**S-100 – Part 18**

**Language Packs**

Page intentionally left blank

**Contents**

[18-1 Scope 1](#_Toc95667192)

[18-2 Normative References 1](#_Toc95667193)

[18-3 General Description 1](#_Toc95667194)

[18-4 Language Pack Lifecycle 2](#_Toc95667195)

[18-4.1 Authoring 2](#_Toc95667196)

[18-4.2 Distribution 2](#_Toc95667197)

[18-4.3 Implementation. 2](#_Toc95667198)

[18-5 Language Pack Structure Overview. 3](#_Toc95667199)

[18-6 Schema Description (Normative) [HB] 3](#_Toc95667200)

[18-7 Annex A(?) S-100 Feature Catalogues (Informative) 5](#_Toc95667201)

Page intentionally left blank

# Scope

This part of S-100 details how multi-lingual support for XML elements of the framework may be implemented. A generic mechanism and structures are described for production of individual language packs which implement translations of any XML content.

This is designed to provide multi-lingual instances of XML resources which support product specifications for provision to end users. Implementing systems are then able to construct translated instances of those supporting resources. This part is not specific to any one individual class of XML resource. It does not detail how multi-lingual support may be added to S-100 product specifications datasets or any external resources they may reference. It provides a generic mechanism which can be applied to any XML based elements of the S-100 framework to adapt them for multi-lingual implementations.

This part of S-100 provides the generic methodology for implementing such support and informative examples for a primary use case, the creation of multi-lingual support for S-100 Feature Catalogues.

# Normative References

The following referenced documents are required for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including amendments) applies:

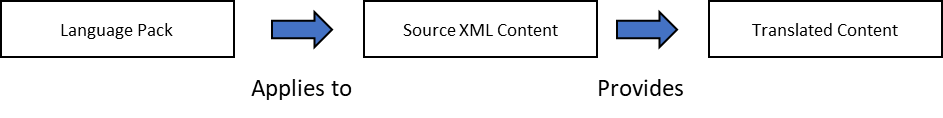
* XML Schema [XML Schema Ref]
* XPath Specification [XPath Ref:]
* ISO 639-2/T for specification of languages. [ISO Language]

# General Description

Under S-100 a number of XML Schemas are defined. These are used in variety of contexts within the framework and by data producers and implementers to define XML content. This part introduces the concept of “language packs” to enable multi-lingual support specific to XML content.

For specific XML “source” content, a language pack is a published XML dataset which provides translations of selected elements of the source content. A language pack is, itself, XML content with a schema defined under S-100, described in section 18-6.

This mechanism enables transformation of selected elements of the source content from one language (the “source language”) to another (the “target language”).



Multiple language packs may also be constructed to define multi-lingual translations of (potentially different) content from the Source XML. For example:

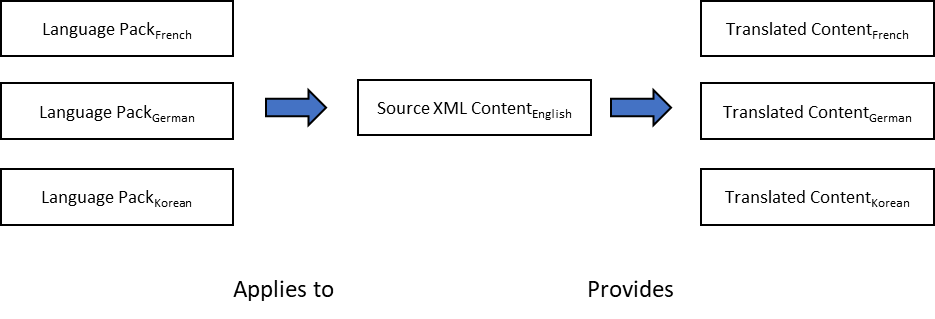
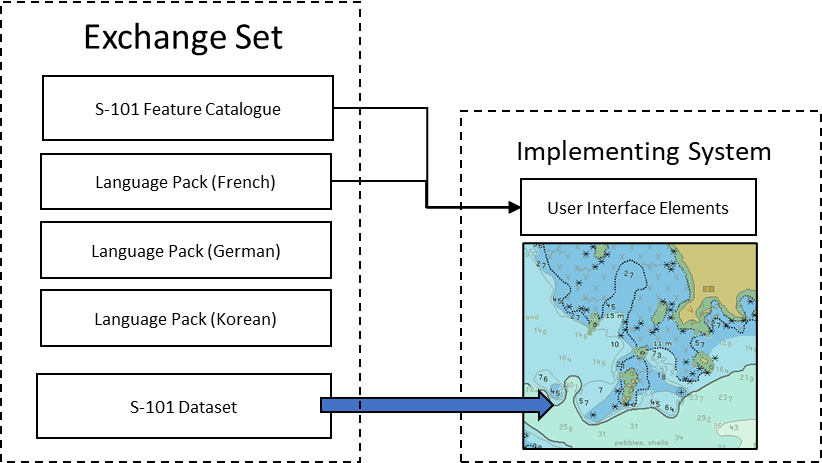


