respond with the SuspendResponseMessage then it MUST send a fault.

Depending on the fault condition the DTI MAY send any of the following faults:

- IncorrectStateFault
- FailedStateTransitionFault
- RequestedStateNotSupportedFault
- CustomFault

4.4 Suspend

A DTI implementation MUST support the message exchange defined in this section. This allows the requestor to manually suspend the DTI.

The format of the request message MUST be:

```
<dmi-plain:SuspendRequestMessage />
```

The wsa:Action for the request MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/SuspendRequest
```

Where:

/dmi-plain:SuspendRequestMessage

defines the message structure for the request message for the Suspend operation as described in [ANTONIOLETTI] section 5.2.4. The message MUST be empty.

The format of a response to such request MUST be:

```
<dmi-plain:SuspendResponseMessage />
```

The wsa:Action for the response MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/SuspendResponse
```

Where:

/dmi-plain:SuspendResponseMessage

defines the message structure for the response message for the Suspend operation as described in [ANTONIOLETTI] section 5.2.4. The message MUST be empty.

4.4.1 Example SOAP encoding

The following is a non-normative example of a SuspendRequestMessage request message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/SuspendRequest
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:SuspendRequestMessage/>
  </s11:Body>
</s11:Envelope>
```

The following is a non-normative example of a SuspendResponseMessage response message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/SuspendResponse
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:SuspendResponseMessage/>
  </s11:Body>
</s11:Envelope>
```

4.4.2 Faults

If a DTI does not respond with the SuspendResponseMessage then it MUST send a fault. Depending on the fault condition the DTI MAY send any of the following faults:

- IncorrectStateFault
- FailedStateTransitionFault
- RequestedStateNotSupportedFault
- CustomFault

4.5 GetState

A DTI implementation MUST support the message exchange defined in this section. This allows the requestor to query the current state of the DTI.

The format of the request message MUST be:

```
<dmi-plain:GetStatusRequestMessage />
```

The wsa:Action for the request MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/GetStatusRequest
```

Where:

/dmi-plain:GetStatusRequestMessage

defines the message structure for the request message for the GetState operation as described in [ANTONIOLETTI] section 5.2.5. The message MUST be empty.

The format of a response to such request MUST follow:

```
<dmi-plain:GetStatusResponseMessage>
  <dmi:State value=”xs:string”>
    <dmi:Detail>
      {any namespace=##other}*
    </dmi:Detail>?
  </dmi:State>
</dmi-plain:GetStatusResponseMessage>
```

The wsa:Action for the response MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/GetStatusResponse
```

Where:

/dmi-plain:GetStatusResponseMessage

defines the message structure for the response message for the GetState operation as described in [ANTONIOLETTI] section 5.2.5.

/dmi-plain:GetStatusResponseMessage/dmi:State

defines the structure of the status information as described in [ANTONIOLETTI] section 5.1.2.

/dmi-plain:GetStatusResponseMessage/dmi:State/@dmi:value

defines the value of the DTI state of type xs:string. It MUST have exactly one of the values defined in [ANTONIOLETTI] section 5.1.3.

/dmi-plain:GetStatusResponseMessage/dmi:State/dmi:Detail

This represents a container for any additional information a DMI implementation MAY want to provide to its clients. This may include non-normative, non-interoperable pieces of information.

/dmi-plain:GetStatusResponseMessage/dmi:State/dmi:Detail/{any}

This represents the extensibility point for additional information a DMI implementation may want to provide to clients.

4.5.1 Example SOAP encoding

The following is a non-normative example of a GetStatusRequestMessage request message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/GetStatusRequest
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:GetStatusRequestMessage/>
  </s11:Body>
</s11:Envelope>
```

The following is a non-normative example of a GetStatusResponseMessage response message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/GetStatusResponse
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:GetStatusResponseMessage>
      <dmi:State dmi:value=”Transferring”/>
    </dmi-plain:GetStatusResponseMessage>
  </s11:Body>
</s11:Envelope>
```

4.5.2 Faults

A DTI MUST NOT send any DMI-defined fault as a response to this operation.

4.6 GetInstanceAttributesDocument

A DTI implementation MUST support the message exchange defined in this section. This allows the requestor to retrieve the attribute values of the addressed DTI.

The format of the request message MUST be:

```
<dmi-plain:GetInstanceAttributesDocumentRequestMessage />
```

The wsa:Action for the request MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/GetInstanceAttributesDocumentRequest
```

Where:

/dmi-plain:GetInstanceAttributesDocumentRequestMessage

defines the message structure of the request message for a GetInstanceAttributesDocument operation as described in [ANTONIOLETTI] section 5.2.6. The message MUST be empty.

The format of a response to such request MUST follow:

```
<dmi-plain:GetInstanceAttributesDocumentResponseMessage>
  <dmi-plain:InstanceAttributes>
    <dmi:StartTime> xs:dateTime </dmi:StartTime>
    <dmi:State value=”xs:string”>
      <dmi:Detail> {any namespace=##other}* </dmi:Detail>?
    </dmi:State>
    <dmi:CompletionTime> xs:dateTime </dmi:CompletionTime>?
    <dmi:TotalDataSize> xs:unsignedLong </dmi:TotalDataSize>?
    <dmi:BytesTransferred> xs:unsignedLong </dmi:BytesTransferred>?
    <dmi:Attempts> xs:unsignedInt </dmi:Attempts>?
    {any namespace=##other}*
  </dmi-plain:InstanceAttributes>
</dmi-plain:GetInstanceAttributesDocumentResponseMessage>
```

The wsa:Action for the response MUST contain the URI:

