@karin: Changes to the CS IP introduced in the SIP are marked RED. When it says “No changes”, just use the exact description and rules of the CS IP.

# METS root element (element mets)

| **Element** | **Definition** | **Explanation** | **Card.** | **METS** | **Changes to CS IP** |
| --- | --- | --- | --- | --- | --- |
| Identity /  Content ID | Identification of the package | A code that uniquely identifies the whole SIP and the digital object/objects being submitted. A UUID or GUID should be used to create globally unique identifiers.  Example:  *“UUID:550e8400-e29b-41d4-a716-446655440004"* | 1 | <mets:  OBJID=”[Identity]”> | No changes. |
| Description /  Package name | Description of the package | Use this description:  An optional short text describing the contents of the package, e.g. “*Accounting records of 2017"*  *Or the following (used on DIP)*  *Optional, if used should be filled with a human-readable description of the package,* e.g. “*Accounting records of 2017"* | 0..1 | <mets:  LABEL=”[Description]”> | New in SIP |
| Content type /  General content type | The content type being submitted with this package | The TYPE attribute must be used for identifying the type of the package (genre), for example ERMS, RDBMS, digitised construction plans.  However, there is no fixed vocabulary and as such implementers are welcome to use values most suitable for their needs.  Example:  ”*ERMS*” | 1 | <mets:  TYPE=”[Content type]” | No changes |
| Profile[[1]](#footnote-2) | Profile name | Identifies the METS profile being used. The name should contain the version number and the version may be captured in the profile file path. | 1 | <mets: PROFILE="[Profile]" | Must point to the SIP profile, no?  In the DIP they say “no change”.  I’m confused. |
| Content Information Type  Specification | Content type  specification used  for the content  type | An attribute added by this specification. It describes which content information type specification is used for the content. Values of the attribute are fixed in the following vocabulary:  1. SMURFERMS  2. SMURFSFSB  3. SIARD1  4. SIARD2  5. SIARDDK  6. GeoVectorGML  7. GeoRasterGeotiff  NB The vocabulary is extensible as additional content information type specifications are developed.  Example:  “SMURFERMS” | 0..1 | <mets:  CONTENTT  YPESPECIFICATION  =”[Content type  specification]” | Karin, please confirm that this element doesn’t need any changes in the SIP.  I know the attribute name changed, but the entire description is still the same, right? |

