

# Entity Identification Service (EIS)

*Version 1.0 – Beta 2 (with change markups)*

---

**OMG Document Number:** dtc/2010-11-22

**Standard document URL:** <http://www.omg.org/spec/EIS/1.0/PDF>

**Associated File(s)\*:**

<http://www.omg.org/spec/EIS/201012/EISAdminEditorInterface.wsdl>

<http://www.omg.org/spec/EIS/201012/EISMetaDataInterface.wsdl>

<http://www.omg.org/spec/EIS/201012/EISMgmtAndQueryInterface.wsdl>

---

\* Original file(s): dtc/2010-11-25 EIS machine consumable files (EIS-archive-201012.zip)

Copyright © 2007-2010, Healthcare Services Specification Project (HSSP)  
Copyright © 2007-2010, Intel Corporation  
Copyright © 2007-2010, Kaiser Permanente  
Copyright © 2007-2010, Object Management Group, Inc.  
Copyright © 2007-2010, Ocean Informatics  
Copyright © 2007-2010, Oracle  
Copyright © 2007-2010, Software Partners, LLC  
Copyright © 2007-2010, WebReach, Inc.

## USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

## LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

## PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

## GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

## DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

## RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c) (1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 140 Kendrick Street, Needham, MA 02494, U.S.A.

## TRADEMARKS

MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™, Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™, MOF™, OMG Interface Definition Language (IDL)™, and OMG SysML™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

## COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

## **OMG's Issue Reporting Procedure**

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page <http://www.omg.org>, under Documents, Report a Bug/Issue (<http://www.omg.org/technology/agreement>.)

# Table of Contents

<u>1 Scope.....</u>	<u>1</u>
<u>2 Compliance.....</u>	<u>1</u>
<u>3 Normative References.....</u>	<u>1</u>
<u>4 Acronyms.....</u>	<u>2</u>
<u>5 Symbols.....</u>	<u>2</u>
<u>6 Additional Information.....</u>	<u>3</u>
<u>6.1 Acknowledgements.....</u>	<u>3</u>
<u>6.2 Internationalization.....</u>	<u>3</u>
<u>6.3 Relation to Existing OMG Specifications.....</u>	<u>3</u>
<u>7 EIS Platform Independent Model.....</u>	<u>5</u>
<u>7.1 Definitions.....</u>	<u>5</u>
<u>7.2 Theory of Solution.....</u>	<u>6</u>
<u>7.2.1 Cascades.....</u>	<u>6</u>
<u>7.2.2 Register and Find.....</u>	<u>6</u>
<u>7.2.3 Duplicates.....</u>	<u>6</u>
<u>8 EISManagementAndQueryInterface.....</u>	<u>7</u>
<u>8.1 RegisterEntityWithIdentity.....</u>	<u>8</u>
<u>8.2 CreateIdentityFromEntity.....</u>	<u>11</u>
<u>8.3 UpdateEntityTraitValues.....</u>	<u>11</u>
<u>8.4 RemoveIdentity.....</u>	<u>13</u>
<u>8.5 GetEntityTraitValues.....</u>	<u>14</u>
<u>8.6 FindIdentitiesByTraits.....</u>	<u>15</u>
<u>8.7 ListLinkedIdentities.....</u>	<u>16</u>
<u>8.8 ListUnlinkedIdentities.....</u>	<u>18</u>
<u>9 EISAdminEditorInterface.....</u>	<u>19</u>
<u>9.1 LinkEntities.....</u>	<u>21</u>
<u>9.2 UnlinkEntities.....</u>	<u>21</u>

9.3 MergeEntities.....	22
9.4 UnMergeEntities.....	23
9.5 ActivateEntity.....	23
9.6 DeactivateEntity.....	24
<b>10 EISMetaDataInterface.....</b>	<b>26</b>
10.1 CreateSemanticSignifier.....	26
10.2 FindSemanticSignifier.....	26
10.3 UpdateSemanticSignifier.....	28
10.4 ListSemanticSignifiers.....	28
10.5 ListDomains.....	28
10.6 ListConformanceProfiles.....	29
<b>11 EIS Data Structures.....</b>	<b>30</b>
11.1 Complex Data Structures.....	30
11.1.1 EISSemanticSignifierDefinition.....	30
11.1.1.1 Overview.....	30
11.1.1.2 Structure.....	30
11.1.2 EISSemanticSignifier.....	30
11.1.2.1 Overview.....	30
11.1.2.2 Structure.....	30
11.1.3 EISSearchQualifier.....	30
11.1.3.1 Overview.....	30
11.1.3.2 Structure.....	30
11.1.4 EISUpdateQualifier.....	31
11.1.4.1 Overview.....	31
11.1.4.2 Structure.....	31
11.1.5 EISStatus.....	31
11.1.5.1 Overview.....	31
11.1.5.2 Structure.....	31
11.1.5.3 StatusCondition component.....	32
11.1.5.4 StatusCode component.....	32
11.1.6 EISState.....	32
11.1.6.1 Overview.....	32
11.1.6.2 Structure.....	33
11.1.7 EISResultSet.....	33
11.1.7.1 Overview.....	33
11.1.7.2 Structure.....	33
11.1.7.3 EISResultSetElement structure.....	33
11.2 Other Simple Data Types.....	33
11.3 Supplemental information on EIS Error Handling.....	34
<b>12 Semantic Profiles.....</b>	<b>35</b>
12.1 Profile alignment by Service Interface and Operations.....	35
12.2 Implementations and Semantic Signifiers.....	35
12.3 Implementations.....	37

12.3.1 Implementation Example – HL7 Version 2.5 Person using IHE Pediatric Demographic option.....	37
12.3.1.1 Definition of Traits using HL7 V2.5 Segments.....	37
12.3.1.2 EISManagementAndQueryInterface.....	39
12.3.1.3 EISAdminEditorInterface.....	47
12.3.1.4 EISMetaDataInterface.....	51
<b>13 Compatibility to Other Standards.....</b>	<b>52</b>
13.1 Compatibility to HL7.....	52
13.2 Compatibility to IHE Standards.....	52

# Preface

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at <http://www.omg.org/>.

## OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. A Specifications Catalog is available from the OMG website at:

[http://www.omg.org/technology/documents/spec\\_catalog.htm](http://www.omg.org/technology/documents/spec_catalog.htm)

Specifications within the Catalog are organized by the following categories:

### OMG Modeling Specifications

- UML
- MOF
- XMI
- CWM
- Profile specifications

### OMG Middleware Specifications

- CORBA/IIOP
- IDL/Language Mappings
- Specialized CORBA specifications
- CORBA Component Model (CCM)

### Platform Specific Model and Interface Specifications

- CORBA services
- CORBA facilities
- OMG Domain specifications
- OMG Embedded Intelligence specifications
- OMG Security specifications

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:



OMG Headquarters  
140 Kendrick Street  
Building A, Suite 300  
Needham, MA 02494  
USA  
Tel: +1-781-444-0404  
Fax: +1-781-444-0320  
Email: [pubs@omg.org](mailto:pubs@omg.org)

Certain OMG specifications are also available as ISO standards. Please consult <http://www.iso.org>

## Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

**Helvetica/Arial - 10 pt. Bold:** OMG Interface Definition Language (OMG IDL) and syntax elements.

**Courier - 10 pt. Bold:** Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

NOTE: Terms that appear in italics are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

# 1 Scope

The scope of this specification is the

- interfaces,
- operations, and
- information structure

required to uniquely identify various kinds of entities within disparate systems within a single enterprise and/or across a set of collaborating enterprises.

These service interfaces are intended to be used within a healthcare setting, in which the security of data exchanges is both important and regulated by laws such as HIPAA in the United States.

However, the following aspects of this service, while both important and required, are addressed elsewhere and are *out of scope* for this specification:

- Security
- Privacy
- Versioning
- Logging
  
- Internationalization

Separating these functions from the EIS service facilitates the application of a uniform policies across a set of services.

## 2 Compliance

Compliance [Conformance] to this specification happens by way of conformance to profiles, which are detailed in Section 8. Conformance Profiles consist of at least one Functional (or Behavioral) Profile and at least one Semantic Profile.

Functional Profiles are named subsets of the overall set of operations defined within the specification that provide a cohesive set of functionality that makes sense from the service client's perspective. Note that this is orthogonal to the internal interface structure of the Service from the service provider's perspective. A reflection style interface is also made available by EIS instances that will identify which profiles it supports at any given point in time.

Similarly, Semantic Profiles are named information models that define the "Semantic Signifiers" that are mapped to and used in the inputs and outputs of operations. This allows the same behavioral interface to be used with different content models, e.g. HL7 V2 data structures, HL7 V3 data structures or OpenEHR Archetypes.

## 3 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, <http://www.w3.org/TR/wsdl>

Simple Object Access Protocol (SOAP) 1.1, W3C Note 08 May 2000, <http://www.w3.org/TR/SOAP>

XML Schema Part 2: Datatypes, W3C Recommendation 02 May 2001, <http://www.w3.org/TR/xmlschema-2>

HL7 Entity Identification Service, Release 1, Service Functional Model Specification, <http://www.hl7.org/dstucomments/index.cfm>

ISO Reference Model of Open Distributed Processing (RM-ODP), <http://www.iso.org>

Service Specifications Framework (SSF), Healthcare Services Specification Project (HSSP), <http://hssp.wikispaces.com>  
 IHE PIX and PDQ including Pediatric Demographics and V3 PIX/PDQ

## 4 Acronyms

There are a number of acronyms used in this document, and in standards or other documents related to this proposal, and here is a brief list of what the most common ones stand for:

<b>Acronym</b>	<b>Full Name</b>
<b>ANSI</b>	<b>American National Standards Institute (U.S.A.)</b>
<b>EIS</b>	<b>Entity Identification Service</b>
<b>HIPAA</b>	<b>Health Insurance Portability and Accountability Act (U.S.A.)</b>
<b>HITSP</b>	<b>Health Information Technology Standards Panel (U.S.A.) of ANSI</b>
<b>HL7</b>	<b>Health Level 7</b>
<b>HSSP</b>	<b>Healthcare Services Specification Project</b>
<b>IHE</b>	<b>Integrating the Healthcare Enterprise</b>
<b>ISO</b>	<b>International Standards Organization</b>
<b>OMG</b>	<b>Object Management Group</b>
<b>PFM</b>	<b>Platform Functional Model</b>
<b>PIM</b>	<b>Platform Independent Model</b>
<b>PIX – PDQ</b>	<b>Patient Index Query – Patient Demographic Query (from IHE)</b>
<b>PSM</b>	<b>Platform Specific Model</b>
<b>RIM</b>	<b>Reference Information Model defined by HL7</b>
<b>RFP</b>	<b>Request for Proposal</b>
<b>RM-ODP</b>	<b>Reference Model of Open Distributed Processing defined by ISO</b>
<b>SDO</b>	<b>Standards Development Organization</b>
<b>SFM</b>	<b>Service Functional Model</b>
<b>UML</b>	<b>Universal Modeling Language</b>

## 5 Symbols

There are no symbols/abbreviations.