Figure 1: Example multi-lingual support

Note that the language pack only provides a list of translations for specific identified elements within the source content. A version of the source content with all the relevant elements translated is not necessarily produced. The language pack and the source content are distributed and used together by implementing systems to provide multi-lingual support to an end user

For example, in the figure below an XML feature catalogue and three language packs, together with a dataset are delivered in an exchange set to an implementing system. The dataset provides the data content and the feature catalogue and its language packs provides multi-lingual support for the user interface elements (e.g legends and textual descriptions of features).



# Model

The model of a language pack is described by the following diagram.

Diagram

Description automatically generated

The ***TranslationPackageType*** consists of a minimal set of header information and a sequence of source files elements. Each of the ***SourceFileType*** elements has a header to identify the resource and a set of items that hold the information for one text element in the resource to be translated.

The details are in the following tables. Note that most classes exist in a namespace **S100\_LA**. This prefix is not part of the class names but logically belongs to them. Exceptions are marked in the tables.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role Name** | **Name** | **Description** | **Mult** | **Type** | **Remarks** |
| Class | TranslationPackageType | The main element of a language package. | - | - |  |
| Role | language | The language in that the translations are made. |  | LanguageType | The type contains an attribute ***language*** that hold language codes according to ISO639-2/T (e.g deu, fra) |
| Attribute | issueDate | The date when the language pack is issued | 1 | Date |  |
| Attribute | issueTime | The time when the language pack is issued | 0..1 | Time |  |
| Attribute | responsibleParty | Meta information about the responsible organisation or individual | 0..\* | CI\_ResponsibleParty | This type conforms to ISO 19115 and is not defined in the namespace ***S100\_LA.***  There may be more than one parties responsible which then have different roles. (e.g, custodian, translator, or publisher) |
| Role | sourceFile | Holding all translation items for one source file | 1..\* | SourceFileType |  |

**TranslationPackageType**

**SourceFileType**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role Name** | **Name** | **Description** | **Mult** | **Type** | **Remarks** |
| Class | SourceFileType | An element holding all information on elements to be translated of a single source file. | - | - | The source file must be in XML format |
| Role | header | Information to identify the source file. | 1 | SourceHeaderType |  |
| Role | translationItem | A list of items each describes an element of the source file the is subject to translation. | 0..\* | TranslationItemType |  |

**SourceHeaderType**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role Name** | **Name** | **Description** | **Mult** | **Type** | **Remarks** |
| Class | SourceHeaderType | The information to identify the source file for that this translation package contains the translations | - | - |  |
| Attribute | resourceIdentification | The identifier of the resource | 1 | URI | This can be either a file URI or a hash URN.  Example:  file:///somefile.xml |
| Role | identification | Information on version information inside the source file | 0..\* | ResourceIdentification |  |

**ResourceIdentification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role Name** | **Name** | **Description** | **Mult** | **Type** | **Remarks** |
| Class | ResourceIdentification | Information where a version information can be found in the source (XML) file and what value is stored there. | - | - |  |
| Attribute | path | An XPath expression to the element or attribute that contains the identification information | 1 | CharacterString | e.g. /S100FC:S100\_FC\_FeatureCatalogue/S100FC:versionNumber |
| Attribute | value | The value that must be in the specified element to match the source file. | 1 | CharacterString | e.g.  1.0.0 |

