

## Paper for Consideration by S100TSM9

## S100\_SupportFileDiscoveryMetadata attributes supportedResource and resourcePurpose

<b>Submitted by:</b>	PRIMAR, NIWC
<b>Executive Summary:</b>	Clarifications of the attributes supportedResource and resourcePurpose are needed.
<b>Related Documents:</b>	S-100 5.0.0, S-100WG7-04.2
<b>Related Projects:</b>	

## Introduction / Background

The attribute supportedResource and resourcePurpose in S100\_SupportFileDiscoveryMetadata (S-100 Part 17) was discussed at S100WG7.

Attribute	supportedResource	Identifier of the resource supported by this support file	0..*	CharacterString	Conventions for identifiers are still to be developed and will be defined later
Attribute	resourcePurpose	The purpose of the supporting resource	0..1	S100_ResourcePurpose	Identifies how the supporting resource is used

1. A common agreement of the intention of supportedResource and resourcePurpose is needed.
2. It was proposed to define conventions for the identifiers to be used, and S100\_DatasetDiscoveryMetadata attributes fileName or fileID was proposed. The discussion revealed that there is a need for more explanation of how this attribute is intended to be used, and also for the TSM9 meeting to decide upon identifier convention.
3. The optionality for supportedResource must be debated. For service providers to ensure consistent support file management it may be opportune that support files always identify the datasets to which they are associated.
4. Further discussions have revealed that there is also a need for revising the enumerates in attribute resourcePurpose type S100\_ResourcePurpose:

S100_ResourcePurpose				
Role Name	Name	Description	Code	Remarks
Enumeration	S100_ResourcePurpose	Defines the purpose of the supporting resource	-	-
Value	dataset	A dataset	1	
Value	featureCatalogue	A Feature Catalogue for an S-100 data product	2	
Value	portrayalCatalogue	A Portrayal Catalogue for an S-100 data product	3	
Value	interoperabilityCatalogue	An Interoperability Catalogue	4	
Value	supportFile	A support file	5	
Value	productVersion	All datasets conforming to a specific version of an S-100 Product Specification	6	
Value	productFamily	All datasets conforming to any active version of an S-100 Product Specification	7	
Value	software	Application software	8	
Value	system	Provides support or common information for a variety of applications and products	9	
Value	exchangeCatalogue	An Exchange Catalogue	10	
Value	ISOMetadata	Dataset metadata in ISO format	11	
Value	languagePack	A Language pack	12	
Value	GML.Schema	GML Application Schema	13	
Value	other	A type of resource not otherwise described	100	

The discussions at S100WG7 culminated in agreement to add a clarifying note to S100\_SupportFileDiscoveryMetadata in S-100 Part 17.

The following topics are further discussed below:

1. Intention of the attribute supportedResource and resourcePurpose..
2. Define conventions for the identifiers to be used for attribute supportedResource.
3. Optionality for the attribute supportedResource.
4. Revision of the enumerates in attribute resourcePurpose type S100\_ResourcePurpose.

## Analysis/Discussion

### 1. Intention of the attributes supportedResource and resourcePurpose

The description of supportedResource is: "identifier of the resource supported by this support file". This means the attribute is encoded to identify whichever resources (e.g., datasets) that use this support file.

The description of resourcePurpose is: "The purpose of the supporting resource". As a result of ongoing discussions, we can assume this means the attribute is encoded to describe the purpose of the support file.

**Conclusion: The intended use of supportedResource is to identify resources (e.g., datasets) that use this support file.**

**Conclusion: The intended use of resourcePurpose is to describe the purpose of the support file.**

### 2. Define conventions for the identifiers to be used for attribute supportedResource

Note: **Dave Grant** input in red - **Jonathan Pritchard** in blue - **PRIMAR** in green

Currently in the remarks field of supportedResource it is stated: "Conventions for identifiers are still to be developed and will be defined later". As a result of ongoing discussions, there seems to be several options to go for here:

- **A file URI per S-100 1-4.6**
  - The examples in Figure 17-5 and Table 17-1 are incorrect (file: is invalid as is file://) references every file with a matching name filenames are unique now so these mappings should be one-to-one and unambiguous. I believe there is a slight wrinkle in the "unique filename" idea though in that GML update datasets will carry the same name (because the file extension remains "GML"). This needs clarification – although the GML update protocol is far simpler than the 8211 one we should clarify that such references are to GML datasets in their updated form. This may modify how updates to support files can be achieved with GML datasets (and is one scenario we should model better).
  - typically a full path should be used to avoid ambiguity
    - ensures the appropriate resource is identified (dataset, catalogue, or support file)
    - otherwise file names would have to be unique across all products and all file types they are supposed to be unique. Because the product type and the data producer code is part of the name they should be unique (notwithstanding the GML update question above). Moreover we stated that content is also unique, so it's impossible to have different content simultaneously with the same filename (because filenames are unique). This will be a migration issue for (e.g.) US – I produced some metrics on this when we wrote Part 17 and I believe this was agreed. As the product and producer code are always part of the filename each data producer has a responsibility to maintain unique filenames. I don't believe there's a "don't reuse filenames after cancellation" rule but we should probably make one (there is such a rule for ENC under S-57 I believe).
    - supports references between different product types
    - file:///S-101/DATASET\_FILES/101AR00530495830.000
      - Note that this does not reference subsequent editions since the file name will change (.000 -> .001)
        - Use an MRN representing the datasetID to reference multiple editions of a dataset.
- **A digital signature MRN per S-100 Table 15-11** agree but needs clarifying which signature this can refer to. The first in the chain signed by the data producer makes sense. These can all be added to datasets under s-164.

- urn:mrn:iho:s100:dsig:dsa:LKJdsIkasj;dlkjk
- references a signature which is specific to the resource contents and signing certificate used
- **A cryptographic hash MRN per S-100 Table 15-12** agree. Support files are never encrypted so these work.
  - per S-100 Table 15-12
  - urn:mrn:iho:s100:hash:sha256:a948904f2f0f479b
  - references a specific instance of a resource
    - the instance could be shared by multiple editions (in the case of a reversion, but this is unlikely because a timestamp or version number would likely change even if the other contents revert)
- **An MRN representing a datasetID per S100\_DatasetDiscoveryMetadata.datasetID** neutral. Seems over complex to me. Stick to the Part 15 ones only would be my suggestion.
  - per S100\_DatasetDiscoveryMetadata.datasetID
  - this should be better defined, perhaps in S-100 3-10?
- **Some other IHO MRN format** as previous point.
- **Some other IHO MRN format**
  - TBD, as defined by the IHO MRN manager

Should only one, a couple or all of the discussed identifiers be supported? Regardless of the decision, it is important that testdata is created for all allowable identifiers.

**Conclusion: A decision of identifier(s) to be used for supportedResource must be made.**

### 3. Optionality for the attribute supportedResource

The attribute supportedResource is currently optional. The use case for the optional nature of it was the README file use case. Sometimes data producers want to distribute supporting information for the entire exchange set. In S-57 this was fulfilled by the README.TXT file (although no mandatory functionality exists on the ECDIS to allow this to be browsed by the user).

From a service provider viewpoint, we believe there should be a requirement that support files always identify the datasets to which they are associated - probably that would make it more manageable to ensure consistent support file management for service providers. For this to work the attribute supportedResource should be made mandatory. As per the discussion below this must be further debated.

Note: Dave Grant input in red - Jonathan Pritchard in blue - PRIMAR in green

Note that *supportedResource* is optional. We believe there should be a requirement that support files always identify the datasets to which they are associated - probably that would make it more manageable to ensure consistent support file management for service providers. As a consequence supportedResource should be mandatory.

In my opinion, it is not intended to be used when the support file is referenced from the supported resource. For instance, when an S-101 dataset references the support file (typically via a file MRN) *supportedResource* is not needed and should not be used. 17-4.3.2 should be removed or updated to indicate that the reference is optional, and will typically not be provided when there is a reference from the dataset to the supporting resource. Don't agree to remove 17-4.3.2, but should be updated instead.