# METS Header (element metsHdr)

| **Element** | **Definition** | **Explanation** | **Card.** | **METS** | **Changes to the CS IP** |
| --- | --- | --- | --- | --- | --- |
| Id | Unique ID for the METS header section | Optional unique ID for the METS header section, e.g. *"uuid-ac0e8400-e29b-41d4-a716-446655440004"* | 0..1  MAY | <metsHdr:  ID=”[ID value]”> | New in the SIP  I would vote to remove it from the SIP. Does this bring any added value? |
| Date and time\*[[2]](#footnote-3) | Time of creation of package[[3]](#footnote-4) | Date and time for creation of the package must be described according to the XML-standard ("YYYY-MM-DDThh:mm:ssZ").[[4]](#footnote-5)  This timestamp states when the whole package and the package file was created.[[5]](#footnote-6)  Example:  *"2012-04-26T12:45:00+01:00"* | 1..1  MUST | <metsHdr:  CREATEDATE=”[Date and time]”> | No changes |
| Status\* | Package status | A way of indicating the status of the package to instruct the OAIS on how to properly handle the.  See also:  Package status vocabulary  If not set, the expected behaviour is equal to NEW. | 0..1  MAY | <metsHdr:  RECORDSTATUS=”[Status]”> | Not found on the CS IP. It’s a new attribute. We should have a controlled vocabulary with the following options:  NEW - a new delivery  SUPPLEMENT - extends the previous delivery  REPLEACEMENT - replaces a previous delivery  TEST - a test delivery. No AIP should be created.  VERSION - a delivery with same content regarding files but one or more files have a new version  DELETE – an order from the Producer to remove an existing AIP  OTHER - status not in list |
| OAIS type of  package | SIP, AIP, DIP, AIU, AIC | @csip:OAISPACKAGETYPE is an attribute added by the CSIP for describing the type of the IP. In the case of an SIP, this value is set to “SIP” | 1..1 | <metsHdr:  PACKAGETYPE=”SIP | Changes made to the value of the PACKAGETYPE.  Please confirm if this is right. |
| DocID\* | METS document ID | A unique identifier for the METS document itself. This identifier may be different from the Identity given in the mets-element. The recommendation is to use the file name given to the METS-document.  Example:  “*SIP20150127.xml*” | 0..1 | <metsDocumentID>[DocID] | Karin, do we need this? I inclined to remove this line. |
| Submission agreement[[6]](#footnote-7) | Reference to the used submission agreement | A reference to the Submission Agreement associated with the package.  Example:  *RA 13-2011/5329; 2012-04-12*  Example:  [*http://submissionagreement.kb.se/dnr331-1144-2011/20120711/*](http://submissionagreement.kb.se/dnr331-1144-2011/20120711/)  **Note:** *It is recommended to use a machine-readable format for better description of a submission agreement.[[7]](#footnote-8)*  Example:  <external\_schema>  <name>Submission Agreement (SA)  </name>  <URL>http://www.dasboard/xmlns/SubmissionAgreement.xsd</URL>  <context> The submission agreement specifies the relations between the producer and the archive as it is described in ISO 20652:2006 (Producer-archive interface -- Methodology abstract standard). </context>  </external\_schema> | 0..1  MAY | <altrecordID:  TYPE=”SUBMISSIONAGREEMENT”>[Submission agreement]  **OR PREFERABLY**  <mets:  <external\_schema>  <name>Submission Agreement (SA)  </name>  <URL>http://www.dasboard/xmlns/SubmissionAgreement.xsd</URL>  <context> The submision agreement specifies the relations between the producer and the archive as it is described in ISO 20652:2006 (Producer-archive interface -- Methodology abstract standard). </context>  </external\_schema> | Element introduced in the SIP.  I’m not sure if this is the best attribute to store this info.  Let’s wait for feedback. |
| Previous submission agreement | The previous submission agreement(s) the information belongs to in the case the information is recorded. | An optional reference to a previous submission agreement(s) which the information may have belonged to.  Example:  *”FM 12-2387/12726, 2007-09-19”*  **Note:** *It is recommended to use a machine-readable format for better description of a submission agreement.[[8]](#footnote-9)* | 0..\*  MAY | <altrecordID:  TYPE=”PREVIOUSSUBMISSIONAGREEMENT”>[Previous submission agreement]  **OR PREFERABLY**  <mets:  <external\_schema>  <name>Previous Submission Agreement (PSA)  </name>  <URL>http://www.dasboard/xmlns/SubmissionAgreement.xsd</URL>  <context> The submision agreement specifies the relations between the producer and the archive as it is described in ISO 20652:2006 (Producer-archive interface -- Methodology abstract standard). </context>  </external\_schema> | Element introduced in the SIP. |
| Archival reference code | Reference code in the archival description | An optional reference code indicating where in the archival hierarchy the package shall be placed in the OAIS.  Example:  *”SE/RA/123456/24/P”* | 0..1  MAY | <altrecordID:  TYPE=”REFERENCECODE”>[Archival reference code] | Element introduced in the SIP. |