**TranslationItemType**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role Name** | **Name** | **Description** | **Mult** | **Type** | **Remarks** |
| Class | TranslationItemType | Information on a single translatable item in the source file. | - | - |  |
| Attribute | path | An XPath expression to the element or attribute that contains the text to be translated | 1 | CharacterString |  |
| Attribute | original | The text as it appears in the source file | 0..1 | CharacterString |  |
| Attribute | status | The status of the item | 0..1 | Status |  |
| Attribute | translation | The translated text. | 1 | CharacterString |  |

**Status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role Name** | **Name** | **Description** | **Remarks** |
| Enumeration | Status | The status of the translation item. | The purpose is to support the life cycle of each translation item. |
| Literal | New | The item is not yet translated. |  |
| Literal | Modified | The content of the item has been changed in the source file. | The translation must be revisited because it may not be valid anymore. |
| Literal | Deleted | The item does not exist in the source file anymore. |  |
| Literal | Translated | The item is translated and ready to use. |  |

# Language Pack Creation

# Authoring

Language Pack authors are responsible for the definition of translated content for a particular language for selected elements of content. While some elements may always require translations (such as definitions of features or information types), others (such as language nonspecific enumeration names) may not. It is the responsibility of the language pack author to determine which elements require translation.

Translations may be revised over time and so the schema makes provision for revisions to, optionally, be contained within a language pack. These are not expected to be implemented by end user systems, however, and each language pack delivered to end user systems defines a complete translation of the required elements in the source XML into the target language.

# Distribution

S-100 Part 17 Exchange Catalogue enables the inclusion of 0 or more language packs within S-100 exchange sets for distribution to end users. Such language packs can be included alongside the content to which they refer or independently to supplement pre-installed content on the end user system.

Where they are delivered alongside the content to which they refer, then they shall be referenced to the content by association in the exchange catalogue using the mechanisms defined for supporting resources within that part. As exchange set content they will be digitally signed.

There is no specific filename convention for language packs but as supporting resources they shall use the filename convention defined in Part 17.

To avoid ambiguity exchange sets shall only contain a single language pack for any given source resource in a given target language.

# Implementation.

The implementer of the S-100 system is responsible for correctly interpreting delivered language packs and performing the content value substitution correctly. The implementer is also responsible for providing support for multiple language packs and any harmonization with multi-lingual support within S-100 product datasets. This can provide the end user with a harmonized experience where data content and associated user interface elements are configured using the same language specifiers.

# Language Pack Creation.

Schema Overview

Under this part of S-100 a language pack is created as XML content conforming to the language pack schema documented in section 18-6. This schema is comprised of two components:

1. A header defining
   1. The language implemented in the language pack
   2. The language pack producer and time/date of issue
2. Translation entries for one (or more) source XML resources. Each source XML resource is identified by a URI and a single element which identifies its revision. A sequence of translation entries for each source is given, each containing (as a minimum):
   1. A location specifier of an element in the source XML. This is specified using an XPath specification.
   2. The translated text of the element

For the purposes of assisting the process of language pack creation, and maintenance the language pack schema may also optionally contain, for each translation entry:

1. The source content, from which the translation is made
2. A state value for the translation

The root translationPackage element defines the language supported by the language pack (the destination language), the issue date and time and language pack producer. The translationPackage then contains any number of sourceFile translations providing translations into the destination language. All languages are specified using ISO639-2/T with descriptors contained in the S-100 Schema codelists. This ensures languages specified in language packs use the same descriptors as languages contained in dataset encodings.

The sourceFile element contains a header identifying the source xml file for which the translation is provided. Unique identification of the source resource is accomplished by matching defined XPath resources in the source with defined values in the language pack translationPackage identification element. This is because such unique identifiers may differ between different S-100 XML schemas and so a general path specification is used to enable unique identification of any XML content within S-100.

TranslationItems define the translations themselves. The *path* attribute in an instance of a *TranslationItemType* will point to either an element or an attribute in the source file using an XPath expression. The expression must uniquely define one element, hence a query on the XML DOM tree must return exactly one node.