Agree supportedResource is optional. The use case for the optional nature of it was (my recollection but I think I have the notes somewhere) the README file use case. Sometimes data producers want to distribute supporting information for the entire exchange set. In S-57 this was fulfilled by the README.TXT file (although no mandatory functionality exists on the ECDIS to allow this to be browsed by the user).

I don't think I agree that supportedResource is not needed when the S-101 dataset references the support file. This causes difficulties with encrypted datasets for distributors (and RENCs). Given an exchange set with 100

cells and 10 support files there's no explicit mapping at the exchange set level between the support files and the cells. Agree that because all filenames are now unique (ish, see later) this mapping can be gleaned from the datasets but if they're encrypted there's no way for a distributor or re-packager to split the exchange set up into coherent packages unless they decrypt the cells. Even for unencrypted cells trying to come up with such mappings, for the purposes of breaking up large exchange sets into usable chunks is non-trivial (particularly once you take updates into account). The way datasets are packaged/exported by the production systems (normally one or more cells at a time which are then aggregated by RENCs, distributors etc...) would be made fundamentally more error-prone (and impossible with locally encrypted data) if these mappings are not explicit in the catalogue.

- *supportedResource* should typically not be used since the support file will be referenced from the dataset(s). disagree for reasons given above.
- I do not agree with David, as this is exactly what we ask as a service provider - we need this functionality to update support files without a dataset added to the exchange set. How then otherwise, if this is not populated, the service provider would know to send this file to customers who have to have this support file because of what dataset files they are subscribing to?
- If you know which datasets the customer is subscribing to then you can scan those datasets and compile the list of support files which must be kept up to date for that customer. You don't need an association in the exchange set to provide this information. This assumes that the service provider is able to decrypt the datasets.
- can't make this assumption. RENCs and providers frequently break up exchange sets into regions and component parts and you can't assume data is always unencrypted
- If the association is provided in both directions (dataset->support file in the encoding, and support file->dataset in the exchange set) then we would have to account for the possibility that the associations could get out of sync with one another (when either the dataset or support file is updated independent of one another).
- this can be addressed with validation tests on the datasets in S-158 (as they are for S-58). The reason for having the association both ways is that the support file->dataset mapping (in the CATALOG.XML) is merely there to place the resource in a particular location - this enables sharing of support files (through dataset aggregation by providers) between multiple datasets (which was identified by several stakeholders as a requirement). The dataset identifies which resource it requires, the CATALOG.XML defines where each named resource is located (through the path) AND which datasets it refers to so that logical packages of regions etc can be made. I know the association exists in both directions but it's for good reason.
- Recommend that the association is described in only a single direction. This recommendation applies to all support files, not just dataset support files.
- this was always thoroughly broken in S-57 where the map of dataset->support file has to be done through physical location resulting in many 000s of extraneous folders for some providers and a lot of ambiguity. Some providers spent years coming up with folder naming schemes.

**Conclusion: A decision if the attribute supportedResource should be mandatory must be made.**

#### 4. Revision of the enumerates in attribute resourcePurpose type S100\_ResourcePurpose.

There is a need for revising the enumerates in attribute resourcePurpose type S100\_ResourcePurpose. Following is a listing of all the enumerates with discussion notes:

Dave Grant input in red - Jonathan Pritchard in blue - PRIMAR in green

- **1 – dataset** agree agree
  - Recommend this value is removed
  - Files associated with a dataset should use “5 – supportFile”
- **2 – featureCatalogue / 3 – portrayalCatalogue / 4 – interoperabilityCatalogue**
  - Recommend these values are removed agree agree
  - Files associated with a catalogue should use “12 – languagePack” or “100 – other”

