



Danish Broadcasting Corporation

DR Metadata Standard

version 2.0

release date 01.03.2005

Abstract

This document is the main document for the DR Metadata standard. The document provides specifications and guidelines for the logical data model, the data dictionary, taxonomies and thesaurus, search and exchange standards.

1.Introduction

The purpose with the DR Metadata standard is to facilitate and enhance the IT-based content production and sharing of all digital media content, both throughout the organization and in relation to end users. The way to achieve this is that all DR production systems work in concert and exchange information transparently. This requires that DR defines and implements a common data model and standards for system-to-system (S2S), business-to-consumer (B2C) and business-to-business (B2B) exchange.

When implemented, results include improved workflow and increased availability of content. In addition, all of DR benefits from time and cost savings when procuring and implementing new systems.

The DR Metadata standard is defined as an internal DR Metadata standard for DRAMS (DR Asset Management Systems), relations to international standards for S2S, B2B and B2C exchange and a set of requirements and guidelines.

The internal DRAMS specifications are developed for managing material in both the production and archiving domain. The DRAMS specifications describe multimedia content such as productions, items (pieces of material), programs and articles with focus on core descriptive metadata.

The DR Metadata standard is based on a logical data model. The intention isn't necessarily to implement one physical DRAMS database in DR. DRAMS could be implemented as many independent databases based on the same concept but serving different purposes. An important property is to make it possible to search data across different instances of DRAMS.

The international S2S, B2C and B2B exchange standards are intended to support business processes inside and outside DR, respectively. These standards consist of normative definitions and recommended use.

The requirements and guidelines cover taxonomies, thesaurus, search, essence management, terminologies, conditions and best practice.

1.1.Standard development procedure

The development of this standard was conducted as a part-time project running from December 2001 to July 2002. The work has been based on actual DR needs enhanced with influences from international standards and related projects.

DR Technology Management, DR Archive and externals have delivered project management, data modelling and a significant part of the document preparation work. A project organization consisting of multiple DR departments and persons have also contributed to this project through requirements, documentation and review efforts.

1.2.Contacts

The DR Metadata Standard is maintained by DR Technology Management

Version 1.0 released 20-07-2002.

Version 1.1 released 01-09-2002.

Version 1.2 released 01-12-2002

Version 2.0 released 01-04-2005

You find the DR Metadata Standard on

Intranet:

http://inline01/inline/Almen/Service/teknologisk_stab/metadata/default.htm

Internet:

www.dr.dk/metadata

Send a mail to Ole Hybæk (DR Technology Management), oh@dr.dk for any question or comments to the DR Metadata Standard. If you have specific questions related to the semantic send a mail to Birgitte Stannius (DR Archive) bsu@dr.dk

2.Intended Audience

The intended audience for this document is DR staff and vendors who wish to participate in DR system design and implementation. This document is technically oriented and general knowledge in IT data management in combination with librarian skills is required to fully understand the material in the following sections.

3.Scope

The DR Metadata standard is intended to serve as a framework for DR asset management with a strong focus on core descriptive metadata. The standard is intentionally restricted and does not cover all system- and implementation specific descriptions. The motivation for these restrictions is to enable cost-effective implementations and allow the use of standard products to the highest possible degree.

3.1.In scope

The following topics are addressed by the DR metadata standard.

- DR data model (entity-relationship diagram)
- DR data dictionary including essential business rules and conditions
- Taxonomies and thesaurus
- Guidelines for search
- Guidelines for essence management
- S2S,B2B and B2C standards
- Use of the DR integration platform

3.2.Not in scope

The following topics are **not** addressed by the DR Metadata standard.

3.2.1. Access restrictions and user management

- Note that most implementation will require fine-grained access control list mechanisms.
- This information is not considered to be descriptive metadata and is therefore not in the scope of this standard.
- Guidelines for security should be based on the DR IT security standard and correlated with specific system and user requirements.
- We expect to be able to get logging information like user-id for all authorized operations such as create and last update for entities like Item, MediaObjectGroup, MediaObjectInstance, Publication, Program and RunningOrder.

