**OpenO&M™ CIR Specification**

**A Common Interoperability Registry**

**V1.0**

This document defines the Open O&M Common Interoperability Registry (CIR). It defines the underlying object model, an XML implementation of the model, and services for a registry. The services are based on the common transaction models defined in OAGIS and B2MML and specified in the ANSI/ISA 95.00.05 Transaction and IEC 62264-5 standards.

OpenO&M™ Common Interop Registry (CIR) Model

Copyright 2010. MIMOSA

All Rights Reserved. http://www.mimosa.org

Parts derived from WBF B2MML-V0401

"The Business To Manufacturing Markup Language (B2MML) is used courtesy of WBF."

**Table of Contents**

[1 Common Interoperability Registry Model 3](#_Toc244410157)

[1.1 Interface Model 3](#_Toc244410158)

[1.2 CIRRegistry 4](#_Toc244410159)

[1.3 CIRCategory 4](#_Toc244410160)

[1.4 CIRRegistryEntry 5](#_Toc244410161)

[1.5 CIRProperty 5](#_Toc244410162)

[2 Service Definitions 7](#_Toc244410163)

[2.1 Type Definitions 7](#_Toc244410164)

[2.1.1 actionCode 7](#_Toc244410165)

[2.2 CIR Services 8](#_Toc244410166)

[2.2.1 Create Registry 8](#_Toc244410167)

[2.2.2 Create Unique Registry 10](#_Toc244410168)

[2.2.3 Combine Registries 11](#_Toc244410169)

[2.2.4 Get Full Registry 12](#_Toc244410170)

[2.2.5 Get Registry Entries 14](#_Toc244410171)

[2.2.6 Find Equivalent Entry 15](#_Toc244410172)

[2.2.7 Add Equivalent Entry 16](#_Toc244410173)

[2.2.8 Define Equivalent Entries 16](#_Toc244410174)

[2.2.9 Delete Registry Entries 16](#_Toc244410175)

[2.2.10 Change Registry Entries 16](#_Toc244410176)

[2.3 Wildcard Specification 18](#_Toc244410177)

[3 XML Implementation 19](#_Toc244410178)

[3.1 CIRRegistry 19](#_Toc244410179)

[3.1.1 Data Element 19](#_Toc244410180)

[3.1.2 Transactions 19](#_Toc244410181)

[3.2 CIRRegistryCategory 19](#_Toc244410182)

[3.2.1 Data Element 19](#_Toc244410183)

[3.2.2 Transactions 20](#_Toc244410184)

[3.3 CIRRegistryEntry 20](#_Toc244410185)

[3.3.1 Data Element 20](#_Toc244410186)

[3.3.2 Transactions 20](#_Toc244410187)

[3.4 CIRProperty 21](#_Toc244410188)

[3.4.1 Data Element 21](#_Toc244410189)

[Appendix A: OpenO&M-defined Properties 22](#_Toc244410190)

[CIRParentEntityID 22](#_Toc244410191)

[CIRChildEntityID 22](#_Toc244410192)

[CIRPossibleEquivalentEntryID: 22](#_Toc244410193)

[Appendix B: WSDL 23](#_Toc244410194)

# Common Interoperability Registry Model

## Interface Model

The CIR defines a common model for registration of entity identifiers and cross references of the identifiers for multiple systems.