```
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
```

```
erInstance/GetInstanceAttributesDocumentResponse
```

Where:

/dmi-plain:GetInstanceAttributesDocumentResponseMessage
defines the message structure of the response message for a GetInstanceAttributesDocument operation as described in [ANTONIOLETTI] section 5.2.6.

/dmi-plain:GetInstanceAttributesDocumentResponseMessage/dmi-plain:InstanceAttributes
defines the structure of the wrapper XML element containing all DTI attributes, as described in [ANTONIOLETTI] section 5.2.6.1.

/dmi-plain:GetInstanceAttributesDocumentResponseMessage/dmi-plain:InstanceAttributes/dmi:StartTime
describes the start time of the DTI as described in [ANTONIOLETTI] section 5.1.1.

/dmi-plain:GetInstanceAttributesDocumentResponseMessage/dmi-plain:InstanceAttributes/dmi:State
describes the state of the DTI as described in [ANTONIOLETTI] section 5.1.2.

/dmi-plain:GetInstanceAttributesDocumentResponseMessage/dmi-plain:InstanceAttributes/dmi:CompletionTime
describes the completion time of the DTI as described in [ANTONIOLETTI] section 5.1.4.

/dmi-plain:GetInstanceAttributesDocumentResponseMessage/dmi-plain:InstanceAttributes/dmi:TotalDataSize
describes the total size of the data in bytes as described in [ANTONIOLETTI] section 5.1.5.

/dmi-plain:GetInstanceAttributesDocumentResponseMessage/dmi-plain:InstanceAttributes/dmi:BytesTransferred
describes the amounts of bytes transferred at that time as described in [ANTONIOLETTI] section 5.1.6.

/dmi-plain:GetInstanceAttributesDocumentResponseMessage/dmi-plain:InstanceAttributes/dmi:Attempts
describes amount of attempts that have already failed plus the current attempt as described in [ANTONIOLETTI] section 5.1.7.

/dmi-plain:GetInstanceAttributesDocumentResponseMessage/{any}
denotes an extensibility point where implementations MAY add additional structured content describing other DTI attributes.

4.6.1 Example SOAP encoding

The following is a non-normative example of a GetInstanceAttributesDocument request message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/GetInstanceAttributesDocumentRequest
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:GetInstanceAttributesDocumentRequestMessage/>
  </s11:Body>
</s11:Envelope>
```

The following is a non-normative example of a GetInstanceAttributesDocument response message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/DataTransf
erInstance/GetInstanceAttributesDocumentResponse
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <dmi-plain:GetInstanceAttributesDocumentResponseMessage>
      <dmi-plain:InstanceAttributes>
        <dmi:StartTime> 2008-10-12T09:44:20.137+01:00 </dmi:StartTime>
        <dmi:State value=”Transferring”/>
        <dmi:BytesTransferred> 123456789 <dmi:BytesTransferred>
        <dmi:Attempts> 1 </dmi:Attempts>
      </dmi-plain:InstanceAttributes>
    </dmi-plain:GetInstanceAttributesDocumentResponseMessage>
  </s11:Body>
</s11:Envelope>
```

4.6.2 Faults

A DTI MUST NOT send any DMI-defined fault as response to this operation.

4.7 Data Transfer Instance Operation Faults

This section defines the faults that a DTI MAY send back to a client. Previous sections define which fault may be sent back for each defined DTI operation. The following sections define the exact message formats of each defined fault.

4.7.1 IncorrectStateFault

If a DTI received a request message for an operation, which is not allowed in the current state of the DTI, then the DTI MUST send an IncorrectStateFault. For example, if the DTI is in Suspended state when it receives a StartRequestMessage the DTI MUST send an IncorrectStateFault.

The format of the IncorrectStateFault MUST follow:

```
<dmi-plain:IncorrectFault>
  <dmi-plain:Message> xs:string </dmi-plain:Message>
  <dmi-plain:Timestamp> xs:dateTime </dmi-plain:Timestamp>
  {any namespace=##other}*
</dmi-plain:IncorrectStateFault>
```

The wsa:Action for the fault MUST contain the URI:

```
http://www.w3.org/2005/08/addressing/soap/fault
```

Where:

/dmi-plain:IncorrectStateFault

defines the message structure for the IncorrectStateFault as described in [ANTONIOLETTI] section 5.3.1.

/dmi-plain:IncorrectStateFault/dmi-plain:Message

is a descriptive, user-readable text describing the exact fault conditions. Its type is `xs:string`.

/dmi-plain:IncorrectStateFault/dmi-plain:Timestamp

describes the point in time when the fault condition has been detected (as opposed to when the message is generated, or sent).

/dmi-plain:IncorrectStateFault/{any}

denotes an extensibility point where implementations MAY add additional structured information to the fault.

4.7.1.1 Example SOAP encoding

The following is a non-normative example of a IncorrectStateFault fault message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://www.w3.org/2005/08/addressing/soap/fault
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <s11:Fault>
      <faultcode>s11:Server</faultcode>
      <faultstring>
        DTI received StartRequestMessage while in state Transferring.
      </faultstring>
      <detail>
        <dmi-plain:IncorrectStateFault>
          <dmi-plain:Message>
            DTI received StartRequestMessage while in state
            Transferring.
          </dmi-plain:Message>
          <dmi-plain:Timestamp>
            2008-10-12T09:44:20.137+01:00
          </dmi-plain:Timestamp>
        </dmi-plain:IncorrectStateFault>
        ...
      </detail>
    </s11:Fault>
  </s11:Body>
</s11:Envelope>
```

4.7.2 FailedStateTransitionFault

If a DTI received a request message causing a state transition (e.g. StartRequestMessage), but is unable to fulfill that state transition then the DTI MUST send a FailedStateTransitionFault.

The format of the FailedStateTransitionFault MUST follow:

```
<dmi-plain:FailedStateTransitionFault>
  <dmi-plain:Message> xs:string </dmi-plain:Message>
  <dmi-plain:Timestamp> xs:dateTime </dmi-plain:Timestamp>
  {any namespace=##other}*
</dmi-plain:FailedStateTransitionFault>
```

The wsa:Action for the fault MUST contain the URI:
`http://www.w3.org/2005/08/addressing/soap/fault`

Where:

/dmi-plain:FailedStateTransitionFault
defines the message structure for the FailedStateTransitionFault as described in [ANTONIOLETTI] section 5.3.2.