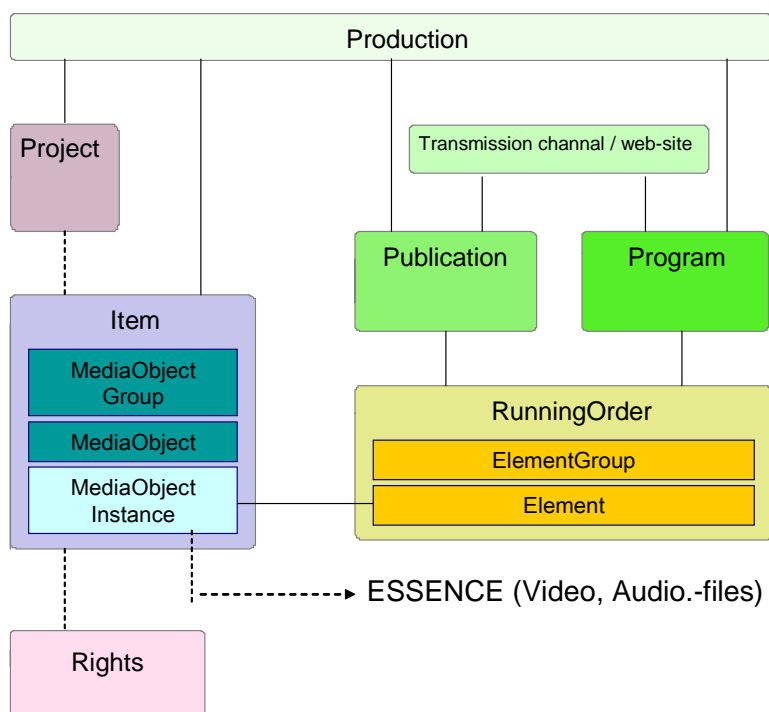
3.2.2. System-specific information

- Technical and control metadata.
- Content-specific details outside the scope of the common metadata model.

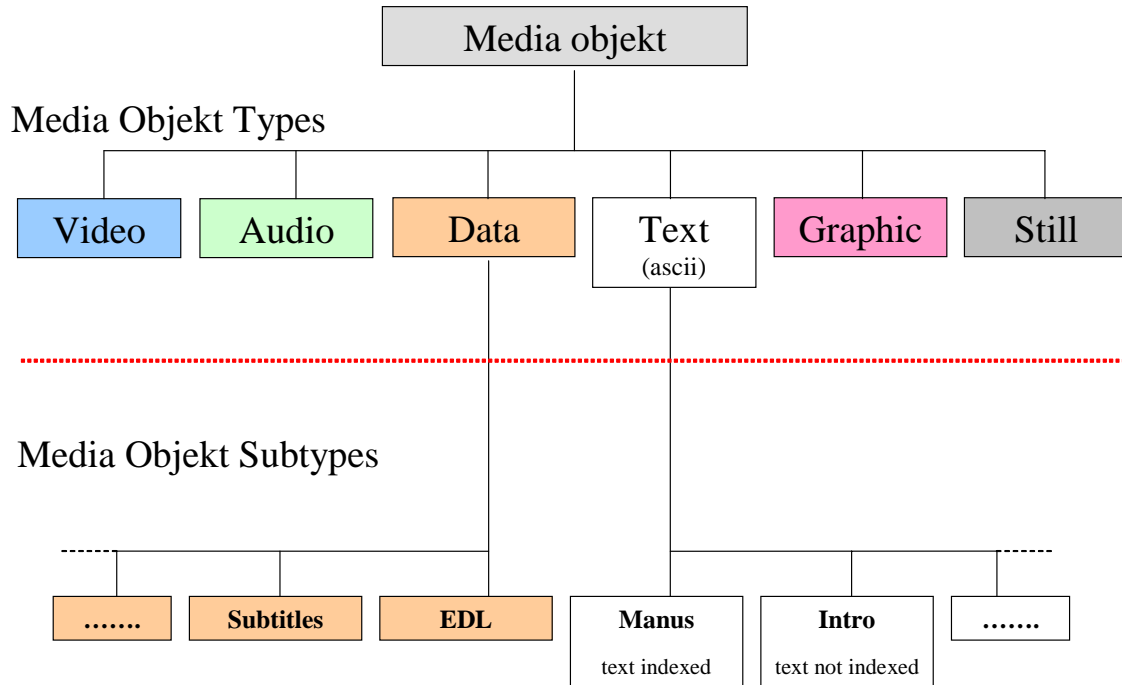
This information is implementation-specific and should not be exchanged outside the systems responsible for this data. Examples include system-specific ids, web and print publishing details such font sizes, style sheets and paper quality, etc.

