MULTILATERAL INTEROPERABILITY PROGRAMME (MIP)

MIP4 Information Exchange Specification (MIP4-IES)
Implementation Guidance

29 April 2021

DOCUMENT CONTROL

Change Management

Change Control Authority	MIP Integrated Product Team 4 (IPT4) Change Manager, webmaster@mip-interop.org
Release Authority	MIP Program Management Group (PMG), pmgchair@mip-interop.org
Latest Public Release	www.mip-interop.org | Public Document Library | 07-MIP4-IES
Latest MIP Release	www.mip-interop.org | Members Only site | CM Portal | MIP4 (account required)
Current Working Draft (for comments)	MIP4-IES Collaborative Environment (account required)

Version History

Version	Author	Date	Reason for Change
1.0.0	IPT4 CM	19 Apr 2018	MIP4.1 official release. Details of internal versions removed from Version History.
1.1.0	SEPT	20 Sep 2018	MIP4.2-CR1: updates applied during W67.
1.2.0	SEPT	14 Feb 2019	MIP4.2-CR3: updates applied during W68-69.
1.3.0	SEPT	11 Apr 2019	MIP4.2-CR4: updates applied during WG70.
1.4.0	IPT4 CM	21 Nov 2019	MIP4.3-CR2: applied CP72201v2
1.5.0	SEPT	29 Apr 2021	MIP 4.3.1-CR2 Updates during WG 71-78

CONTENTS

1 Overview

This document provides implementation guidance to organisations who are developing or maintaining a MIP4-IES interface. The information herein is intended to support and complement the core MIP4 specification, not to override it.

This document consists of the following sections:

a) General Advice
b) Developer Advice - General: a collection of helpful advice for developing a MIP4IES capability irrespective of the development platform used.
c) Developer Advice - .NET: a collection of helpful advice with the Microsoft .NET development framework, based on feedback from MIP members.
d) Developer Advice - Java: a collection of helpful advice with the Java development framework, based on feedback from MIP members.
e) Procedures: a description of any manual steps or routines that are recommended to accomplish typical scenarios related to MIP4-IES configuration and operation.
f) An Annex which refers to the Reference Implementations that can be used to get started quickly.

2 General Advice

2.1 Development approaches

Depending on the priorities of your development there are two general approaches:
Mapping-focussed: The aim here is to get the mapping of the C2IS objects to the BSO as complete and perfect as possible without focussing on the exchange. The suggested order in which to develop is then:

- File Exchange
- IDT-C2IS mapping
- Request-Response
- Publish-Subscribe

Exchange-focussed: The aim here is to implement all the exchange logic properly before worrying too much about the exact data that is exchanged. The suggested order in which to develop is then:

- File Exchange
- Request-Response
- Publish-Subscribe
- IDT-C2IS mapping.

Obviously the IDT-C2IS mapping could be developed in parallel to the full exchange stack. Either way the suggested order in which to implement the exchanges is as follows:

- File Exchange
- Request-Response
- Publish-Subscribe

2.2 Management Shared Concepts

MIP4IES allows concepts to reference other Concepts, either directly or through an association. An Implementation should ensure that all referenced Concepts can be requested as long as their references exist.
In a Concept is deleted:

- The Producer is strongly encouraged to explicitly delete all the associations to that Concept;

In case all the Concepts referencing (or referencing by association) a Concept are deleted:

- The Producer is strongly encouraged to explicitly delete the no longer referenced Concept;

2.3 Handling of Overlays

The systems should not expect to get a name element of overlays. If a consumer will get an overlay without a name element, it still should be able to process it. However the producer shall use the name element with overlays, even if it is not mandatory.

2.4 Limit the payload size

Systems are encouraged to implement mechanisms that may reduce the size of their payloads.

2.4.1 Producers

A system acting as a Producer should reduce the size of their messages by:

- Ensuring that HTTP Compression is used in the exchanges.
- Large information structures should prefer referencing over embedding content.
  NOTE: The nature of non-editable StaffConcepts like Organisation Structures prevent this approach.
  NOTE: This behavior should be governed by a configuration parameter like:
    - The maximum number of BattlespaceConcepts that may be embedded in an Overlay;
    - The maximum number of StaffConcepts that may be published on a single Topic.

2.4.2 Consumers

A system acting as a Producer should reduce the size of their messages by:

- Ensuring that HTTP Compression is used in the exchanges.
- When resolving references a maximum amount of identifiers is passed.
  NOTE: This behavior should be governed by a configuration parameter like:
    - The maximum number of references that can be included in a filter request.

2.5 Handling Implicit Deletes of StaffConcepts

The MIP4IES specification relias on an implicit deletion of StaffConcepts published on a Topic. This is accomplished by the “Synchronisation” R/R Request Operation executed directly following a P/S Subscribe Request. The basic concept means that any StaffConcept which is not returned by the Synchronisation is deleted from the Topic(s) provided in the Request.

This may lead to a ‘flickering’ of overlays available on the Topics after a Provider reboots its gateway, where the WSMP stack may be operational before the gateway has repopulated its data. If in this scenario, the Consumer Subscribes/Synchronizes before the data was populated, the data would not be provided in the Synchronisation message and removed from the consumers system, only to be reinserted once the Update Notifications have been received.

This may be undesirable due to the following reasons:

- A stateful gateway would delete information from a physical datastore and reinsert it later, depending on the datastore that may lead to substantial data pollution.
- Once data is deleted, the history may have been deleted with it.
- An operator may observe objects disappearing from its view and reappearing later.
- An operator may observe objects disappearing without reappearing, if an explicit action is required to make the reappeared StaffConcept visible again.

The following Advice should be followed to avoid the behaviour above:

Producer

If possible, a producer is strongly encouraged to defer accepting Subscriptions and Synchronisation events until the data is available and the StaffConcepts have been repopulated.

Consumer

A Consumer is encouraged to apply a ‘grace-period’ before actually performing an implicit delete of StaffConcepts based on a Synchronisation Response. The timeout could be different based on the systems that are providing the data, 10 minutes should be an appropriate default.
NOTE: This behavior should be governed by a configuration parameter like:

- The minimum number of minutes before implying deletes on StaffConcepts.

3 Developer Advice - General

This chapter contains advice for implementers starting out in developing a MIP4IES interface. Before starting to implement the full stack any such implementer is **strongly advised** to see if they can re-use otherwise available implementations of the stack. The implementation is quite time-consuming, and a lot of time and resources can be saved using an existing Library. Aside from the reference libraries provided by MIP Nations, commercial implementation may also be available.

This section contains advice/guidance that is generic, and not related to any platform (.NET, Java etc.)

3.1 Artifacts

Copy the following artifacts into a single folder in your workspace:

- All MIP4-IES Exchange Mechanism Artifacts (refer to section 3.3 in the *MIP4-IES Exchange Mechanism Overview* document and [REF-MIP-06]). These include the MIP4 information exchange XSDs, the WSMP WSDL, and WSMP message XSDs.
- The following referenced schema namespace XSD and WSDL definitions (refer to section 4 of the *MIP4-IES Exchange Mechanism Overview* document):
    - b-2.xsd
    - bf-2.xsd
    - br-2.xsd
    - brw-2.wsdl
    - bw-2.wsdl
    - r-2.xsd
    - rp-2.xsd
    - rpw-2.wsdl
    - rw-2.wsdl
    - t-1.xsd
    - ws-addr.xsd
    - xml.xsd
    - genericode.xsd

These artifacts will be used for generating web services interfaces and classes for MIP.

3.2 DateTime format in XML

For writing datetime to string in generated classes (e.g. `InitialTerminationTime` in `Subscribe`) use the same format as described in `Primitives.xsd`. The format should be `xsd:datetime` restricted with this pattern: `(.+\+00:00)|(.+Z)`.