# 6 Additional Information

## 6.1 Acknowledgements

The following organizations contributed to this specification:

Intel Corporation  
Kaiser Permanente  
Ocean Informatics  
Oracle  
Software Partners  
Web Reach

## 6.2 Internationalization

This EIS submission primarily reflects the contributions of U.S.-based submitters and supporters. However, certain measures were taken to maximize international alignment. First, some submitters and supports do business internationally and will have their initial implementations in these environments. Second, by aligning with the trends of the internationally-based SDOs mentioned above, specifically, HL7 and IHE, we hope to indirectly harmonize with international requirements. Finally, the use of semantic signifiers as traits parameters introduces a degree of flexibility that we hope will optimally facilitate international differences.

## 6.3 Relation to Existing OMG Specifications

The following is extracted from Section 6.3 of the OMG EIS RFP:

Reference	Description	Relationship to EIS
<b>OMG Person Identification Service (PIDS) Specification Version 1.1, April 2001</b>	<b>A specification of a Person Identification Service with UML models and a complete specification in CORBA IDL. This includes most of the functions for managing a patient identity. The traits of an identity in this specification used the HL7 v2.5 and the vCard specifications.</b>	<b>This specification has been used as the basis for many implementations in different industries. This has been used as a primary source for many of the ideas in this specification. This specification could be seen as bringing PIDS up to date, although the additional "Entity" abstraction provides for additional flexibility of use.</b>

EIS defines 'generic' interfaces that would allow name-value pairs to be associated with an entity, similar to its fore-runner, the Person Identification Service (PIDS). In fact, for person related functionality, the service interfaces defined in this document could be implemented over an implementation of the PIDS module as shown in Figure 1:

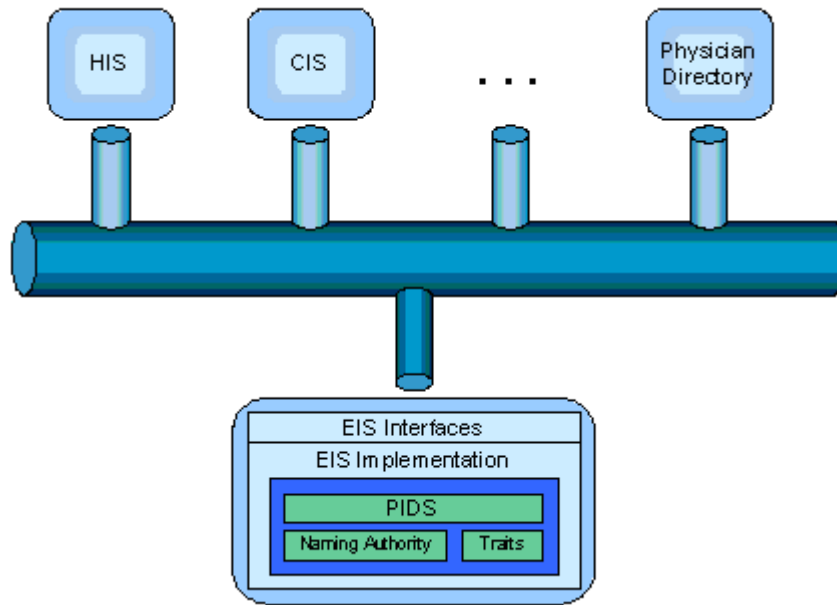


Figure 2: The PIDS module

Other emerging OMG Specifications that may be related to this specification are the Organization Structure Metamodel and the UML Profile and Metamodel for Services.

# 7 EIS Platform Independent Model

The Platform Independent Model (PIM) for EIS represents a wire and payload implementation independent means to describe the EIS interfaces.

The PIM is detailed in the sub-sections below containing a UML model and associated descriptions which depicts the structure of the interfaces provided by EIS. It is important to note that the structure of interfaces is specifically defined for the Service provider and reflects a natural structure of components that may provide the EIS functionality. These are effectively orthogonal to the groupings of capabilities defined in Functional Profiles that form the basis for conformance and for consumption of the service by Clients. In some cases, the functional profiles may approximate the interface structure.

## 7.1 Definitions

EIS embodies a number of key concepts that should be understood up front before attempting to read and understand this specification. Key terms are described below:

Term	Description
Semantic Signifier	This represents a schema definition and an associate set of validation instructions for the EIS Traits. It defines the structure and the means by which that structure should be validated. In XML for example, it might be an XSD and associated Schematron.
Logical Record	An instance of the semantic signifier. It could either be in document or message format depending on the implementation.
Real World Entity (RWE)	Represents the actual physical thing itself, e.g. the actual Person, the actual Device etc.
Entity	The software representation of a RWE - a record.
Domain	A set of values in which each is unique.
Source	A system generating an Entity.
Identifier (ID)	A value within a Domain that is associated with an object – an Entity, a Source, etc. - and uniquely identifies it within the scope of the Domain.
Entity ID	An Identifier associated with an Entity.
Source ID	An Identifier associated with a Source.
Identity	A single (Entity ID, Source ID) pair, or set of linked (Entity ID, Source ID) pairs.
EIS ID	An Identifier within the Domain of an EIS that is associated by the EIS with an Identity. It is not the same as an Entity ID, since it represents the entire set of linked Identities that represent a single Real World Entity.
Trait	A data element used by an EIS matching algorithm to link a particular (Source ID, Entity ID) pair with an existing Identity, based upon a conclusion that the Entities they are identifying represent the same RWE.

## 7.2 Theory of Solution

EIS primarily deals with the maintenance and use of data from the Entity class of the HL7 v3 RIM (Reference Information Model). It may optionally deal with the Role class of the RIM, but it does not store information related to Acts or Participations.

EIS is based upon the creation and maintenance of an index consisting of a linked set of Source ID/Entity ID pairs representing the same Real World Entity (RWE). Although the subject of this technical specification is a service, the acronym “EIS” is also used here to refer to the index itself and associated Traits. A Source ID and Entity ID are supplied in pairs in order that they may uniquely identify an Entity with *the Domain of the EIS*. (An Entity ID alone uniquely identifies an Entity *within the Domain of the Source*).

### 7.2.1 Cascades

EISs may be cascaded. This is done through:

- the assignment of an EIS ID to each Identity that the EIS maintains, plus
- the existence of a Source ID associated with the EIS itself.

In this case, the Source ID representing the EIS and the EIS ID representing the Entity can be passed as parameters to another EIS, creating a cascade. This is also known as a parent-child relationship between EISs. Cascades are referred to in the SFM as “XEIS”.

EISs may also be associated in a reciprocal relationship by registering their (Source ID, Entity ID) pairs with each other. This is also known as a peer-to-peer relationship between EISs.

### 7.2.2 Register and Find

The bulk of the work of the EIS is performed by two operations, the register (`RegisterEntityWithIdentity`) operation and the find (`FindIdentitiesByTraits`) operation. In the register operation, a Source ID/Entity ID pair is supplied to the EIS for matching and linking with existing Identities. In the query operation, Identities are retrieved based upon matches of Identities or their associated Traits to query parameters.

The semantics of both the register and find operations depends largely upon the state of the EIS at the time the operation is invoked. The state in which the EIS was found is returned in the operation parameter of type `EISState`. Details of operation semantics associated with different EIS states are discussed in the descriptions of the behavior of operations, below.

### 7.2.3 Duplicates

Each RWE should be associated with only one Identity by an EIS. A state in which two or more Identities may represent the same RWE indicates a possible duplicate. Unlink, link and merge operations should be used to correct this situation.

## 8 EISManagementAndQueryInterface

The EIS Management and Query Interface is modeled below:

Operation	Returns	Parameters
RegisterEntityWithIdentity	EISStatus	in entityId: string(idl) in sourceId : string(idl) in semanticSignifierName : string(idl) in traits: EISSemanticSignifier out eisId : string(idl)
CreateIdentityFromEntity	EISStatus	in sourceId : string(idl) in semanticSignifierName : string(idl) in traits: EISSemanticSignifier out entityId : string(idl) out eisId : string(idl)
UpdateEntityTraitValues	EISStatus	in entityId: string(idl) in sourceId : string(idl) in updateMethod : EISUpdateMethod in semanticSignifierName : string(idl) in traits: EISSemanticSignifier
RemoveIdentity	EISStatus	in entityId: string(idl) in sourceId : string(idl) in semanticSignifierName : string(idl) in traits: EISSemanticSignifier
GetEntityTraitValues	EISStatus	in entityId: string(idl) in sourceId : string(idl) in semanticSignifierName : string(idl) in traits: EISSemanticSignifier out traits: EISSemanticSignifier
FindIdentitiesByTraits	EISStatus	in entityId: string(idl) in sourceId : string(idl) In eisId : string(idl) in semanticSignifierName : string(idl) in traits: EISSemanticSignifier in searchQualifier : EISSearchQualifier out matchingResultSet : EISResultSet
ListLinkedIdentities	EISStatus	in entityId: string(idl) in sourceId : string(idl)



		in searchQualifier : EISSearchQualifier in semanticSignifierName : string(idl) in traits: EISSemanticSignifier out matchingResultSet : EISResultSet
ListUnlinkedIdentities	EISStatus	in entityId: string(idl) in sourceId : string(idl) in searchQualifier : EISSearchQualifier in semanticSignifierName : string(idl) in traits: EISSemanticSignifier out matchingResultSet : EISResultSet

## 8.1 RegisterEntityWithIdentity

### Behavior

This operation inserts a Source ID/Entity ID pair and supplied Traits into the EIS with implicit linking to other matching Source ID/ Entity ID pairs, based on the configured internal matching algorithm.