/dmi-plain:FailedStateTransitionFault/dmi-plain:Message
is a descriptive, user-readable text describing the exact fault conditions. Its type is `xs:string`.

/dmi-plain:FailedStateTransitionFault/dmi-plain:Timestamp
describes the point in time when the fault condition has been detected (as opposed to when the message is generated, or sent).

/dmi-plain:FailedStateTransitionFault/{any}
denotes an extensibility point where implementations MAY add additional structured information to the fault.

4.7.2.1 Example SOAP encoding

The following is a non-normative example of a FailedStateTransitionFault fault message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://www.w3.org/2005/08/addressing/soap/fault
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <s11:Fault>
      <faultcode>s11:Server</faultcode>
      <faultstring>
        Could not start the DTI: Network is down.
      </faultstring>
      <detail>
```

```
        <dmi-plain:FailedStateTransitionFault>
          <dmi-plain:Message>
            Could not start the DTI: Network is down.
          </dmi-plain:Message>
          <dmi-plain:Timestamp>
            2008-10-12T09:44:20.137+01:00
          </dmi-plain:Timestamp>
        </dmi-plain:FailedStateTransitionFault>
        ...
      </detail>
    </s11:Fault>
  </s11:Body>
</s11:Envelope>
```

4.7.3 TransferProtocolNotInstantiatableFault

If a DTI was unable to instantiate the underlying data transfer it MUST send a TransferProtocolNotInstantiatableFault.

The format of the TransferProtocolNotInstantiatableFault MUST follow:

```
<dmi-plain:TransferProtocolNotInstantiatableFault>
  <dmi-plain:Message> xs:string </dmi-plain:Message>
  <dmi-plain:Timestamp> xs:dateTime </dmi-plain:Timestamp>
  <dmi-plain:Protocol> xs:anyURI </dmi-plain:Protocol>
  <dmi-plain:SourceDEPR>
    wsa:EndpointReferenceType
  </dmi-plain:SourceDEPR>?
  <dmi-plain:SinkDEPR>
    wsa:EndpointReferenceType
  </dmi-plain:SinkDEPR>?
  {any namespace=##other}*
</dmi-plain:RequestedStateNotSupportedFault>
```

The wsa:Action for the fault MUST contain the URI:
`http://www.w3.org/2005/08/addressing/soap/fault`

Where:

/dmi-plain:TransferProtocolNotInstantiatableFault
defines the message structure for the TransferProtocolNotInstantiatableFault as described in [ANTONIOLETTI] section 5.3.3.

/dmi-plain:TransferProtocolNotInstantiatableFault/dmi-plain:Message
is a descriptive, user-readable text describing the exact fault conditions. Its type is `xs:string`.

/dmi-plain:TransferProtocolNotInstantiatableFault/dmi-plain:Timestamp
describes the point in time when the fault condition has been detected (as opposed to when the message is generated, or sent).

/dmi-plain:TransferProtocolNotInstantiatableFault/dmi-plain:Protocol
describes the URI of the protocol that the DTI attempted to instantiate. The type is `xs:anyURI`.

/dmi-plain:TransferProtocolNotInstantiatableFault/dmi-plain:SourceDEPR

describes the DataEPR for the source of the data transfer, as described in [ANTONIOLETTI] section 5.3.3.

/dmi-plain:TransferProtocolNotInstantiatableFault/dmi-plain:SinkDEPR

describes the DataEPR for the sink of the data transfer, as described in [ANTONIOLETTI] section 5.3.3.

/dmi-plain:TransferProtocolNotInstantiatableFault/{any}

denotes an extensibility point where implementations MAY add additional structured information to the fault.

4.7.3.1 Example SOAP encoding

The following is a non-normative example of a TransferProtocolNotInstantiatableFault fault message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://www.w3.org/2005/08/addressing/soap/fault
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <s11:Fault>
      <faultcode>s11:Server</faultcode>
      <faultstring>
        No data found at data URL.
      </faultstring>
      <detail>
        <dmi-plain:TransferProtocolNotInstantiatableFault>
          <dmi-plain:Message>
            No data found at data URL.
          </dmi-plain:Message>
          <dmi-plain:Timestamp>
            2008-10-12T09:44:20.137+01:00
          </dmi-plain:Timestamp>
          <dmi-plain:Protocol>
            http://www.ogf.org/ogsa-dmi/2006/03/im/protocol/ftp
          </dmi-plain:Protocol>
        </dmi-plain:TransferProtocolNotInstantiatableFault>
        ...
      </detail>
    </s11:Fault>
  </s11:Body>
</s11:Envelope>
```

4.7.4 RequestedStateNotSupportedFault

If a DTI received a request message causing a state transition that it does not support (as opposed to being unable to fulfill the transition) then the DTI MUST send a RequestedStateNotSupportedFault.

The format of the RequestedStateNotSupportedFault MUST follow:

```
<dmi-plain:RequestedStateNotSupportedFault>
  <dmi-plain:Message> xs:string </dmi-plain:Message>
  <dmi-plain:Timestamp> xs:dateTime </dmi-plain:Timestamp>
  {any namespace=##other}*
</dmi-plain:RequestedStateNotSupportedFault>
```

The wsa:Action for the fault MUST contain the URI:

```
      http://www.w3.org/2005/08/addressing/soap/fault
```

Where:

/dmi-plain:RequestedStateNotSupportedFault
defines the message structure for the RequestedStateNotSupportedFault as described in [ANTONIOLETTI] section 5.3.4.

/dmi-plain:RequestedStateNotSupportedFault/dmi-plain:Message
is a descriptive, user-readable text describing the exact fault conditions. Its type is `xs:string`.

