

RosettaNet TPIR-PIP for Engineering Information Specification

Specification Information	
Name	RosettaNet TPIR-PIP for Engineering Information Specification
Version	V11.00.00
Date	19 March 2008

Table of Contents

1	DOCUMENT MANAGEMENT	5
1.1	Legal Disclaimer.....	5
1.2	Copyright	5
1.3	Trademarks	5
1.4	Document Version History	5
1.5	Related Documents.....	5
1.6	Audience.....	5
1.7	Document Conventions	6
1.8	Document Structure	6
1.9	Acknowledgement	6
2	OVERVIEW	7
2.1	TPIR-PIP for Engineering Information Development Requirements	8
2.1.1	The Necessity for Adding Private Properties (Private-EIPS)	8
2.2	Process and Deliverables Matrix.....	9
3.	REQUIREMENTS AND RULES	10
3.1.	TPIR-EIPS	10
3.1.1.	Basic Rules and Requirements.....	10
3.1.2.	TPIR-EIPS File, Namespace, Versioning and Documentation Guidelines.....	11
3.2.	Private-EIPS	13
3.2.1.	Basic Rules and Requirements.....	13
3.2.2.	Example of Private-EIPS.....	14
3.2.3.	Private-EIPS File, Namespace, Versioning and Documentation Guidelines.....	15
3.3.	Creating a TPIR-PIP from a PIP with entry points	16
3.3.1.	Basic Rules and Requirements.....	16
3.3.2.	EIPS Insertion Process	21

3.3.3. TPIR-PIP File, Namespace, Versioning and Documentation Guidelines..... 24

4. REFERENCES 25

5. GLOSSARY 26

6. RULES AND RATIONALE..... 27

List of Tables

Table 2-1: Process and Deliverables Matrix 9

List of Figures

Fig. 2.1: TPIR-PIP creation flow using TPIR-PIP FOR ENGINEERING INFORMATION method..... 7
Fig. 2.2: Private-EIPS 8

1 Document Management

1.1 Legal Disclaimer

RosettaNet™, its members, officers, directors, employees, or agents shall not be liable for any injury, loss, damages, financial or otherwise, arising from, related to, or caused by the use of this document or the specifications herein, as well as associated guidelines and schemas. The use of said specifications shall constitute your express consent to the foregoing exculpation.

1.2 Copyright

©2008 RosettaNet. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

1.3 Trademarks

RosettaNet, Partner Interface Process, PIP and the RosettaNet logo are trademarks or registered trademarks of "RosettaNet," a non-profit organization. All other product names and company logos mentioned herein are the trademarks of their respective owners. In the best effort, all terms mentioned in this document that are known to be trademarks or registered trademarks have been appropriately recognized in the first occurrence of the term.

1.4 Document Version History

Version	Date	Description
V11.00.00	March 19, 2008	Published as Validated specification

1.5 Related Documents

- RosettaNet EIPS/XSD Specification
- RosettaNet TPIR-PIP Design Specification

1.6 Audience

This document's primary audience is Solution Providers and PIP Implementers that need to create TPIR-PIPs as specialized versions of existing RosettaNet released PIPs in order to better support their business processes. The secondary audience is RosettaNet Engineering who creates PIP with entry points.

1.7 Document Conventions

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119] as quoted here:

<i>MUST</i>	This word, or the terms "REQUIRED" or "SHALL", means that the definition is an absolute requirement of the specification.
<i>MUST NOT</i>	<i>This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the specification.</i>
<i>SHOULD</i>	<i>This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.</i>
<i>SHOULD NOT</i>	<i>This phrase, or the phrase "NOT RECOMMENDED", means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.</i>
<i>MAY</i>	<i>This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides).</i>

1.8 Document Structure

This document describes details of the TPIR-PIP for Engineering Information and the development methodology referred to as TPIR-PIP for Engineering Information process, and consists of:

- Overview of the TPIR-PIP for Engineering Information methodology
- Process and deliverables description
- Process and deliverables specification
- PIP design requirement specification (for RosettaNet Engineering)
- Other information (Glossary, Reference documents)

1.9 Acknowledgement

This document has been prepared by EIM Foundational Program TPIR-PIP for Engineering Information prototyping team consisting of voluntary RosettaNet Japan members with significant contributions from the EIM Milestone Program Participants, the RosettaNet Global Architecture Team and RosettaNet members.

2 Overview

TPIR-PIP for Engineering Information is the methodology to extend a TPIR-PIP with a dictionary extension point(s) referred to as a TPIR-EIPS and/or Private-EIPS. The ultimate deliverable from this process is referred to as a TPIR-PIP for Engineering Information. To utilize this extension feature, the PIP MUST support specific requirements as described in this document.

The illustration below identifies the TPIR-PIP for Engineering Information creation process using the TPIR-PIP for Engineering Information method. An additional step to the initial method defined by the TPIR-PIP creation process is to add a TPIR-EIPS and Private-EIPS as inputs to the TPIR-PIP. To realize this process, TPIR-PIP for Engineering Information introduces the concept of a “PIP with an Entry Point (PIP-w/EP)”. Those PIPs designed for exchanging engineering information would have an “entry point” to which trading partners are allowed to embed customized engineering information structures.