- **5 – supportFile**
  - Recommend rename to *datasetSupportFile* *neutral. agree, this would provide clarity. Currently there is confusion if e.g. the ISOMetadataFile is a support file. With this change we can assume it is, and that datasetSupportFile is specifically what we traditionally know as supportfiles (TXT, TIFF and so on).*
  - *supportedResource* should typically not be used since the support file will be referenced from the dataset(s). *disagree for reasons given above.*
  - May be used to associate an otherwise unreferenced support file with a dataset. For example:
    - Provide notes associated with the dataset for exchange between HO's
    - Extended metadata
    - etc.
  - A file URI can only reference a specific dataset edition (since the filename changes with the edition number) not sure I agree. ENC names don't change with edition numbers? *No they dont.* Support files can be updated independently of dataset updates (agreed during Part 17 drafting I thought I understood).
  - To reference multiple editions of a dataset a *datasetID* should be used / referenced.
- **6 - productVersion**
  - *Is this needed?*
- **7 - productFamily**
  - *Is this needed?*
- **8 – software** *neutral. There has been discussion in the past of delivery of software patches to specific ECDIS by service providers. Ultimately I think the security risk can be mitigated by digital signatures which would be required. This would be a private agrgeement between a service provider and a specific OEM or customer. Agree to remove - to avoid added complexity. I believe OEMs can share software patches using other channels.*
  - Recommend this value is removed
  - Not sure why application software would be delivered via a support file in an exchange set
    - Seems like a security risk
  - What is the receiving system supposed to do with the software?
    - Windows/Mac/Linux?
    - OS version?
    - etc.
- **9 – system** *unknown. I would delete unless there's a use case. Agree to remove if no use case.*
  - Not sure what the use case is for this
    - Assuming there is a use case, the applications and products should have MRN formats registered which allow them to be referenced
- **10 – exchangeCatalogue** *agree. Remove. Agree to remove*
  - Recommend value is remove d.
  - Exchange catalogues are not delivered as a support file.
- **11 – ISOMetadata** *no ECDIS use case. Could have application outside ECDIS though, particularly to prevent extension of CATALOG.XML Schemas by product specs. Suggest to keep.*
  - Could be referenced from the dataset: See 10b-11.8.1, 10c-12
  - Reference to the associated dataset is the same as for 5 - *supportFile*
- **12 – languagePack** *agree to keep. Agree*
  - References one or more catalogues
    - Typically a feature catalogue
    - file:///S-101/CATALOGUES/FeatureCatalog.xml
    - urn:mrn:iho:s100:hash:sha256:a948904f2f0f479b
- **13 – GMLSchema** *keep. I don't believe it's necessary as there's (currently) no requirement for an ECDIS to validate datasets against a schema, only against a feature catalogue. This needs clarification though. I believe Part 10b is prescriptive enough for such validation to take place and this can be tested in S-164 Agree*
  - References are always to datasets encoded per Part 10b
  - I don't see that the schema is referenced directly from the dataset – should check with Raphael to confirm

- Reference to the associated dataset is the same as for 5 - *supportFile*
- 100 – **other** keep, always need a “doesn’t fit” category. *Agree*
  - Could reference nothing or anything.
  - Usage of the support file is defined somewhere other than S-100
    - In the applicable product specification
    - By the transmitting authority
    - By the receiving OEM
    - etc.

**Conclusion: The enumerates in attribute resourcePurpose type S100\_ResourcePurpose must be reviewed. Decisions to be made based on ongoing discussions.**

When the above mentioned issues are sorted out, clarifying information should be added in S-100 Part 17. This could for example be in the form of a note in the S100\_SupportFileDiscoveryMetadata table.

### **Conclusions**

- The intentions of the attributes supportedResource and resourcePurpose:
  - The intended use of supportedResource is to identify resources (e.g., datasets) that use this support file.
  - The intended use of resourcePurpose is to describe the purpose of the support file.
- A decision of identifiers(s) to be used for supportedResource must be made.
- A decision if the attribute supportedResource should be mandatory must be made.
- The enumerates in attribute resourcePurpose type S100\_ResourcePurpose must be reviewed.
- Add explanatory note to supportedResource and resourcePurpose in S100\_SupportFileDiscoveryMetadata table.

### **Action Required of S100TSM9**

The S100TSM is invited to:

Note the paper and discuss proposed changes.