For example, given the following XML source content:

<S100FC:S100\_FC\_FeatureCatalogue xmlns:S100FC=<http://www.iho.int/S100FC> …  
…  
 <S100FC:S100\_FC\_SimpleAttributes>  
 <S100FC:S100\_FC\_SimpleAttribute>  
 <S100FC:name>Category of Topping</S100FC:name>  
 <S100FC:definition>Topping on a pizza (a maximum of four)</S100FC:definition>  
 <S100FC:code>categoryOfTopping</S100FC:code>

A simplified tree structure of the XML file would look like:

The XPath expression:

/S100FC:S100\_FC\_FeatureCatalogue/S100FC:S100\_FC\_SimpleAttributes/S100FC:S100\_FC\_SimpleAttribute[./S100FC:code/text()='categoryOfTopping']/S100FC:name

would locate the marked element by locating wlements with the name ***S100FC:S100\_FC\_Simple\_Attribute*** with parent ***S100FC:S100\_FCSimpleAttributes*** and then ***S100FC:S100\_FC\_FeatureCatalog*** that has a child ***S100FC:code*** with a text of “*categoryOfTopping*”. The child **S100FC:name** is selected and used as the reference for translation. In the case of more elements have the same code the Xpath expression would be ambiguous. In feature catalogues the codes are always unique so using XPath expressions referring to codes guarantees an unambiguous path specification in the TranslationItem. The TranslationItem then contains a translation of the text of the element (the name) in the destination language and any optional state or revision information.

# Schema Description

The schema is defined in the namespace:

xmlns="<http://www.iho.int/s100/la>"

It uses types from two ISO 19115 schemas

xmlns:gco="http://standards.iso.org/iso/19115/-3/gco/1.0"

xmlns:cit="<http://standards.iso.org/iso/19115/-3/cit/2.0>"

The following settings are made for this schema

targetNamespace="http://www.iho.int/S100/la"

elementFormDefault="qualified"

attributeFormDefault="unqualified"

In the schema the type **Status** is implemented as a simple type as follows:

<xs:simpleType name="Status">

<xs:annotation>

<xs:documentation>

The status of the translation item. The purpose is manly to support the  
 functionality of an translation tool.

</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:enumeration value="New">

<xs:annotation>

<xs:documentation>

The item is new, there is no translation available yet.

</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="Modified">

<xs:annotation>

<xs:documentation>

The original text has been changed in the source

</xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="Deleted">

<xs:annotation>

<xs:documentation>

The text defined by path is not longer available in the source  
 document.  
 </xs:documentation>

</xs:annotation>

</xs:enumeration>

<xs:enumeration value="Translated">

<xs:annotation>

<xs:documentation>

The item has a valid translation.

</xs:documentation>

</xs:annotation>

</xs:enumeration>

</xs:restriction>

</xs:simpleType>

For the definition of the language a complex type **LanguageType** is defined that is using an element of the type **gco:CodeListValue\_Type** (from ISO 19115)

<xs:complexType name="LanguageType">

<xs:sequence>

<xs:element name="languageCode" type="gco:CodeListValue\_Type"/>

</xs:sequence>

</xs:complexType>

Each translation item is defined by the type **TranslationItemType**:

<xs:complexType name="TranslationItemType">

<xs:annotation>

<xs:documentation>

One item to be translated. This will be a uniquely identifiable element or

attribute in a source (XML) file

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="path" type="xs:string">

<xs:annotation>

<xs:documentation>

The XPath that defines the 'source' text.

</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="original" type="xs:string" minOccurs="0">

<xs:annotation>

<xs:documentation>

The original text as exists in the source document.

</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="status" type="Status" minOccurs="0">

<xs:annotation>

<xs:documentation>

The status of the translation item

</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="translation" type="xs:string">

<xs:annotation>

<xs:documentation>

The translated text.

</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

The type **ResourceIdentification** contains the information to identify a source file e.g. by defining the XPath to the version element and the value that this element in the source file must have.

<xs:complexType name="ResourceIdentification">

<xs:annotation>

<xs:documentation>

Information to identify a specific version of an (XML) file.

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="path" type="xs:string">

<xs:annotation>

<xs:documentation>

The XPath to the element or attribute that allows the unique identification

of the source file.