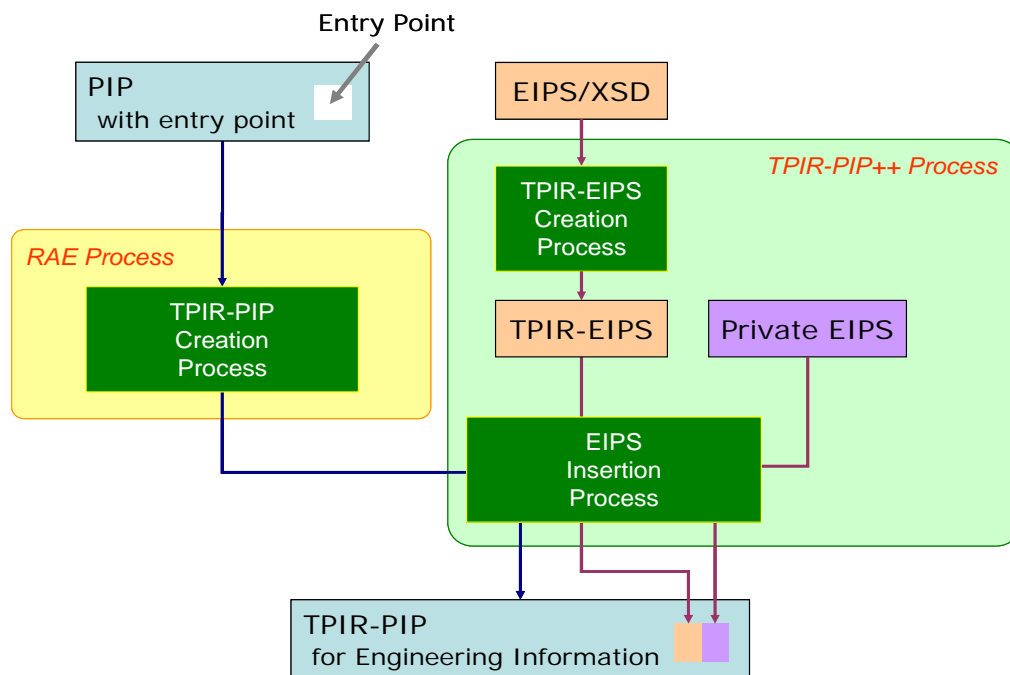


Fig. 2.1: TPIR-PIP for Engineering Information Creation Flow Using the TPIR-PIP for Engineering Information Method

Artifacts created or used in the process identified above:

- PIP with entry point (specifically designed for additional engineering information) is available on the RosettaNet web site
- EIPS/XSD is available on the RosettaNet web site
- TPIR-EIPS, developed by partner, is a restriction of the content of the EIPS/XSD
- Private-EIPS, developed by partner, addresses engineering needs not contained in the EIPS/XSD
- TPIR-PIP for Engineering Information, developed by partner, is the result of the RAE process based upon the TPIR-PIP Design Specification and the TPIR-PIP for Engineering Information process.

Note: The immediate output of the RAE process may result in a temporary file (when the entry point replacement process is done some time after the RAE process). This is not a permanent artifact and thus the naming is up to the partner.

2.1 TPIR-PIP for Engineering Information Development Requirements

The TPIR-PIP for Engineering Information methodology stems from the requirements of the EIM Milestone Program:

- Ability to add private properties
- Flexibility for a single PIP to exchange a variety of engineering information, such as by kind of electronic component
- Use of community agreed property definition subset(s) rather than the entire RNTD due to its size
- Automated, richer validation by XML Schema

2.1.1 The Necessity for Adding Private Properties (Private-EIPS)

An EIPS is a group of properties that are commonly agreed by a number of buyers and suppliers to be the minimum number of properties necessary to define a certain set of engineering information. It may not include all required properties for a given partner. The technology used in technical products changes frequently, which in turn requires frequent changes to how those products are specified technically. New products often have special characteristics. Some technical properties are not so new, but are not so common. These properties are not contained in a Community Agreed Public EIPS. If a Community Agreed Public EIPS lacks certain properties that an organization must exchange with its partner, then the organization will create these properties in a “partner agreed” Private-EIPS utilizing the same fundamental rules used to create a Community Agreed Public EIPS (Refer to EIPS Maintenance document).

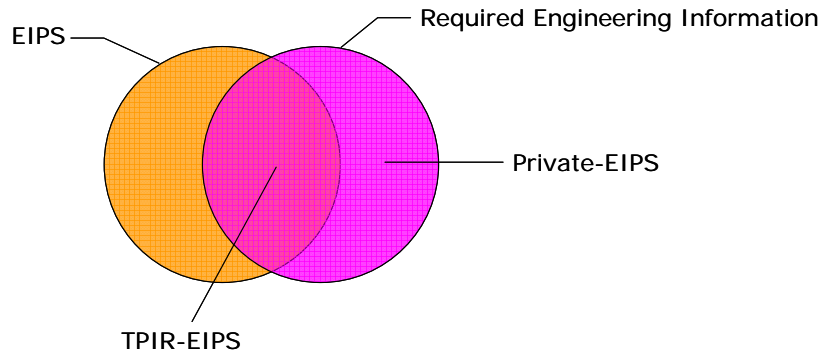


Fig. 2.2: Private-EIPS

2.2 Process and Deliverables Matrix

TPIR-PIP for Engineering Information Creation Process using TPIR-PIP for Engineering Information Method

Input	Process	Output	Section in this Document
<ul style="list-style-type: none">• EIPS/XSD	Create TPIR-EIPS	TPIR-EIPS	3.1
---	Create Private-EIPS	Private-EIPS	3.2
<ul style="list-style-type: none">• TPIR-EIPS• Private-EIPS• PIP-w/EP	Create TPIR-PIP EIPS Insertion	TPIR-PIP for Engineering Information	3.3

Table 2-1: Process and Deliverables Matrix

3. Requirements and Rules

3.1. TPIR-EIPS

3.1.1. Basic Rules and Requirements

Process description: TPIR-EIPS creation process

TPIR-EIPS is created from EIPS/XSD by restriction. This process is optional. If partners agree to use the EIPS/XSD without restriction, then this process can be bypassed.

Input : EIPS/XSD (Community agreed, schema format)

Output : TPIR-EIPS (Constrained, schema format)

An organization creates a TPIR-EIPS Schema by constraining the community agreed EIPS/XSD to limit EIPS/XSD content. Constraining may be done utilizing an ordinary text editor; an XML editor that directly manipulates the XSD text and validates it; or a more sophisticated XML editor that manipulates the XSD in a visual way and validates the text.

Rule 3-1-1: TPIR-EIPS MUST be created from an EIPS/XSD.

Rationale: TPIR-EIPS is by definition a constrained version of an EIPS/XSD.

Rule 3-1-2: TPIR-EIPS MUST be XML Schema-based.

Rationale: TPIR-EIPS is a constrained version of an EIPS/XSD, which is XML Schema-based.

Rule 3-1-3: All EIPS/XSD restrictions MUST be compliant with the TPIR-PIP Design Specification.

Rationale: To ensure harmonization with the latest RosettaNet practices and improve enablement across the supply chain.