/dmi-plain:RequestedStateNotSupportedFault/dmi-plain:Timestamp
describes the point in time when the fault condition has been detected (as opposed to when the message is generated, or sent).

/dmi-plain:RequestedStateNotSupportedFault/{any}
denotes an extensibility point where implementations MAY add additional structured information to the fault.

4.7.4.1 Example SOAP encoding

The following is a non-normative example of a RequestedStateNotSupportedFault fault message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://www.w3.org/2005/08/addressing/soap/fault
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <s11:Fault>
      <faultcode>s11:Server</faultcode>
      <faultstring>
        Suspended state not supported for FTP transfers.
      </faultstring>
      <detail>
        <dmi-plain:RequestedStateNotSupportedFault>
          <dmi-plain:Message>
            Suspended state not supported for FTP transfers.
          </dmi-plain:Message>
          <dmi-plain:Timestamp>
            2008-10-12T09:44:20.137+01:00
          </dmi-plain:Timestamp>
        </dmi-plain:RequestedStateNotSupportedFault>
        ...
      </detail>
    </s11:Fault>
  </s11:Body>
</s11:Envelope>
```

4.7.5 CustomFault

If a DTI encounters any other fault condition that is not captured in the defined faults above then it MUST send a CustomFault.

The format of the CustomFault MUST follow:

```
<dmi-plain:CustomFault>
  <dmi-plain:Message> xs:string </dmi-plain:Message>
  <dmi-plain:Timestamp> xs:dateTime </dmi-plain:Timestamp>
  {any namespace=##other}*
</dmi-plain:CustomFault>
```

The wsa:Action for the fault MUST contain the URI:
`http://www.w3.org/2005/08/addressing/soap/fault`

Where:

/dmi-plain:CustomFault
defines the message structure for the CustomFault as described in [ANTONIOLETTI] section 4.3.4.

/dmi-plain:CustomFault/dmi-plain:Message
is a descriptive, user-readable text describing the exact fault conditions. Its type is `xs:string`.

/dmi-plain:CustomFault/dmi-plain:Timestamp
describes the point in time when the fault condition has been detected (as opposed to when the message is generated, or sent).

/dmi-plain:CustomFault/{any}
denotes an extensibility point where implementations MAY add additional structured information to the fault.

4.7.5.1 Example SOAP encoding

The following is a non-normative example of a CustomFault fault message using SOAP 1.1:

```
<s11:Envelope>
  <s11:Header>
    ...
    <wsa:Action>
      http://www.w3.org/2005/08/addressing/soap/fault
    </wsa:Action>
    ...
  </s11:Header>
  <s11:Body>
    <s11:Fault>
      <faultcode>s11:Server</faultcode>
      <faultstring>
        Instantiation error for transfer protocol
      </faultstring>
      <detail>
        <dmi-plain:CustomFault>
          <dmi-plain:Message>
            Instantiation error for transfer protocol
          </dmi-plain:Message>
          <dmi-plain:Timestamp>
            2008-10-12T09:44:20.137+01:00
          </dmi-plain:Timestamp>
        </dmi-plain:CustomFault>
        ...
      </detail>
    </s11:Fault>
  </s11:Body>
</s11:Envelope>
```

5. Security Considerations

This Rendering specification defines the message formats for the operations defined in [ANTONIOLETTI]. Implementations of the Rendering specification must reflect whatever security requirements the target deployment environment requires to be present. The authors expect that implementations will compose the Rendering specification with security related specifications to achieve the required security level of the target environment.

From a functional perspective, section 6 in [ANTONIOLETTI] fully applies to this document, and need not repeated here.

6. Author Information

Mario Antonioletti
EPCC
JCMB
The King’s Buildings
Mayfield Road
Edinburgh EH9 3JZ

Michel Drescher
Fujitsu Laboratories of Europe, Ltd.
Hayes Park Central, Hayes End Road
Hayes, Middlesex UB4 8FE
United Kingdom

Steven Newhouse
Microsoft Corporation
One Microsoft Way
Redmond
USA
WA 98053

Updated addresses for authors at time of publication:
‘

Michel Drescher
European Grid Initiative (EGI.eu)
140 Science Park,
1098 XG Amsterdam
The Netherlands

Steven Newhouse
European Grid Initiative (EGI.eu)
140 Science Park,
1098 XG Amsterdam
The Netherlands

7. Acknowledgements

We would like to acknowledge the contributions from Allen Luniewski, Ravi Madduri, Mohammad Shahbaz Memon.

8. Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

9. Disclaimer

This document and the information contained herein is provided on an “As Is” basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

10. Full Copyright Notice

OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

11. References

[ANTONIOLETTI] Antonioletti, M., Drescher, M., Luniewski, A., Newhouse, S., Madduri, R., OGSA-DMI Functional Specification 1.0, GFD.134, Open Frid Forum 15 August 2008.

[BRADNER1] Bradner, S. Key Words for Use in RFCs to Indicate Requirement Levels, RFC 2119. March 1997.

[BRAY] Bray, T., Hollander, D., Laymann, A., Tobin, R. Namespaces in XML 1.0, http://www.w3.org/TR/REC-xml-names/. W3C, 16 August 2006.

12. Reference XML Schema

This section defines the reference XML Schema for the OGSA-DMI WSDL rendering for plain Web Services. All XML messages that are involved in OGSA-DMI communications for plain Web Services MUST validate against this XML Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  The OGF takes no position regarding the validity or scope of any
  intellectual property or other rights that might be claimed to
  pertain to the implementation or use of the technology described
  in this document or the extent to which any license under such
  rights might or might not be available; neither does it represent
  that it has made any effort to identify any such rights.  Copies
  of claims of rights made available for publication and any
  assurances of licenses to be made available, or the result of an
  attempt made to obtain a general license or permission for the use
  of such proprietary rights by implementers or users of this
  specification can be obtained from the OGF Secretariat.

  The OGF invites any interested party to bring to its attention any
  copyrights, patents or patent applications, or other proprietary
  rights which may cover technology that may be required to practice
  this recommendation.  Please address the information to the OGF
  Executive Director.

  This document and the information contained herein is provided on
  an "As Is" basis and the OGF disclaims all warranties, express or
  implied, including but not limited to any warranty that the use
  of the information herein will not infringe any rights or any
  implied warranties of merchantability or fitness for a particular
  purpose.

  Copyright (C) Open Grid Forum (2008). All Rights Reserved.

  This document and translations of it may be copied and furnished to
  others, and derivative works that comment on or otherwise explain
  it or assist in its implementation may be prepared, copied, published
  and distributed, in whole or in part, without restriction of any
  kind, provided that the above copyright notice and this paragraph are
  included on all such copies and derivative works. However, this
  document itself may not be modified in any way, such as by removing
  the copyright notice or references to the OGF or other organizations,
  except as needed for the purpose of developing Grid Recommendations
  in which case the procedures for copyrights defined in the OGF
  Document process must be followed, or as required to translate it
  into languages other than English.

  The limited permissions granted above are perpetual and will not be
  revoked by the OGF or its successors or assignees.
-->
<schema

targetNamespace="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain
"
      xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:dmi="http://schemas.ogf.org/dmi/2008/05/dmi"
  xmlns:dmi-
plain="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain"
      xmlns:wsa="http://www.w3.org/2005/08/addressing"
  elementFormDefault="qualified">


  <!-- ======================================================== -->
  <!-- ==                                                    == -->
  <!-- == OGSA DMI WS-I Message model                        == -->
  <!-- ==                                                    == -->
  <!-- == This XML Schema document defines the XML elements  == -->
  <!-- == necessary to define WS-I compliant WS operations   == -->
  <!-- == and messages.                                      == -->
  <!-- ======================================================== -->

  <import namespace="http://www.w3.org/2005/08/addressing"
    schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>

  <import namespace="http://schemas.ogf.org/dmi/2008/05/dmi"
     schemaLocation="./dmi-draft-64.xsd"/>


  <!-- ============================================ -->
  <!-- == Factory / Instance Fault message model == -->
  <!-- ============================================ -->

  <!-- Common type for all OGSA-DMI specific faults -->
  <complexType name="DMIFaultType">
    <sequence>
      <element name="Message" type="string" />
      <element name="Timestamp" type="dateTime" />
      <any namespace="#other" minOccurs="0" maxOccurs="unbounded" />
    </sequence>
  </complexType>

  <!-- UnsatisfiableRequestOptionsFault message -->
  <element name="UnsatisfiableRequestOptionsFault"
           type="dmi-plain:UnsatisfiableRequestOptionsFaultType" />
  <complexType name="UnsatisfiableRequestOptionsFaultType">
    <complexContent>
      <extension base="dmi-plain:DMIFaultType"/>
    </complexContent>
  </complexType>

  <!-- NoTransferProtocolAgreementFault message -->