4. The information model

4.1. The main compositions

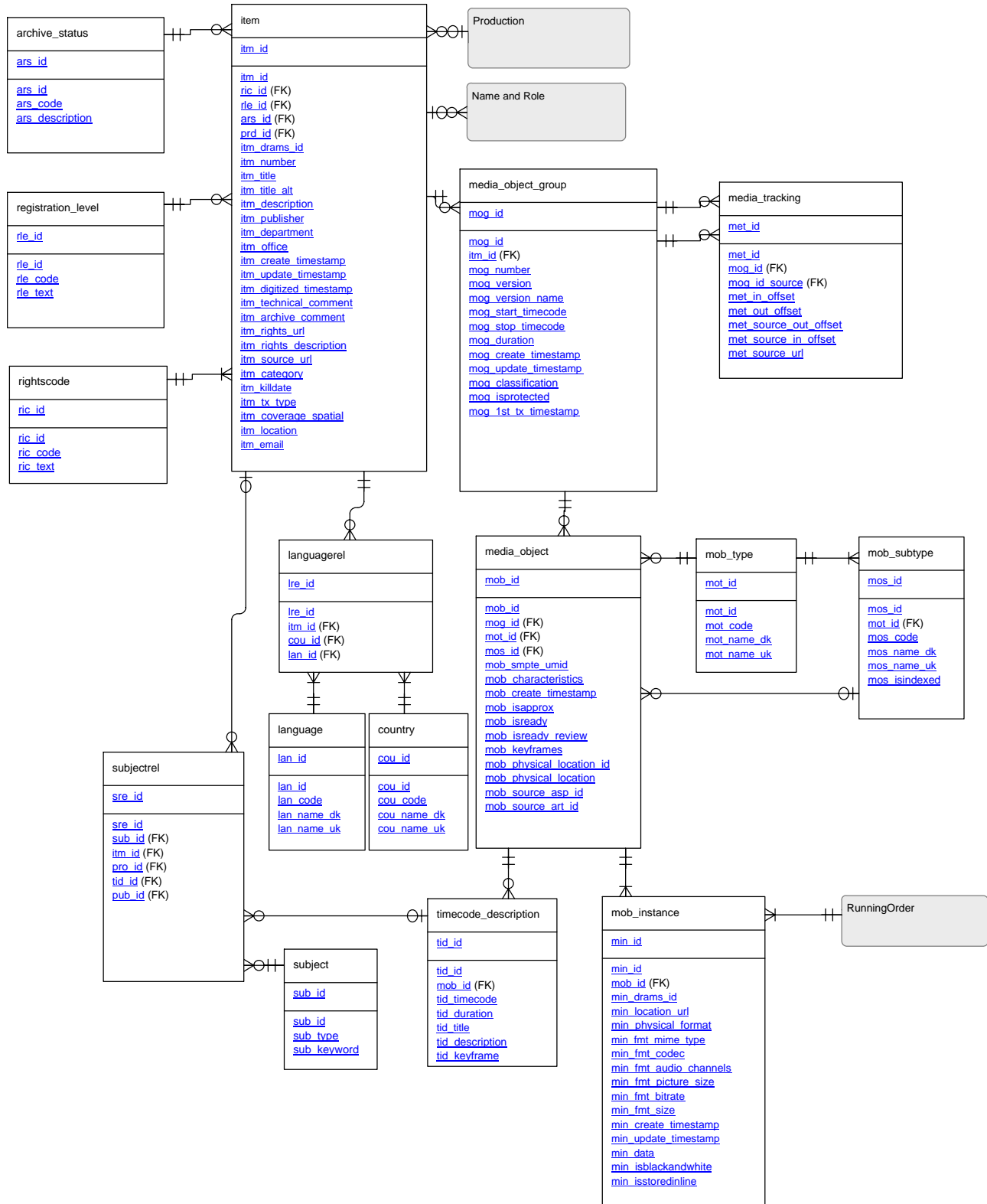


4.2. The Media Object types and subtypes

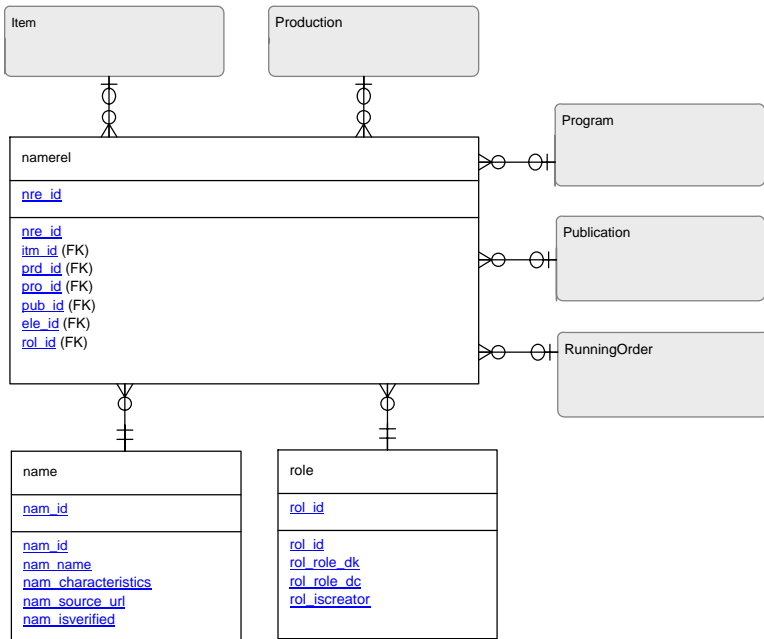