Examples: The restriction rules for a TPIR-EIPS are fully aligned with the restriction rules for a TPIR-PIP, a typical example of restriction of cardinality:

EIPS/XSD (before restriction)

```
<xs:complexType name="AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
  <xs:annotation>
    :
  </xs:annotation>
  <xs:sequence>
    :
    <xs:element name="RatedCapacitance" type="edt:RatedCapacitanceType"
      minOccurs="0"/>
    :
  </xs:sequence>
</xs:complexType>
```

TPIR-EIPS (after restriction)

```
<xs:complexType
name="MNC_AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
  <xs:annotation>
    :
  </xs:annotation>
  <xs:sequence>
    :
    <xs:element name="RatedCapacitance" type="edt:RatedCapacitanceType"/>
    :
  </xs:sequence>
</xs:complexType>
```

This is an example where the partner restricts the cardinality of the RatedCapacitance element from optional to mandatory.

3.1.2. TPIR-EIPS File, Namespace, Versioning and Documentation Guidelines

Rule 3-1-4: The TPIR-EIPS file name MUST comply with the following requirements:

Requirement 3-1-4-1: The original EIPS/XSD MUST be easily identified.

Requirement 3-1-4-2: The TPIR-EIPS MUST identify the organization owner.

Requirement 3-1-4-3: The TPIR-EIPS MUST identify that it is an organization's representation of its requirements to easily identify that the file is not the original EIPS/XSD.

Rationale: Compliance with TPIR-PIP practices.

Rule 3-1-5: The TPIR-EIPS file name MUST carry both the version of the input EIPS/XSD, as well as, the version of the TPIR-EIPS itself.

Rationale: This complies with the TPIR-PIP design guideline rules.

Example: The format of the EIPS file name is

"[EIPS_Category_Name]_[EIPSMajorVersion_EIPSMajorVersion].xsd"

While the format of the TPIR-EIPS file name is

"[MNC]_[EIPS_Category_Name]_[EIPSMajorVersion_EIPSMajorVersion]
[TPIR-EIPSMajorVersion_TPIR-EIPSMajorVersion].xsd"

Rule 3-1-6: An organization MUST assign the version number of TPIR-EIPS (TPIR-EIPS major version and TPIR-EIPS minor version) by following the versioning rules of EIPS/XSD (Refer to the EIPS/XSD Specification, Section 5.1.7).

Rationale: Compliance with EIPS/XSD practice.

Example: If the EIPS/XSD file name is:

Capacitor_01_01.xsd

Then the TPIR-EIPS Schema file name should look like:

MNC_Capacitor_PartInfo_01_01_01_00.xsd

where "MNC" denotes the organization that created the TPIR-EIPS Schema, "PartInfo" the business process identifier, "01_01" the version of the EIPS/XSD, and "01_00" the version assigned to the TPIR-EIPS Schema by the organization.

Rule 3-1-7: The TPIR-EIPS MUST comply with TPIR-PIP namespace and namespace prefix rules.

Rationale: Since the TPIR-EIPS is created based upon a TPIR-PIP, the TPIR-EIPS namespace and namespace prefix must comply with TPIR-PIP rules.

Example: When EIPSCapacitor_01_01.xsd is used, the namespace appears as follows:
xmlns:eca="urn:rosettanet:specification:domain:Design:Capacitor:xsd:eips:01.01" where "eca" identifies the namespace prefix.

When MNC_Capacitor_PartInfo_01_01_01_00.xsd is used, it uses the original namespace prefix for the new namespace:

xmlns:eca="urn:rosettanet:partner:external:123456789:PartInfo:domain:Design:Capacitor:xsd:eips:01.01:01.00"

Rule 3-1-8: The TPIR-EIPS namespace prefix MUST start with an "e" for EIPS then followed by two characters representing the EIPS category (i.e. capacitor, connector, etc.). If the initial two characters of an EIPS category overlap, then the overlap is removed by selecting the next unused letter in the name of the newly overlapping EIPS. If all letters have already been used in the name, choose the first unused letter in the alphabet.

Rationale: The namespace prefix should be generated using simple rules and should be persistent for a specific EIPS.

Example: When a TPIR-EIPS file name is MNC_EIPSCapacitor_PartInfo_01_01_01_00.xsd, the namespace would be "eca". When other EIPS categories begin with "ca", the namespace of one would be "eca" and other is "ecb".

Rule 3-1-9: The TPIR-EIPS file MUST be stored under the Design domain.

Rationale: These definitions are only used in the domain of design engineering and are not used in any other domain.

3.2. Private-EIPS

3.2.1. Basic Rules and Requirements

Process description: Private-EIPS creation process

A Private-EIPS is created from scratch manually by using an XML Editor. This process is optional. If partners agreed not to use a Private-EIPS, then this process MAY be bypassed. The output from this process is expected to follow RosettaNet rules and practices, but is a private artifact under the governance of the company that developed it and is not part of the RosettaNet standards.

Input : none

Output : Private-EIPS (Private, schema format)

Rule 3-2-1: A Private-EIPS MUST comply with the technical requirements of the EIPS/XSD Specification.

Rationale: Although the content is defined by the MNC, the format must comply with RosettaNet general practices and rules to prevent the system from having to process the public and private extensions differently.

3.2.2. Example of Private-EIPS

The example is the case where an organization defines a "PartSpecificationCode" and a "PolarityType" as private properties within its Private-EIPS named "AluminumElectrolyticCapacitorsWithSolidElectrolyteType" (based on the rules of EIPS/XSD), it would look as follows.

```
<xs:element name="AlmininumElectrolyticCapacitorsWithSolidElectorolyte"
type="tns:AlmininumElectrolyticCapacitorsWithSolidElectorolyteType"/>
<xs:complexType
name="AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
  <xs:annotation>
    <xs:appinfo>
      <urss:Code>MNC003</urss:Code>
      <urss:CreationDate>2006-03-17</urss:CreationDate>
      <urss:Definition>Aluminum Electrolytic Capacitors with Solid Electrolyte.
</urss:Definition>
      <urss:LastUpdatedDate>2006-03-17</urss:LastUpdatedDate>
      <urss:ReferenceSource>MNC003-001</urss:ReferenceSource>
      <urss:TypeVersion>01.00</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="PartSpecificationCode" type="zebf:EIPSString32BasicType"/>
    <xs:annotation>
      <xs:appinfo>
        <urss:Definition>The code of the part specification number or drawing
number.</urss:Definition>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
    <xs:element name="PolarityType" type="zebf:EIPSString32BasicType">
      <xs:annotation>
        :
      </xs:annotation>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="schemaVersion" type="xs:token">
  </xs:attribute>
</xs:complexType>
```

3.2.3. Private-EIPS File, Namespace, Versioning and Documentation Guidelines

Rule 3-2-2: A Private-EIPS file name MUST comply with following requirements:

Requirement 3-2-2-1: Easily discriminate the group or category of the Private-EIPS.