  <element name="NoTransferProtocolAgreementFault"
           type="dmi-plain:NoTransferProtocolAgreementFaultType" />
  <complexType name="NoTransferProtocolAgreementFaultType">
    <complexContent>
      <extension base="dmi-plain:DMIFaultType"/>
    </complexContent>
  </complexType>

  <!-- IncorrectStateFault message -->
  <element name="IncorrectStateFault"
           type="dmi-plain:IncorrectStateFaultType" />
  <complexType name="IncorrectStateFaultType">
    <complexContent>
      <extension base="dmi-plain:DMIFaultType"/>
    </complexContent>
  </complexType>

  <!-- FailedStateTransitionFault message -->
  <element name="FailedStateTransitionFault"
           type="dmi-plain:FailedStateTransitionFaultType" />
  <complexType name="FailedStateTransitionFaultType">
    <complexContent>
      <extension base="dmi-plain:DMIFaultType"/>
    </complexContent>
  </complexType>

  <!-- RequestedStateNotSupportedFault message -->
  <element name="RequestedStateNotSupportedFault"
           type="dmi-plain:RequestedStateNotSupportedFaultType" />
  <complexType name="RequestedStateNotSupportedFaultType">
    <complexContent>
      <extension base="dmi-plain:DMIFaultType"/>
    </complexContent>
  </complexType>

  <!-- CustomFault message -->
  <element name="CustomFault"
           type="dmi-plain:CustomFaultType" />
  <complexType name="CustomFaultType">
    <complexContent>
      <extension base="dmi-plain:DMIFaultType"/>
    </complexContent>
  </complexType>

  <!-- TransferProtocolNotInstantiatableFault message -->
  <element name="TransferProtocolNotInstantiatableFault"
           type="dmi-plain:TransferProtocolNotInstantiatableFaultType"
/>
  <complexType name="TransferProtocolNotInstantiatableFaultType">
    <complexContent>
      <extension base="dmi-plain:DMIFaultType">
        <sequence>
          <element name="Protocol" type="anyURI" />
          <element name="SourceDEPR" type="wsa:EndpointReferenceType"
                   minOccurs="0" />
          <element name="SinkDEPR" type="wsa:EndpointReferenceType"
                   minOccurs="0" />
        </sequence>
      </extension>
    </complexContent>
  </complexType>


  <!-- ================================================ -->
  <!-- == Data Transfer Factory related message model == -->
  <!-- ================================================ -->

  <!--
    Operation "GetFactoryAttributesDocument"
    Input:  n/a
    Output: <dmi:FactoryAttributes />
    Faults: n/a
  -->
  <!-- Request Message -->
  <element name="GetFactoryAttributesDocumentRequestMessage" />
  <!-- Response Message -->
  <element name="GetFactoryAttributesDocumentResponseMessage"
           type="dmi-plain:GetFactoryAttributesDocumentResponseType"/>
  <complexType name="GetFactoryAttributesDocumentResponseType">
    <sequence>
      <element name="FactoryAttributes"
               type="dmi-plain:FactoryAttributesType" />
    </sequence>
  </complexType>
  <!-- Supplementary elements -->
  <complexType name="FactoryAttributesType">
    <sequence>
      <element name="SupportedProtocol"
               type="dmi:SupportedProtocolType"
          maxOccurs="unbounded" />
      <any namespace="##other"  minOccurs="0" maxOccurs="unbounded" />
    </sequence>
  </complexType>

  <!--
    Operation "GetDataTransferInstance"
    Input:  <dmi:SourceDEPR />
            <dmi:SinkDEPR />
            <dmi:TransferRequirements />
    Output: <dmi:ServiceInstance />
    Faults: <dmi:UnsatisfiableRequestOptionsFault />
            <dmi:NoTransferProtocolAgreementFault />
            <dmi:CustomFault />
  -->
  <!-- Request Message -->
```

```
<element name="GetDataTransferInstanceRequestMessage"
         type="dmi-plain:GetDataTransferInstanceRequestType"/>
<complexType name="GetDataTransferInstanceRequestType">
  <sequence>
    <element name="SourceDEPR" type="wsa:EndpointReferenceType"/>
    <element name="SinkDEPR" type="wsa:EndpointReferenceType"/>
    <element name="TransferRequirements"
             type="dmi:TransferRequirementsType" />
  </sequence>
</complexType>
<!-- Response Message -->
<element name="GetDataTransferInstanceResponseMessage"
         type="dmi-plain:GetDataTransferInstanceResponseType"/>
<complexType name="GetDataTransferInstanceResponseType">
  <sequence>
    <element name="ServiceInstance"
             type="wsa:EndpointReferenceType"/>
  </sequence>
</complexType>


<!-- ================================================== -->
<!-- == Data Transfer Instance related message model == -->
<!-- ================================================== -->

<!--
  Operation "GetInstanceAttributesDocument"
  Input:  n/a
  Output: <dmi:InstanceAttributes />
  Faults: n/a
-->
<!-- Request Message -->
<element name="GetInstanceAttributesDocumentRequestMessage" />
<!-- Response Message -->
<element name="GetInstanceAttributesDocumentResponseMessage"
         type="dmi-plain:GetInstanceAttributesDocumentResponseType"/>
<complexType name="GetInstanceAttributesDocumentResponseType">
  <sequence>
    <element name="InstanceAttributes"
             type="dmi-plain:InstanceAttributesType" />
  </sequence>
</complexType>
<!-- Supplementary elements -->
<complexType name="InstanceAttributesType">
  <sequence>
    <element ref="dmi:StartTime" minOccurs="0"/>
    <element ref="dmi:State" />
    <element ref="dmi:CompletionTime" minOccurs="0" />
    <element ref="dmi:TotalDataSize" minOccurs="0" />
    <element ref="dmi:BytesTransferred" minOccurs="0" />
    <element ref="dmi:Attempts" />
    <any namespace="##other" minOccurs="0" maxOccurs="unbounded" />
  </sequence>
</complexType>

<!--
```

```
  Operation "GetStatus"
  Input:  n/a
  Output: <dmi:Status />
  Faults: n/a
-->
<!-- Request Message -->
<element name="GetStatusRequestMessage" />
<!-- Response Message -->
<element name="GetStatusResponseMessage"
         type="dmi-plain:GetStatusResponseType" />
<complexType name="GetStatusResponseType">
  <sequence>
    <element ref="dmi:State" />
  </sequence>
</complexType>

<!--
  Operation "Start"
  Input:  n/a
  Output: n/a
  Faults: <dmi:IncorrectStateFault />
          <dmi:FailedStateTransitionFault />
          <dmi:TransferProtocolNotInstantiatableFault />
          <dmi:CustomFault />
-->
<!-- Request Message -->
<element name="StartRequestMessage" />
<!-- Response Message -->
<element name="StartResponseMessage" />

<!--
  Operation "stop"
  Input:  n/a
  Output: n/a
  Faults: <dmi:IncorrectStateFault />
          <dmi:FailedStateTransitionFault />
          <dmi:CustomFault />
-->
<!-- Request Message -->
<element name="StopRequestMessage" />
<!-- Response Message -->
<element name="StopResponseMessage" />

<!--
  Operation "restart"
  Input:  n/a
  Output: n/a
  Faults: <dmi:IncorrectStateFault />
          <dmi:FailedStateTransitionFault />
          <dmi:CustomFault />
-->
<!-- Request Message -->
<element name="ResumeRequestMessage" />
<!-- Response Message -->
<element name="ResumeResponseMessage" />
```

```
  <!--
    Operation "suspend"
    Input:  n/a
    Output: n/a
    Faults: <dmi:IncorrectStateFault />
            <dmi:FailedStateTransitionFault />
            <dmi:CustomFault />
  -->
  <!-- Request Message -->
  <element name="SuspendRequestMessage" />
  <!-- Response Message -->
  <element name="SuspendResponseMessage" />

</schema>
```