3.3 Using SOAPUI

This section explains how SOAPUI can be used to test some WSMP related exchanges.

3.3.1 Download the WSMP Artifacts

The artifacts are actually contained in the WSMP specification document which (once ratified) can be downloaded from the NATO Standardisation Organisation (NSO) website.
NOTE: You may find locations hosting these files as a developer courtesy.
For example the **'MIP4supp' SVN repository**: where the files are included at the following path: "**artifacts\nmrr.nc3a.nato.int\rest\doc\NATO\DM\WSMP\1.3**".

The required files are:

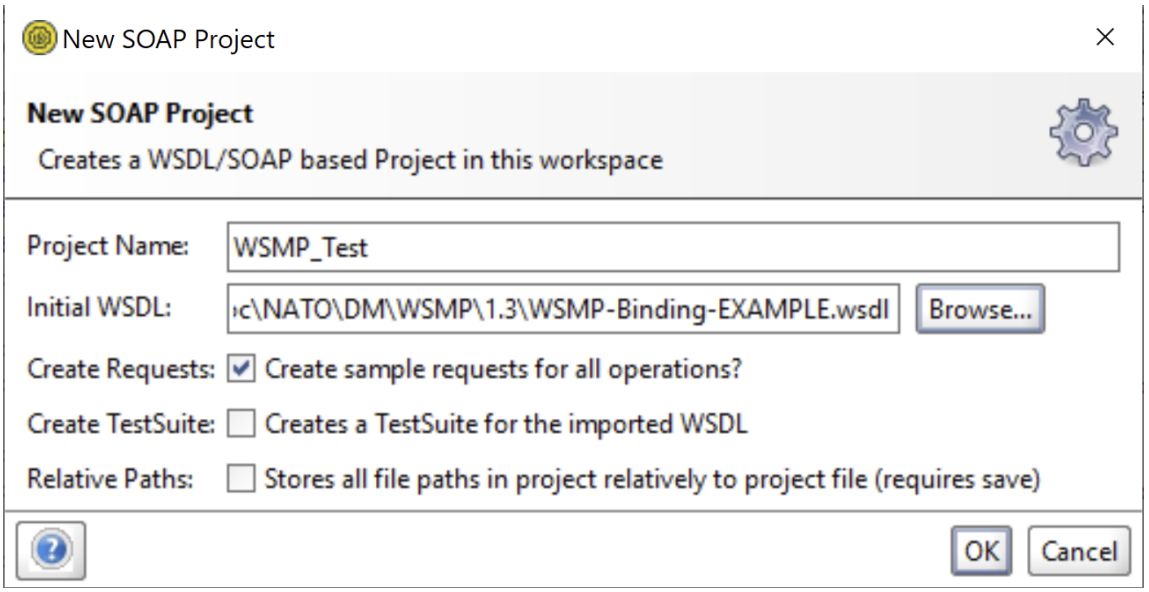
- **WSMP-Service-EXAMPLE.wsdl** *[non-normative]*
- **WSMP-Binding-EXAMPLE.wsdl** *[non-normative]*
- **WSMP-PortTypes.wsdl** *[normative]*
- **WSMP-Common.xsd** *[normative]*
- **WSMP-FaultMessages.xsd** *[normative]*
- **WSMP-ResourceMessages.xsd** *[normative]*
- **WSMP-SoapParameters.xsd** *[normative]*

3.3.2 Get SOAPUI

Goto the website and and install SOAPUI 5.3.0 or later

3.3.3 Create a new SOAP project

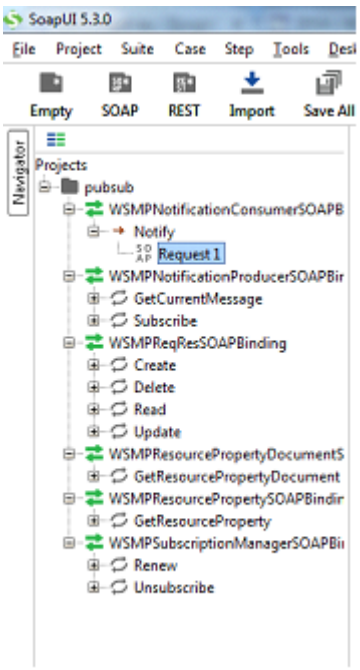
Click "Browse", elect a WSDL file that contains a WSMP service definition.



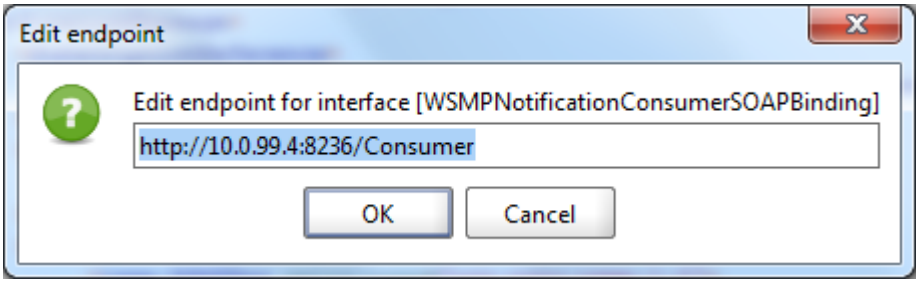
Click OK.

3.3.4 Test a Notify Message

Create a new Request for the Notify on the NotificationProducer endpoint.



Set the endpoint for your NotificationConsumer



Copy-paste a SOAP call on the request of SOAPUI and execute.
Example of a Notify SOAP call

XML Message

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsa:To env:mustUnderstand="true">http://localhost:8088/WSMPNotificationConsumer</wsa:To>
    <wsa:Action>http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify</wsa:Action>
    <wsa:MessageID>urn:uuid:77a99084-4aa5-445d-a2b8-dae01833ba6a</wsa:MessageID>
  </env:Header>
  <env:Body>
    <ns4:Notify xmlns:ns12="https://mip-interop.org/data/v4.3/Base"
                xmlns:ns14="https://mip-interop.org/data/v4.3/BattlespaceConcept"
                xmlns:ns15="https://mip-interop.org/data/v4.3/BattlespaceConcept/Location"
                xmlns:ns17="https://mip-interop.org/data/v4.3/Concept"
                xmlns:ns19="https://mip-interop.org/data/v4.3/Concept/Metadata"
                xmlns:ns24="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object"
                xmlns:ns28="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Actor/Organisation"
                xmlns:ns3="http://www.w3.org/2005/08/addressing"
                xmlns:ns38="https://mip-interop.org/data/v4.3/BattlespaceConcept/Affiliation"
                xmlns:ns4="http://docs.oasis-open.org/wsn/b-2"
                xmlns:ns52="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Actor/Organisation/Unit"
                xmlns:ns8="urn:nato:stanag:5644:wsmp:1:3">
```