| Previous reference code | An earlier used reference code in the archival description | In case where the SIP originates from other institutions maintaining a reference code structure, this element can be used to record these reference codes and therefore support the provenance of the package when a whole archival description is not submitted with the submission.  Example:  *”SE/FM/123/123.1/123.1.3”* | 0..\* | <altrecordID:  TYPE=”PREVIOUSREFERENCECODE”>[Previous reference code] | Element introduced in the SIP. |
| Archival creator | Name of archival creator | The name of the organisation that originally created the data being transferred.  Please note that this might be different from the organisation which has been charged with preparing and sending the SIP to the archives.  Example:  *”The Swedish health agency”* | 1..1  MUST | <agent:  ROLE=” ARCHIVIST”  TYPE=  ”ORGANIZATION”>  <name>  [Archival creator organisation] | Element introduced in the SIP.  I propose setting this as optional (0..1), as in most cases this information will come in the descriptive metadata (e.g. EAD or DC)  Please remember that there are cases where the data being transferred is produced by an individual. We have several cases of that, e.g. presidents, famous writers, etc. |
| Archival creator identification code | A unique identification code for the archival creator | The unique identifier of the organisation that originally created the data being transferred.  The code uses a prefix followed by a ”:”  Prefix[[9]](#footnote-10) according to vcTypeOfIdentificationCode.  Example:  *”VAT:SE201345098701”* | 0..1 | <agent:  ROLE=” ARCHIVIST”  TYPE=  ”ORGANIZATION”>  <note>  [Archival creator identifier code] | Element introduced in the SIP.  Again, the TYPE could be an INDIVIDUAL. |
| Submitting organisation name[[10]](#footnote-11) | Name of the organisation submitting the package to the archive | The name of the agent responsible for creating and delivering the SIP.  For additional agents add more agent elements according to the METS schema.  Example:  *”The agency, Personnel* | 1..1  MUST | <agent:  ROLE=”CREATOR”  TYPE=”ORGANIZATION”>  <name>  [Submitting organisation name] | Element introduced in the SIP.  Again, the TYPE could be an INDIVIDUAL so I wonder if we should add a restriction such as a CREATOR must exist, but it can be an ORGANIZATION or an INDIVIDUAL. |
| Delivering organisation identification code | A unique identification code for the delivering organisation | The unique identifier of the agent responsible for creating and delivering the SIP.  The code uses a prefix followed by a ”:”  Prefix according to vcTypeOfIdentificationCode.  Example:  *”* *VAT:SE2098109810-AF87”* | 0..1 | <agent:  ROLE=”CREATOR”  TYPE=  ”ORGANIZATION”>  <note>  [Delivering organisation identification code] | Element introduced in the SIP.  Again, the TYPE could be an INDIVIDUAL. |
| Contact person name | Contact person for the submission | The name of the person responsible for the creation and delivery of the SIP.  This information is only valid during a short time period.  Example:  *”Sven Svensson”* | 0..\* | <agent  ROLE=”CREATOR”  TYPE=  ”INDIVIDUAL”>  <name>  [Contact person name] | Element introduced in the SIP. |
| Contact person contact information | Contact information for the contact person | The contact information of the person responsible for the creation and delivery of the SIP.  This information is only valid during a short time period.  Example:[[11]](#footnote-12)  *”08-12 34 56”*  *“sven.svensson@fm.se”* | 0..\* | <agent  ROLE=”CREATOR”  TYPE=  ”INDIVIDUAL”>  <note>  [contact information of the person responsible for the creation and delivery of the SIP] | Element introduced in the SIP. |
| Software\* | The software which has been used to create the package | The metsHdr must include at least one agent describing the software which has been used to create the package. | 1 | <agent ROLE=”CREATER” TYPE=”OTHER” OTHERTYPE=”SOFTWARE”> | No changes. Inherited from the CS IP. Do not include in the SIP profile. |
| Preservation organisation name | Name of organisation preserving the package | The name of the organisation responsible for the preservation of the data delivered in the SIP.  Example:  *”National Archives of Hungary”* | 1..1  MUST  NOTE:  I suggest changing it to MAY | <agent  ROLE=  ”PRESERVATION”  TYPE=  ”ORGANIZATION”>  <name>  [Preservation organisation name] | Element introduced in the SIP.  I would make this optional (0..1). In most cases, the organisation responsible for preserving the package is the one we are submitting the package to. So this is redundant in most cases |
| Preservation organisation identification code | Identification code of organisation preserving the package | The unique identifier of the organisation responsible for the preservation of the data delivered in the SIP  A unique identification code for the organisation responsible for preservation. The code uses a prefix followed by ”:”  Prefix according to vcTypeOfIdentificationCode.  Example:  *”ORG:2010340987”* | 0..1 | <agent  ROLE=  ”PRESERVATION”  TYPE=  ”ORGANIZATION”>  <note>[Preservation organisation identification code] | Element introduced in the SIP. |