13. Reference WSDL Port Type

This section defines the reference WSDL port type for plain Web Services. It omits y design the WSDL Binding definition and WSDL Service definition to allow flexibility in binding the normative port type to alternative transport protocols, such as SOAP over XMPP or direct TCP. All WSDL messages that are involved in OGSA-DMI communications for plain Web Services MUST validate against this WSDL file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  The OGF takes no position regarding the validity or scope of any
  intellectual property or other rights that might be claimed to
  pertain to the implementation or use of the technology described
  in this document or the extent to which any license under such
  rights might or might not be available; neither does it represent
  that it has made any effort to identify any such rights.  Copies
  of claims of rights made available for publication and any
  assurances of licenses to be made available, or the result of an
  attempt made to obtain a general license or permission for the use
  of such proprietary rights by implementers or users of this
  specification can be obtained from the OGF Secretariat.

  The OGF invites any interested party to bring to its attention any
  copyrights, patents or patent applications, or other proprietary
  rights which may cover technology that may be required to practice
  this recommendation.  Please address the information to the OGF
  Executive Director.

  This document and the information contained herein is provided on
  an "As Is" basis and the OGF disclaims all warranties, express or
  implied, including but not limited to any warranty that the use
  of the information herein will not infringe any rights or any
  implied warranties of merchantability or fitness for a particular
  purpose.

  Copyright (C) Open Grid Forum (2008). All Rights Reserved.

  This document and translations of it may be copied and furnished to
  others, and derivative works that comment on or otherwise explain
  it or assist in its implementation may be prepared, copied, published
  and distributed, in whole or in part, without restriction of any
  kind, provided that the above copyright notice and this paragraph are
  included on all such copies and derivative works. However, this
  document itself may not be modified in any way, such as by removing
  the copyright notice or references to the OGF or other organizations,
  except as needed for the purpose of developing Grid Recommendations
  in which case the procedures for copyrights defined in the OGF
  Document process must be followed, or as required to translate it
  into languages other than English.

  The limited permissions granted above are perpetual and will not be
  revoked by the OGF or its successors or assignees.
-->
<wsdl:definitions name="dmi-plain-rendering"

targetNamespace="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain
"
  xmlns:dmi="http://schemas.ogf.org/dmi/2008/05/dmi"
  xmlns:dmi-
plain="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">


  <!-- ========================== -->
  <!-- == Importing XML Schema == -->
  <!-- ========================== -->
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain
">
      <xsd:include schemaLocation="./dmi-messages-draft-7.xsd" />
    </xsd:schema>
  </wsdl:types>


  <!-- ========================== -->
  <!-- == DTF related messages == -->
  <!-- ========================== -->

  <!-- getFactoryAttributesDocument -->
  <wsdl:message name="GetFactoryAttributesDocumentRequest">
    <wsdl:part name="parameters"
      element="dmi-plain:GetFactoryAttributesDocumentRequestMessage" />
  </wsdl:message>

  <wsdl:message name="GetFactoryAttributesDocumentResponse">
    <wsdl:part name="parameters"
      element="dmi-plain:GetFactoryAttributesDocumentResponseMessage"
/>
  </wsdl:message>

  <!-- GetDataTransferInstance -->
  <wsdl:message name="GetDataTransferInstanceRequest">
    <wsdl:part name="parameters"
      element="dmi-plain:GetDataTransferInstanceRequestMessage" />
  </wsdl:message>

  <wsdl:message name="GetDataTransferInstanceResponse">
    <wsdl:part name="parameters"
      element="dmi-plain:GetDataTransferInstanceResponseMessage" />
  </wsdl:message>


  <!-- ========================== -->
  <!-- == DTI related messages == -->
  <!-- ========================== -->

  <!-- getInstanceAttributesDocument -->
  <wsdl:message name="GetInstanceAttributesDocumentRequest">
    <wsdl:part name="parameters"
      element="dmi-plain:GetInstanceAttributesDocumentRequestMessage"
/>
  </wsdl:message>

  <wsdl:message name="GetInstanceAttributesDocumentResponse">
    <wsdl:part name="parameters"
      element="dmi-plain:GetInstanceAttributesDocumentResponseMessage"
/>
  </wsdl:message>

  <!-- getStatus -->
  <wsdl:message name="GetStatusRequest">
    <wsdl:part name="parameters"
      element="dmi-plain:GetStatusRequestMessage" />
  </wsdl:message>

  <wsdl:message name="GetStatusResponse">
    <wsdl:part name="parameters"
      element="dmi-plain:GetStatusResponseMessage" />
  </wsdl:message>

  <!-- start -->
  <wsdl:message name="StartRequest">
    <wsdl:part name="parameters"
      element="dmi-plain:StartRequestMessage" />
  </wsdl:message>

  <wsdl:message name="StartResponse">
    <wsdl:part name="parameters"
      element="dmi-plain:StartResponseMessage" />
  </wsdl:message>

  <!-- stop -->
  <wsdl:message name="StopRequest">
    <wsdl:part name="parameters" element="dmi-plain:StopRequestMessage"
/>
  </wsdl:message>
  <wsdl:message name="StopResponse">
    <wsdl:part name="parameters"
      element="dmi-plain:StopResponseMessage" />
  </wsdl:message>

  <!-- resume -->
  <wsdl:message name="ResumeRequest">
    <wsdl:part name="parameters"
      element="dmi-plain:ResumeRequestMessage" />
  </wsdl:message>

  <wsdl:message name="ResumeResponse">
    <wsdl:part name="parameters"
      element="dmi-plain:ResumeResponseMessage" />
  </wsdl:message>

  <!-- suspend -->
  <wsdl:message name="SuspendRequest">
    <wsdl:part name="parameters"
      element="dmi-plain:SuspendRequestMessage" />
  </wsdl:message>

  <wsdl:message name="SuspendResponse">
    <wsdl:part name="parameters"
      element="dmi-plain:SuspendResponseMessage" />
  </wsdl:message>


  <!-- =========================== -->
  <!-- == Fault related messages == -->
  <!-- ============================ -->

  <wsdl:message name="UnsatisfiableRequestOptionsFault">
    <wsdl:part name="parameters"
      element="dmi-plain:UnsatisfiableRequestOptionsFault" />
  </wsdl:message>

  <wsdl:message name="NoTransferProtocolAgreementFault">
    <wsdl:part name="parameters"
      element="dmi-plain:NoTransferProtocolAgreementFault" />
  </wsdl:message>

  <wsdl:message name="RequestedStateNotSupportedFault">
    <wsdl:part name="parameters"
      element="dmi-plain:RequestedStateNotSupportedFault" />
  </wsdl:message>