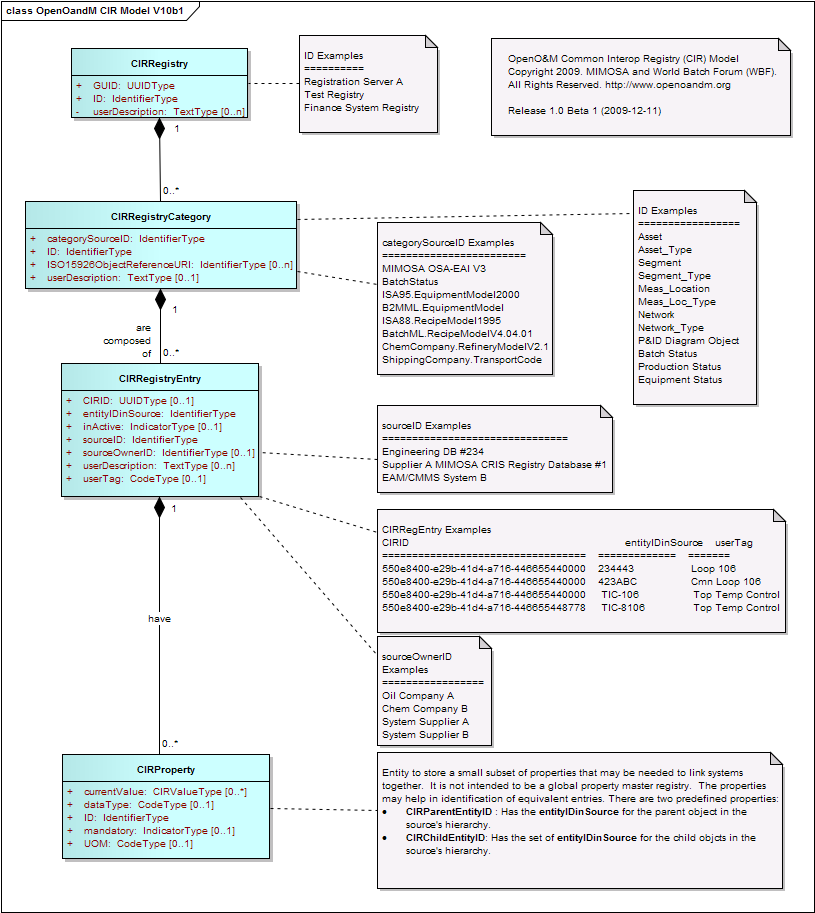


Figure - Common Interoperability Object Model

## CIRRegistry

A CIRRegistry object is the container object for a set of registration categories. Examples of multiple registries include: test registry, active registry, local site registry, global corporate registry.

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | | |
| **Attribute** | **Description** | **Restrictions** |
| ID | User supplied ID of the registry. This must be unique within the registry server. | Required |
| GUID | System assigned globally unique ID for the registry. This is unique across all registry servers. | Required |
| userDescription | User description of the registry and expected use of the registry. | Multiple values allowed for multiple languages or alternate descriptions |

## CIRCategory

A CIRCategory object is the container object for a set of registry entries. Registry categories define sets of related, or potentially related registry entries. For example, a registry category may be defined for equipment hierarchy level names (Enterprise, Site, Area, Work Center, Work Unit), which have alternate names on different systems. The combination of ID and categorySourceID must be unique.

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | | |
| **Attribute** | **Description** | **Restrictions** |
| ID | User supplied ID of the category | Required |
| categorySourceID | Identification of the category. May define the organization and specification name for the category, for example:   * MIMOSA OSA-EAI V3 * ISA 88 BatchStatus * ISA 95-2000 EquipmentModel * B2MML.EquipmentModel * ISA88.RecipeModel1995 * BatchML.RecipeModelV4.04.01 * ChemCompany.RefineryModelV2.1 * ShippingCompany.TransportCode | Required |
| userDescription | User description of the category and expected use of the category | Multiple values allowed for multiple languages or alternate descriptions |
| ISO15926ObjectReferenceURI | Defines the associated part of the ISO 15926 that defines the registry category. | Optional |

## CIRRegistryEntry

A CIRRegistryEntry object defines a registry entry. Registry entries define named element and properties with an identifier local to the owning application and a possible global ID (CIRID) that defined equivalent entries in other applications.

For example the tag TC101 in system A may be the equivalent of tag UNIT101.TOP\_TEMP in system B.

The assumption is that the combination of entityIDinSource and sourceID form a unique composite key within a registry category.

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | | |
| **Attribute** | **Description** | **Restrictions** |
| entityIDinSource | User defined identification of the entry in the source system | Required  Unique within the source system |
| sourceID | Identification of the source system | Required |
| CIRID | System assigned globally unique ID for the entry | Optional |
| sourceOwnerID | Organization that has responsibility for the source system or entity name space | Optional |
| userDescription | User description of the entry | Multiple values allowed for multiple languages or alternate descriptions |
| userTag | Shortcut identification of the entry, may not be unique within the source system. | Optional |
| inActive | Flag, if FALSE or missing indicates the entry is active and available for use. Examples of inactive entries may be data that is entered but the source system is not yet available or in use. | Optional |

## CIRProperty

A CIRProperty object defines a property of a registry entry. Properties may be used to help identify equivalent registry entries. The properties should be a small set of properties that may be needed to link systems together, and not intended to be a global property master registry.