```
<ns4:NotificationMessage>
  <ns4:SubscriptionReference>
    <ns3:Address>
      http://localhost:9091/services/Producer?id=subscriptions-192-168-2-120-17627cd08f2-0-0
    </ns3:Address>
  </ns4:SubscriptionReference>
  <ns4:Topic xmlns:xxxxx="https://insane.fkie.fraunhofer.de/models/datapool"
       Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple" xxxxx:ignore="me">
    xxxxx:SAD
  </ns4:Topic>
  <ns4:Message>
    <ns8:WSMPMsg>
      <ns8:Update>
        <ns8:Data Dialect="https://mip-interop.org/data/v4.3/Dialect" >
          <ns12:Context>
            <ns12:ContextIdentifier>/Overlay/3a9b275a-07d2-42df-a6c0-92394e46b19b/Content
            </ns12:ContextIdentifier>
            <ns12:Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:type="ns52:InfantryUnitType">
              <ns12:ID>6c8ced0a-359b-4b7d-8263-f8370578712c</ns12:ID>
              <ns17:ConceptName>Infantry_GER</ns17:ConceptName>
              <ns14:BattlespaceConceptGeographicLocation>
                <ns15:LocationGeometry xsi:type="ns15:AbsolutePointType">
                  <ns15:AbsolutePointLatitudeCoordinate>
                    <ns15:LatitudeCoordinateCoordinate>49.92
                    </ns15:LatitudeCoordinateCoordinate>
                  </ns15:AbsolutePointLatitudeCoordinate>
                  <ns15:AbsolutePointLongitudeCoordinate>
                    <ns15:LongitudeCoordinateCoordinate>8.50
                    </ns15:LongitudeCoordinateCoordinate>
                  </ns15:AbsolutePointLongitudeCoordinate>
                </ns15:LocationGeometry>
              </ns14:BattlespaceConceptGeographicLocation>
              <ns14:BattlespaceConceptHostilityCode>Hostile
              </ns14:BattlespaceConceptHostilityCode>
              <ns14:BattlespaceConceptMetadata>
                <ns19:ConceptMetadataCommentText>This is a Comment
                </ns19:ConceptMetadataCommentText>
                <ns19:ConceptMetadataReportingData>
                  <ns19:ReportingDataCategoryCode>Reported
                  </ns19:ReportingDataCategoryCode>
                  <ns19:ReportingDataObservationDateTime>2020-09-14T08:19:42.941Z
                  </ns19:ReportingDataObservationDateTime>
                  <ns19:ReportingDataReportingDateTime>2020-09-14T08:19:42.941Z
                  </ns19:ReportingDataReportingDateTime>
                  <ns19:ReportingDataReporter>
                    <ns19:ReporterName>reporter</ns19:ReporterName>
                  </ns19:ReportingDataReporter>
                </ns19:ConceptMetadataReportingData>
              </ns14:BattlespaceConceptMetadata>
              <ns14:BattlespaceConceptStaffCommentsText>shortcomment
              </ns14:BattlespaceConceptStaffCommentsText>
              <ns24:ObjectAffiliation xsi:type="ns38:GeopoliticalAffiliationType">
                <ns38:GeopoliticalAffiliationCode>DEU</ns38:GeopoliticalAffiliationCode>
              </ns24:ObjectAffiliation>
              <ns28:OrganisationHasCommandFunctionIndicator>true
              </ns28:OrganisationHasCommandFunctionIndicator>
              <ns28:MilitaryOrganisationAvailabilityCode>After30Days
              </ns28:MilitaryOrganisationAvailabilityCode>
              <ns28:MilitaryOrganisationServiceCode>Army
```

```
                        </ns28:MilitaryOrganisationServiceCode>
                        <ns24:ArmyEchelonCode>Battalion</ns24:ArmyEchelonCode>
                      </ns52:UnitEchelon>
                      <ns52:UnitTransportationCode>Mechanised</ns52:UnitTransportationCode>
                    </ns12:Data>
                    <ns12:ContextLastModificationDateTime>2020-09-15T08:28:04.094Z
                    </ns12:ContextLastModificationDateTime>
                  </ns12:Context>
                  </ns8:Data>
                </ns8:Update>
              </ns8:WSMPMsg>
            </ns4:Message>
          </ns4:NotificationMessage>
        </ns4:Notify>
      </env:Body>
    </env:Envelope>
```

4 Developer Advice - .NET

This section contains advice/guidance that is related to the Microsoft .NET platform.

4.1 .NET - Choose between reference library or manual implementation

MIP4 is building a .NET Reference Library for the MIP4-IES. The code and binaries may be used by any nation/implementer. For further information refer to Annex A.1.

4.1.1 Services (EMT):

Currently the implementation of the services in the reference library are not complete. Anyone that is able and willing may contribute to complete this.

It is also possible to generate all the services manually using the WSDL, as explained below.

4.1.2 Data structures (IDT/EMT):

The steps below describe how to generate all the classes manually. Note that there are many advantages of looking at the code provided by the reference library. Even if the services layer is generated manually the data structures could still be used.

4.2 .NET - First steps how to create the web services

4.2.1 Generate the web services interfaces and classes from the MIP4 artifacts

See section **3.1.1 Artifacts** for more information on the required artifacts and where to get them. Ensure all required artifacts are located in a single flat folder. Also check that any schemalocation attribute in the files points to that directory.

From a Visual Studio Command line, in the folder path, execute:

Command
`svcutil.exe /out:IMIP4Exchange.cs /ixt /s /language:c# *.wsdl *.xsd 1> _log.txt 2> _logErrors.txt`

NOTE: /ixt is shorthand /importXmlTypes which means “Configures the DataContract serializer to import non-DataContract types as `IXmlSerializable` types”.

Open the _logErrors.txt file after running the command to check if there were errors. Note: some errors are expected (e.g. `namespaces` already declared). If there are any errors referring to unreferenced schemas, add any missing artifacts as needed and re-run the command until those errors are cleared.

4.2.2 Renaming Code Classes, Properties, Interfaces

Please note it is Strongly advised to change the names (some of) the generated `classes`, `properties`

and `interfaces`, to be shorter, and conform to your Microsoft, (and your Company's) naming conventions. Take into account that the .NET attribute based serialization, uses the names of these code elements, unless the code-attributes contain a `(Type)Name` parameter. If this parameter is missing it must be inserted with the value to contain the name that was produced during generation.

Take special note with:

- `Properties`: If there is no `Xml.....` code-attribute the default of `XmlElement` is used by the serialization, this means that when renaming the property the `XmlElement` with the `Name` parameter must be inserted.
- `Types`: Aside from the `XmlType` attribute where the `TypeName` parameter may need to be added, also the `XmlRoot` attribute must be taken into account when renaming, possibly adding the `Name` parameter.
- `Interfaces`: Aside from the `XmlType` attribute where the `TypeName` parameter may need to be added, also the `ServiceContract` attribute must be taken into account when renaming, possibly adding the `Name` parameter.

4.2.3 Create a console application hosting MIP4 web services

Using Visual Studio, create a new project as a ConsoleApplication.
Add these references to the project:

- `System.Runtime.Serialization`
- `System.ServiceModel`

Add `IMIP4Exchange.cs`, the file generated using svcutil.exe, to the project.
Edit the file `IMIP4Exchange.cs`, add a `namespace` that includes all the code in this file.
Build. You will see compilation errors.

Comment out duplicated classes from `IMIP4Exchange.cs`

- CreatePullPoint
- DeleteResourceProperties
- InsertResourceProperties
- QueryResourceProperties
- RegisterPublisher
- UpdateResourceProperties

Add `XmlRootAttribute` attributes for each of the following `class` declarations, using the `element` name without the "Type" suffix for the following:

- TopicExpressionType,
- QueryExpressionType,
- TopicSetType,
- CodeListDocument,
- WSMPMsgType,
- MIP4FilterType,
- IdentifiableType,

E.g. for the class `TopicExpressionType`, add the following:

Command

```
[System.Xml.Serialization.XmlRootAttribute(ElementName = "TopicExpression", Namespace = "http://docs.oasis-open.org/wsn/b-2", IsNullable = false)]
```

Refactor the interfaces to start with an 'I', take special note to the rules in paragraph 3.3.2 Renaming Code Classes, Properties, Interfaces.

Create a new class named **`WsmpService`**. This class will implement the MIP4 interfaces:

C# Code

```
using System;

namespace MIP4IESTest
{
   /// <summary>
   /// Class WsmpService.
   /// </summary>
   public class WsmpService : MarshalByRefObject, IGetResourceProperty, IGetResourcePropertyDocument, INotificationConsumer, INotificationProducer, ISubscriptionManager, IWSMP, IDisposable
   {
……..
   }
}
```

Implement functions and methods for all the interfaces:
In Visual Studio, right-click on each interface and select "Implement Interface".

Create a new class named **`WsmpHost`** which will instantiate the **`ServiceHost`** and **`ServiceEndpoint`**:

C# Code