Requirement 3-2-2-2: Easily identify the organization that creates the Private-EIPS.

Rationale: Compliance with the naming practices within the TPIR-PIP Design Specification.

Rule 3-2-3: Private-EIPS version SHOULD comply with the versioning rules of EIPS/XSD.

Rationale: Using the same versioning rules for a TPIR-EIPS and applying it to a Private-EIPS reduces the versioning management by an organization for EIPS'.

Example: The Private-EIPS file name looks like:

MNC_Capacitor_01_00.xsd

where "MNC" is the organization that creates Private-EIPS, "Capacitor" a descriptor of the contents of the Private-EIPS, and "01_00" the Private-EIPS Schema version.

Note: The Private-EIPS file name may or may not be based on the related TPIR-EIPS.

Rule 3-2-4: Private-EIPS namespace prefix SHOULD begin with "pe" (Private-EIPS) then followed by two characters identifying the category (capacitor, connector, etc.). If the initial two characters of an EIPS category overlap, then the overlap is removed by selecting the next unused letter in the name of the newly overlapping EIPS. If all letters have already be used in the name, choose the first unused letter in the alphabet.

Rationale: The namespace prefix should be generated using simple rules and should be persistent for a specific EIPS (Note: Same as **Rule 3-1-7**).

Example: When a Private-EIPS file name is MNC_Capacitor_01_00.xsd, the namespace for it is "peca". If the organization uses other Private-EIPS with the category beginning with "ca", then the namespace of the original one continues as "peca" and other would be something like "pecb".

Rule 3-2-5: A Private-EIPS MUST comply with the URN format defined in the RosettaNet Namespace Specification and Management document:

*urn:rosettanet:partner:external:{partner-id}:{specification-class}:{specification-subclass}
{:specification-id}?:{type}{:subtype}?:{version-id}:{instance-id }*

partner-id: = {DUNS | DUNS+4}:{business-process-id}

Rationale: Complies with the RosettaNet Namespace Specification.

Example:

**xmlns:peca="urn:rosettanet:partner:external:123456789:PartInfo:domain:Design:MNC
Capacitor:xsd:eips:01.00"**

Rule 3-2-6: A Private-EIPS file MUST be stored under Design domain.

Rationale: Private properties are only used in the domain of design engineering and are not used in any other domain, like a TPIR-EIPS.

3.3. Creating a TPIR-PIP from a PIP with entry points

3.3.1. Basic Rules and Requirements

Process description: TPIR-PIP creation process
 EIPS insertion process

A TPIR-PIP for Engineering Information is created from a PIP-w/EP (for details on this process see Figure 2-1) by inserting a TPIR-EIPS or EIPS/XSD and/or Private-EIPS. EIPS insertion process is optional. If partners agreed to utilize the original (community) PIP without any engineering property sets, then the artifact of the RAE method is a TPIR-PIP. In TPIR-PIP creation process and EIPS insertion process, the partner could use an XML Editor or any other appropriate tool.

If a RosettaNet Milestone Program or any other standards development activity has requirements to exchange engineering information by a singular PIP and EIPS(s), the PIP with an entry point has been developed by clarifying the position of entry point(s).

The PIP tags are defined by the requirements of the business/engineering process. The EIPS(s) merely define replaceable chunks of engineering information (not process information).

Input : PIP-w/EP (Community agreed, schema format)
 TPIR-EIPS (Private, schema format)
 (or EIPS/XSD (Community agreed, schema format))
 Private-EIPS (Private, schema format)

Output : TPIR-PIP for Engineering Information (Private – Partner agreed PIP, schema format)
 (or TPIR-PIP (PIP without any EIPSS, Private, Partner agreed, schema format))

Rule 3-3-1: Restriction of the community-agreed PIP MUST comply with the rules defined in the TPIR-PIP Design Specification.

Rationale: The restriction process of a RosettaNet community PIP is defined in the TPIR-PIP Design Specification.

Rule 3-3-2: The PIP applied to the TPIR-PIP for Engineering Information method MUST contain entry points for additional engineering information.

Rationale: TPIR-PIP prohibits the extension of a community agreed PIP XML Schema as described in "TPIR-PIP Design Specification". Entry points serve as a denotation of the placement for engineering information.

Rule 3-3-3: An engineering information entry point type is defined as a complexType that MUST contain a special xs:annotation/urss:Definition element with the value of "Entry point type for additional engineering information".

Rationale: This is a method to allow an application processor to distinguish an engineering information entry point type from other types.

Example:

```
<xs:complexType name="EngineeringInformationType">
  <xs:annotation>
    <xs:appinfo>
      <urss:CreationDate>2007-05-23</urss:CreationDate>
      <urss:Definition>Entry point type for additional engineering information
      </urss:Definition>
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>
      <urss:TypeVersion>01.00</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType>
```

Rule 3-3-4: The placement and the cardinality of an engineering information entry point MUST be specified by a declaration of an element that corresponds to the definition of a complexType as defined under [Rule 3-3-3]. This element MUST contain a special xs:annotation/urss:Definition element with the value of "Entry point for additional engineering information".

Rationale: One or more entry points could be placed in various places of the enclosing engineering information structures. At the same time the declaration of the element for an entry point allows for the replacement of an entry point with various TPIR-EIPS/Private-EIPS pairs in a final engineering information TPIR-PIP.

Note: It is possible (but not likely) that several EIPS' of either type could be inserted at the same place in an engineering structure.