# METS descriptive metadata (dmdSec)

No special changes introduced in the SIP

# Administrative metadata (amdSec)

No special changes introduced in the SIP

# Files (fileSec)

| **Element** | **Definition** | **Explanation** | **Card.** | **METS** | **Changes to the CS IP** |
| --- | --- | --- | --- | --- | --- |
| Identification of the file | Identification of the file object | A code that uniquely identifies the file inside the METS-document for referencing in the structMap element. Suggested use is a prefix “ID” directly followed by an UUID or GUID or own local identification code. ID follows the rules of the xml attribute XML:ID.  Example:  *"ID550e8400-e29b-41d4-a716-4466554400bg"* | 1 | <file  ID=”[Identification of the file]”> | No change = CSIP59 or CSIP65 |
| File location | Name of the file | Name of the file and the path to locate it in the package. The file name must use the prefix file://. The attribute LOCTYPE is mandatory to describe how to find the file and uses a value list present in METS.  Example:  *"file://personnelexport.xml"* | 1 | <file  <flocat:  LOCTYPE=”URL”  xlink:href=”file://[File name]”  xlink:type=”simple”> | No changes = CSIP77 |
| Date and time | Timestamp for the file[[12]](#footnote-13) | The timestamp seen on the file and used for validating the file. In most cases this is the last modification date. Described using xml type datetime rules.  Example:  *"2012-04-20T13:30:00+01:00"* | 1 | <file  CREATED=”[Date and time]”> | No changes = CSIP70 |
| MIME type[[13]](#footnote-14) | Simplest way of describing a file type | The simplest way of describing a file type.  Example:  *"text/xml"* | 1 | <file  MIMETYPE=”[MIME type]”> | No changes = CSIP68 |
| File format name[[14]](#footnote-15) | Name of file format | A more detailed file format definition when the use of PREMIS has not been agreed upon in the submission agreement.  Example:  *"Extensible Markup Language”*  Example:  *”PDF/A”*  Example:  *”ISO/IEC 26300:2006”* | 0..1 | <file  ext:FILEFORMATNAME=”[File format name]” | This is a new element not found on the CS IP.  I wonder if we should keep it here or add it to the CS IP. |
| File format version | Version of file format | The version of the file format when the use of PREMIS has not been agreed upon in the submission agreement.  Example:  *"1.0"* | 0..1 | <file  ext:FILEFORMATVERSION=”[File format version]” | This is a new element not found on the CS IP.  I wonder if we should keep it here or add it to the CS IP. |
| Format registry name | Name of the format registry identifying the file format | The name of the format registry used to identify the file format when the use of PREMIS has not been agreed upon in the submission agreement.  According to vocabulary[[15]](#footnote-16) vcTypeOfFormatregistry  Example:  *" PRONOM”"* | 0..1 | <file  ext:FORMATREGISTRY=”[Format registry name]” | This is a new element not found on the CS IP. |
| Format key | Key of the file format in the registry | Key of the file format in the registry when use of PREMIS has not been agreed upon in the submission agreement.  Example:  *“fmt/101"* | 0..1  1  If Format registry name is used | <file  ext:FORMATREGISTRYKEY=”[Format key]” | This is a new element not found on the CS IP. |
| File size | Size of the file in bytes | Size of the file in bytes.  Example:  *”8765324”* | 1 | <file  SIZE=”[File size]”> | No changes = CSIP69 |
| Function | Identifies the function of the file | the intended use of the file (e.g., "master", "reference", "thumbnail" for image files).  A vocabulary for stating the “intended use” is recommended in and the vocabulary should be set in the METS profile.  Example:  *”thumbnail”* | 0..1  MAY | <file:  USE=”[Function]”> | This is a new element not found on the CS IP.  I wonder if this is even feasible to implement on most agencies….  I would remove it completely or add it to the CS IP |
| Checksum type | Used algorithm for creating the checksum | Algorithm used for creating the checksum. Values are predefined in METS. The algorithm to use is to be stated in the submission agreement.  Example:  *”SHA-256”* | 1 | <file:  CHECKSUMTYPE=”[Checksum type]”> | No changes = CSIP72 |
| Checksum value | Calculated checksum for the file | Check sum for the file.  Example:  *”574b69cf71ceb5534c8a2547f5547d”* | 1 | <file:  CHECKSUM=”[Checksum]”> | No changes = CSIP71 |
| Transformation algorithm | Transformation algorithm used for transformation | The type of transformation needed to render content of a file accessible (including unpacking a file into subsidiary files/streams). This can be used either for compressed or encrypted files  When used, the transformation algorithm must be to defined, i.e. a string describing the specific decompression or decryption routine used to access the contents of the file.  Example:  TRANSFORMTYPE=”decryption”  TRANSFORMALGORITHM=”DES” | 0..1 | <file  <transformFile  TRANSFORMTYPE=” decompression  | decryption”  TRANSFORMALGORITHM=  ”[Transformation algorithm]”> | This is a new element not found on the CS IP. |
| Transformation key | Transformation key for a transformed file | The key to be used with the transform algorithm for accessing the file's contents.  Example:  *”574b69cf71ceb5534c8a2547f5547d”* | 0..1  1  If Transformation algorithm is used | <file  <transformFile  TRANSFORMKEY=  ”[Transformation key]”> | This is a new element not found on the CS IP. |