  <wsdl:message name="CustomFault">
    <wsdl:part name="parameters" element="dmi-plain:CustomFault" />
  </wsdl:message>

  <wsdl:message name="TransferProtocolNotInstantiatableFault">
    <wsdl:part name="parameters"
      element="dmi-plain:TransferProtocolNotInstantiatableFault" />
  </wsdl:message>
  <wsdl:message name="IncorrectStateFault">
    <wsdl:part name="parameters"
      element="dmi-plain:IncorrectStateFault" />
  </wsdl:message>

  <wsdl:message name="FailedStateTransitionFault">
    <wsdl:part name="parameters"
      element="dmi-plain:FailedStateTransitionFault" />
  </wsdl:message>
```

```
  <!-- ===================================== -->
  <!-- == Data Transfer Factory port type == -->
  <!-- ===================================== -->
  <wsdl:portType name="DataTransferFactory">
    <wsaw:UsingAddressing />

    <!-- getFactoryAttributesDocument -->
    <wsdl:operation name="GetFactoryAttributesDocument">
      <wsdl:input
        message="dmi-plain:GetFactoryAttributesDocumentRequest"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferFactory/GetFactoryAttributesDocumentRequest" />
      <wsdl:output
        message="dmi-plain:GetFactoryAttributesDocumentResponse"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferFactory/GetFactoryAttributesDocumentResponse" />
    </wsdl:operation>

    <!-- GetDataTransferInstance -->
    <wsdl:operation name="GetDataTransferInstance">
      <wsdl:input message="dmi-plain:GetDataTransferInstanceRequest"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferFactory/GetDataTransferInstanceRequest" />
      <wsdl:output message="dmi-plain:GetDataTransferInstanceResponse"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferFactory/GetDataTransferInstanceResponse" />
      <wsdl:fault name="UnsatisfiableRequestOptionsFault"
        message="dmi-plain:UnsatisfiableRequestOptionsFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="NoTransferProtocolAgreementFault"
        message="dmi-plain:NoTransferProtocolAgreementFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="CustomFault" message="dmi-plain:CustomFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
    </wsdl:operation>
  </wsdl:portType>
  <!-- ====================================== -->
  <!-- == Data Transfer Instance port type == -->
  <!-- ====================================== -->
  <wsdl:portType name="DataTransferInstance">
    <wsaw:UsingAddressing />

    <!-- getInstanceAttributesDocument -->
    <wsdl:operation name="GetInstanceAttributesDocument">
      <wsdl:input
        message="dmi-plain:GetInstanceAttributesDocumentRequest"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/GetInstanceAttributesDocumentRequest" />
      <wsdl:output
        message="dmi-plain:GetInstanceAttributesDocumentResponse"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/GetInstanceAttributesDocumentResponse" />
    </wsdl:operation>

    <!-- getStatus -->
    <wsdl:operation name="GetStatus">
      <wsdl:input message="dmi-plain:GetStatusRequest"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/GetStatusRequest" />
      <wsdl:output message="dmi-plain:GetStatusResponse"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/GetStatusResponse" />
    </wsdl:operation>

    <!-- start -->
    <wsdl:operation name="Start">
      <wsdl:input message="dmi-plain:StartRequest"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/StartRequest" />
      <wsdl:output message="dmi-plain:StartResponse"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/StartResponse" />
      <wsdl:fault name="IncorrectStateFault"
        message="dmi-plain:IncorrectStateFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="FailedStateTransitionFault"
        message="dmi-plain:FailedStateTransitionFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="TransferProtocolNotInstantiatableFault"
        message="dmi-plain:TransferProtocolNotInstantiatableFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="RequestedStateNotSupportedFault"
        message="dmi-plain:RequestedStateNotSupportedFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
```

```
/>
      <wsdl:fault name="CustomFault" message="dmi-plain:CustomFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
    </wsdl:operation>

    <!-- stop -->
    <wsdl:operation name="Stop">
      <wsdl:input message="dmi-plain:StopRequest"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/StopRequest" />
      <wsdl:output message="dmi-plain:StopResponse"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/StopResponse" />
<wsdl:fault name="FailedStateTransitionFault"
        message="dmi-plain:FailedStateTransitionFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
<wsdl:fault name="CustomFault" message="dmi-plain:CustomFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
    </wsdl:operation>

    <!-- resume -->
    <wsdl:operation name="Resume">
      <wsdl:input message="dmi-plain:ResumeRequest"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/ResumeRequest" />
      <wsdl:output message="dmi-plain:ResumeResponse"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/ResumeResponse" />
      <wsdl:fault name="IncorrectStateFault"
        message="dmi-plain:IncorrectStateFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="FailedStateTransitionFault"
        message="dmi-plain:FailedStateTransitionFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="RequestedStateNotSupportedFault"
        message="dmi-plain:RequestedStateNotSupportedFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="CustomFault" message="dmi-plain:CustomFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
    </wsdl:operation>

    <!-- suspend -->
    <wsdl:operation name="Suspend">
      <wsdl:input message="dmi-plain:SuspendRequest"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
```

```
aTransferInstance/SuspendRequest" />
      <wsdl:output message="dmi-plain:SuspendResponse"

wsaw:Action="http://schemas.ogf.org/dmi/2008/06/dmi/rendering/plain/Dat
aTransferInstance/SuspendResponse" />
      <wsdl:fault name="IncorrectStateFault"
        message="dmi-plain:IncorrectStateFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="FailedStateTransitionFault"
        message="dmi-plain:FailedStateTransitionFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="RequestedStateNotSupportedFault"
        message="dmi-plain:RequestedStateNotSupportedFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
      <wsdl:fault name="CustomFault" message="dmi-plain:CustomFault"
        wsaw:Action="http://www.w3.org/2005/08/addressing/soap/fault"
/>
    </wsdl:operation>
  </wsdl:portType>

</wsdl:definitions>
```