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | | |
| **Attribute** | **Description** | **Restrictions** |
| ID | User defined identification of the property | Required  Unique within the list of entry properties |
| currentValue | Current value of the property | Multiple values with value key to provide for alternate values for identification |
| dataType | Data type of the current value | Optional |
| UOM | Unit of measure of the current value | Optional |
| mandatory | Flag that indicates if a value is required | Optional |

# Service Definitions

This section defines the detailed format for the *CIR Service* definitions.

## Type Definitions

### actionCode

0 = Success

1 = Duplicate Registry or Category definition

2 = Invalid channel URI

3 = Invalid channel type

## CIR Services

### Create Registry

|  |  |
| --- | --- |
| **Name** | CreateRegistry |
| **Description** | Creates a new registry, new category in a registry, or new entries in a category. Returns the system assigned GUID for the new registry. |
| **Input Parameters** | ProcessCIRRegistry as ProcessCIRRegistryType  If no registry exists with the specified ID or GUID, then a new registry is created and the registry GUID is returned.  If no category exists with the specified ID, then a new category is created.  If registry entries are defined, then adds the registry entries. May include the properties for the registry entries. No CIRIDs are assigned to the registry entries.  processCITRegistry |
| **Returns** | AcknowledgeCIRRegistry as AcknowledgeCIRRegistryType  Response status in:  AcknowledgeCIRRegistry/DataArea/Acknowledge/ResponseCriteria/ResponseExpression@actionCode  Registry GUID in:  AcknowledgeCIRRegistry/DataArea/CIRRegistry/GGUD  AckCIRReg |

### Create Unique Registry

|  |  |
| --- | --- |
| **Name** | CreateUniqueRegistry |
| **Description** | Creates a new registry, new category in a registry, or new entries in a category. Returns the system assigned GUID for the new registry and for the new registry entries. |
| **Input Parameters** | ProcessCIRRegistry as ProcessCIRRegistryType  If no registry exists with the specified ID or GUID, then a new registry is created and the registry GUID is returned.  If no category exists with the specified ID, then a new category is created.  If registry entries are defined, then adds the registry entries. May include the properties for the registry entries. New CIRIDs are assigned to the registry entries.  processCITRegistry |
| **Returns** | AcknowledgeCIRRegistry as AcknowledgeCIRRegistryType  Response status in:  AcknowledgeCIRRegistry/DataArea/Acknowledge/ResponseCriteria/ResponseExpression@actionCode  Registry GUID in:  AcknowledgeCIRRegistry/DataArea/CIRRegistry/GGUD  Entry CIRIDs in:  AcknowledgeCIRRegistry/DataArea/CIRRegistry/category/entry/CIRID  AckCIRReg |

### Combine Registries

|  |  |
| --- | --- |
| **Name** | CombineRegistries |
| **Description** | Adds a list of a registry categories and registry entries to an existing registry.  Return an error if the registry is not defined. |
| **Input Parameters** | * Registry ID or registry GUID as string * Complete registry as ProcessCIRRegistryType |
| **Returns** | AcknowledgeCIRRegistry as AcknowledgeCIRRegistryType  Response status in  AcknowledgeCIRRegistry/DataArea/Acknowledge/ResponseCriteria/ResponseExpression@actionCode  AckCIRReg |

### Get Full Registry

|  |  |
| --- | --- |
| **Name** | GetFullRegistry |
| **Description** | Returns all categories and entries in a registry. |
| **Input Parameters** | GetCIRRegistry as GetCIRRegistryType  If ID or GUID specified, then the identified registry is returned.  If no ID or GUID is specified, then the service determines which registry information to return.  GetCIRRegistry |
| **Returns** | ShowCIRRegistry as ShowCIRRegistryType  Response status in  ShowCIRRegistry/DataArea/Acknowledge/ResponseCriteria/ResponseExpression@actionCode  ShowCIRReg |