</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="value" type="xs:string">

<xs:annotation>

<xs:documentation>

The value of the element or attribute that describes the identification of

the source file. e.g. the version or issue date

</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

The type **SourceHeaderType** contains the information on the source file as the filename and an identification mechanism.

<xs:complexType name="SourceHeaderType">

<xs:annotation>

<xs:documentation>

Information to identyfing the source file. It supports mechanism to distinguish

different versions of a source file.

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="resourceIdentifier" type="xs:anyURI"/>

<xs:element name="identification" type="ResourceIdentification" minOccurs="0"

maxOccurs="unbounded">

<xs:annotation>

<xs:documentation>

Identification by one or more elements or attributes in the source

file.

</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

The type **SourceFileType** contains the header information of the file and a list of translation items for that file.

<xs:complexType name="SourceFileType">

<xs:sequence>

<xs:element name="header" type="SourceHeaderType"/>

<xs:element name="translationItem" type="TranslationItemType" minOccurs="0"

maxOccurs="unbounded">

<xs:annotation>

<xs:documentation>

The list of translation items.

</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

The last type defined by the schema is the type **TranslationPackageFile**

<xs:complexType name="TranslationPackageType">

<xs:sequence>

<xs:element name="language" type="LanguageType">

<xs:annotation>

<xs:documentation>

The language that will be supported by this translation

file.

</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="issueDate" type="xs:date"/>

<xs:element name="issueTime" type="xs:time" minOccurs="0"/>

<xs:element name="responsibleParty" type="cit:CI\_Responsibility\_PropertyType"   
 minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="sourceFile" type="SourceFileType" maxOccurs="unbounded">

<xs:annotation>

<xs:documentation>

The list of source files for that this translation file provides

translations.

</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

this type defines some elements for the metadata of the translation file as

* language
* issue date
* issue time
* and responsible parties

and a list of source files with their translations.

Note that for the responsible party the type **cit:CI\_Responsibility\_PropertyType** is used.

A possible encoding would look like:

<S100LA:responsibleParty>

<cit:CI\_Responsibility>

<cit:role>

<cit:CI\_RoleCode codeList="codeListLocation#CI\_RoleCode"

codeListValue="custodian">custodian</cit:CI\_RoleCode>

</cit:role>

<cit:party>

<cit:CI\_Individual>

<cit:name>

<gco:CharacterString>Max Mustermann</gco:CharacterString>

</cit:name>

</cit:CI\_Individual>

</cit:party>

</cit:CI\_Responsibility>

</S100LA:responsibleParty>

Alternative to the element <cit:CI\_Individual> the element <cit:CI\_Organisation> can be used in case the producer is an organisation rather than an individual.

Note that the predefined list of roles for a responsible party are:

* *resourceProvider*
* *custodian*
* *owner*
* *user*
* *distributor*
* *originator*
* *pointOfContact*
* *principalInvestigator*
* *processor*
* *publisher*
* *author*
* *sponsor*
* *coAuthor*
* *collaborator*
* *editor*
* *mediator*
* *rightsHolder*
* *contributor*
* *funder*
* *stakeholder*

There is no value for ‘*translator’*. ‘*contributor’* could be used for this role if required.

Finally, the schema defines the root element of a translation file.

<xs:element name="translationPackage" type="TranslationPackageType">

<xs:annotation>

<xs:documentation>

The root element of a translation file.