It is the normal operation for the insertion of new Entity data into the EIS.

A full discussion of the structure and content of EISState (the OUT parameter) and EISStatus (the Return value) can be found in Chapter 6 of this specification.

### Parameters

- entityId: String (IN)
- sourceId : String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)
- eisId : STRING (OUT)
- eisState : EISState (OUT)

**Issue 13934 – here: replace the following table**

### Semantics of OUT Parameters

The following truth table defines all the possible state changes resulting from operations in an EIS which create or update Identities:

EIS starting state (eisState)						EIS ending state			Semantics
Input matches single Identity						Input matches multiple Identities	input Source ID/ input Entity ID	Duplicate Detected?	Meaning
EIS ID	Input Source ID /input Entity ID	Input Source ID/ other Entity ID	Other Source ID/ any Entity ID		EIS ID				
1	0	0	0	0	0	1	1	0	Record does not exist in the EIS currently. Identity is created with EIS ID. Input Source

									ID/input Entity ID and input Traits are recorded in EIS.
2	0	0	0	1	0	1	1	0	Indicates a match to an "unlinked" input record, such as one waiting for manual review. Put record in a queue for manual review.
3	0	0	1	0	0	1	1	1	
4	0	0	1	1	0	1	1	1	
5	0	1	0	0	0	1	1	0	
6	0	1	0	1	0	1	1	0	
7	0	1	1	0	0	1	1	1	
8	0	1	1	1	0	1	1	1	
9	1	0	0	0	0	1	1	0	This can happen if meta-traits are associated with EIS ID and the input matched the meta-traits but not any actual trait set from which the meta-traits were derived. Input Source ID/input Entity ID is linked with EIS Identity.
10	1	0	0	1	0	1	1	0	Single match found. Input Source ID/input Entity ID is linked with EIS Identity.
11	1	0	1	X	0	1	1	1	Single match found. Input Source ID/input Entity ID is linked with EIS Identity.
12	1	1	0	X	0	1	1		Single match found. Traits are updated.
13	1	1	1	X	0	1	1	1	Single match found. Traits are updated.
14	X	X	0	X	1	1	1	0	Multiple matches found. Queue for manual review.
15	X	X	1	X	1	1	1	1	Multiple matches found. Queue for manual review.

**Return**

- EISStatus

**Status Conditions**

<b>code</b>	<b>severity</b>	<b>Message</b>
1002	Info	Entity has been successfully registered with EIS ID '%1' in the EIS
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS
2001	Error	The mandatory Trait '%1' for the Entity Type '%2' was not provided
2002	Error	The Trait identified by '%1' does not exist in the EIS
2003	Error	The Identifying Trait list provided contains a duplicate Trait '%1'
2004	Error	An invalid value was passed for the Trait '%1'
2005	Error	An entity with that ID already exists
3001	Warning	The Entity referenced by the ID information provided may already exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 8.2 CreateIdentityFromEntity

### Behavior

This operation generates an Entity ID on behalf of a Source that does not generate Entity IDs. It inserts the Source ID/newly-generated Entity ID pair and supplied Traits into the EIS with implicit linking to other matching Source ID/Entity ID pairs, based on the configured internal matching algorithm.

This operation may be used in environments where the RegisterEntityWithIdentity cannot be used because the Source does not assign Identifiers to Entities.

### Parameters

- systemIdSourceId : SYSTEMID (IN)String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)
- entityId : String (OUT)
- eisState : EISState (OUT)

### Return

- EISStatus

### Status Conditions

code	severity	Message
1001	Info	An Entity ID was successfully created in the EIS
1002	Info	Entity has been successfully registered with EIS ID '%1' in the EIS
2001	Error	The mandatory Trait '%1' for the Entity Type '%2' was not provided
2002	Error	The Trait identified by '%1' does not exist in the EIS
2003	Error	The Identifying Trait list provided contains a duplicate Trait '%1'
2004	Error	An invalid value was passed for Trait '%1'
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS
3001	Warning	The Entity referenced by the ID information provided may already exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 8.3 UpdateEntityTraitValues

### Behavior

This operation updates the Traits stored in the EIS for the Entity identified by the supplied Source ID/Entity ID pair. Whether this entails re-analyzing matching identities is implementation specific, although it would be done in most cases. How it would be done is also implementation specific.

### Parameters

- entityId: String (IN)
- sourceId : String (IN)
- eisUpdateQualifier : EISUpdateQualifier(IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)
- eisState : EISState (OUT)

**Return**

- EISStatus

**Status Conditions**

code	severity	Message
1003	Info	Entity Trait Values have been successfully updated in the EIS
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS
2002	Error	The Trait identified by '%1' does not exist in the EIS
2003	Error	The Identifying Trait list provided contains 1 or more duplicates
2004	Error	An invalid value was passed for the Trait '%1'
2005	Error	Attempt to delete 1 or more mandatory Traits for the Entity Type '%1'
2006	Error	1 or more Traits passed in the list are not valid or do not exist for Entity Type '%1'
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 8.4 RemoveIdentity

### Behavior

This function deletes a Source ID/Entity ID pair and its associated Traits from the EIS repository. This operation is designed to correct a true invalid data condition, such as the data entry of practice or testing data.

All of the operations in the EISAdminInterface are the normal operations to deal with changes and corrections to otherwise valid data.

### Parameters

- entityId: String (IN)
- sourceId : String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)

### Return

- EISStatus

### Status Conditions

code	severity	message
1004	Info	Entity reference has been successfully removed in the EIS
2007	Error	Entity Referenced does not exist in the EIS
3002	Error	Wrong existing state for deletion
2100	Error	Identity linked to other identities
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 8.5 GetEntityTraitValues

### Behavior

Retrieves the Traits associated with a Source ID/Entity ID pair.

### Parameters

- entityId: String (IN)
- sourceId : String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (OUT)

### Return

- EISStatus

### Status Conditions

code	severity	message
1005	Info	Entity Trait values retrieved successfully
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS
3001	Error	Entity Referenced does not exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 8.6 FindIdentitiesByTraits

### Behavior

This operation provides the means to perform a broad search of all records in the EIS whose Traits match some criteria in the supplied search criteria (such as find all records who match the name “Jones, Bob”).

This operation is equivalent to the IHE PDQ transaction.

**Issue 13934 – here: remove spurious hyperlink “[ds1]”**

### Parameters

- entityId: String (IN)
- sourceId : String (IN)
- eisId : String (IN)
- searchQualifier : EISSearchQualifier (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)
- matchingResultSet : EISResultSet (OUT)
- eisState : EISState (OUT)

**Issue 13934 – here: replace the following table**

### Semantics of OUT Parameters

What is matched					semantics			
Input matches single EIS ID					meaning	ID re- turned	Other IDs returned	(demographic) traits returned
input EIS ID matches EIS ID	input source ID /input Entity ID matches EIS Identity	traits match those linked with any Identity	Input matches additional EIS ID					
1	1	X	X	0	EIS ID was submitted & match is found	depends upon request - can return this EIS ID	depends upon request - can return any or all other IDs	depends upon request - can return any or all traits, at different (parent-child) levels
2	X	1	X	0	source ID/Entity ID was submitted & match is found			
3	X	X	1	X	Traits were submitted & match is found			
4	0	0	0	0	no matches	nothing returned	nothing returned	nothing returned
5	1	X	0	1	error			



6	X	1	0	1	error			
---	---	---	---	---	-------	--	--	--

### Return

- EISStatus

### Status Conditions

code	severity	message
1006	Info	'%1' Entities successfully retrieved
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS
2002	Error	The Trait identified by '%1' does not exist in the EIS
2007	Error	Entity Referenced does not exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 8.7 ListLinkedIdentities

### Behaviors

This operation retrieves all the Source ID/Entity ID pairs that are linked to the supplied Source ID/Entity ID pair. The operation can be filtered with the sourceConstraintSet property of the EISSearchQualifier to only return entities within specified Source domains.

This operation is equivalent to the IHE PIX transaction.

### Parameters

- entityId: String (IN)
- sourceId : String (IN)
- searchQualifier : EISSearchQualifier (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)
- matchingResultSet : EISResultSet (OUT)

### Return

- EISStatus

### Status Conditions

code	severity	message
1007	Info	'%1' Linked Entities successfully retrieved
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS

<b>2007</b>	<b>Error</b>	<b>Entity Referenced does not exist in the EIS</b>
<b>2002</b>	<b>Error</b>	<b>The Trait identified by '%1' does not exist in the EIS</b>
<b>2011</b>	<b>Error</b>	<b>Identity Qualifier '%1' does not exist in the EIS</b>
<b>2977</b>	<b>Error</b>	<b>The Domain identified by SourceId '%1' does not exist in the EIS</b>

## 8.8 ListUnlinkedIdentities

### Behavior

This operation retrieves dublicately specified, yet unlinked records in the EIS repository. It filters the record set to only return matching, but unlinked records. The operation can be filtered with sourceConstraintSet to only return entities within specified Source domains.