Example:

Here is an example of more than one engineering information entry point sequentially placed in one place of the enclosing structure:

```
<xs:element name="EngineeringInformation"
  type="edt:EngineeringInformationType" minOccurs="0"
  maxOccurs="unbounded">
  <xs:annotation>
    <xs:appinfo>
      <urss:CreationDate>2007-05-23</urss:CreationDate>
      <urss:Definition>Entry point for additional engineering information
      </urss:Definition>
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>
      <urss:TypeVersion>01.00</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
</xs:element/>
```

Rule 3-3-5: The concrete element that is inserted under an entry point MUST be declared as a combination of 0 or 1 TPIR-EIPS and 0 or 1 Private-EIPS. TPIR-EIPS and Private-EIPS MUST be placed in alphabetical order.

Rationale: Combining a TPIR-EIPS with its corresponding Private-EIPS follows a good abstraction pattern. Regarding sorting, RosettaNet XML Design Guideline requires that components of an XML Schema SHOULD be sorted in alphabetical order.

Possible combinations of entry points are as follows:

- 1) None of the entry points used
- 2) A single entry point with a single community EIPS
- 3) A single entry point with a single private EIPS
- 4) A single entry point with a single private EIPS and a single community EIPS
- 5) Two entry points with a single community EIPS each

And so on.

Example:

```
<xs:complexType name="EngineeringInformationForCapacitorType">
  <xs:annotation>
    <xs:appinfo>
      <urss:CreationDate>2007-05-23</urss:CreationDate>
      <urss:Definition>Entry point type for additional engineering information
      </urss:Definition>
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>
      <urss:TypeVersion>01.00</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="AluminumElectrolyticCapacitorsWithSolidElectrolyte"
      type="eca:AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
      <xs:annotation>
        :
      </xs:annotation>
    </xs:element>
    <xs:element name="MNC_AluminumElectrolyticCapacitorsWithSolidElectrolyte"
      type="peca:MNC_AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
      <xs:annotation>
        :
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Rule 3-3-6: During the EIPS Insertion Process [Fig. 2.1], an organization MUST rename the element name and type name, which has the entry point, according to the inserted TPIR-EIPS and/or Private-EIPS.

Rationale: Using the name of the EIPS itself facilitates human understanding. Also, by assigning a different name it is possible to place different EIPS' in the same enclosing engineering structure.

Example:

```
<xs:complexType name="EngineeringInformationTypeForCapacitor">
  <xs:annotation>
    <xs:appinfo>
      <urss:CreationDate>2007-05-23</urss:CreationDate>
      <urss:Definition>Entry point type for additional engineering information
      </urss:Definition>
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>
      <urss:TypeVersion>01.00</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="AluminumElectrolyticCapacitorsWithSolidElectrolyte"
      type="eca:AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
      <xs:annotation>
        :
      </xs:annotation>
    </xs:element>
    <xs:element name="MNC_AluminumElectrolyticCapacitorsWithSolidElectrolyte"
      type="peca:MNC_AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
      <xs:annotation>
        :
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Rule 3-3-7: During the EIPS Insertion Process [Fig. 2.1], an organization MUST add namespace declaration and import for reference of inserted TPIR-EIPS and Private-EIPS.

Rationale: In order to use other schemas efficiently by the XML Schema, namespace declaration and import are used.

Rule 3-3-8: When an organization needs to use two or more TPIR-EIPS' and/or two or more Private-EIPS', an organization MUST NOT arbitrarily mix TPIR-EIPS' and/or Private-EIPS'. Each corresponding pair of TPIR-EIPS and Private-EIPS MUST be placed in a separate element as defined under [Rule 3-3-5].

Rationale: Mixing two or more EIPS' under the same entry point causes ambiguity of the content of the entry point.

Example:

In the case of an organization that needs to insert EIPS' for both a capacitor and a resistor:

```
<xs:element name="EngineeringInformationForCapacitor"
  type="tns:EngineeringInformationForCapacitorType" minOccurs="0"/>
<xs:element name="EngineeringInformationForResistor"
  type="tns:EngineeringInformationForResistorType" minOccurs="0"/>
:
<xs:complexType name="EngineeringInformationForCapacitorType">
  <xs:annotation>
    <xs:appinfo>
      <urss:CreationDate>2007-05-23</urss:CreationDate>
      <urss:Definition>Additional engineering information for Capacitor
      </urss:Definition>
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>
      <urss:TypeVersion>01.00</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="AluminumElectrolyticCapacitorsWithSolidElectrolyte"
      type="eca:AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
      <xs:annotation>
        <xs:appinfo>
          :
          </xs:appinfo>
        </xs:annotation>
      </xs:element>
      :
    </xs:sequence>
  </xs:complexType>
  :
<xs:complexType name="EngineeringInformationForResistorType">
  <xs:annotation>
    <xs:appinfo>
      <urss:CreationDate>05/23/2007</urss:CreationDate>
      <urss:Definition>Additional engineering information for Resistor
      </urss:Definition>
      <urss:LastUpdateDate>05/23/2007</urss:LastUpdateDate>
      <urss:TypeVersion>1.0</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="FixedLinearResistor"
      type="ere:FixedLinearResistorType">
      <xs:annotation>
        :
        </xs:annotation>
      </xs:element>
      :
    </xs:sequence>
  </xs:complexType>
```

3.3.2. EIPS Insertion Process

Sample of an entry point: the entry point for TPIR-EIPS and Private-EIPS is indicated according to [Rule 3-3-3, Rule 3-3-4].

```
<xs:element name="EngineeringInformation"
  type="edt:EngineeringInformationType" minOccurs="0"
  maxOccurs="unbounded">
  <xs:annotation>
    <xs:appinfo>
      <urss:CreationDate>2007-05-23</urss:CreationDate>
      <urss:Definition>Entry point for additional engineering
  information</urss:Definition>
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>
      <urss:TypeVersion>01.00</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
:
<xs:complexType name="EngineeringInformationType">
  <xs:annotation>
    <xs:appinfo>
      <urss:CreationDate>2007-05-23</urss:CreationDate>
      <urss:Definition>Entry point type for additional engineering
  information</urss:Definition>
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>
      <urss:TypeVersion>01.00</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
</xs:complexType>
```