### Get Registry Entries

|  |  |
| --- | --- |
| **Name** | GetRegistryEntries |
| **Description** | Returns entries and their properties from a category in a registry. Used to find information about registry entries. |
| **Input Parameters** | GetCIRRegistryEntry as GetCIRRegistryEntryType  getcirregistryentry  If the registry is not defined, then the service selects the registry to use.  If an entry CIRID is specified, than all entries with the specified CIRID are returned,  Else if no category ID and no CategorySourceID is defined, then an error is returned,  Else if an entryIDinSource and sourceID are specified, then all information on the specified entry is returned,  Else if the entryIDinSource contains a wildcard specification (see section ) and sourceID is specified, then all entries matching the wildcard specification are returned,  Else if a userTag and sourceID are specified, then all information on all entries with the userTag are returned,  Else if the userTag contains a wildcard specification (see section ) and sourceID is specified, then all entries matching the wildcard specification are returned,  Else if only a sourceID is specified, then all entries for the source ID are returned,  Else an error is returned. |
| **Returns** | ShowCIRRegistry as ShowCIRRegistryType  Response status in  ShowCIRRegistry/DataArea/Acknowledge/ResponseCriteria/ResponseExpression@actionCode |

### Find Equivalent Entry

|  |  |
| --- | --- |
| **Name** | FindEquivalentEntry |
| **Description** | Finds one or more equivalent registry entries  Return an error if the registry or category is not defined. |
| **Input Parameters** | * Source & Target Registry ID or GUID as string * Source & Target Category ID as string * Source EntityIDinSource (0..\*) as string using the JSON[[1]](#footnote-1) format for the array of IDs * Source Source ID as string * Target Source ID as string |
| **Returns** | * EntityIDinSource (0..\*) in Target Source ID, returns an empty string in case of no equivalent or error. The returned value follows the JSON format for an array of equivalent entities. |

### Add Equivalent Entry

|  |  |
| --- | --- |
| **Name** | AddEquivalentEntry |
| **Description** | Adds a CIRID to a registry entry from an equivalent registry entry. If neither entry has a CIRID, then a new CIRID is created.  Return an error if the registry, category, source entry, or target entry are not defined. |
| **Input Parameters** | * Source & Target Registry ID as string * Source & Target Category ID as string * Source EntityIDinSource as string * Source SourceID as string * Target EntityIDinSource as string * Target SourceID as string |
| **Returns** | * CIRID as string, returns an empty string in case of error. |

### Define Equivalent Entries

|  |  |
| --- | --- |
| **Name** | DefineEquivalentEntries |
| **Description** | Defines two CIRIDs as equivalent by changing the target CIRID in all registries to the source CIRID. |
| **Input Parameters** | * Source CIRID * Target CIRID |
| **Returns** | * Error string, empty string in the case of no error |

### Delete Registry Entries

|  |  |
| --- | --- |
| **Name** | DeleteRegistryEntries |
| **Description** | Deletes one or more registry entries  Return an error if the registry or category is not defined. |
| **Input Parameters** | * Registry ID or GUID as string * Category ID as string * List of Registry entries as GetCIRRegistryEntryListType |
| **Returns** | * Error string, empty string in the case of no error |

### Change Registry Entries

|  |  |
| --- | --- |
| **Name** | ChangeRegistryEntries |
| **Description** | Updates one or more registry entries and associated properties. |
| **Input Parameters** | ChangeCIRRegistry as ChangeCIRRegistryType  If the registry is not defined, then the service selects the registry.  If the category is not defined, then an error is returned.  If an entry CIRID and sourceID are defined, then the other registry information and properties are changed.  Else if a sourceID and sourceOwnerID are defined, then the other registry information and properties are changed.  Else an error is returned.  ChangeCIRRegistry |
| **Returns** | RespondCIRRegistryEntry as RespondShowCIRRegistryEntryType  Response status in  RespondCIRRegistryEntry/DataArea/Acknowledge/ResponseCriteria/ResponseExpression@actionCode  ResponseCIRRegEntry |

## Wildcard Specification

Services utilitizing FilterCIRRegistryType, FilterCIRRegistryCategoryType, FilterCIRRegistryEntryType can utilize wildcards when specifying filter parameters to identify multiple objects. The convention for specifying wildcards in text strings is through limited regular expressions. In a limited regular expression a wildcard value can have the following special characters:

“\*” – Indicates zero or more characters, any character is acceptable

Example 1: The wildcard “ABC\*” would match “ABC”, “ABCD”, “ABCDEF”, “ABC@4!\*“, but not “ABDDEF”

“%” – Indicates one or more characters, any character is acceptable

Example 2: The wildcard “ABC%” would match “ABCD”, “ABCDEF”, “ABC^4^\*“, but not “ABC”

“?” – Indicates zero or one characters at the specified position, any character is acceptable

Example 3: The wildcard “ABC?” would match “ABCX”, “ABCD”, “ABC!“, “ABC”, but not “ABCDE” or “ABDC”

The character following a “\” is considered a literal character, not a wildcard character.

Example 4: An object ID of “ABC\\*” defines the object ID as “ABC\*”.

Example 5: A property ID of “\\\\USM 123” defines the property ID “\\USM 123”.

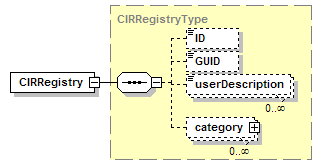
Two consecutive backslash characters, i.e. “\\” are interpreted to be a single backslash character “\”.

# XML Implementation

The XML implementation is based on the UN/CEFACT core component data types and the ISA 95.00.05, IEC 622264-5, and OAGIS 9.0 transaction specifications.

## CIRRegistry

### Data Element

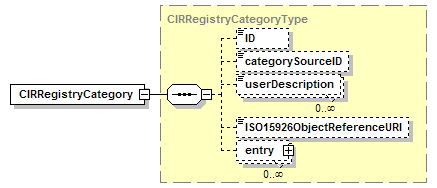


### Transactions

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Returns** |
| GetCIRRegistry | Get registry information for specified registry ID or GUID. If no ID or GUID specified, return all registries. | ShowCIRRegistry |
| ProcessCIRRegistry | Create a new registry. | AcknowledgeCIRRegistry |
| ChangeCIRRegistry | Change a registry ID or userDescription. | RespondCIRRegistry |
| CancelCIRRegistry | Remove a registry |  |
| SyncCIRRegistry | Publish changes to a registry | SyncCIRRegistry |

## CIRRegistryCategory

### Data Element

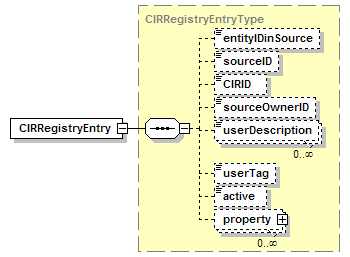


### Transactions

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Returns** |
| GetCIRRegistryCategory |  | ShowCIRRegistryCategory |
| ProcessCIRRegistryCategory |  | AcknowledgeCIRRegistryCategory |
| ChangeCIRRegistryCategory |  | RespondCIRRegistryCategory |
| CancelCIRRegistryCategory |  |  |
| SyncCIRRegistryCategory |  | SyncCIRRegistryCategory |

## CIRRegistryEntry

### Data Element

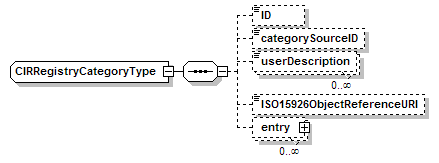


### Transactions

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Returns** |
| GetCIRRegistryEntry |  | ShowCIRRegistryEntry |
| ProcessCIRRegistryEntry |  | AcknowledgeCIRRegistryEntry |
| ChangeCIRRegistryEntry |  | RespondCIRRegistryEntry |
| CancelCIRRegistryEntry |  |  |
| SyncCIRRegistryEntry |  | SyncCIRRegistryEntry |

## CIRProperty

### Data Element



Appendix A: OpenO&M-defined Properties

There are predefined OpenO&M properties that should be used to identify commonly understood relationships between entities.

## CIRParentEntityID

The CIRParentEntity contains the set of entityIDinSource IDs for parent object of the entity in the source's hierarchy.

The property value should follow the JSON format for the set of entityIDinSource IDs.

## CIRChildEntityID

The CIRChildEntityID contains the set of entityIDinSource IDs for child objects of the entity in the source's hierarchy.

