Paper for Consideration by S-100TSM7

S-1xx Exchange Catalogue file structuring for multiple product distribution

Submitted by:	PRIMAR, IC-ENC, IIC, SevenCs, Furuno, CHS, NOAA, DHN, NIWC
Executive Summary:	This is a discussion paper related to the structuring of exchange catalogues for multiple product distribution.
Related Documents:	S-100WG4_2019_4.2_ExchangeCatalogueFileNaming
Related Projects:	

Introduction / Background

After discussions on S-100WG4 on the topic presented in paper S-100WG4_2019_4.2_ExchangeCatalogueFileNaming, the following action was taken:

"No. 1 Item 4.2: Investigate how best to structure exchange catalogues for distributing different Product Specification datasets, (as part of an exchange set) for TSM7".

This paper discusses solutions for how to:
1. register multiple products carried in the exchange set.
2. incorporate necessary S-1xx information in one (1) catalogue (CATALOG.XML) file.
3. organize the exchange set directory structure.

Analysis

1. Currently, the attribute productSpecification defined in S100_ExchangeCatalogue has the multiplicity [0..1].

Extract from S-100 Figure 4a-D-4

When encoding this attribute, the encoder limits the content of the exchange set to containing one product only. When not encoding the attribute, there is no other mechanism except from parsing the whole CATALOG.XML file telling the end user system which products are available within the exchange set. This information is then accessible through the S100_DatasetDiscoveryMetadata attribute productSpecification.

If the attribute productSpecification defined in S100_ExchangeCatalogue changes multiplicity to [1..*], we will have a mechanism defined identifying the products within the catalogue without the need to parse through all of the CATALOG.XML files content.

S100_ExchangeCatalogue

+	identifier: S100_CatalogueIdentifier
+	contact: S100_CataloguePointofContact
+	productSpecification: S100_ProductSpecification [1..*]
+	metadataLanguage: CharacterString
+	exchangeCatalogueName: CharacterString
+	exchangeCatalogueDescription: CharacterString
+	exchangeCatalogueComment: CharacterString [0..1]
+	compressionFlag: Boolean [0..1]
+	sourceMedia: CharacterString [0..1]
+	replacedData: Boolean [0..1]
+	dataReplacement: CharacterString [0..1]

Proposed change

The main benefit for the end user system would be immediately recognizing whether the exchange set contains supported products or not, being able to reject the catalogue immediately without parsing through the content of the CATALOG.XML file if it only contains unsupported products.

This mechanism also supports the inclusion of datasets for one product based on different versions of the product specification, e.g. S-101 1.00 and S-101 1.5.0 being packaged within the same catalogue. The solution is then to individually encode the attribute productSpecification with the different product versions accordingly.

2. All relevant S_1xx information within the exchange catalogue should be accessible through only one CATALOG.XML file. This then means that all product specifications should use the naming convention CATALOG.XML (and not the different variations pointed out in paper S-100WG4_2019_4.2_ExchangeCatalogueFileNaming).

The benefits would be that we do not need to encode metainformation in the CATALOG.XML filename and reduce the number of catalog files necessary in end user system installation process. Also, there will only be one CATALOG.XML that needs a digital signature for cyber security reasons.

A change must be done in S-100:
Part 4a Metadata S100_ExchangeCatalogue current situation:

Attribute	exchangeCatalogueName	Catalogue filename	1	CharacterString	In S-101 it would be CATLOG.101

Part 4a Metadata S100_ExchangeCatalogue proposed solution:

Attribute	exchangeCatalogueName	Catalogue filename	1	CharacterString	Must be named CATALOG.XML

Part 4a Metadata Overview third paragraph (page 26) – Added textual statement (Highlighted):
"The S100_ExchangeCatalogue is an XML instance, which provides the information needed to exploit all the components of an exchange set. It consists of sections for the catalogues and datasets with subsections for support file metadata and a reference to classic ISO 19115-1 dataset metadata. The Exchange Catalogue must be named CATALOG.XML".

3. For future product delivery purposes it would be beneficial to include S-57 files and S-1xx within the same catalogue. Now S-100 defines the directory EXCH_ROOT as the directory of the files.

Role Name	Name	Description	Mult	Type	Remarks
Attribute	filePath	Full path from the exchange set root directory	1	CharacterString	Path relative to the root directory of the exchange set. The location of the file after the exchange set is unpacked into directory <EXCH_ROOT> will be <EXCH_ROOT>/<filePath>/<filename>
Attribute	fileLocation	Full location from the exchange set root directory	1	CharacterString	Path relative to the root directory of the exchange set. The location of the file after the exchange set is unpacked into directory <EXCH_ROOT> will be <EXCH_ROOT>/<filePath>/<filename>

EXCH_ROOT defined in the discovery metadata attribute filePath/fileLocation comments fields

It is anticipated that S-57 and S-101 will coexist for several years. For future data delivery we envisage being able to deliver both S-57 and S-10x datasets in the same exchange set. To ensure backwards compatibility it is essential that S-57s ENC_ROOT and INFO directories are found in the top directory structure of the exchange set. Changing the top directory from ENC_ROOT to EXCH_ROOT could cause issues with backward compatibility for existing end user systems. Existing systems using S-57 have incorporated data access based on this structure.