### Parameters

- entityId: String (IN)
- sourceId : String (IN)
- searchQualifier : EISSearchQualifier (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)
- matchingResultSet : EISResultSet (OUT)

### Return

- EISStatus

### Status Conditions

code	severity	message
1008	Info	'%1' UnLinked Entities successfully retrieved
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS
2007	Error	Entity Referenced does not exist in the EIS
2002	Error	The Trait identified by '%1' does not exist in the EIS
2011	Error	Identity Qualifier '%1' does not exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 9 EISAdminEditorInterface

The EIS Administration Editor Interface is summarized in the model below:

Operation	Returns	Parameters
<b>LinkEntities</b>	<b>EISStatus</b>	in sourceEntityId : string(idl) in sourceSourceId : string(idl) in targetEntityId : string(idl) in targetSourceId : string(idl) in reasonCode : string(idl) in semanticSignifierName: string(idl) in traits : EISSemanticSignifier
<b>UnlinkEntities</b>	<b>EISStatus</b>	in sourceEntityId : string(idl) in sourceSourceId : string(idl) in targetEntityId : string(idl) in targetSourceId : string(idl) in reasonCode : string(idl) in semanticSignifierName: string(idl) in traits : EISSemanticSignifier
<b>MergeEntities</b>	<b>EISStatus</b>	in sourceEntityId : string(idl) in targetEntityId : string(idl) in targetSourceId : string(idl) in reasonCode : string(idl) in semanticSignifierName: string(idl) in traits : EISSemanticSignifier
<b>UnMergeEntities</b>	<b>EISStatus</b>	in sourceEntityId : string(idl) in targetEntityId : string(idl) in targetSourceId : string(idl) in reasonCode : string(idl) in semanticSignifierName: string(idl) in traits : EISSemanticSignifier
<b>ActivateEntity</b>	<b>EISStatus</b>	in entityId : string(idl) in sourceId : string(idl) in reasonCode : string(idl) in semanticSignifierName: string(idl) in traits : EISSemanticSignifier
<b>DeactivateEntity</b>	<b>EISStatus</b>	in entityId : string(idl) in sourceId : string(idl)

		<b>in reasonCode : string(id)</b> <b>in semanticSignifierName: string(id)</b> <b>in traits : EISSemanticSignifier</b>
--	--	-----------------------------------------------------------------------------------------------------------------------------

# 9.1 LinkEntities

## Behavior

This operation provides the means to create an explicit linking between the source and target record IDs in the EIS repository.

### Parameters

- sourceEntityId : String (IN)
- sourceSourceId : String (IN)
- targetEntityId : String (IN)
- targetSourceId : String (IN)
- reasonCode : String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)

### Return

- EISStatus

### Status Conditions

code	severity	message
1010	Info	Entities successfully linked
2020	Error	Source Entity Referenced does not exist in the EIS
2021	Error	Target Entity Referenced does not exist in the EIS
2022	Error	Reason Code is not in a valid format or does not exist in the EIS
2023	Error	The Entity Type identified by '%1' does not exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

# 9.2 UnlinkEntities

## Behavior

This operation provides the means for explicitly breaking the link between the source and target record IDs in the EIS repository.

### Parameters

- sourceEntityId : String (IN)
- sourceSourceId : String (IN)
- targetEntityId : String (IN)
- targetSourceId : String (IN)
- reasonCode : String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)

**Return**

- EISStatus

**Status Conditions**

code	severity	message
1011	Info	Entities successfully unlinked
2020	Error	Source Entity Referenced does not exist in the EIS
2021	Error	Target Entity Referenced does not exist in the EIS
2022	Error	Reason Code is not in a valid format or does not exist in the EIS
2023	Error	The SemanticSignifier identified by '%1' does not exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 9.3 MergeEntities

**Behavior**

This operation provides the means to explicitly consolidate EIS member records in the EIS repository. The source record is effectively merged into the target, leaving only the resulting target record in the EIS repository. Identifying attributes in the target which are empty are filled from the source, and existing attributes in the target remain AS-IS. The source record is inactivated in the EIS repository at the successful end of this operation.

**Parameters**

- sourceEntityId : String (IN)
- sourceSourceId : String (IN)
- targetEntityId : String (IN)
- targetSourceId : String (IN)
- reasonCode : String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)

**Return**

- EISStatus

**Status Conditions**

code	severity	message
1012	Info	Entities successfully merged
2020	Error	Source Entity Referenced does not exist in the EIS
2021	Error	Target Entity Referenced does not exist in the EIS

2022	Error	Reason Code is not in a valid format or does not exist in the EIS
2023	Error	The SemanticSignifier identified by '%1' does not exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 9.4 UnMergeEntities

### Behavior

#### Parameters

- sourceEntityId : String (IN)
- sourceSourceId : String (IN)
- targetEntityId : String (IN)
- targetSourceId : String (IN)
- reasonCode : String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)

#### Return

- EISStatus

### Status Conditions

code	severity	message
1013	Info	Entities successfully unmerged
2020	Error	Source Entity Referenced does not exist in the EIS
2021	Error	Target Entity Referenced does not exist in the EIS
2022	Error	Reason Code is not in a valid format or does not exist in the EIS
2023	Error	The SemanticSignifier identified by '%1' does not exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 9.5 ActivateEntity

### Behavior

This operation marks the record as “active” in the EIS repository, effectively marking it as logically un-deleted.

#### Parameters



- entityId : String (IN)
- sourceId : String (IN)
- reasonCode : String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)

#### Return

- EISStatus

#### Status Conditions

code	severity	message
1014	Info	Entity successfully activated
2007	Error	Entity Referenced does not exist in the EIS
2002	Error	The Trait identified by '%1' does not exist in the EIS
2022	Error	Reason Code is not in a valid format or does not exist in the EIS
2023	Error	The SemanticSignifier identified by '%1' does not exist in the EIS
2977	Error	The Domain identified by SourceId '%1' does not exist in the EIS

## 9.6 DeactivateEntity

#### Behavior

This operation marks the record as “inactive” in the EIS repository, effectively marking it available for Get() or List() operations at the entity level but not from the EIS repository.

**Issue 13934 – here: remove spurious hyperlink “[ds2]”**

#### Parameters

- entityId : String (IN)
- sourceId : String (IN)
- reasonCode : String (IN)
- semanticSignifierName : String (IN)
- traits: EISSemanticSignifier (IN)

#### Return

- EISStatus

#### Status Conditions

Code	severity	message
1015	Info	Entity successfully deactivated

<b>2002</b>	<b>Error</b>	<b>The Trait identified by '%1' does not exist in the EIS</b>
<b>2007</b>	<b>Error</b>	<b>Entity Referenced does not exist in the EIS</b>
<b>2022</b>	<b>Error</b>	<b>Reason Code is not in a valid format or does not exist in the EIS</b>
<b>2023</b>	<b>Error</b>	<b>The SemanticSignifier identified by '%1' does not exist in the EIS</b>
<b>2977</b>	<b>Error</b>	<b>The Domain identified by SourceId '%1' does not exist in the EIS</b>

## 10 EISMetaDataInterface

The EIS Meta Data Interface is summarized in the model below:

Operation	Returns	Parameters
CreateSemanticSignifier	EISStatus	in semanticSignifierDefinition : EISSemanticSignifierDefinition
FindSemanticSignifier	EISStatus	in semanticSignifierName : string(idl) out semanticSignifierDefinition : EISSemanticSignifierDefinition
UpdateSemanticSignifier	EISStatus	inout semanticSignifierDefinition : EISSemanticSignifierDefinition
ListSemanticSignifiers	EISStatus	In entityTypeName : string(idl) out listOfSemanticSignifierNames : sequence(idl)
ListDomains	EISStatus	out listOfDomains : sequence(idl)
ListConformanceProfiles	EISStatus	out conformanceProfiles : arrayOf[string(idl)]

### 10.1 CreateSemanticSignifier

#### Behavior

Creates a semantic signifier definition in the EIS meta-data repository.

#### Parameters

- semanticSignifierDefinition: EISSemanticSignifierDefinition (IN)

#### Return

- EISStatus

#### Status Conditions

code	severity	message
1032	Info	SemanticSignifier successfully created
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS (This condition is optional and based on if the SemanticSignifier is passed)
2022	Error	A SemanticSignifier identified by '%1' already exists in the EIS

### 10.2 FindSemanticSignifier

#### Behavior

Locates and returns a semantic signifier definition from the EIS meta-data repository.

#### Parameters

- semanticSignifierName : String (IN)

- semanticSignifierDefinition : EISSemanticSignifierDefinition (OUT)

**Return**

- EISStatus

**Status Conditions**

<b>code</b>	<b>severity</b>	<b>message</b>
<b>1033</b>	<b>Info</b>	<b>Semantic Signifier successfully retrieved</b>
<b>2023</b>	<b>Error</b>	<b>A Semantic Signifier identified by '%1' does not exist in the EIS</b>

## 10.3 UpdateSemanticSignifier

### Behavior

Updates a semantic signifier definition in the EIS meta-data repository.

### Parameters

- semanticSignifierName : String (IN)
- semanticSignifierDefinition: EISSemanticSignifierDefinition (IN)

### Return

- EISStatus

### Status Conditions

code	severity	message
1034	Info	SemanticSignifier successfully updated
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS (This condition is optional and based on whether the semanticSignifier is passed)

## 10.4 ListSemanticSignifiers

### Behavior

List the traits registered with the system.

### Parameters

- entityTypeName : String (IN)
- listOfSemanticSignifiers : List<String> (OUT)

### Return

- EISStatus

### Status Conditions

code	severity	message
1035	Info	SemanticSignifiers successfully listed
2023	Error	The Semantic Signifier identified by '%1' does not exist in the EIS (This condition is optional and based on whether the semanticSignifier is passed)

## 10.5 ListDomains

### Behavior

List the domain hierarchy configured in the EIS meta-data repository.

**Parameters**

- listOfDomains : List<String> (OUT)

**Return**

- EISStatus

**Status Conditions**

code	severity	message
1024	Info	Domains successfully listed

## 10.6 ListConformanceProfiles

**Behavior**

Returns the list of conformance profile names and versions that this implementation of EIS supports.

**Parameters**

- listOfConformanceProfiles: arrayOf[ConformanceProfile (OUT)

**Issue 13934 – here: remove spurious hyperlink “[ds3]”**

**Return**

- EISStatus

**Status Conditions**

code	severity	message
1025	Info	Conformance Profiles successfully listed

# 11 EIS Data Structures

## 11.1 Complex Data Structures

### 11.1.1 EISSemanticSignifierDefinition

#### 11.1.1.1 Overview

Meta-data structure for definition of a data type processed by EIS.