</xs:documentation>

</xs:annotation>

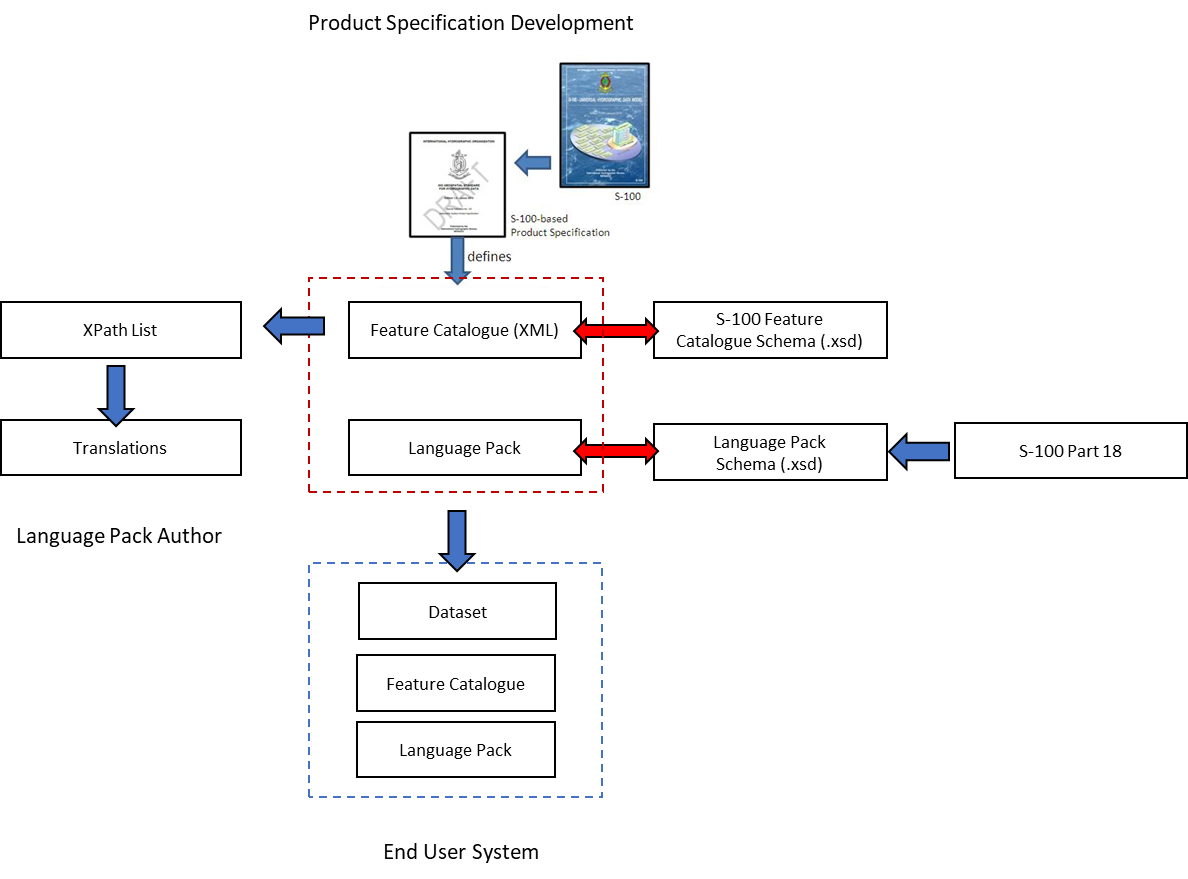
</xs:element>

# Annex A : S-100 Feature Catalogues (Informative)

Although, as stated in the introduction to this section, language pack production can be applied to any S-100 XML content, the primary use case is the production of language packs to support feature catalogues for implementing systems. This allows systems (either for end users or data producers) to use translated versions of feature catalogue entries such as features, information types, attributes and relationships.

In this example implementation an individual feature catalogue, defined by an S-100 product specification process contains a number of elements which could, potentially, be translated.

A set of translations is prepared of those fields required and a language pack is constructed, conformant with the language pack schema, which implements a particular language for the given feature catalogue.



This language pack, together with the feature catalogue can then be distributed to end users for implementation by the end user system. The end user system is able to either construct a version of the feature catalogue in the language given, or interleave the translated items as required, at runtime. This is implementation-specific.

As certain elements are used as references in feature catalogues to provide bindings between entries (e.g. feature attribute binding references) care should be taken to ensure such mappings remain intact, should end user systems choose to create complete translations of the feature catalogue at runtime. This would preclude the S100\_FC\_Item code field from translation as it provides the reference in attribute and sub-attribute bindings.