The solution could be to add a new directory to the top structure for S-1xx data.

The dataset files, support files and catalogues from S-57 and S-1xx products could be structured as the following example illustrates. ENC_ROOT and INFO (as currently in S-57) and S100_ROOT are top level directories. S100_ROOT serves as the directory for all S-100 product dataset files, support files and catalogues. As the example demonstrates, it can carry catalogues from different versions of the same product specification. Further on, the structuring of support files in a separate directory opposed to today's S-57 solution, is also demonstrated. The CATALOG.XML file should be placed in the S100_ROOT directory, together with its signature (CATALOG.SIGN). All other signatures are included in the CATALOG.XML.

A proposed example of the files structured in directories:

```
ENC_ROOT
INFO
S100_ROOT
    I----S101
    I       I----DATASET_FILES
    I       I        I----AR00
    I       I        I      I----101AR001234567890
    I       I        I      I            I----1
    I       I        I      I                 I----0
    I       I        I      I                 I    I----101AR001234567890.000
    I       I        I      I                 I----1
    I       I        I      I                 I    I----101AR001234567890.001
    I       I        I      I                 I----2
    I       I        I      I                      I----101AR001234567890.002
    I       I        I      I----101AR00ABCDEFGHIJ
    I       I        I      I            I----1
    I       I        I      I                 I----0
    I       I        I      I                      I----101AR00ABCDEFGHIJ.000
    I       I        I----CL00
    I       I        I----NO00
    I       I----SUPPORT_FILES
    I       I        I----AR00
    I       I        I      I----101AR00QWERTYUIOP.TXT
    I       I        I      I----101AR00ASDFGHJKLO.TIF
    I       I        I      I----101AR00ZXCVBNMJKL.HTM
    I       I        I----CL00
    I       I        I----NO00
    I       I----CATALOGUES
    I       I        I----101_1_0_0_FC.XML
    I       I        I----101_1_5_0_FC.XML
    I       I        I----101_1_0_0_PC.XML
    I       I        I----101_1_5_0_PC.XML
    I----S102
    I       I----DATASET_FILES
    I       I        I----AR00
    I       I        I      I----102AR001234567890
    I       I        I      I            I----1
    I       I        I      I                 I----0
    I       I        I      I                 I    I----102AR001234567890.000
    I       I        I      I----102AR00ABCDEFGHIJ
    I       I        I----CL00
    I       I        I----NO00
    I       I----CATALOGUES
    I       I        I----102_2_0_0_FC.XML
    I       I        I----102_2_0_0_PC.XML
    I----S104
    I----CATALOG.XML
    I----CATALOG.SIGN
```

Because of this structure the references to EXCH_ROOT in S-100 should be replaced with updated information.
S-100 refers to EXCH_ROOT in the following sections:

Part 4a Metadata S100_DatasetDiscoveryMetadata:

Attribute	filePath	Full path from the exchange set root directory	1	CharacterString	Path relative to the root directory of the exchange set. The location of the file after the exchange set is unpacked into directory <EXCH_ROOT> will be <EXCH_ROOT>/<filePath>/<filename>

Part 4a Metadata S100_SupportFileDiscoveryMetadata:

Attribute	fileLocation	Full location from the exchange set root directory	1	CharacterString	Path relative to the root directory of the exchange set. The location of the file after the exchange set is unpacked into directory <EXCH_ROOT> will be <EXCH_ROOT>/<filePath>/<filename>

Part 4a Metadata S100_CatalogueMetadata:

Attribute	fileLocation	Full location from the exchange set root director	1..*	CharacterString	Path relative to the root directory of the exchange set. The location of the file after the exchange set is unpacked into directory <EXCH_ROOT> will be <EXCH_ROOT>/<filePath>/<filename>

The proposal would be to replace each instance of ENC_ROOT with S100_ROOT.

Discussion

The solutions proposed will ensure that if an exchange catalogue contains multiple S-1xx products, the information of which product types will be easily accessible for end user systems.

Implementing the use of one CATALOG.XML file for all S-1xx product types helps to avoid encoding metainformation in the catalog filename. The end user system then only needs to relate to one CATALOG.XML file within an exchange set.

Organizing the structure of all information in the top level directories ENC_ROOT, INFO and S100_ROOT ensures that both S-57 and S-1xx data can be delivered in one exchange set. Proposed structure would mean that S-100 references to EXCH_ROOT for filepath/FileLocation attributes must be changed.

Conclusions

A mechanism for including multiple product types in one exchange catalogue is proposed.
The name of the catalog is proposed to be CATALOG.XML, and this file should carry all necessary information for all S-1xx product types within the catalogue.
A directory structure identifying how dataset files, support files and catalogues should be organized is proposed.

Action Required of S-100TSM7

The S-100TSM7 is invited to:
Discuss the proposals within this paper and decide whether they could be forwarded to S100WG5 for further consideration.