5. Logical models

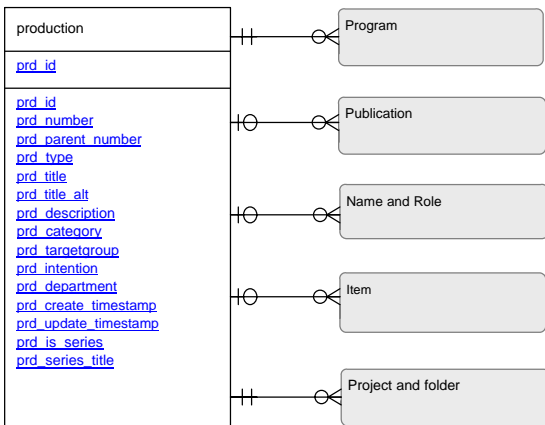
5.1. Item



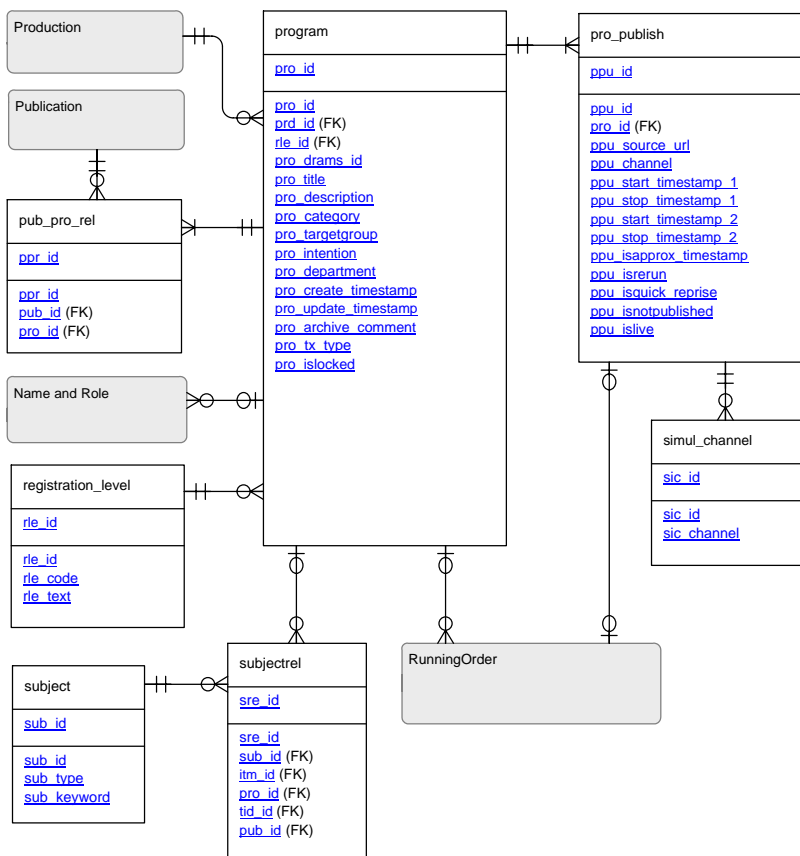
5.2.Name and Role



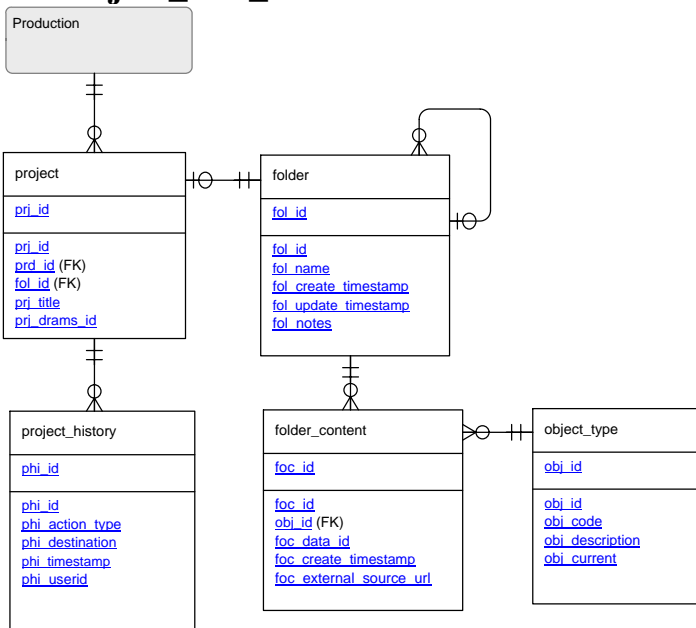
5.3.Production