```
using System;
using System.ServiceModel;
using System.ServiceModel.Description;

namespace MIP4IESTest
{
   /// <summary>
   /// Class WsmpHost.
   /// </summary>
   /// <seealso cref="System.IDisposable" />
   public class WsmpHost : IDisposable
   {
      private readonly ServiceHost _serviceHost;
      private readonly WsmpService _instance;

      /// <summary>
      /// Initializes a new instance of the <see cref="WsmpHost"/> class.
      /// </summary>
      /// <param name="baseAddresses">An <see cref="T:System.Array" /> of type <see cref="T:System.Uri" /> that contains the base addresses for the hosted service.</param>
      public WsmpHost(params Uri[] baseAddresses)
      {
         _instance = new WsmpService();
         _serviceHost = new ServiceHost(_instance, baseAddresses);
         var binding = WcfHelper.CreateWsiBp20Binding();
         _serviceHost.CreateServiceEndpoint<IWSMP>("wsmp", binding);
         _serviceHost.CreateServiceEndpoint<INotificationConsumer>("consumer", binding);
         _serviceHost.CreateServiceEndpoint<INotificationProducer>("producer", binding);
         _serviceHost.CreateServiceEndpoint<ISubscriptionManager>("manager", binding);
```

```
            _serviceHost.CreateServiceEndpoint<IGetResourceProperty>("property", binding);
            _serviceHost.CreateServiceEndpoint<IGetResourcePropertyDocument>("document", binding);
            _serviceHost.Description.Behaviors.Add(new ServiceMetadataBehavior { HttpGetEnabled = true });
            _serviceHost.SetIncludeExceptionDetailInFaults(true);
            _serviceHost.Description.Behaviors.Add(new MessageInspectorBehavior());
            _serviceHost.Description.Name = "WSMP";
        }

        /// <summary>
        /// Starts this Service.
        /// </summary>
        public void Start()
        {
            _serviceHost.Open();
        }

        #region IDisposable
        protected virtual void Dispose(bool disposing)
        {
            if (disposing)
            {
                _instance?.Dispose();
                ((IDisposable)_serviceHost)?.Dispose();
            }
        }

        public void Dispose()
        {
            Dispose(true);
            GC.SuppressFinalize(this);
        }

        ~WsmpHost()
        {
            Dispose(false);
        }
        #endregion
    }
}
```

Add the following **`WcfHelper Class`**:

C# Code

```
using System;
using System.ServiceModel;
using System.ServiceModel.Channels;
using System.ServiceModel.Description;
using System.Text;
using System.Xml;

namespace MIP4IESTest
{
    /// <summary>
    /// Class Extensions.
    /// </summary>
    public static class WcfHelper
    {
        /// <summary>
        /// Creates a ServiceEndpoint on the specified ServiceHost for the specified type of ServiceInterface.
        /// </summary>
        /// <typeparam name="TServiceInterface">The type of Service to create an Endpoint for.</typeparam>
        /// <param name="host">The ServiceHost to create the Endpoint On.</param>
        /// <param name="addressSuffix">The suffix to be added to the ServiceHost's BaseAddress.</param>
        /// <param name="binding">The binding to be used.</param>
        /// <returns>The created ServiceEndpoint.</returns>
        public static ServiceEndpoint CreateServiceEndpoint<TServiceInterface>(this ServiceHost host, string addressSuffix, Binding
binding)
        {
```

```
    var serviceType = typeof(TServiceInterface);
    var serviceEndpoint = host.AddServiceEndpoint(serviceType, binding, $"/{addressSuffix}");
    serviceEndpoint.Name = serviceType.Name.Substring(1);
    serviceEndpoint.Behaviors.Add(new MustUnderstandBehavior(false));
    return serviceEndpoint;
}

/// <summary>
/// Creates a binding compatible with WS-I Basic Profile 2.0.
/// </summary>
/// <param name="timeout">The timeout. (20 minutes by default)</param>
/// <param name="maximumSize">The maximum size. (5MB as default)</param>
/// <returns>The Binding.</returns>
public static Binding CreateWsiBp20Binding(int timeout = 20 * 60 * 1000, int maximumSize = 5 * 1024 * 1024)
{
    var timing = TimeSpan.FromMilliseconds(timeout);
    var binding = new CustomBinding
    {
        ReceiveTimeout = timing,
        OpenTimeout = timing,
        CloseTimeout = timing,
        SendTimeout = timing,
        Name = "WS-I BP 2.0"
    };
    binding.Elements.Add(new TextMessageEncodingBindingElement
    {
        MessageVersion = MessageVersion.Soap12WSAddressing10,
        WriteEncoding = Encoding.UTF8,
        ReaderQuotas = { MaxNameTableCharCount = maximumSize }
    });
    binding.Elements.Add(new HttpTransportBindingElement { MaxReceivedMessageSize = maximumSize });
    return binding;
}

/// <summary>
/// Sets the 'IncludeExceptionDetailInFaults' setting on the specified ServiceHost.
/// </summary>
/// <param name="serviceHost">The ServiceHost.</param>
/// <param name="value">The value for the 'IncludeExceptionDetailInFaults'</param>
public static void SetIncludeExceptionDetailInFaults(this ServiceHost serviceHost, bool value)
{
    var serviceDebugBehavior = serviceHost.Description.Behaviors.Find<ServiceDebugBehavior>();
    if (serviceDebugBehavior == null)
    {
        serviceHost.Description.Behaviors.Add(new ServiceDebugBehavior { IncludeExceptionDetailInFaults = true });
    }
    else
    {
        serviceDebugBehavior.IncludeExceptionDetailInFaults = true;
    }
}

/// <summary>
/// Creates an EndpointReference with the specified URI and referenceParameters
/// </summary>
/// <param name="uri">The URI of the EndpointReference.</param>
/// <param name="referenceParameters">The reference parameters.</param>
/// <returns>The EndpointAddress for the EndpointReference.</returns>
public static EndpointAddress CreateEndpointAddress(Uri uri, params Tuple<XmlQualifiedName, object>[] referenceParameters)
{
    var addressBuilder = new EndpointAddressBuilder
    {
        Uri = uri
    };
    foreach (var referenceParameter in referenceParameters)
    {
        addressBuilder.Headers.Add(AddressHeader.CreateAddressHeader(referenceParameter.Item1.Name,
            referenceParameter.Item1.Namespace, referenceParameter.Item2));
    }
    return addressBuilder.ToEndpointAddress();
```

```
    }
  }
}
```

Create the **`Console Application`** below:

C# Code

```
using System;

namespace MIP4IESTest
{
  internal static class Program
  {
    private static void Main(string[] args)
    {
      try
      {
        using (var serviceHost = new WsmpHost(new Uri($"http://localhost:82")))
        {
          serviceHost.Start();
          Console.WriteLine("Running....");
          Console.ReadLine();
        }
      }
      catch (Exception ex)
      {
        Console.WriteLine(ex);
        Console.ReadLine();
      }
    }
  }
}
```

Once you see the "Running…." on the console, open a browser and try the following URL

- http://localhost:82

It works...congratulations, you now have your first MIP4 implementation !

4.2.4 Create a Service Endpoint for a Service

Action: Create a **`ServiceEndpoint`** for a specific Service:

C# Code