The property value should follow the JSON format for the set of entityIDinSource IDs.

## CIRPossibleEquivalentEntryID:

The CIRPossibleEquivalentEntryID contains a set of target entities which are possibly equivalent to the entity. This allows for automated equivalency determination. Each returned target entry contains the following set of information:

1. EntityIDinSource
2. SourceID
3. percentLikelihood [Optional]

The property value should follow the JSON format for the array of objects or equivalent IDs.

Examples:

[ {"EntityIDinSource" : "TIC101", "SourceID" : "Thor"} ]

[ {"EntityIDinSource" : "TIC101", "SourceID" : "Thor" , "percentLikelihood" : 20} ]

[ {"EntityIDinSource" : "TIC101", "SourceID" : "Thor"} ,

{"EntityIDinSource" : "T101" , "SourceID" : "Apollo"} ]

Appendix B: WSDL

<?xml version="1.0" encoding="utf-8"?>

<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"

xmlns:tns="http://www.openoandm.org/xml/CIRML/"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"

xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"

targetNamespace="http://www.openoandm.org/xml/CIRML/"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

<wsdl:types>

<xsd:schema elementFormDefault="qualified" targetNamespace="http://www.openoandm.org/xml/CIRML/">

<xsd:element name="CreateRegistry">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="processCIRRegistry" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="CreateRegistryResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="CreateRegistryResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="CreateUniqueRegistry">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="processCIRRegistry" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="CreateUniqueRegistryResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="CreateUniqueRegistryResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="CombineRegistry">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="registryID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="processCIRRegistry" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="CombineRegistryResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="CombineRegistryResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="GetFullRegistry">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="getCIRRegistry" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="GetFullRegistryResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="GetFullRegistryResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="GetRegistryEntries">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="getCIRRegistry" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="GetRegistryEntriesResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="GetRegistryEntriesResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="FindEquivalentEntry">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="registryID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="categoryID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="entryIDinSource" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="sourceOwnerID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="targetOwnerID" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="FindEquivalentEntryResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="FindEquivalentEntryResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="AddEquivalentEntry">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="registryID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="categoryID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="entryIDinSource" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="sourceOwnerID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="entryIDinTarget" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="targetOwnerID" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="AddEquivalentEntryResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="AddEquivalentEntryResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="DefineEquivalentEntries">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="sourceCIRID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="targetCIRID" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="DefineEquivalentEntriesResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="DefineEquivalentEntriesResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="DeleteRegistryEntries">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="registryID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="categoryID" type="xsd:string" />

<xsd:element minOccurs="0" maxOccurs="1" name="getCIRRegistry" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="DeleteRegistryEntriesResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="DeleteRegistryEntriesResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="ChangeRegistryEntries">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="changeCIRRegistry" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="ChangeRegistryEntriesResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" maxOccurs="1" name="ChangeRegistryEntriesResult" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