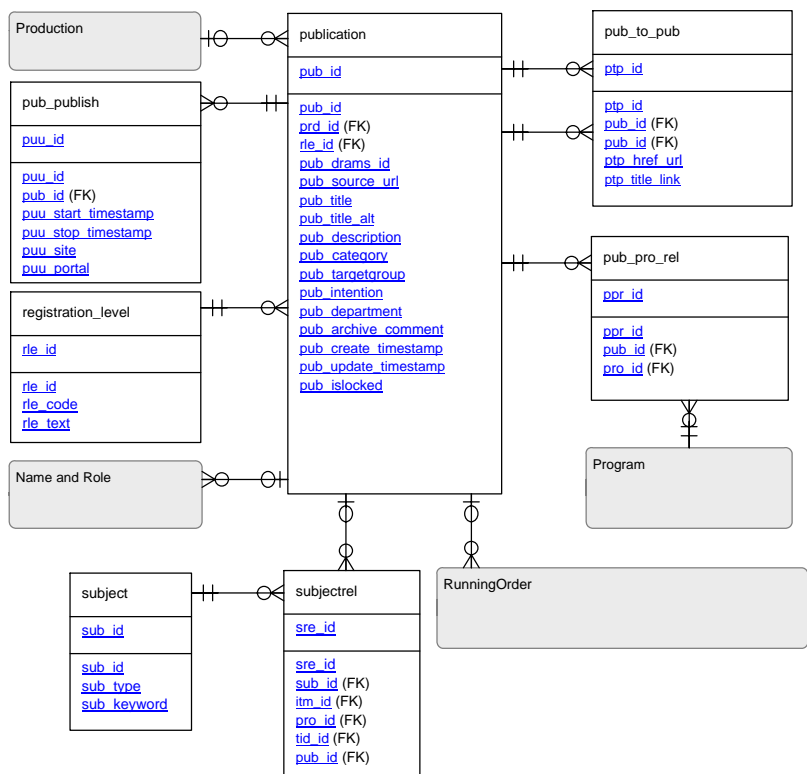
5.4.Program



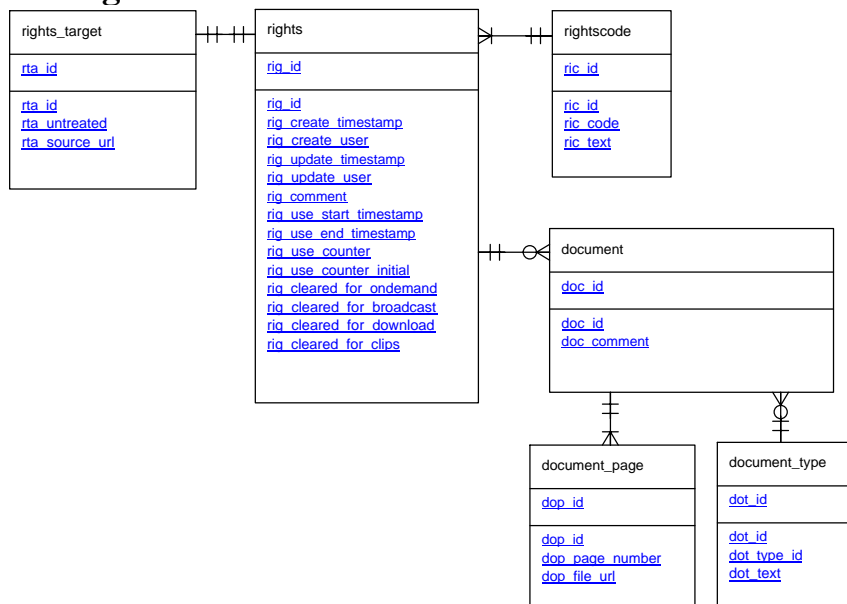
5.5.Project and folder



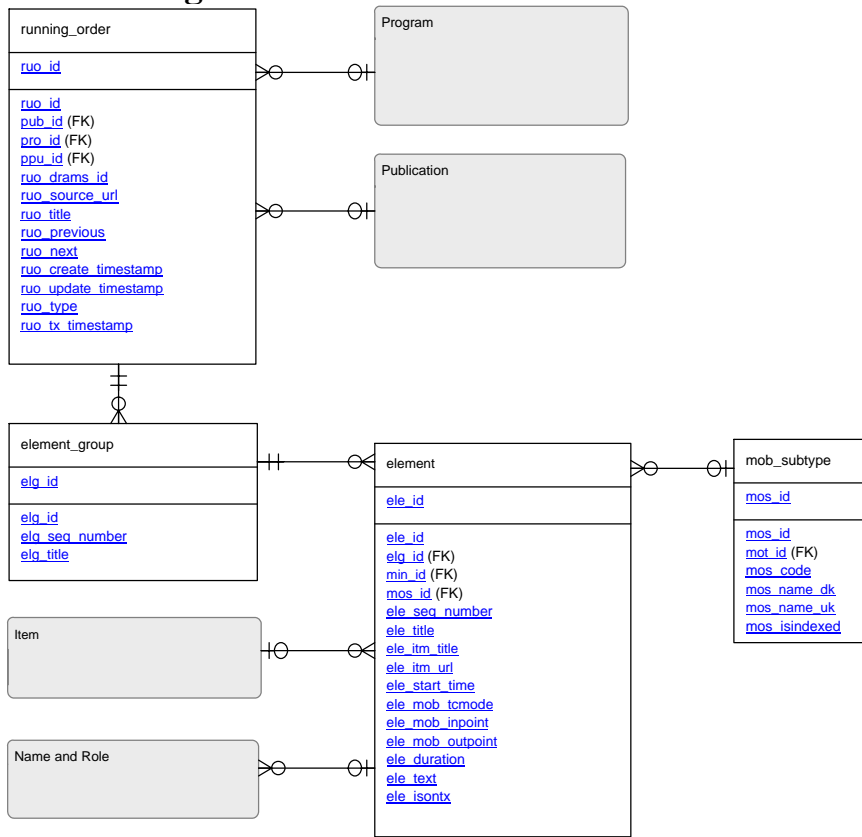
5.6.Publication



5.7.Rights



5.8. RunningOrder



6.Data dictionary

6.1.archive_status

A list of valid values for archive_status, which describes the status of the content file and it's meta data.