# Structure structMap

No special changes introduced in the SIP

1. This element is not used for representation METS XML files. [↑](#footnote-ref-2)
2. Header elements marked with an asterisk \* represent elements which are required only for the root METS.xml. All other elements can be recorded in the representation METS.xml files. [↑](#footnote-ref-3)
3. The Commona Specification for Information Packages also defines the last modification date, but this is not relevant for SIPs. [↑](#footnote-ref-4)
4. XML Schema Part 2: Datatypes Second Edition, https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/#isoformats [↑](#footnote-ref-5)
5. The package file is the METS-document which describes the content of the whole package. [↑](#footnote-ref-6)
6. Reference Model for an Open Archival Information System (OAIS), 2012,

   public.ccsds.org/publications/archive/650x0m2.pdf [↑](#footnote-ref-7)
7. For example, the submission agreement developed by Docuteam GmbH http://www.loc.gov/standards/mets/profiles/00000041.xml [↑](#footnote-ref-8)
8. For example, the submission agreement developed by Docuteam GmbH http://www.loc.gov/standards/mets/profiles/00000041.xml [↑](#footnote-ref-9)
9. All prefixes referred in this table are described in the SIP METS Profile [↑](#footnote-ref-10)
10. All similar <agent> elements (the archival creator, delivering organisation, submitting organisation and producing organisation) may not be needed in some implementations of the SIP METS Profile. [↑](#footnote-ref-11)
11. As the SIP profile allows for 0..\* fields then it is possible to have e-mail, phone, physical address details etc in separate fields. The choice is up to specific implementations. [↑](#footnote-ref-12)
12. It is the timestamp recorded inside the file (i.e. information we can read with JHove and similar tools). [↑](#footnote-ref-13)
13. Media Types, 2015, <https://www.iana.org/assignments/media-types/media-types.xhtml> [↑](#footnote-ref-14)
14. This and all following file elements can be recorded in the SIP by using extension schemas (as shown here), but also by using PREMIS or some other metadata standard. [↑](#footnote-ref-15)
15. All vocabularies referred in this table are described in the SIP METS Profile [↑](#footnote-ref-16)