```
/// <summary>
/// Creates a ServiceEndpoint on the specified ServiceHost for the specified type of ServiceInterface.
/// </summary>
/// <typeparam name="TServiceInterface">The type of Service to create an Endpoint for.</typeparam>
/// <param name="host">The ServiceHost to create the Endpoint On.</param>
/// <param name="addressSuffix">The suffix to be added to the ServiceHost's BaseAddress.</param>
/// <param name="binding">The binding to be used.</param>
/// <returns>The created ServiceEndpoint.</returns>
private static ServiceEndpoint CreateServiceEndpoint<TServiceInterface>(this ServiceHost host, string addressSuffix, Binding binding)
{
  var serviceType = typeof(TServiceInterface);
  var serviceEndpoint = host.AddServiceEndpoint(serviceType, binding, $"/{addressSuffix}");
  serviceEndpoint.Name = serviceType.Name.Substring(1);
  serviceEndpoint.Behaviors.Add(new MustUnderstandBehavior(false));
  return serviceEndpoint;
}
```

4.2.5 Create a WS-I BP 2.0 Binding

Problem: the .NET Framework doesn't provide a binding compatible with WS-I Basic Profile 2.0
Solution: create a custom binding. See the method below that creates the **`Binding`**.

C# Code

```
/// <summary>
/// Creates a binding compatible with WS-I Basic Profile 2.0.
/// </summary>
/// <param name="timeout">The timeout. (20 minutes by default)</param>
/// <param name="maximumSize">The maximum size. (5MB as default)</param>
/// <returns>The Binding.</returns>
public static Binding CreateWsiBp20Binding(int timeout = 20 * 60 * 1000, int maximumSize = 5 * 1024 * 1024)
{
   var timing = TimeSpan.FromMilliseconds(timeout);
   var binding = new CustomBinding
   {
      ReceiveTimeout = timing,
      OpenTimeout = timing,
      CloseTimeout = timing,
      SendTimeout = timing,
      Name = "WS-I BP 2.0"
   };
   binding.Elements.Add(new TextMessageEncodingBindingElement
   {
      MessageVersion = MessageVersion.Soap12WSAddressing10,
      WriteEncoding = Encoding.UTF8,
      ReaderQuotas = { MaxNameTableCharCount = maximumSize }
   });
   binding.Elements.Add(new HttpTransportBindingElement { MaxReceivedMessageSize = maximumSize });
   return binding;
}
```

4.2.6 Expose Exception Details in Faults

Problem: when implementing a system it is sometimes difficult to get details of problems that occur on the `ServiceHost` when processing requests from Clients.
Solution: the host can be configured to send exception details as part of the responses to the client. This can be accomplished by configuring a `ServiceDebugBehavior`. **Note:** it may be advisable to disable this functionality as the system is deployed. When the code block below is included in an extension class the method `'SetIncludeExceptionDetailInFaults'` can be used on a Service host to enable/disable the exposing of `Exception` Details.

C# Code

```
/// <summary>
/// Sets the 'IncludeExceptionDetailInFaults' setting on the specified ServiceHost.
/// </summary>
/// <param name="serviceHost">The ServiceHost.</param>
/// <param name="value">The value for the 'IncludeExceptionDetailInFaults'</param>
public static void SetIncludeExceptionDetailInFaults(this ServiceHost serviceHost, bool value)
{
   var serviceDebugBehavior = serviceHost.Description.Behaviors.Find<ServiceDebugBehavior>();
   if (serviceDebugBehavior == null)
   {
      serviceHost.Description.Behaviors.Add(new ServiceDebugBehavior { IncludeExceptionDetailInFaults = true });
   }
   else
   {
      serviceDebugBehavior.IncludeExceptionDetailInFaults = true;
   }
}
```

4.2.7 Finding namespaces for prefixes

Problem: sometimes systems may not declare the namespace prefixes on the elements that use them. For instance if an XPath is received as part of one of the SOAPmessages, the prefix used may be declared on the soap:envelope element.
Solution: in this case the only way to access the whole message that was received as the source of the operation is through the Current Operation Context, see the statement below.

```
C# Code
var wholeMsg = OperationContext.Current.RequestContext.RequestMessage.ToString();
```

NOTE: Be careful, depending on where you are in the document the prefix may represent a different Xml Namespace.

4.3 .NET - Manually Generate the IDT model classes using XSD.exe

As mentioned in the beginning of this chapter an implementer has the choice to manually generate the classes for the MIP4 data schema, instead of using the .NET Reference Library. This paragraph provides some advice on how to proceed with this.

4.3.1 Generate C# classes using xsd.exe

Create a batch file to call **xsd.exe**. There are a lot of parameters to provide and this will help to avoid typos or mistakes when launching the generation command.

Here is an example of a batch file command. This example generates the classes for the MIP4 service types only. This could, for example, be used in combination with the data classes provided by the .NET reference library, so that full manual generation of all the classes is not required.

The full list of files for the MIP4 XSD data type artifacts can be found in Section 8 of the MIP4-IES Information Definition Overview document and [REF-MIP-06]). The batch file can refer to the directory paths as structured in the released artifacts, or you can copy all of the necessary files into a single flat folder.

```
Command
xsd.exe /l:CS /n:MIP4SchemasNS /nologo /c AppInfo.xsd Base.xsd Common.xsd Extension.xsd
Primitives.xsd .\Filter.xsd 1> _log.txt 2> _logErrors.txt
```

Note: Notice the last XSD file in the list is prepended with a relative directory path, e.g. "**.\File.xsd**". This is the way to control the output filename. Using the example of "**.\File.xsd**", the output file would be "**File.cs**". Otherwise, the tool generates a filename that combines the names of ALL the input files. I.e. if you call **xsd.exe** using a list of parameters: "**xsd.exe /l:CS /n:MIPSchemasNS /nologo /c Schema1.xsd Schema2.xsd Schema3.xsd**", the output file name would be Schema1_Schema2_Schema3.cs. Considering the number of XSD files included in the MIP4 release, the output file generated would result in an error because the file name would be too long for Windows.

Launch the batch file to run **`xsd.exe`**. Once completed, check the **`_logErrors.txt`** file for any errors. If necessary, adjust the inputs to the XSD command and rerun the batch file. Finally, rename the output code file as needed.

NOTE: If you generate the WSDL using **`svcutil.exe`** manually there will be duplication of classes since the EMT also generates classes from IDT. This overlap needs to be resolved. Note you'll have several classes suffixed with a number (e.g. Metadata1) since there are multiple MetaData elements. When cleaning up duplicates ensure that you remove the correct classes.

4.4 .NET - Geometry conversion samples from JC3IEDM to MIP 4.3+

4.5 .NET - Generating XmlSerializers using sgen.exe.

Note: This improves run-time performance!

4.6 .NET - XmlSerializer out of memory issue

See

Online Documentation

https://blogs.msdn.microsoft.com/asiatech/2013/10/22/case-study-outofmemory-exception-caused-by-xmlserializer-is-not-used-properly/

4.6.1 Solution 1:

Cache the **`XmlSerializer`** in order to instantiate only one instance for each **`type`**.

C# Code

```
private static readonly Dictionary<string, XmlSerializer> cache =
                            new Dictionary<string, XmlSerializer>();
…

private static XmlSerializer GetSerializer(Type type)
{
   if (!cache.ContainsKey(type.Name))
   {
      cache.Add(type.Name, new XmlSerializer(type));
   }
   return cache[type.Name];
}
```

NOTE: This is only a partial solution as each time an **`XmlSerializer`** is created for the first time a performance hit is taken.

4.6.2 Solution 2

Cache the XmlSerializer in order to instantiate only one instance for each type.

Online Documentation

https://docs.microsoft.com/en-us/dotnet/standard/serialization/xml-serializer-generator-tool-sgen-exe

- Create a new class library project containing only the C# file generated by XSD.exe.
  - For this example we will name this assembly MIP4Schemas.dll
- Compile the MIP4Schemas.dll
- Using a Visual Studio command line execute the following in the same directory where the assembly MIP4Schemas.dll is located.

Command