Attribute name	Comment	Data type	Mand.
ars_id	ID for the archive_status.	id_number	Yes
ars_code	Contains value for archive_status. Examples: Archived, Archive candidate, Locked for registration...	char100	Yes
ars_description	Extended description of the archive_status.	description	No

Comments:

The system provider maintains the table.

Relations:

Refers to 0:M item

6.2.country

A list of valid codes and names for countries.

Attribute name	Comment	Data type	Mand.
cou_id	ID for Country.	id_number	Yes
cou_code	Contains the International country code. Examples: dk, gb...	char100	Yes
cou_name_dk	Danish name for country.	char100	Yes
cou_name_uk	English name for country.	char100	Yes

Comments:

The valid countries are specified in standard ISO 3166-1.

Country is used in connection with Language through Language rel.

The system administrator maintains the table.

Relations:

Refers to 1:M languagerel

6.3.document

Document is the wrapper, which relates the specific files with the document_type.

Attribute name	Comment	Data type	Mand.
doc_id	ID for Document.	id_number	Yes
doc_comment	Comment to the document. Example: The Document is in bad condition...	description	No

Relations:

Refers to 1:M document_page

Refers to 1:1 document_type

Refers to 1:1 rights

6.4.document_page

Document_page points to or the specific files containing the rights documents.

Attribute name	Comment	Data type	Mand.
dop_id	ID for document_page.	id_number	Yes
dop_page_number	Pagenumber of the document.	number	No
dop_file_url	The source url for the rights document.	URL	Yes

Comments:

Document_page handles both electronic documents and paper documents.

Dop_page_number is used to manage the paper documents, which are scanned into the system in individual pages.

Relations:

Refers to 1:1 document

6.5.document_type

A valid list of values for document_type, which describes the type of document used in connection with rights.

Attribute name	Comment	Data type	Mand.
dot_id	ID for referenced Document.	id_number	Yes
dot_type_id	ID for the specific documenttype.	char100	Yes
dot_text	Contains the vlaue of document_type. Exaples: Reports to composers' society, Fees to be paid according to agreement codes, Contracts (contracts with external companies) ...	char100	Yes

Comments:

The system administrator maintains the table.

Relations:

Refers to 0:M document

6.6.element

An element is the smallest part of a Running Order. The instance description contains basic properties such as title, item URL and text.

Attribute name	Comment	Data type	Mand.
ele_id	ID for Element.	id_number	Yes
elg_id	ID for element_group.	id_number	Yes
min_id	ID for mob_instance.	id_number	No
mos_id	ID for mob_subtype.	id_number	No
ele_seq_number	Element sequence number.	number	No
ele_title	Displayed title for element in running order. Default value ele_itm_title if exist. Example: Arafat This field is compliant to Dublin Core.	title	Yes
ele_itm_title	Copy of referenced itm_title. Example: Arafat and Bush This field is compliant to Dublin Core.	title	Yes
ele_itm_url	Reference to an external item. Example: Dalet+\Music\2354678-0102	URL	No

ele_start_time	The start time is relative to start time for the actual Element in the element_group. Example: 00:01:05:000	time	No
ele_mob_tcmode	Mode of time code. Two kinds are used W (wall clock) or Z (zero based).	number	Yes
ele_mob_inpoint	Timecode for inpoint on the referenced video/audio file. Example: 00:00:17:000	timecode	No
ele_mob_outpoint	Timecode for outpoint on the referenced video/audio file. Example: 00:00:39:000	timecode	No
ele_duration	Estimated or exact duration in seconds. Example: 00:00:22:000	number	No
ele_text	ASCII Text. Could be the content of description an Intro. This field is compliant to Dublin Core.	description	No
ele_isontx	Flag is set if the element is available on the transmission server.	boolean	No

Comments:

An Element is typically imported from a NCS (News Computer System) or from a Computer Aided Radio or TV scheduling system but an Element may also be created and maintained in DRAMS. An Element may point to a DRAMS Item through a Mob Instance or it may point to external Item in another proprietary or DRAMS structured database. An Element could also be a text describing an Intro to next element. In this case the Element must not point to an internal or external item.