#### 11.1.1.2 Structure

An Instance of EISSemanticSignifierDefinition has the following properties:

- semanticSignifierName : String The label for the SemanticSignifier
- identifier : string(idl) A Globally unique alphanumeric ID or GUID that uniquely identifies a SemanticSignifier
- versionIdentifier : String A Version label for the SemanticSignifier
- entityTypeName : String The EntityName that the SemanticSignifier represents. Also serves to categorize SemanticSignifiers by *type*.
- schemaDefinition : String A URI to an XSD or other meta-document that describes the definition of the SemanticSignifier schema. A given instance of a SemanticSignifier may be optionally validated against the document pointed to by this URI
- renderingDefinition : String A URI to an XSL or other suitable resource (e.g., a Java Class) that can be utilized to extract Traits from an instance of this SemanticSignifier
- integrityRulesetDefinition : String A URI to a Schematron or WSDL reference that can serve to make integrity assertions against a given instance of a SemanticSignifier

### 11.1.2 EISSemanticSignifier

#### 11.1.2.1 Overview

Instance of the Schema Definition in EISSemanticSignifierDefinition

#### 11.1.2.2 Structure

An Instance of EISSemanticSignifier has the following single element:

- document : String The document or source (e.g., raw XML, HTML, structured text, etc) of an instance of a SemanticSignifier

### 11.1.3 EISSearchQualifier

#### 11.1.3.1 Overview

This data structure contains optional search qualifiers that can be used by the Find operation to more finely control the behavior of the search algorithm. It is an optional parameter and if not supplied, the internal configuration of the EIS engine will dictate all search behavior.

Structure

#### 11.1.3.2 Structure

An instance of EISSearchQualifier is a complex data structure with the following elements:

- **matchingModuleName** : String The name of the EIS Matching algorithm name to be used. If this value is blank, the default configured for the EIS will be used
- **searchConfidenceLevel** : String The search confidence level to be used for matching (0..1, or % threshold). For search operations, this value defaults to blank which indicates to the EIS that the default value configured for the EIS is to be used
- **maxResultSetSize** : number The maximum number of results to be returned from the given API call. This value defaults to 0 which indicates that the EIS may arbitrarily set the limit to some configured value
- **raiseErrorIfLimitReached** : Boolean A flag that indicates that if the value set in the **maxResultSetSize** property is reached, an Error StatusCondition is to be returned, indicating to the caller that the threshold they set was reached. This value defaults to *false*
- **sourceConstraintSet**: List<String> A list of SourceId strings that serve to constrain the results of a Search or List operation to a set of existing identity domains

## 11.1.4 EISUpdateQualifier

### 11.1.4.1 Overview

The EISUpdateQualifier is meant to contain elements that are used to limit, constrain or otherwise alter the behavior of the API update operation involved

### 11.1.4.2 Structure

The EISUpdateQualifier structure consists of the following elements:

- **updateMode** : enum An enumeration indicating a mode to be utilized during the update process. Values include:
  - OVERWRITE (0) Replaces the entire set of data elements with the entire supplied set. Null values in the submitted data will replace existing non-null values.
  - VALUED (1) Replaces only those data elements that are valued in the input data. Null values in the submitted data will not replace existing non-null values.
  - UNSPECIFIED (2) Implementation specific.
- **updateSchemaDefinition** : String A URI to an XSL or other suitable resource (e.g., a Java Class) that can be utilized to extract Traits from an instance of this SemanticSignifier

## 11.1.5 EISStatus

### 11.1.5.1 Overview

All operations executed via the EIS PIM API have an OUTPUT parameter of type EISStatus. Upon return from an API operation, the returned EISStatus structure will contain information that indicates the outcome (success or failure) of the preceding API operation.

### 11.1.5.2 Structure

The EISStatus structure consists of 3 attributes. They are as follows:

- **statusSuccess** : Boolean A flag representing the success or failure of the API operation. True indicates that the operation was successful and false indicates a non-successful operation
- **statusConditions** : Array<StatusCondition> An set of the StatusCondition structure representing all the conditions that occurred during the API call



### 11.1.5.3 StatusCondition component

- **statusCode** : String A 4 digit numeric code representation of the outcome of the API operation.
- **statusSeverity** : enum An enumeration that indicates the severity of the status-code returned. Possible values are:
  - Informational (0) – Used to relay to the client useful information about the outcome of the call. Informational messages are usually indications of successful operations, but other useful purposes do exist.
  - Warning (1) – Operation completed, but some portion did not complete as expected.
  - Error (2) – Operation was unable to complete for identifiable data or software problems.
  - Severe (3) – Operation was unable to complete for system problems, such as “resource allocation failure”, or “hardware failure”, or unidentifiable data or software problems.
- **statusMessage**: String An alpha-numeric message describing the outcome of the API operation
- **statusDetail** : String A detailed message describing the outcome of the API operation. This element is optional

### 11.1.5.4 StatusCode component

EISStatus StatusCode instances have two dimensions. First, each StatusCode is unique and represents a single specific condition that may occur within the EIS. Second, StatusCodes are numeric and their range will indicate a class of conditions that has occurred. For example, status-codes in the range of 1000-1999 indicate a successful operation.

Since there will be a number of implementers of the EIS specification, there must be the capability for implementers to indication server conditions that are specific to a given implementation.

Range	Condition Class	Example
1000-1999	Successful operations	<b>1032 - A SemanticSignifier was successfully created in EIS</b> (In the case where an implementer is opting to standardize on using a single message to indicate a successful operation, the status-code of 1000 is to be used and the text of the message should be standardized as per the EIS spec. “Operation Succeeded”, for instance)
2000-2999	Client side or parameter specific conditions	<b>2002 - The Trait identified by ‘trait_ssn’ does not exist in the EIS</b>
3000-3999	Service side conditions not related to parameter data	<b>3001 - The Entity referenced by the provided information may already exist in the EIS</b>
4000-4999	Implementer specific error conditions not covered by specification covered errors	<b>4001 – The specified query hint ‘use-fuzzy’ is not a valid hint</b>

## 11.1.6 EISState

### 11.1.6.1 Overview

Certain operations executed via the EIS PIM API have an OUTPUT parameter of type EISState. Upon return from an API operation that registers or queries an Entity, this structure will contain detailed information about certain starting (or pre-operation) and ending (or post-operation) states of the EIS with respect to the identifiers of the Entity passed in the API operation.

Each attribute of the EISState structure, referred to individually as *sub-states*, may have take on one of three values: Exists (1), Doesn't Exist (0), May or may not exist (-1)

### 11.1.6.2 Structure

The EISState structure consists of the following elements each a *flag* or *sub-state* indicator:

- eisMediatingIdStartState : Boolean
- inputSourceIdStartState : Boolean
- inputSourceOtherIdStartState : Boolean
- otherSourceAnyIdStartState : Boolean
- eisMediatingIdEndState : Boolean
- inputSourceIdEndState : Boolean
- duplicateDetected : Boolean

### 11.1.7 EISResultSet

#### 11.1.7.1 Overview

A structure that contains the results of a Search or Find operation. EISResultSet consist of a list of the EISResultSetElement structure. Each instance of EISResultSetElement contains the information about an

#### 11.1.7.2 Structure

The EISResultSet structure consists of the following element:

- eisResultSetElements : List<EISResultSetElement>

#### 11.1.7.3 EISResultSetElement structure

- entityId : String The entityId of the Identity
- sourceId : String The sourceId of the Identity
- eisId : String The identifier used by the EIS to correlate between identities
- semanticSignifierName : String The name of the SemanticSignifier which describes the content of the EISSemanticSignifier instance of the *traits* element below
- traits : EISSemanticSignifier An instance of a SemanticSignifier containing the Traits for the identity

## 11.2 Other Simple Data Types

EIS Data Type	Description
<b>EntityTypeName</b>	<b>This is a string representing an enumeration of Entity Types supported by the EIS. A simple example would be "PERSON," but other entities such as "BEDS" or "SPECIMENS" could also be supported within a single EIS.</b>
<b>Identifiers</b>	<b>All identifiers referenced are intended to be GUIDs, including Source ID, Entity ID, and EIS Entity ID.</b>
<b>ConformanceProfiles</b>	<b>As defined by relevant standards</b>

### 11.3 Supplemental information on EIS Error Handling

- All EIS API calls will return the specified status object that represents the outcome (success/failure, code, message, etc) of what occurred during the API call.
- All EISAPI calls will return an instance of an EISStatus structure. Example method signature: *EISStatus API\_call(param1, param2, ... paramN)*
- The EISStatus structure will indicate the success or failure of the API call
- The EISStatus structure will return a set of StatusCondition structures in a property named *statusConditions* that summarize in greater detail the success or failure of the API call. Each instance of the StatusCondition structure will contain information pertaining to an individual condition encountered during the API call.
- For successful API calls, while there will typically be a single StatusCondition instance returned in the *statusConditions* property of the EISStatus structure
- For successful API calls, implementers may choose to return a single generic INFO level StatusCondition that indicates a successful operation (e.g., *OPERATION\_SUCCESSFUL*) or they may utilize the provided API call specific INFO level messages
- For unsuccessful API calls, implementers must return at least one instance of StatusCondition corresponding one of the total set of conditions encountered during the call. The implementer may also choose to return (n) StatusCondition instances that fully communicate all the individual conditions encountered during the call
- Implementers choosing to implement a single StatusCondition for each API call should give priority to *Errors* over the *Info* and *Warning* severities. However, this choice is up to the implementer. They may choose to simply return the first error encountered by whatever error checking algorithm they implement.
- Each StatusCondition will have a *statusSeverity* enumeration property. The severity enumeration will consist of INFO, WARN, ERROR and SEVERE. INFO is considered to have the lowest severity and SEVERE to have the highest
- Error conditions are to be ordered by *statusSeverity* and within severity from generic to specific. Example: An EIS CreateIdentityFromEntity operation:
  - Condition 0 - Severe: storage error occurred
  - Condition 1 - Error: invalid SemanticSignifier value (specific)
  - Condition 2 - Error: missing one or more required SemanticSignifier fields (generic)
- There is no inherent mechanism provided within the array of StatusCondition instances returned within the EISStatus structure to relate any given StatusCondition instance to another. Implementers may choose to imply or hint at these types of relationships via the *statusDetail* property of the StatusCondition structure
- EIS implementers can optionally provide localized the *StatusMessage* values and in the absence of a locale, return messages in the default locale (language) for that server instance

## 12 Semantic Profiles

**Issue 13934 – here: remove spurious hyperlink “[ds4]”**

This is a normative reference of implementing the EIS service interface using either HL7 v3 RIM-based structures or HL7 v2.x structures:

- a) A Person EIS in HL7 v2.x
- b) A Person EIS in HL7 v2.x using the IHE Pediatric Demographic Option

**NOTE: All machine readable files are available in OMG document number health/xxx**

### 12.1 Profile alignment by Service Interface and Operations

**Issue 13934 – here: remove spurious hyperlink “[ds5]”**

What this profile does is line up the platform independent model to create a specific service interface which supports the various EIS operations but does so by creating a specific WSDL for each data type (Provider, Patient, etc.) and then types the semantic signifier payload to the specific operations.