</wsdl:types>

<wsdl:message name="CreateRegistrySoapIn">

<wsdl:part name="parameters" element="tns:CreateRegistry" />

</wsdl:message>

<wsdl:message name="CreateRegistrySoapOut">

<wsdl:part name="parameters" element="tns:CreateRegistryResponse" />

</wsdl:message>

<wsdl:message name="CreateUniqueRegistrySoapIn">

<wsdl:part name="parameters" element="tns:CreateUniqueRegistry" />

</wsdl:message>

<wsdl:message name="CreateUniqueRegistrySoapOut">

<wsdl:part name="parameters" element="tns:CreateUniqueRegistryResponse" />

</wsdl:message>

<wsdl:message name="CombineRegistrySoapIn">

<wsdl:part name="parameters" element="tns:CombineRegistry" />

</wsdl:message>

<wsdl:message name="CombineRegistrySoapOut">

<wsdl:part name="parameters" element="tns:CombineRegistryResponse" />

</wsdl:message>

<wsdl:message name="GetFullRegistrySoapIn">

<wsdl:part name="parameters" element="tns:GetFullRegistry" />

</wsdl:message>

<wsdl:message name="GetFullRegistrySoapOut">

<wsdl:part name="parameters" element="tns:GetFullRegistryResponse" />

</wsdl:message>

<wsdl:message name="GetRegistryEntriesSoapIn">

<wsdl:part name="parameters" element="tns:GetRegistryEntries" />

</wsdl:message>

<wsdl:message name="GetRegistryEntriesSoapOut">

<wsdl:part name="parameters" element="tns:GetRegistryEntriesResponse" />

</wsdl:message>

<wsdl:message name="FindEquivalentEntrySoapIn">

<wsdl:part name="parameters" element="tns:FindEquivalentEntry" />

</wsdl:message>

<wsdl:message name="FindEquivalentEntrySoapOut">

<wsdl:part name="parameters" element="tns:FindEquivalentEntryResponse" />

</wsdl:message>

<wsdl:message name="AddEquivalentEntrySoapIn">

<wsdl:part name="parameters" element="tns:AddEquivalentEntry" />

</wsdl:message>

<wsdl:message name="AddEquivalentEntrySoapOut">

<wsdl:part name="parameters" element="tns:AddEquivalentEntryResponse" />

</wsdl:message>

<wsdl:message name="DefineEquivalentEntriesSoapIn">

<wsdl:part name="parameters" element="tns:DefineEquivalentEntries" />

</wsdl:message>

<wsdl:message name="DefineEquivalentEntriesSoapOut">

<wsdl:part name="parameters" element="tns:DefineEquivalentEntriesResponse" />

</wsdl:message>

<wsdl:message name="DeleteRegistryEntriesSoapIn">

<wsdl:part name="parameters" element="tns:DeleteRegistryEntries" />

</wsdl:message>

<wsdl:message name="DeleteRegistryEntriesSoapOut">

<wsdl:part name="parameters" element="tns:DeleteRegistryEntriesResponse" />

</wsdl:message>

<wsdl:message name="ChangeRegistryEntriesSoapIn">

<wsdl:part name="parameters" element="tns:ChangeRegistryEntries" />

</wsdl:message>

<wsdl:message name="ChangeRegistryEntriesSoapOut">

<wsdl:part name="parameters" element="tns:ChangeRegistryEntriesResponse" />

</wsdl:message>

<wsdl:portType name="CIRServiceSoap">

<wsdl:operation name="CreateRegistry">

<wsdl:documentation>Creates a new registry, new category in a registry, or new entries in a category. Returns the system assigned GUID for the new registry.</wsdl:documentation>

<wsdl:input message="tns:CreateRegistrySoapIn" />

<wsdl:output message="tns:CreateRegistrySoapOut" />

</wsdl:operation>

<wsdl:operation name="CreateUniqueRegistry">

<wsdl:documentation>Creates a new registry, new category in a registry, or new entries in a category. Returns the system assigned GUID for the new registry and for the new registry entries.</wsdl:documentation>

<wsdl:input message="tns:CreateUniqueRegistrySoapIn" />

<wsdl:output message="tns:CreateUniqueRegistrySoapOut" />

</wsdl:operation>

<wsdl:operation name="CombineRegistry">

<wsdl:documentation>Adds a list of a registry categories and registry entries to an existing registry. Return an error as AcknowledgeCIRRegistry if the registry is not defined.</wsdl:documentation>

<wsdl:input message="tns:CombineRegistrySoapIn" />

<wsdl:output message="tns:CombineRegistrySoapOut" />

</wsdl:operation>

<wsdl:operation name="GetFullRegistry">

<wsdl:documentation>Returns all categories and entries in a registry.</wsdl:documentation>

<wsdl:input message="tns:GetFullRegistrySoapIn" />

<wsdl:output message="tns:GetFullRegistrySoapOut" />

</wsdl:operation>

<wsdl:operation name="GetRegistryEntries">

<wsdl:documentation>Returns entries and their properties from a category in a registry. Used to find information about registry entries.</wsdl:documentation>

<wsdl:input message="tns:GetRegistryEntriesSoapIn" />

<wsdl:output message="tns:GetRegistryEntriesSoapOut" />

</wsdl:operation>

<wsdl:operation name="FindEquivalentEntry">

<wsdl:documentation>Finds an equivalent registry entry. Return an error if the registry or category is not defined.</wsdl:documentation>