```
"C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin\sgen.exe" /keep /debug /force
/verbose /c:"/platform:x86" MIP4Schemas.dll 1> _log.txt 2> _logError.txt
```

NOTE: It will look like nothing is happening but it works. This process can last for about 10 to 15 hours due to the large number of classes.

- sgen.exe will generate a class library assembly containing specific serializers for each class from MIP4Schemas.dll. Using the MIP4Schemas.dll, the assembly generated will be named MIP4Schemas.XmlSerializers.dll
- Use those specific serializers instead of using XmlSerializer.
  - Instantiation of those serializers is milliseconds compared to XmSerializer which can take nearly a minute to instantiate.

4.7 .NET - Support of HTTP and HTTPS

4.7.1 Client-side

Use HttpTransportBindingElement or HttpsTransportBindingElement

4.7.2 Server-side

Online Documentation

https://docs.microsoft.com/en-us/dotnet/framework/wcf/feature-details/how-to-configure-an-iis-hosted-wcf-service-with-ssl

4.8 .NET - Handling Endpoint References

The `EndPoint URL` of a Web Service is retrieved by evaluating the `WSDLs` exposed.
An `EndPointReference` can contain different types of `URI` for the address attribute:

- A `URN` like ‘nation:c2is:mip4:subscription’;
- A `URL` that is **different from** the `URL` the service is exposed at;
- A `URL` that is identical to the `URL` the service is exposed at.

In the first two cases `WCF` needs to 'know' the `URL` the service is exposed at, this can be accomplished by adding the `ClientViaBehaviour` to the `Behaviours` of a `WcfClient`.

Code

```
using (SubscriptionManagerClient manager = new SubscriptionManagerClient(binding, subscriptionReference.Endpoint))
{
    if (providerUsingWSResource)
        manager.Endpoint.Behaviors.Add(new System.ServiceModel.Description.ClientViaBehavior(FixedEndpoint.Uri));

    await manager.UnsubscribeAsync(new Unsubscribe());
}
```

5 Developer Advice - Java

This section contains advice/guidance that is related to the Java platform.

5.1 Java - First steps how to create the web services

In the Java ecosystem there are multiple ways to deal with web service creation. Depending on the build tool, plugins might have different options and characteristics

5.1.1 Generate the web services interfaces and classes from the MIP4 artifacts

MIP provides XSD and WSDL files for generating classes and services. See section **3.1.1 Artifacts** for more information on artifacts and where to get them.

5.1.2 Generate MIP4 class libraries

For generating the MIP4 library only XSD (XML Schema Definition) files are needed.
There are different libraries for generating java classes from XML. The most popular is probably an implementation of JAXB supported by eclipse.
For jaxb to work it needs to resolve the schemaLocations in the XSDs. This can be achieved either by editing the XSDs manually or by using a catalog..The latter is described in further detail elsewhere.[1].

A binding.xml file is needed with the entry: <jxb:globalBindings
typesafeEnumMaxMembers="2000"/>
Without xjc will throw an EnumMemberSizeCap error.

Command
`xjc -b binding.xml -encoding UTF-8 IES.xsd`

After this command the MIP4 class library should be available.

[1] http://www.sagehill.net/docbookxsl/WriteCatalog.html

5.1.3 Generate web service libraries

The web service libraries are a bit more elaborate to generate. Libraries based on both XSD and WSDL (Web Service Definition Language) have to be generated. It is necessary to do the same schemaLocation changes to these files as it was with files for generating the MIP4 class libraries.

Commands	
MIP4-IES Filtering Profile	`xjc -encoding UTF-8 Filter.xsd`
MIP4-IES Common	`xjc -encoding UTF-8 Common.xsd`
WSMP Common	`xjc -encoding UTF-8 WSMP-Common.xsd`
WSMP Resources	`xjc -encoding UTF-8 WSMP-ResourceMessages.xsd`
WSMP Soap parameters	`xjc -encoding UTF-8 WSMP-SoapParameters.xsd`
WS Addressing	`xjc -encoding UTF-8 ws-addr.xsd`
WS Base Notification	`xjc -encoding UTF-8 b-2.xsd`
WS Resource Framework	`xjc -encoding UTF-8 r-2.xsd & xjc -encoding UTF-8 p-2.xsd`
WS Topics	`xjc -encoding UTF-8 t-1.xsd`
WSMP services	`wsdl2java -catalog catalog.xml WSMP-Services-example.wsdl`

When these commands have been executed, the web service library should be available.

5.1.4 Getting started with JAXB

Now that libraries have been generated one can start using them. The code snippet below reads a XML file with a WSMPmessage. This is probably one of the first steps one would like to achieve if one has chosen to go for the Exchange-focused approach described in **section 2.1**.

Java Code

```
public class WSMPMsgTest {

   @Test
   public void testThatAWsmpMsgCanBeDeserializedFromResource() throws FileNotFoundException, JAXBException
{
      FileInputStream fileStream = new FileInputStream("src/test/resources/DNK-LandPicture.xml");
      JAXBContext jaxbContext = JAXBContext.newInstance(WSMPMsgType.class);
      Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();
      @SuppressWarnings("unchecked")
      JAXBElement<WSMPMsgType> unitWrapperForUnmarshalling = (JAXBElement<WSMPMsgType>)
unmarshaller.unmarshal(fileStream);
      WSMPMsgType unit = unitWrapperForUnmarshalling.getValue();
      Assert.assertNotNull(unit);

      nato.stanag._5644.wsmp._1._3.ObjectFactory objectFactory = new nato.stanag._5644.wsmp._1._3.ObjectFactory();
      JAXBElement<WSMPMsgType> unitWrapperForMarshalling = objectFactory.createWSMPMsg(unit);
```