This could also be done more generically, with an EIS-HSSP interface with only generic signifiers to represent the data payloads and then the caller could dynamically determine the required schema definition by inspecting it at run-time via a Describe() operation. This is similar to using an *any or variant* data type instead of a specific structure to represent the data payload.

The method used below is more type-safe and has some advantages when using modern web service development tools since the more type explicit approach allows the development tools to create object wrappers around the explicit data types, where the generic approach requires the developer to write code against the XML text format of the semantic signifier directly.

### 12.2 Implementations and Semantic Signifiers

The EIS interface in WSDL as shown in the next section can process any of the following (in document or message form) which can be normalized in an XSD with an optionally associated Schematron:

- Care Record
- CDA
- Patient Administration data

The key differences in supporting another semantic signifier type include:

- a) the URI XSD which is held in the `EIStypes:EISsemanticSignifier` meta-data structure which describe the structure of the data to be handled by the interface
- b) The `<xs:element ref="cda:ClinicalDocument" minOccurs="1">` element in the Get(), Put(), List() parameter which corresponds to the `traits(s)` parameter in the platform independent model.

So, for example to support the care record message or a MS-Lab content profile, an EIS service would:

- Have its metadata registry configured to point to a URI of an appropriate defined XSD to meet requirement (a) and then
- Have a specific WSDL would be defined incorporating that XSD as a specific type of the `traits(s)` parameter in each applicable SOAP request and response satisfying the requirement listed in (b) above.

This approach allows for a flexible and loosely coupled deployment of an EIS service which supports many different kinds of data types.

## 12.3 Implementations

The necessary WSDLs for supporting EIS are including as separate machine-readable documents for the example shown below.

### 12.3.1 Implementation Example – HL7 Version 2.5 Person using IHE Pediatric Demographic option

#### 12.3.1.1 Definition of Traits using HL7 V2.5 Segments

A mapping of Entity Traits to HL7 V2.5 fields is given in Table 1 and Table 2. The SFM included an initial Semantic Profile which identified a list of traits, modeled after those required by IHE PDQ Query, and allows for profiles to define additional traits. This profile defines additional traits as shown.

FLD	ELEMENT NAME
<b>Traits (as defined in SFM)</b>	
PID.3	Patient Identifier List
PID.5	Patient Name
PID.7	Date/Time of Birth
PID.8	Administrative Sex
PID.11	Patient Address
<b>Additional Traits</b>	
PID.6	Mother's Maiden Name
PID.13	Patient Home Telephone Number
PID.24	Patient Multiple Birth Indicator
PID.25	Patient Birth Order
PID.33	Last Update Date / Time
PID.34	Last Update Facility

Table 1: Traits for Patient Feed

FLD	ELEMENT NAME
<b>Traits (as defined in SFM)</b>	
<b>PID.3</b>	<b>Patient Identifier List</b>
<b>PID.5</b>	<b>Patient Name</b>
<b>PID.7</b>	<b>Date/Time of Birth</b>
<b>PID.8</b>	<b>Administrative Sex</b>
<b>PID.11</b>	<b>Patient Address</b>
<b>PID.18</b>	<b>Patient Account Number</b>
<b>Additional Traits</b>	
<b>PID.6</b>	<b>Mother's Maiden Name</b>
<b>PID.13</b>	<b>Patient Home Telephone Number</b>

Table 2: Traits Used by Query

See The IHE discussions in IHE IT Infrastructure Technical Framework, vol. 2 (ITI TF-2): Transactions, pp. 199-213, Transaction “ITI-21” for the Patient Demographics Query (PDQ).

Note that the Patient Identifier List PID.3 will support the identifiers of Table 3. Also note that these identifiers are a CX data type, and the issuing authority must be included, particularly for identifiers that don't imply one particular issuing authority. In this Implementation Example, OIDs are used to identify the issuing authority as specified by IHE standards.

ID Type (from table User Defined 0203)	Description
<b>SS</b>	<b>Social Security Number</b>
<b>MA</b>	<b>Medicaid Number</b>
<b>MR</b>	<b>Medical Record Number</b>
<b>BR</b>	<b>Birth Registry Number</b>
<b>LR</b>	<b>Local Registry Number</b>
<b>RRI</b>	<b>Regional Registry Number</b>
<b>SR</b>	<b>State Registry Number</b>
<b>WC</b>	<b>WIC ID</b>
<b>XX</b>	<b>Organization Identifier</b>

Table 3: Identifiers Supported By PID.3

The list of patients returned by queries will be ordered, in descending sequence, by whatever means is available to the query fulfiller, so that the most likely match is shown first.

## 12.3.1.2 EISManagementAndQueryInterface

### 12.3.1.2.1 RegisterEntityWithIdentity

The RegisterEntityWithIdentity operation uses parameters structured as an HL7 ADTA28 message with ACK response.

It has the same message structure as CreateIdentityFromEntity (see below), but requires that the Entity ID be furnished.

See The IHE discussions in IHE IT Infrastructure Technical Framework, vol. 2 (ITI TF-2): Transactions, pp. 41-50 about Transaction “ITI-30” which can be found at [www.ihe.net](http://www.ihe.net) for full details.

Also See HL7 discussion of the ADTA28 message on page 3-36.

ADTA28 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- EVN
- PID
- [PD1] – for Publicity Code, Protection Indicator and Effective Date, Immunization Registry Status and Date
- [{NK1}] – repeats for mother / father & related info, as required
- PV1 – required for Patient Class, Point of Care, Primary Provider
- PV2 – for Patient Status

	PIM	PSM
<b>Inputs</b>	<ul style="list-style-type: none"> <li>➤ SourceID</li> <li>➤ EntityID.</li> <li>➤ Unordered set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</li> <li>➤</li> </ul>	<ul style="list-style-type: none"> <li>➤ This is mapped to MSH.4 in HL7 v2.5.</li> <li>➤ This is mapped to PID.3, first element in the list</li> <li>➤ Use a standard HL7 v2.5 ADT message to hold the traits.</li> <li>➤</li> </ul>
<b>Outputs</b>	<ul style="list-style-type: none"> <li>➤ Acknowledgement</li> </ul>	<ul style="list-style-type: none"> <li>➤ HL7 ACK.mapped into EISStatus</li> </ul>
<b>Error Conditions</b>	<ul style="list-style-type: none"> <li>➤ Entity reference already exists (error based on ID match)</li> <li>➤ Entity reference may already exist (warning) – returns ID(s) and set of traits for matched identity</li> <li>➤ Mandatory traits not supplied</li> <li>➤ A trait in the list does not exist in the system metadata for the Entity Type</li> <li>➤ There are duplicates in the trait list</li> <li>➤ Invalid data type or code value for trait</li> </ul>	<p>Error Conditions will be reported in an ERR segment in the ACK, using definitions from the PIM above. Status condition codes returned will only support Error, Warn, and Info</p> <p>All error and status reporting will conform to the description in Chapter 6.</p>



	<b>PIM</b>	<b>PSM</b>
<b>Additional Details</b>		

### 12.3.1.2.2 CreateIdentityFromEntity

The CreateIdentityFromEntity operation uses parameters structured as an HL7 ADTA28 message with ACK response.

It has the same message structure as RegisterEntityWithIdentity (see above), but requires that the EIS generate and return an Entity ID for the record.

See The IHE discussions in IHE IT Infrastructure Technical Framework, vol. 2 (ITI TF-2): Transactions, pp. 41-50 about Transaction “ITI-30” which can be found at [www.ihe.net](http://www.ihe.net) for full details on the PIX Query.

Also See HL7 discussion of the ADTA28 message on page 3-36.

ADTA28 Segments Needed:

- MSH
- SFT] – used to report software vendor ID & version
- EVN
- PID
- [PD1] – for Publicity Code, Protection Indicator and Effective Date, Immunization Registry Status and Date
- [{NK1}] – repeats for mother / father & related info, as required
- PV1 – required for Patient Class, Point of Care, Primary Provider
- PV2 – for Patient Status

	<b>PIM</b>	<b>PSM</b>
<b>Inputs</b>	<ul style="list-style-type: none"> <li>➤ <b>Source ID (Entity ID not submitted):</b></li> <li>➤ <b>Unordered set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</b></li> <li>➤</li> </ul>	<ul style="list-style-type: none"> <li>➤ <b>This is mapped to MSH.4</b></li> <li>➤ <b>use a standard HL7 v2.5 ADT message to hold the traits in segments shown above</b></li> <li>➤</li> </ul>
<b>Outputs</b>	<ul style="list-style-type: none"> <li>➤ <b>Entity ID</b></li> </ul>	<ul style="list-style-type: none"> <li>➤ <b>Entity ID and ACK mapped into EISStatus</b></li> </ul>
<b>Error Conditions</b>	<ul style="list-style-type: none"> <li>➤ <b>Entity reference may already exist (warning) – returns ID(s) and set of traits for matched identity</b></li> <li>➤ <b>Mandatory traits not supplied</b></li> <li>➤ <b>A trait in the list does not exist in the system metadata for the Entity Type</b></li> <li>➤ <b>There are duplicates in the trait list</b></li> </ul>	<p><b>Error Conditions will be reported in an ERR segment in the ACK, using definitions from the PIM above. Status condition codes returned will only support Error, Warn, and Info.</b></p> <p><b>All error and status reporting will conform to the description in</b></p>

	PIM	PSM
	➤ Invalid data type or code value for trait	Chapter 6.
<b>Additional Details</b>		

### 12.3.1.2.3 UpdateEntityTraitValues

The UpdateEntityTraitValues operation uses parameters structured as an HL7 ADTA31 message with ACK response.