Sample of TPIR-EIPS

```
<xs:complexType name="AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
  <xs:annotation>
    :
  </xs:annotation>
  <xs:sequence>
    <xs:element name="CategoryTemperature"
  type="zebf:EIPSSignedFloat6BasicType">
      <xs:annotation>
        :
      </xs:annotation>
    </xs:element>
    <xs:element name="RatedCapacitance" type="zebf:EIPSSignedFloat6BasicType">
      <xs:annotation>
        :
      </xs:annotation>
    </xs:element>
    :
  </xs:sequence>
</xs:complexType>
```

Sample of Private-EIPS

```
<xs:complexType
name="MNC_AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
  <xs:annotation>
    :
  </xs:annotation>
  <xs:sequence>
    :
    <xs:element name="PartSpecificationCode" type="zebf:EIPSString32BasicType">
      <xs:annotation>
        :
      </xs:annotation>
    </xs:element>
    :
  </xs:sequence>
</xs:complexType>
```

The procedure which inserts the above-mentioned TPIR-EIPS and Private-EIPS under an entry point is as follows:

Example of EIPS insertion steps

Step 1: Insert element declarations of TPIR-EIPS and Private-EIPS inside an entry point type [Rule 3-3-5]

```
<xs:complexType name="EngineeringInformationType">
  <xs:annotation>
    <xs:appinfo>
      <urss:CreationDate>2007-05-23</urss:CreationDate>
      <urss:Definition>Entry point for additional engineering
information</urss:Definition>
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>
      <urss:TypeVersion>01.00</urss:TypeVersion>
    </xs:appinfo>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="AluminumElectrolyticCapacitorsWithSolidElectrolyte"
type="eca:AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
      <xs:annotation>
        <xs:appinfo>
          <urss:Definition>EIPS for alminum electrolytic capacitors with solid
electrolyte.</urss:Definition>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
    <xs:element name="MNC_AluminumElectrolyticCapacitorsWithSolidElectrolyte"
type="peca:MNC_AluminumElectrolyticCapacitorsWithSolidElectrolyteType">
      <xs:annotation>
        <xs:appinfo>
          <urss:Definition>MNC's EIPS for alminum electrolytic capacitors with
solid electrolyte.</urss:Definition>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

```
</xs:element>  
</xs:sequence>  
</xs:complexType>
```

An organization replaces the element declaration(s) in the proper place.

Step 2: Rename the element of an entry point according to the inserted TPIR-EIPS and Private-EIPS [Rule 3-3-6]

In community agreed PIP, entry point is specified as an element named "EngineeringInformation".

```
<xs:element name="EngineeringInformation"  
  type="edt:EngineeringInformationType" minOccurs="0"  
  maxOccurs="unbounded">  
  <xs:annotation>  
    <xs:appinfo>  
      <urss:CreationDate>2007-05-23</urss:CreationDate>  
      <urss:Definition>Entry point for additional engineering  
information</urss:Definition>  
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>  
      <urss:TypeVersion>01.00</urss:TypeVersion>  
    </xs:appinfo>  
  </xs:annotation>  
</xs:element/>
```

After the EIPS insertion, an entry point is renamed according to the EIPS by adding "for____" where "____" stands for the chosen relevant part of the name of the EIPS. For example, in the case when the organization needs to insert an EIPS for a capacitor, the entry point is renamed as "EngineeringInformationForCapacitor"

```
<xs:element name="EngineeringInformationForCapacitor"  
  type="tns:EngineeringInformationForCapacitorType" minOccurs="0"  
  maxOccurs="unbounded">  
  <xs:annotation>  
    <xs:appinfo>  
      <urss:CreationDate>2007-05-23</urss:CreationDate>  
      <urss:Definition>Engineering information for Capacitor</urss:Definition>  
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>  
      <urss:TypeVersion>01.00</urss:TypeVersion>  
    </xs:appinfo>  
  </xs:annotation>  
</xs:element>  
:  
<xs:complexType name="EngineeringInformationForCapacitorType">  
  <xs:annotation>  
    <xs:appinfo>  
      <urss:CreationDate>2007-05-23</urss:CreationDate>  
      <urss:Definition>Engineering information type for  
Capacitor</urss:Definition>  
      <urss:LastUpdateDate>2007-05-23</urss:LastUpdateDate>  
      <urss:TypeVersion>01.00</urss:TypeVersion>  
    </xs:appinfo>  
  </xs:annotation>  
</xs:complexType>
```

Step 3: Add namespace declaration and schema location of TPIR-EIPS and Private-EIPS into the TPIR-PIP schema [Rule 3-3-7]

When the TPIR-EIPS and/or Private-EIPS are inserted under the entry point, the namespace declaration is added to the schema.

```
xmlns:eca="urn:rosettanet:partner:external:123456789:PartInfo:domain:Design:Ca  
pacitor:xsd:eips:01.01:01.00"  
xmlns:peca="urn:rosettanet:partner:external:123456789:PartInfo:domain:Design:M  
NC_Capacitor:xsd:eips:01.00"
```

For reference to the TPIR-EIPS and Private-EIPS, "import" is described in the TPIR-PIP Schema.