```
        StringWriter stringWriter = new StringWriter();
        Marshaller marshaller = jaxbContext.createMarshaller();
        marshaller.marshal(unitWrapperForMarshalling, stringWriter);
        String unitXml = stringWriter.toString();
        Assert.assertNotNull(unitXml);
    }
}
```

The XML file below is used in combination with the code snippet above.
This WSMPMsg includes one layer with one unit in it.

XML Message

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns1:WSMPMsg
xmlns:ns6="urn:nato:stanag:5644:wsmp:1:3:resources"
xmlns:ns5="http://docs.oasis-open.org/wsn/t-1" xmlns:ns8="http://docs.oasis-open.org/wsrf/rp-2"
xmlns:ns7="urn:nato:stanag:5644:wsmp:1:3:fault"
xmlns:ns9="https://mip-interop.org/data/v4.3/Extension"
xmlns:ns12="https://mip-interop.org/exchange/v4.3/Common"
xmlns:ns11="https://mip-interop.org/data/v4.3/AppInfo"
xmlns:ns10="https://mip-interop.org/data/v4.3/Base"
xmlns:ns2="http://docs.oasis-open.org/wsn/b-2" xmlns:ns1="urn:nato:stanag:5644:wsmp:1:3"
xmlns:ns4="http://www.w3.org/2005/08/addressing"
xmlns:ns3="http://docs.oasis-open.org/wsrf/bf-2"><ns1:Create><ns1:Data
ns1:Dialect="https://mip-interop.org/data/v4.3/Dialect"><ns2:Context
xmlns:ns2="https://mip-interop.org/data/v4.3/Base"
xmlns:ns1="https://mip-interop.org/data/v4.3/Extension"
xmlns:ns4="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
xmlns:ns3="https://mip-interop.org/data/v4.3/Concept"
xmlns:ns31="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Materiel/Equipment"
xmlns:ns6="https://mip-interop.org/data/v4.3/AppInfo"
xmlns:ns30="https://mip-interop.org/data/v4.3/StaffConcept/OrganisationStructure"
xmlns:ns5="https://mip-interop.org/data/v4.3/Concept/Metadata"
xmlns:ns8="https://mip-interop.org/data/v4.3/BattlespaceConcept"
xmlns:ns7="https://mip-interop.org/data/v4.3/StaffConcept/InformationGroup"
xmlns:ns13="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Materiel"
xmlns:ns35="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Materiel/ConsumableM
ateriel" xmlns:ns12="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Actor"
xmlns:ns34="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Feature/GeographicFeat
ure/Vegetation" xmlns:ns9="https://mip-interop.org/data/v4.3/BattlespaceConcept/Location"
xmlns:ns11="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object"
xmlns:ns33="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Materiel/Equipment/Ves
sel" xmlns:ns10="https://mip-interop.org/data/v4.3/BattlespaceConcept/Address"
xmlns:ns32="https://mip-interop.org/data/v4.3/BattlespaceConcept/Mobility"
xmlns:ns17="https://mip-interop.org/data/v4.3/BattlespaceConcept/Action"
xmlns:ns39="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Materiel/Equipment/We
apon"
xmlns:ns16="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Actor/Organisation"
xmlns:ns38="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Materiel/Equipment/Veh
icle" xmlns:ns15="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Facility"
xmlns:ns37="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Materiel/Equipment/Sen

sor" xmlns:ns14="https://mip-interop.org/data/v4.3/BattlespaceConcept/Generic"
xmlns:ns36="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Materiel/UnexplodedOrdnance"
xmlns:ns19="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Feature/GeographicFeature" xmlns:ns18="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Feature"
xmlns:ns20="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Feature/ControlFeature"
xmlns:ns42="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Actor/Organisation/Unit"
xmlns:ns41="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Actor/Organisation/MilitaryPost"
xmlns:ns40="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Actor/Organisation/MilitaryOrganisationReadiness" xmlns:ns24="https://mip-interop.org/data/v4.3/StaffConcept/Metadata"
xmlns:ns23="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/InformationResource"
xmlns:ns22="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Actor/Person"
xmlns:ns21="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Feature/MeteorologicFeature"
xmlns:ns43="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/Facility/CulturalPropertyProtectionSite"
xmlns:ns28="https://mip-interop.org/data/v4.3/BattlespaceConcept/Object/NetworkService"
xmlns:ns27="https://mip-interop.org/exchange/v4.3/Common"
xmlns:ns26="https://mip-interop.org/data/v4.3/StaffConcept/Overlay"
xmlns:ns25="https://mip-interop.org/data/v4.3/StaffConcept"
xmlns:ns29="https://mip-interop.org/data/v4.3/BattlespaceConcept/Affiliation"><ns2:ContextIdentifier>/Overlay</ns2:ContextIdentifier><ns2:Data xsi:type="ns26:OverlayType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><ns2:ID>bcb7e1f6-787d-35f4-9612-14101c8942e4</ns2:ID><ns3:ConceptName>LandPicture</ns3:ConceptName><ns25:StaffConceptMetadata><ns5:ConceptMetadataAlternativeConfidentialityLabel
xsi:nil="true"/><ns5:ConceptMetadataAppraisal xsi:nil="true"/><ns5:ConceptMetadataCorrelation
xsi:nil="true"/><ns5:ConceptMetadataMetadataConfidentialityLabel
xsi:nil="true"/><ns5:ConceptMetadataOriginatorConfidentialityLabel
xsi:nil="true"/><ns5:ConceptMetadataReportingData
xsi:nil="true"/><ns24:StaffConceptMetadataIsAcknowledgementRequiredIndicator
xsi:nil="true"/><ns24:StaffConceptMetadataIssuingDateTime>2020-12-01T14:28:40.813Z</ns24:StaffConceptMetadataIssuingDateTime><ns24:StaffConceptMetadataLifecycleCode
xsi:nil="true"/><ns24:StaffConceptMetadataVersionValue
xsi:nil="true"/><ns24:StaffConceptMetadataOriginator><ns5:OriginatorName>DNK</ns5:OriginatorName><ns5:OriginatorActorReference
xsi:nil="true"/></ns24:StaffConceptMetadataOriginator></ns25:StaffConceptMetadata><ns26:OverlayIsUncorrelatedIndicator xsi:nil="true"/><ns26:OverlayContent
xsi:type="ns42:GroundBasedAirDefenceUnitType"><ns2:Amplification><ns2:AmplificationLocationText>MilitaryOrganisationServiceCodeType</ns2:AmplificationLocationText><ns2:AmplificationText>Missing</ns2:AmplificationText></ns2:Amplification><ns2:Amplification><ns2:AmplificationLocationText>Boolean</ns2:AmplificationLocationText><ns2:AmplificationText>Missing</ns2:AmplificationText></ns2:Amplification><ns2:ID>07348dc6-51a8-37be-b0ee-11c02ca73ef6</ns2:ID><ns3:ConceptName>Test Unit</ns3:ConceptName><ns8:BattlespaceConceptIsTypeIndicator
xsi:nil="true"/><ns8:BattlespaceConceptTypeReference
xsi:nil="true"/><ns8:BattlespaceConceptAdditionalInformationText
xsi:nil="true"/><ns8:BattlespaceConceptGeographicLocation><ns9:LocationGeometry
xsi:type="ns9:AbsolutePointType"><ns9:AbsolutePointLatitudeCoordinate><ns9:LatitudeCoordinateCoordinate>55.28617</ns9:LatitudeCoordinateCoordinate><ns9:LatitudeCoordinatePrecisionC

ode
xsi:nil="true"/></ns9:AbsolutePointLatitudeCoordinate><ns9:AbsolutePointLongitudeCoordinate>
<ns9:LongitudeCoordinateCoordinate>10.39388</ns9:LongitudeCoordinateCoordinate><ns9:Long
itudeCoordinatePrecisionCode
xsi:nil="true"/></ns9:AbsolutePointLongitudeCoordinate><ns9:AbsolutePointVerticalDistance
xsi:nil="true"/></ns9:LocationGeometry><ns9:LocationHorizontalAccuracyDimension
xsi:nil="true"/><ns9:LocationMeaningCode xsi:nil="true"/><ns9:LocationName
xsi:nil="true"/><ns9:LocationVerticalAccuracyDimension
xsi:nil="true"/></ns8:BattlespaceConceptGeographicLocation><ns8:BattlespaceConceptHostilityC
ode>Friend</ns8:BattlespaceConceptHostilityCode><ns8:BattlespaceConceptMetadata><ns5:Conc
eptMetadataAlternativeConfidentialityLabel xsi:nil="true"/><ns5:ConceptMetadataAppraisal