Also See HL7 discussion of the ADTA31 message on page 3-40.

ADTA31 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- EVN
- PID
- [PD1] – for Publicity Code, Protection Indicator and Effective Date, Immunization Registry Status and Date
- [{NK1}] – repeats for mother / father & related info, as required
- PV1 – for Patient Class, Primary Point of Care, Primary Provider

	PIM	PSM
<b>Inputs</b>	<ul style="list-style-type: none"> <li>➤ Source ID</li> <li>➤ Entity ID</li> <li>➤ Unordered set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</li> <li>➤</li> </ul>	<ul style="list-style-type: none"> <li>➤ This is mapped to MSH.4 in HL7 v2.5.</li> <li>➤ This is mapped to PID.3, first element in the list</li> <li>➤ Use a standard HL7 v2.5 ADT message to hold the traits in segments shown above.</li> <li>➤</li> </ul>
<b>Outputs</b>	➤ Acknowledgement	➤ HL7 ACK.
<b>Error Conditions</b>	<ul style="list-style-type: none"> <li>➤ Entity reference does not exist (error based on ID match)</li> <li>➤ Mandatory traits deleted</li> <li>➤ A trait in the list does not exist in the system metadata for the Entity Type</li> <li>➤ There are duplicates in the trait list</li> <li>➤ Invalid data type or code value for trait</li> </ul>	<p>Error Conditions will be reported in an ERR segment in the RSP, using definitions from Section 6 above. Status condition codes returned will only support Error, Warn, and Info.</p> <p>All error and status reporting will conform to the description in Chapter 6.</p>
<b>Additional Details</b>		

### 12.3.1.2.4 RemoveIdentity

The RemoveIdentity operation uses parameters structured as an HL7 v2.5 ADTA29 message with ACK response. This operation will set a flag to remove it from all future user queries.

This operation is permanent.

This operation is intended for administrator use only. The identifying information in the PID segment must resolve to a single person Identity record which is not already flagged as “Removed”, or this operation must fail.

See HL7 discussion on page 3-38.

ADTA29 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- EVN
- PID
- PV1 – required by standard, but ignored here
- 

	PIM	PSM
Inputs	➤ Source ID Entity ID	➤ mapped to MSH.4 in HL7 v2.5. ➤ It is mapped to PID.3, first element in the list.
Outputs	➤ An acknowledgement that the Entity has been removed:	➤ HL7 ACK message.
Error Conditions	➤ Entity reference does not exist ➤ Wrong existing state for deletion ➤ Identity linked to other identities	➤ Error Conditions will be reported in an ERR segment in the ACK, using definitions from Section 6 above. Status condition codes returned will only support Error, Warn, and Info.
Additional Details		In this PSM, the Entity Identifier cannot be reused. This implementation will remove records by flagging them as removed.

### 12.3.1.2.5 GetEntityTraitValues

The GetEntityTraitValues operation uses parameters structured as an HL7 QBPQ22 message with RSPK21 response. It returns a list of linked identifiers of patient records from all other domains that are linked to the queried information about one particular patient in a specified domain.

The parameters for Source ID and Entity ID are required for this operation. This operation will only return the Traits for the specified record or it will fail.

See IHE ITI TF volume 2 for a discussion of supported PID query parameters and for response parameters in the ITI-21.

Also see HL7 discussion of the QBPQ22 message and the RSPK21 response on page 3-64 with an example on page 3-67.

QBPQ22 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- QPD
- RCP

RSPK21 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- MSA
- [ERR] – repeats, as required
- QAK
- QPD
- [{PID [PD1] [{NK1}] [QRI] }] repeats, as required for each set of demographic data
- [DSC] – continuation indicator, repeats, as required

	PIM	PSM
<b>Inputs</b>	<ul style="list-style-type: none"> <li>➤ Source ID</li> <li>➤ Entity ID</li> </ul>	<ul style="list-style-type: none"> <li>➤ This is specified in the QPD segment.</li> <li>➤ This is also specified in the QPD segment</li> </ul>
<b>Outputs</b>	<ul style="list-style-type: none"> <li>➤ Unordered set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</li> <li>➤ Entity Status</li> </ul>	<ul style="list-style-type: none"> <li>➤ In this implementation, the trait values are returned in an HL7 v2.5 RSP message.</li> <li>➤</li> <li>➤</li> </ul>
<b>Error Conditions</b>	<ul style="list-style-type: none"> <li>➤ Entity Identifier does not exist (error based on ID match)</li> </ul>	<p>Error Conditions will be reported in an ERR segment in the RSP, using definitions from Section 6 above. Status condition codes returned will only support Error, Warn, and Info.</p>
<b>Additional Details</b>		

### 12.3.1.2.6 FindIdentitiesByTraits

The FindIdentitiesByTraits operation uses parameters structured as an HL7 QBPQ22 message with RSPK21 response. It returns a list of patient records which have a “best fit” match to the submitted Search Qualifiers.

It differs from the GetEntityTraitValues operation described above in that the query can include any available Traits as criteria, as well as combinations of the Source ID, Entity ID, and EIS ID. This operation may return no Entities, exactly one Entity, or a list of Entities. This list can be limited in length, and it can be repeatedly queried to get additional rows, as required.

Given a list of Traits (Trait Identifier or Name, Trait Value pairs), this allows for a search for matching Entities. Outputs include a quality of match.

	PIM	PSM
<b>Inputs</b>	<ul style="list-style-type: none"> <li>➤ Unordered set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</li> <li>➤ Trait Identifiers or Names to be returned (can be different than input -default)</li> <li>➤ Matching algorithm (Optional)</li> </ul>	<ul style="list-style-type: none"> <li>➤ This implementation uses a standard HL7 v2.5 QBPQ22 message to hold the traits.</li> <li>➤ Not supported in this implementation.</li> <li>➤</li> <li>➤ Not supported in this implementation.</li> <li>➤</li> </ul>
<b>Outputs</b>	<ul style="list-style-type: none"> <li>➤ Set of: Source ID, Entity ID, Quality of match tuples together with Status and other traits OR “zero matches found”</li> <li>➤ SFM also identified a possible output as a list of domains that were covered by the search ****</li> </ul>	<ul style="list-style-type: none"> <li>➤ In this implementation, the list of identifiers are returned in an HL7 v2.5 RSP message. In this implementation, an HL7 ACK message is returned when there are no matches.</li> <li>➤ The searchQualifier can include a list of domains to restrict the search to</li> </ul>
<b>Error Conditions</b>	<ul style="list-style-type: none"> <li>➤ A Trait Identifier/Name in the input list does not exist in metadata</li> </ul>	<ul style="list-style-type: none"> <li>➤ Error Conditions will be reported in an ERR segment in the RSP, using definitions from Section 6 above. Status condition codes returned will only support Error, Warn, and Info.</li> </ul>
<b>Additional Details</b>		

### 12.3.1.2.7 ListLinkedIdentities

The ListLinkedIdentities operation uses parameters structured as an HL7 QBPQ23 message with RSPK23 response. It returns a list of linked identifiers of patient records from all other domains that are linked to the queried information about one particular patient in a specified domain.

This corresponds to the IHE PIX Query.

See HL7 discussion of the QBPQ23 message and the RSPK23 response on page 3-67 of the HL7 2.5 specification with an example on page 3-69.

QBPQ23 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- QPD

- RCP

RSPK23 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- MSA
- [ERR] – repeats, as required
- QAK
- QPD
- [PID] - repeats, as required
- [DSC] – continuation indicator, repeats, as required

	PIM	PSM
Inputs	<ul style="list-style-type: none"> <li>➤ Source ID</li> <li>➤ Entity ID</li> <li>➤ SearchQualifiers (optional)</li> </ul>	<ul style="list-style-type: none"> <li>➤ This is specified in the QPD segment.</li> <li>➤ This is also specified in the QPD segment</li> <li>➤ Domains constraints are submitted as specified by IHE PIX Query.</li> </ul>
Outputs	<ul style="list-style-type: none"> <li>➤ List of Entities (Source ID, Entity ID) that are linked to the specified Entity</li> </ul>	<ul style="list-style-type: none"> <li>➤ HL7 RSP, with Continuation Pointer as appropriate.</li> </ul>
Error Conditions	<ul style="list-style-type: none"> <li>➤ Entity ID does not exist</li> <li>➤ Source ID not known to the system</li> </ul>	<p>Error Conditions will be reported in an ERR segment in the RSP, using definitions from Section 6 above. Status condition codes returned will only support Error, Warn, and Info.</p>
Additional Details		

### 12.3.1.2.8 ListUnlinkedEntities

The parameters are structured exactly like the GetAllEntityTraits operation described above.

Given an Entity ID, list all other entities that are not linked to the Entity (optionally constrained within one or more Entity Domains) but whose traits match the traits of the Entity

	PIM	PSM
Inputs	<ul style="list-style-type: none"> <li>➤ Source ID</li> <li>➤ Entity ID</li> <li>➤ Search Qualifiers</li> </ul>	<ul style="list-style-type: none"> <li>➤ This is specified in the QPD segment</li> <li>➤ This is also specified in the QPD segment</li> <li>➤ The search can be restricted to domains specified in the QPD segment..</li> </ul>
Outputs	<ul style="list-style-type: none"> <li>➤ List of Entities (IdentityQualifier, Entity Identifier, Status, Trait List) that could be linked to the specified Entity</li> </ul>	<ul style="list-style-type: none"> <li>➤ In this implementation, the list of entities are returned in an HL7 v2.5 RSP message.</li> </ul>
Error Conditions	<ul style="list-style-type: none"> <li>➤ Entity reference does not exist</li> </ul>	<ul style="list-style-type: none"> <li>➤ Error Conditions will be reported in an ERR segment in the RSP, using definitions from Section 6</li> </ul>

	PIM	PSM
	>IdentityQualifier not known to the system	above. Status condition codes returned will only support Error, Warn, and Info.
<b>Additional Details</b>		

### 12.3.1.2.9 DescribeEntityTraits

This operation is not supported in this profile.