Language packs for feature catalogues should use the following fields for product and version identification of the source feature catalogue:

* /S100FC:S100\_FC\_FeatureCatalogue/S100FC:name[1]
* /S100FC:S100\_FC\_FeatureCatalogue/S100FC:versionNumber[1]

The following fields are candidates for translation in a feature catalogue language pack (feature catalogue XPath locations)

* /S100FC:S100\_FC\_FeatureCatalogue/S100FC:scope[1]
* /S100FC:S100\_FC\_FeatureCatalogue/S100FC:fieldOfApplication[1]
* Name and Definition in any element derived from S100\_FC\_Item. E.g.
  1. /S100\_FC\_SimpleAttributes/S100\_FC\_SimpleAttribute/name
  2. /S100\_FC\_SimpleAttributes/S100\_FC\_SimpleAttribute/definition
  3. /S100\_FC\_SimpleAttributes/S100\_FC\_SimpleAttribute/remarks
  4. Simple Attribute enumeration labels and definitions:
     1. /listedValues/listedValue/S100FC:label
     2. /:listedValues/listedValue/definition
  5. Complex Attributes
  6. Roles
  7. Information Associations
  8. FeatureAssociations
  9. InformationAssociations
  10. InformationTypes
  11. FeatureTypes

As stated in Section XXX, Feature Catalogues may produce translations of some or all of these fields, depending on the preference of the language pack author.

An example of some of these fields are shown in the extract from the S-101 feature catalogue below:

<S100FC:S100\_FC\_SimpleAttribute>  
 <S100FC:name>Category of Light</S100FC:name>  
 <S100FC:definition>Classification of different light types.</S100FC:definition>  
 <S100FC:code>categoryOfLight</S100FC:code>  
 <S100FC:remarks>All lights are considered to be marine lights unless the category of light indicates otherwise.</S100FC:remarks>  
 <S100FC:alias>CATLIT</S100FC:alias>  
 <S100FC:definitionReference>  
 <S100FC:sourceIdentifier>88</S100FC:sourceIdentifier>  
 <S100FC:definitionSource ref="IHOREG"/>  
 </S100FC:definitionReference>  
 <S100FC:valueType>enumeration</S100FC:valueType>  
 <S100FC:listedValues>  
 <S100FC:listedValue>  
 <S100FC:label>Leading Light</S100FC:label>  
 <S100FC:definition>A light associated with other lights so as to form a leading line to be followed.</S100FC:definition>  
 <S100FC:code>4</S100FC:code>  
 <S100FC:definitionReference>  
 <S100FC:sourceIdentifier>863</S100FC:sourceIdentifier>  
 <S100FC:definitionSource ref="IHOREG"/>  
 </S100FC:definitionReference>  
 </S100FC:listedValue>  
 <S100FC:listedValue>  
 <S100FC:label>Aero Light</S100FC:label>  
 <S100FC:definition>An aero light is established for aeronautical navigation and may be of higher power than marine lights and visible from well offshore.</S100FC:definition>  
 <S100FC:code>5</S100FC:code>  
 <S100FC:definitionReference>  
 <S100FC:sourceIdentifier>864</S100FC:sourceIdentifier>  
 <S100FC:definitionSource ref="IHOREG"/>  
 </S100FC:definitionReference>  
 </S100FC:listedValue>

This section of the S-101 feature catalogue would result in the following translations included in a French S-101 feature catalogue language pack:

|  |  |  |
| --- | --- | --- |
| **Xpath** | **English Text** | **French Text** |
| **Name** | Category of Light | [TBD] |
| **Definition** | Classification of different light types |  |
| **Remarks** | All lights are considered to be marine lights unless the category of light indicates otherwise |  |
|  |  |  |
| **listedValue/label** | Leading Light |  |
| **listedValue/definition** | A light associated with other lights so as to form a leading line to be followed. |  |
|  |  |  |
| **listedValue/label** | Aero Light |  |
| **listedValue/definition** | An aero light is established for aeronautical navigation and may be of higher power than marine lights and visible from well offshore |  |

The values in the table would be encoded in the French S-101 language pack, as demonstrated in the following XML fragment.

[Insert French XML text conformant with language pack XML]

This language pack would be named [FILEREF:LANG] and would support a feature catalogue using the following language pack XML content:

[Language Pack XML content identifying the revision of the feature catalogue]

Note use of the product and version number to identify the feature catalogue. Both feature catalogue and language pack can be aggregated in an S-100 conformant exchange catalogue for distribu

tion to end users.