xsi:nil="true"/><ns5:ConceptMetadataCorrelation
xsi:nil="true"/><ns5:ConceptMetadataMetadataConfidentialityLabel
xsi:nil="true"/><ns5:ConceptMetadataOriginatorConfidentialityLabel
xsi:nil="true"/><ns5:ConceptMetadataReportingData><ns5:ReportingDataCategoryCode>Reporte
d</ns5:ReportingDataCategoryCode><ns5:ReportingDataIsBasedOnCountIndicator
xsi:nil="true"/><ns5:ReportingDataIsRealDataInExerciseIndicator
xsi:nil="true"/><ns5:ReportingDataObservationDateTime>2020-12-01T14:28:30.682Z</ns5:Repor
tingDataObservationDateTime><ns5:ReportingDataReportingDateTime>2020-12-01T14:28:30.68
2Z</ns5:ReportingDataReportingDateTime><ns5:ReportingDataObserver
xsi:nil="true"/><ns5:ReportingDataReporter><ns5:ReporterAddress
xsi:nil="true"/><ns5:ReporterName>Org4810</ns5:ReporterName><ns5:ReporterActorReference
xsi:nil="true"/></ns5:ReportingDataReporter></ns5:ConceptMetadataReportingData></ns8:Battles
paceConceptMetadata><ns8:BattlespaceConceptStaffCommentsText
xsi:nil="true"/><ns11:ObjectIsBoobyTrappedIndicator
xsi:nil="true"/><ns11:ObjectIsDecoyIndicator xsi:nil="true"/><ns11:ObjectOrientation
xsi:nil="true"/><ns11:ObjectTypeName xsi:nil="true"/><ns11:ObjectVelocity
xsi:nil="true"/><ns11:ObjectAssignedEstablishmentRef
xsi:nil="true"/><ns11:ObjectAssignedEstablishment
xsi:nil="true"/><ns16:OrganisationCbrnDressStateCode
xsi:nil="true"/><ns16:OrganisationDescriptionText
xsi:nil="true"/><ns16:OrganisationHasCommandFunctionIndicator
xsi:nil="true"/><ns16:OrganisationOperationalStatusCode>Operational</ns16:OrganisationOperati
onalStatusCode><ns16:OrganisationOperationalStatusQualifierCode
xsi:nil="true"/><ns16:OrganisationRadiationDoseMeasure
xsi:nil="true"/><ns16:GovernmentOrganisationMainActivityCode
xsi:nil="true"/><ns16:MilitaryOrganisationAvailabilityCode
xsi:nil="true"/><ns16:MilitaryOrganisationCommandAndControlCategoryCode
xsi:nil="true"/><ns16:MilitaryOrganisationCommandAndControlRoleCode
xsi:nil="true"/><ns16:MilitaryOrganisationEmissionControlCode
xsi:nil="true"/><ns16:MilitaryOrganisationFireModeCode
xsi:nil="true"/><ns16:MilitaryOrganisationIdentifier
xsi:nil="true"/><ns16:MilitaryOrganisationIsCommittedIndicator
xsi:nil="true"/><ns16:MilitaryOrganisationIsInActionIndicator
xsi:nil="true"/><ns16:MilitaryOrganisationIsInReserveIndicator
xsi:nil="true"/><ns16:MilitaryOrganisationReadiness
xsi:nil="true"/><ns16:MilitaryOrganisationReadinessDuration
xsi:nil="true"/><ns16:MilitaryOrganisationServiceCode
xsi:nil="true"/><ns16:MilitaryOrganisationTrainingLevelCode xsi:nil="true"/><ns42:UnitEchelon
xsi:type="ns11:ArmyEchelonType"><ns11:ArmyEchelonCode

xsi:nil="true"/></ns42:UnitEchelon><ns42:UnitEmploymentCaveatText
xsi:nil="true"/><ns42:UnitEquipmentScalingCode
xsi:nil="true"/><ns42:UnitEquipmentTeardownDuration
xsi:nil="true"/><ns42:UnitFormalAbbreviatedName xsi:nil="true"/><ns42:UnitGeneralMobility
xsi:nil="true"/><ns42:UnitHigherFormationName xsi:nil="true"/><ns42:UnitIsArmouredIndicator
xsi:nil="true"/><ns42:UnitIsNotionalIndicator xsi:nil="true"/><ns42:UnitIsTaskForceIndicator
xsi:nil="true"/><ns42:UnitPersonnelStrength xsi:nil="true"/><ns42:UnitReinforcementCode
xsi:nil="true"/><ns42:UnitSupplementarySpecialisationCode
xsi:nil="true"/><ns42:UnitTransportationCode
xsi:nil="true"/><ns42:UnitUsesUnmannedVehiclesIndicator
xsi:nil="true"/><ns42:UnitSupportedMilitaryOrganisationRef
xsi:nil="true"/><ns42:UnitSupportedMilitaryOrganisation
xsi:nil="true"/><ns42:GroundBasedAirDefenceUnitAirDefenceAltitudeCode
xsi:nil="true"/><ns42:GroundBasedAirDefenceUnitOperatingLevelCode
xsi:nil="true"/><ns42:GroundBasedAirDefenceUnitRangeCode
xsi:nil="true"/></ns26:OverlayContent></ns2:Data><ns2:ContextLastModificationDateTime>202
0-12-01T14:06:06.733Z</ns2:ContextLastModificationDateTime></ns2:Context></ns1:Data></ns
1:Create></ns1:WSMPMsg>

Documentation reference

For more information on Exchange of information see the document *MIP4-IES_File_Exchange_Pattern_v*.pdf* in the mip subversion. For information on publish/subscribe and request/response see documents *MIP4-IES_Publish-Subscribe_Exchange_Pattern_v*.pdf* and *MIP4-IES_Request-Response_Exchange_Pattern_v1.*.pdf*.

Implementation reference

A Java reference implementation is available through the MIP community. See further details on the SEPT page[2]..

[2] https://groups.google.com/g/sept-mip4-ies

Annex-A Reference Implementations

This annex provides an overview of the available Reference Libraries, their capabilities, and a brief description on their API, including samples on how they should be used.

A.1 Microsoft .NET

This section describes the Reference Libraries that have been made available to assist development in the .NET framework.

A.1.1 Acquiring the Reference Libraries

Links to the .NET Reference implementation:
Note, the access controlled areas require a microsoft ID, for access, contact fabian boumeester.

- project (access controlled)
- code (access controlled)
- nuget packages (access controlled)
- sonarqube (open)

A.1.2 Capabilities

Some of the features of the Reference Library are:

- Code generation process:
  - Initial code is generated by the standard svcutil.exe and/or xsd.exe
  - The few large generated files split-up following .NET Coding Standards.
  - Classes, Interfaces, Properties, Fields etc are renamed using .NET Coding Standards.
  - The XmlSerialization attributes are corrected, where required.
  - XmlRootAttribute removed from Types with multiple top-level-elements.
  - XmlIncludeAttributes removed in favor of another Serialization Mechanism
  - Classes/Interfaces/Enumerations are distributed over Namespaces.
  - Known issues of the Microsoft tooling are corrected
    - SubstitutionGroups on Substituting elements are now detected.
  - WS-Addressing Actions are corrected for OASIS services.
  - SGen is executed during the build process, *(no longer takes more than 6 hours)*
  - Nuget Packages are created during the build process, and pushed to a nuget server.
  - Code Quality is guarded by a sonarqube server.
- Code is maintained and improved by the community.
  - Contributions/Feedback are welcome and encouraged
- Code updated with MIP4-IES releases.
- Code for both 4.3 and 4.4 (latest CR)
- Implicit casts for value types reducing the code required to assign values.
- .NET Attributes attached to most Classes and Properties which reflect the AppData provided in the IDT Xml Schema's (CR3)

- Specialized serialization for classes with an XmlAny.
- Factory Methods for Identifiables based on SemanticId and XmlQualifiedName (CR3)
- A growing set of UnitTests that cover a range of functionality
- Current version using .NET 4.0 for WindowsXP compatibility.
- Contains additional classes for helpful standards (Among others)
    - STANAG 4774: Confidentiality Labelling for Information Sharing
    - STANAG 4778: Metadata Binding Mechanism
    - Xml-SPIF: Security Policy Information File (Security validation)
    - KML: Keyhole Markup Language (Filter)
- Genericode: OASIS Code List Representation (Extended Lists)
- Pluggable WSMP engine with a plugin for the MIP4 profile. (CR4)

A.1.3 API Documentation

TBD

A.2 Java

This section describes the Reference Libraries that have been made available to assist development in the Java framework.

A.2.1 Acquiring the Reference Libraries

Links to the java Reference implementation:
Note, the access controlled areas require registration and approval.

- Zip Archive on MIM-World (access controlled)

A.2.2 Capabilities

TBD

A.2.3 API Documentation

TBD