### 12.3.1.3 EISAdminEditorInterface

The following operations are defined for this interface.

#### 12.3.1.3.1 LinkEntities

The LinkEntities operation uses parameters structured as an HL7 ADTA24 message. It returns an ACK indicating the success of the action.

Also See HL7 discussion of the ADTA24 message on page 3-33.

ADTA24 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- EVN – required, but not used
- PID – Patient #1
- PID – Patient #2

	PIM	PSM
<b>Inputs</b>	<ul style="list-style-type: none"> <li>➤Source – Source ID – Entity ID</li> <li>➤target – Source ID - Entity ID</li> <li>➤Unordered Set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</li> </ul>	<ul style="list-style-type: none"> <li>➤This is mapped to the PID segment[1]</li> <li>➤This is mapped to the PID segment[2]</li> <li>➤Use a standard HL7 v2.5 ADT message to hold the source and target ID values. No other entity Traits are required.</li> </ul>
<b>Outputs</b>	<ul style="list-style-type: none"> <li>➤Acknowledgement</li> </ul>	<ul style="list-style-type: none"> <li>➤HL7 ACK.</li> </ul>
<b>Error Conditions</b>	<ul style="list-style-type: none"> <li>➤Entity reference does not exist (error based on ID match)</li> <li>➤Mandatory traits deleted</li> <li>➤A trait in the list does not exist in the system metadata for the Entity Type</li> <li>➤There are duplicates in the trait list</li> <li>➤Invalid data type or code value for trait</li> </ul>	<p>Error Conditions will be reported in an ERR segment in the ACK. Status condition codes returned will only support Error, Warn, and Info. All error and status reporting will conform to the description in Chapter 6.</p>
<b>Additional Details</b>		

#### 12.3.1.3.2 UnLinkEntities

The UnLinkEntities operation uses parameters structured as an HL7 ADTA37 message. It returns an ACK indicating the success of the action.

Also See HL7 discussion of the ADTA37 message on page 3-45.

ADTA37 Segments Needed:

- MSH



- [SFT] – used to report software vendor ID & version
- EVN – required, but not used
- PID – Patient #1
- PID – Patient #2

	PIM	PSM
<b>Inputs</b>	<ul style="list-style-type: none"> <li>➤ Source – Source ID – Entity ID</li> <li>➤ Target – Source ID - Entity ID</li> <li>➤ Unordered Set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</li> </ul>	<ul style="list-style-type: none"> <li>➤ This is mapped to the PID segment[1]</li> <li>➤ This is mapped to the PID segment[2]</li> <li>➤ Use a standard HL7 v2.5 ADT message to hold the source and target ID values. No other entity Traits are required</li> </ul>
<b>Outputs</b>	➤ Acknowledgement	➤ HL7 ACK.
<b>Error Conditions</b>	<ul style="list-style-type: none"> <li>➤ Entity reference does not exist (error based on ID match)</li> <li>➤ Mandatory traits deleted</li> <li>➤ A trait in the list does not exist in the system metadata for the Entity Type</li> <li>➤ There are duplicates in the trait list</li> <li>➤ Invalid data type or code value for trait</li> </ul>	Error Conditions will be reported in an ERR segment in the ACK. Status condition codes returned will only support Error, Warn, and Info. All error and status reporting will conform to the description in Chapter 6.
<b>Additional Details</b>		

### 12.3.1.3.3 MergeEntities

The MergeEntities operation uses parameters structured as an HL7 ADTA40 message. It returns an ACK indicating the success of the action.

Also See HL7 discussion of the ADTA40 message on page 3-47.

ADTA40 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- EVN – required, but not used
- PID
- MRG – required

	PIM	PSM
<b>Inputs</b>	➤ Source – Source ID – Entity ID	➤ This is mapped to the MRG segment

	PIM	PSM
	<ul style="list-style-type: none"> <li>➤Target – Source ID - Entity ID</li> <li>➤Unordered Set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</li> </ul>	<ul style="list-style-type: none"> <li>➤This is mapped to MSH.4 and PID.3, first element in the list</li> <li>➤Use a standard HL7 v2.5 ADT message to hold the source and target ID values. No other entity Traits are required</li> </ul>
<b>Outputs</b>	➤Acknowledgement	➤HL7 ACK.
<b>Error Conditions</b>	<ul style="list-style-type: none"> <li>➤Entity reference does not exist (error based on ID match)</li> <li>➤Mandatory traits deleted</li> <li>➤A trait in the list does not exist in the system metadata for the Entity Type</li> <li>➤There are duplicates in the trait list</li> <li>➤Invalid data type or code value for trait</li> </ul>	Error Conditions will be reported in an ERR segment in the ACK. Status condition codes returned will only support Error, Warn, and Info. All error and status reporting will conform to the description in Chapter 6.
<b>Additional Details</b>		

### 12.3.1.3.4 UnMergeEntities

This operation is not supported in this profile.

### 12.3.1.3.5 ActivateEntity

The ActivateEntity operation uses parameters structured as an HL7 ADTA08 message to set a status flag in the EIS to make the Entity unavailable to general queries. The record will remain available to specific queries by Source ID and Entity ID, but will not be available in record linking and queries by traits other than Source ID and Entity ID The operation returns an ACK indicating the success of the action.

Also See HL7 discussion of the ADTA31 message on page 3-40.

ADTA31 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- EVN – required, but not used
- PID
- PV1 – required, but not used
- PV2 – required to set PV2-24 Patient Status

	PIM	PSM
<b>Inputs</b>	<ul style="list-style-type: none"> <li>➤Source ID</li> <li>➤Entity ID</li> </ul>	<ul style="list-style-type: none"> <li>➤This is mapped to MSH.4 in HL7 v2.5.</li> <li>➤This is mapped to PID.3, first</li> </ul>

	PIM	PSM
	➤ <b>Unordered Set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</b>	element in the list ➤ Use a standard HL7 v2.5 ADT message to hold the status flag. Set PV2-24 Patient Status Code to "A" to indicate Activated Patient. Set PV2-46 to Patient Status Effective Date.
<b>Outputs</b>	➤ <b>Acknowledgement</b>	➤ <b>HL7 ACK.</b>
<b>Error Conditions</b>	➤ <b>Entity reference does not exist (error based on ID match)</b> ➤ <b>Mandatory traits deleted</b> ➤ <b>A trait in the list does not exist in the system metadata for the Entity Type</b> ➤ <b>There are duplicates in the trait list</b> ➤ <b>Invalid data type or code value for trait</b>	<b>Error Conditions will be reported in an ERR segment in the ACK. Status condition codes returned will only support Error, Warn, and Info. All error and status reporting will conform to the description in Chapter 6.</b>
<b>Additional Details</b>		

### 12.3.1.3.6 DeactivateEntity

The DeactivateEntity operation uses parameters structured as an HL7 ADTA08 message to set a status flag in the EIS to make the Entity unavailable to general queries. The record will remain available to specific queries by Source ID and Entity ID, but will not be available in record linking and queries by traits other than Source ID and Entity ID. The operation returns an ACK indicating the success of the action.

Also See HL7 discussion of the ADTA08 message on page 3-15.

ADTA31 Segments Needed:

- MSH
- [SFT] – used to report software vendor ID & version
- EVN – required, but not used
- PID
- PV1 – required, but not used
- PV2 – required to set PV2-24 Patient Status

	PIM	PSM
<b>Inputs</b>	➤ <b>Source ID</b> ➤ <b>Entity ID</b> ➤ <b>Unordered Set of Traits (Data Structures consisting of a single or hierarchic structure of name-value pairs)</b>	➤ <b>This is mapped to MSH.4 in HL7 v2.5.</b> ➤ <b>This is mapped to PID.3, first element in the list</b> ➤ <b>Uses a standard HL7 v2.5 ADT message to hold the status flag. Set PV2-24 Patient Status Code to "D" to indicate Deactivated Patient. Set PV2-46 to Patient</b>

	PIM	PSM
		Status Effective Date.
Outputs	➤ Acknowledgement	➤ HL7 ACK.
Error Conditions	<ul style="list-style-type: none"> <li>➤ Entity reference does not exist (error based on ID match)</li> <li>➤ Mandatory traits deleted</li> <li>➤ A trait in the list does not exist in the system metadata for the Entity Type</li> <li>➤ There are duplicates in the trait list</li> <li>➤ Invalid data type or code value for trait</li> </ul>	Error Conditions will be reported in an ERR segment in the ACK. Status condition codes returned will only support Error, Warn, and Info. All error and status reporting will conform to the description in Chapter 6
Additional Details		

#### 12.3.1.4 EISMetaDatumInterface

No interfaces for the Meta Data are supported in this Semantic Profile.

# 13 Compatibility to Other Standards

## 13.1 Compatibility to HL7

EIS provides the means to exchange HL7 Version 2.x and Version 3 messages and documents as semantic signifier and logical record payloads.

## 13.2 Compatibility to IHE Standards

**Issue 13934 – here: remove spurious hyperlink “[ds6]”**

Correspondence between EISMgmtAndQueryInterface methods and IHE transactions is given here.

<b>EISMgmtAndQueryInterface</b>	<b>IHE PIX/PDQ</b>
<b>CreateIdentityFromEntity</b>	<b>Patient Identity Feed</b>
<b>RegisterEntityWithIdentity</b>	<b>Patient Identity Feed</b>
<b>UpdateEntityTraitValues</b>	<b>Patient Identity Feed</b>
<b>ListLinkedIdentities</b>	<b>PIX Query</b>
<b>None</b>	<b>PIX Notify</b>
<b>GetEntityTraitValues</b>	<b>Patient Demographics Query</b>
<b>FindIdentitiesByTraits</b>	<b>Patient Demographics Query</b>
<b>ListUnlinkedIdentities</b>	<b>Patient Demographics Query</b>
<b>None</b>	<b>Patient Demographics and Visit Query</b>

Table 4: EIS Correspondence to IHE PIX and PDQ