```
<xs:import  
namespace="urn:rosettanet:partner:external:123456789:PartInfo:domain:Design:Ca  
pacitor:xsd:eips:01.01:01.00"  
schemaLocation=" ../Domain/Design/Capacitor/MNC_Capacitor1_PartInfo_01_01_01_00  
.xsd"/>  
<xs:import  
namespace="urn:rosettanet:partner:external:123456789:PartInfo:domain:Design:MN  
C_Capacitor:xsd:eips:01.00"  
schemaLocation=" ../Domain/Design/Capacitor/MNC_Capacitor1_01_00.xsd"/>
```

3.3.3. TPIR-PIP File, Namespace, Versioning and Documentation Guidelines

Rule 3-3-9: The TPIR-PIP using for Engineering Information MUST comply with the rules for file, namespace, versioning and documentation guidelines of TPIR-PIP Design Specification.

Rationale: The TPIR-PIP for Engineering Information methodology is based upon the methods defined by the RAE Program. The ultimate deliverable from the TPIR-PIP for Engineering Information must conform to the guidelines specified in those practices.

4. References

Source	Description
EIM Engineering Report	Title : EIM Investigation Report V01.00 (15June2005) RosettaNet Retrieved March 03, 2008 from: http://members.rosettanet.org/dnn_rose/Standards/RosettaNetPrograms/FoundationalPrograms/ActiveFoundationalPrograms/EIMFoundational/InvestigationPhaseMaterials/tabid/1431/Default.aspx
[EIPS/XSD]	Title: RosettaNet EIPS XSD Specification V11.00.00 (05Mar2008) RosettaNet Retrieved March 05, 2008 from: http://members.rosettanet.org/dnn_rose/DMX/tabid/2979/DMXModule/624/Command/Core_ViewDetails/Default.aspx?EntryId=325
[NSM]	Title: RosettaNet Namespace Specification and Management Issue 01.00.00 (11Dec2003) RosettaNet Retrieved March 05, 2008 from: http://members.rosettanet.org/dnn_rose/DMX/tabid/2979/DMXModule/624/Command/Core_ViewDetails/Default.aspx?EntryId=4335
[RFC2119]	Author: Scott Bradner Title: Key words for use in RFCs to Indicate Requirement Levels The Internet Engineering Task Force Retrieved March 05, 2008 from: http://www.ietf.org/rfc/rfc2119.txt
[TPIR-PIP-DG]	Title: RosettaNet Automated Enablement, Trading Partner Implementation Requirements-Partner Interface Process™ (TPIR-PIP) Design Specification – V11.00.00 RosettaNet Retrieved March 05, 2008 from: http://members.rosettanet.org/Standards/RosettaNetStandards/TradingPartnerImplementationRequirements/tabid/480/Default.aspx
[XDG]	Title: RosettaNet XML Design Guideline Issue 01.00.00 (11Dec2003) RosettaNet Retrieved March 05, 2008 from: http://members.rosettanet.org/dnn_rose/DMX/tabid/2979/DMXModule/624/Command/Core_ViewDetails/Default.aspx?EntryId=4335

5. Glossary

Terms	Definition
Community Agreed	This phrase means PIPs, Specifications and other files, documents and tools released by RosettaNet Global in RosettaNet.
Complex Type	An XML element type that allows nested elements in their content and may carry attributes.
EIPS	Engineering Information Property Set. The community-agreed subset of the full set of characteristics for a specific piece of technical information.
EIPS/XSD	EIPS content represented in W3C XML Schema. Used wherever a distinction between the EIPS content in general and its XML Schema rendering is important.
Element	A fundamental unit of XML information, which has an element name, optional attributes, optional data value, and an associated type definition. Elements may be nested, one inside another.
Element declaration	An element declaration is an association of a name with a type definition, either simple or complex, an (optional) default value and a (possibly empty) set of identity-constraint definitions.
Entry Point	A structure that designates the point of extension within a PIP.
Namespace	An XML namespace is a collection of names identified by a URI reference, which are used in XML documents as element types and attribute names.
Namespace prefix	Namespace prefix is mapped to a URI reference, selects a namespace.
Organization	In the context of TPIR-PIP or TPIR-PIP for Engineering Information, it means organization which creates TPIR-PIP.
PIP with entry point	A PIP provided by RosettaNet that support the TPIR-PIP for Engineering Information methodology requirements specified by this document.
PIP-w/EP	See the term PIP with entry point.
Private EIPS	A partner-agreed EIPS (Engineering Information Property Set) contains technical property (ies) that is (are) not contained in community-agreed) EIPS/XSD.
RAE	RosettaNet Automated Enablement. It is a method to ease RosettaNet implementation for Small Medium Industry (SMI)/ Small Medium Enterprises (SME).
RNBD	The repository for business artifacts definitions.
RNTD	The repository for technical property and property set definitions.
TPIR-EIPS	Trading Partner Implementation Requirements- Engineering Information Property Set. A customized (restricted) EIPS/XSD in accordance with the partner requirements.
TPIR-PIP	A constrained business community agreed schema created by removing ambiguity and limiting PIP content.
TPIR-PIP for Engineering Information	A combination of a constrained business community schema with entry point and EIPS(s) (TPIR-EIPS and/or Private-EIPS).
XML Schema	An XML document that defines the allowable content of a class of XML documents. A class of documents refers to all possible permutations of structure in documents that will still confirm to the rules of the Schema.

6. Rules and Rationale

<p>Rule 3-1-1: TPIR-EIPS MUST be created from an EIPS/XSD.</p>	<p>Rationale: TPIR-EIPS is by definition a constrained version of an EIPS/XSD.</p>
<p>Rule 3-1-2: TPIR-EIPS MUST be XML Schema-based.</p>	<p>Rationale: TPIR-EIPS is a constrained version of an EIPS/XSD, which is XML Schema-based.</p>
<p>Rule 3-1-3: All EIPS/XSD restrictions MUST be compliant with the TPIR-PIP Design Specification.</p>	<p>Rationale: To ensure harmonization with the latest RosettaNet practices and improve enablement across the supply chain.</p>
<p>Rule 3-1-4: TPIR-EIPS file name MUST comply with the following requirements: 1) Requirement 3-1-4-1: The original EIPS/XSD MUST be easily identified. 2) Requirement 3-1-4-2: The TPIR-EIPS MUST identify the organization owner. 3) Requirement 3-1-4-3: The TPIR-EIPS MUST identify that it is an organization's representation of its requirements. Easily find out that the file is not the original EIPS/XSD</p>	<p>Rationale: Compliance with TPIR-PIP practices.</p>
<p>Rule 3-1-5: The TPIR-EIPS file name MUST carry both the version of the input EIPS/XSD, as well as, the version of the TPIR-EIPS itself.</p>	<p>Rationale: This complies with the TPIR-PIP design guideline rules.</p>
<p>Rule 3-1-6: An organization MUST assign the version number of TPIR-EIPS (TPIR-EIPS major version and TPIR-EIPS minor version) by following the versioning rules of EIPS/XSD.</p>	<p>Rationale: Compliance with EIPS/XSD practice.</p>