<wsdl:input message="tns:FindEquivalentEntrySoapIn" />

<wsdl:output message="tns:FindEquivalentEntrySoapOut" />

</wsdl:operation>

<wsdl:operation name="AddEquivalentEntry">

<wsdl:documentation>Adds a CIRID to a registry entry from an equivalent registry entry. If neither entry has a CIRID, then a new CIRID is created. Return an error if the registry, category, source entry, or target entry are not defined.</wsdl:documentation>

<wsdl:input message="tns:AddEquivalentEntrySoapIn" />

<wsdl:output message="tns:AddEquivalentEntrySoapOut" />

</wsdl:operation>

<wsdl:operation name="DefineEquivalentEntries">

<wsdl:documentation>Defines two CIRIDs as equivalent by changing the target CIRID in all registries to the source CIRID.</wsdl:documentation>

<wsdl:input message="tns:DefineEquivalentEntriesSoapIn" />

<wsdl:output message="tns:DefineEquivalentEntriesSoapOut" />

</wsdl:operation>

<wsdl:operation name="DeleteRegistryEntries">

<wsdl:documentation>Deletes one or more registry entries. Return an error if the registry or category is not defined.</wsdl:documentation>

<wsdl:input message="tns:DeleteRegistryEntriesSoapIn" />

<wsdl:output message="tns:DeleteRegistryEntriesSoapOut" />

</wsdl:operation>

<wsdl:operation name="ChangeRegistryEntries">

<wsdl:documentation>Updates one or more registry entries and associated properties.</wsdl:documentation>

<wsdl:input message="tns:ChangeRegistryEntriesSoapIn" />

<wsdl:output message="tns:ChangeRegistryEntriesSoapOut" />

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name="CIRServiceSoap" type="tns:CIRServiceSoap">

<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />

<wsdl:operation name="CreateRegistry">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/CreateRegistry" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="CreateUniqueRegistry">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/CreateUniqueRegistry" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="CombineRegistry">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/CombineRegistry" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="GetFullRegistry">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/GetFullRegistry" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="GetRegistryEntries">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/GetRegistryEntries" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="FindEquivalentEntry">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/FindEquivalentEntry" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="AddEquivalentEntry">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/AddEquivalentEntry" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="DefineEquivalentEntries">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/DefineEquivalentEntries" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="DeleteRegistryEntries">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/DeleteRegistryEntries" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="ChangeRegistryEntries">

<soap:operation soapAction="http://www.openoandm.org/xml/CIRML/ChangeRegistryEntries" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:binding name="CIRServiceSoap12" type="tns:CIRServiceSoap">

<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />

<wsdl:operation name="CreateRegistry">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/CreateRegistry" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="CreateUniqueRegistry">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/CreateUniqueRegistry" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="CombineRegistry">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/CombineRegistry" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="GetFullRegistry">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/GetFullRegistry" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="GetRegistryEntries">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/GetRegistryEntries" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="FindEquivalentEntry">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/FindEquivalentEntry" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="AddEquivalentEntry">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/AddEquivalentEntry" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="DefineEquivalentEntries">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/DefineEquivalentEntries" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="DeleteRegistryEntries">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/DeleteRegistryEntries" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="ChangeRegistryEntries">

<soap12:operation soapAction="http://www.openoandm.org/xml/CIRML/ChangeRegistryEntries" style="document" />

<wsdl:input>

<soap12:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap12:body use="literal" />

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="CIRService">

<wsdl:port name="CIRServiceSoap" binding="tns:CIRServiceSoap">

<soap:address location="http://localhost:49574/CIRService.asmx" />

</wsdl:port>

<wsdl:port name="CIRServiceSoap12" binding="tns:CIRServiceSoap12">

<soap12:address location="http://localhost:49574/CIRService.asmx" />

</wsdl:port>

</wsdl:service>

</wsdl:definitions>

1. **JSON** (JavaScript Object Notation) [www.json.org] is a lightweight data-interchange format that is easy to read, write, parse and generate. It is based on a subset of the [JavaScript Programming Language](http://javascript.crockford.com/), [Standard ECMA-262 3rd Edition - December 1999](http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf). JSON is a text format that is language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, and Python. JSON is built on two structures:

   A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table

   An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence. [↑](#footnote-ref-1)