<p>Rule 3-1-7: The TPIR-EIPS MUST comply with TPIR-PIP namespace and namespace prefix rules.</p>	<p>Rationale: Since the TPIR-EIPS is created based upon a TPIR-PIP, the TPIR-EIPS namespace and namespace prefix must comply with TPIR-PIP rules.</p>
<p>Rule 3-1-8: The TPIR-EIPS namespace prefix MUST start with an "e" for EIPS then followed by two characters representing the EIPS category (i.e. capacitor, connector, etc.). If the initial two characters of an EIPS category overlap, then the overlap is removed by selecting the next unused letter in the name of the newly overlapping EIPS. If all letters have already been used in the name, choose the first unused letter in the alphabet.</p>	<p>Rationale: The namespace prefix should be generated using simple rules and should be persistent for a specific EIPS.</p>
<p>Rule 3-1-9: The TPIR-EIPS file MUST be stored under the Design domain.</p>	<p>Rationale: These definitions are only used in the domain of design engineering and are not used in any other domain.</p>
<p>Rule 3-2-1: A Private-EIPS MUST comply with the technical requirements of the EIPS/XSD Specification.</p>	<p>Rationale: Although the content is defined by the MNC, the format must comply with RosettaNet general practices and rules to prevent the system from having to process the public and private extensions differently.</p>

<p>Rule 3-2-2: A Private-EIPS file name MUST comply with following requirements: 1) Requirement 3-2-2-1: Easily discriminate the group or category of the Private-EIPS. 2) Requirement 3-2-2-2: Easily identify the organization that creates the TPIR-EIPS.</p>	<p>Rationale: Compliance with the naming practices within the TPIR-PIP Design Specification.</p>
<p>Rule 3-2-3: Private-EIPS version SHOULD comply with the versioning rules of EIPS/XSD.</p>	<p>Rationale: Using the same versioning rules for a TPIR-EIPS and applying it to a Private-EIPS reduces the versioning management by an organization for EIPS'.</p>
<p>Rule 3-2-4: Private-EIPS namespace prefix SHOULD begin with "pe" (Private-EIPS) then followed by two characters identifying the category (capacitor, connector, etc.). If the initial two characters of an EIPS category overlap, then the overlap is removed by selecting the next unused letter in the name of the newly overlapping EIPS. If all letters have already be used in the name, choose the first unused letter in the alphabet.</p>	<p>Rationale: The namespace prefix should be generated using simple rules and should be persistent for a specific EIPS (Note: Same as Rule 3-1-7).</p>
<p>Rule 3-2-5: A Private-EIPS MUST comply with the URN format defined in the RosettaNet Namespace Specification and Management document.</p>	<p>Rationale: Comply with RosettaNet Namespace Specification.</p>
<p>Rule 3-2-6: A Private-EIPS file MUST be stored under Design domain.</p>	<p>Rationale: Private properties are only used in the domain of design engineering and are not used in any other domain, like a TPIR-EIPS.</p>
<p>Rule 3-3-1: Restriction of the community-agreed PIP MUST complies with the rules defined in the TPIR-PIP Design Specification.</p>	<p>Rationale: The restriction process of a RosettaNet community PIP is defined in the TPIR-PIP Design Specification.</p>
<p>Rule 3-3-2: The PIP applied to the TPIR-PIP for Engineering Information method MUST contain entry points for additional engineering information.</p>	<p>Rationale: TPIR-PIP prohibits the extension of a community agreed PIP XML Schema as described in "TPIR-PIP Design Specification". Entry points serve as a denotation of the placement for engineering information.</p>
<p>Rule 3-3-3: An engineering information entry point type is defined as a complexType that MUST contain a special xs:annotation/urss:Definition element with the value of "Entry point type for additional engineering information".</p>	<p>Rationale: This is a method to allow an application processor to distinguish an engineering information entry point type from other types.</p>
<p>Rule 3-3-4: The placement and the cardinality of an engineering information entry point MUST be specified by a declaration of an element that corresponds to the definition of a complexType as defined under [Rule 3-3-3]. This element MUST contain a special xs:annotation/urss:Definition element with the value of "Entry point for additional engineering information".</p>	<p>Rationale: One or more entry points could be placed in various places of the enclosing engineering information structures. At the same time the declaration of the element for an entry point allows for the replacement of an entry point with various TPIR-EIPS/Private-EIPS pairs in a final engineering information TPIR-PIP.</p>

<p>Rule 3-3-5: The concrete element that is inserted under an entry point MUST be declared as a combination of 0 or 1 TPIR-EIPS and 0 or 1 Private-EIPS. TPIR-EIPS and Private-EIPS MUST be placed in alphabetical order.</p>	<p>Rationale: Combining a TPIR-EIPS with its corresponding Private-EIPS follows a good abstraction pattern. Regarding sorting, RosettaNet XML Design Guideline requires that components of an XML Schema SHOULD be sorted in alphabetical order.</p>
<p>Rule 3-3-6: During the EIPS Insertion Process [Fig. 2.1], an organization MUST rename the element name and type name, which has the entry point, according to the inserted TPIR-EIPS and/or Private-EIPS.</p>	<p>Rationale: Using the name of the EIPS itself facilitates human understanding. Also, by assigning a different name it is possible to place different EIPS' in the same enclosing engineering structure.</p>
<p>Rule 3-3-7: During the EIPS Insertion Process [Fig. 2.1], an organization MUST add namespace declaration and import for reference of inserted TPIR-EIPS and Private-EIPS.</p>	<p>Rationale: In order to use other schemas efficiently by the XML Schema, namespace declaration and import are used.</p>
<p>Rule 3-3-8: When an organization needs to use two or more TPIR-EIPS' and/or two or more Private-EIPS', an organization MUST NOT arbitrarily mix TPIR-EIPS' and/or Private-EIPS'. Each corresponding pair of TPIR-EIPS and Private-EIPS MUST be placed in a separate element as defined under [Rule 3-3-5].</p>	<p>Rationale: Mixing two or more EIPS' under the same entry point causes ambiguity of the content of the entry point.</p>
<p>Rule 3-3-9: The TPIR-PIP using for Engineering Information MUST comply with the rules for file, namespace, versioning and documentation guidelines of TPIR-PIP Design Specification.</p>	<p>Rationale: The TPIR-PIP for Engineering Information methodology is based upon the methods defined by the RAE Program. The ultimate deliverable from the TPIR-PIP for Engineering Information must conform to the guidelines specified in those practices.</p>