Ele_text's are texts associated to the OnAir situation. Text valuable for reuse in future productions should be copied to production data (Item) as part of transmission post processing. Available text content types are maintained in the mob_subtype entity and only for mob_type = 'Text'.

Relations:

Refers to 0:M namerel

Refers to 1:1 element_group

Refers to 0:1 mob_subtype

Refers to 0:1 mob_instance

6.7.element_group

A section of a running order that constitute a composed entity. Example: a news story.

Attribute name	Comment	Data type	Mand.
elg_id	ID for element_group.	id_number	Yes
elg_seq_number	Describes the order of appearance for element_groups as well as individual Elements.	number	Yes
elg_title	A common title for element_group. This field is compliant with Dublin Core.	title	Yes

Comments:

An Element Group is typically imported from a NCS (News Computer System) or from a CAR (Computer Aided Radio) or TV system but a ElementGroup may also be created and maintained in DRAMS. It may refer to a story containing several Elements like an intro, a video clip and an outro. The elg_seq_num attribute is generated by the RunningOrder Parser. The elg_seq_num describes the order of appearance for Element groups. If an element_group is moved the elg_seq_num shall be changed according to the altered running order sequence.

Example:

1 Report from Afghanistan (elg_seq_num=1)

2 Report from Oslo (elg_seq_num=2)

3 Report from Copenhagen (elg_seq_num=3)

After change of order of stories the elg_seq_num are updated.

1 Report from Afghanistan (elg_seq_num=1)

2 Report from Copenhagen (elg_seq_num=2)

3 Report from Oslo (elg_seq_num=3)

Relations:

Refers to 0:M element

Refers to 1:1 running_order

6.8.folder

A folder is a wrapper for the Folder content.

Attribute name	Comment	Data type	Mand.
fol_id	ID for Folder.	id_number	Yes
fol_name	Folder name.	title	Yes
fol_create_timestamp	Folder create timestamp. Example: 22-06-2004 10:48:20	date_time	Yes
fol_update_timestamp	Folder update timestamp. Example: 27-06-2004 12:00:20	date_time	No
fol_notes	Folder additional comment.		No

Comments:

Depending on the type of implementation a folder structure can be established.

Relations:

Refers to 0:M folder

Refers to 0:M folder_content

Refers to 0:1 folder

Refers to 0:1 project

6.9.folder_content

Folder_content references a specific document in the folder.

Attribute name	Comment	Data type	Mand.
foc_id	ID for Folder Content.	id_number	Yes
obj_id	ID for the object _type.	id_number	Yes
foc_data_id	ID for the referenced document. Example: DR\Mediearkivet\ITM\266187 Syntax: Organisation \ Instance of archive \ object type \ serial number DRAMS ID	char100	Yes
foc_create_timestamp	Create timestamp for the folder_content. Example: 24-12-2005 09:18:20	date_time	Yes
foc_external_source_url	This entity is used to reconstruct a project in an external environment. Example: File location URL (UNC) for external editing facility.	char100	No

Relations:

Refers to 1:1 folder

Refers to 1:1 object_type

6.10.item

An item is the smallest unit that carries descriptive metadata (including rights). Each item contains zero or more media object groups.

Attribute name	Comment	Data type	Mand.
itm_id	ID for Item.	id_number	Yes
ric_id	ID for Rightscode.	id_number	Yes
rle_id	ID for registration_level.	id_number	Yes
ars_id	ID for the archive_status.	id_number	Yes
prd_id	ID for Production.	id_number	No
itm_drams_id	Unique identifier for a DRAMS item Example: DR\Mediearkivet\ITM\266187 Syntax: Organisation \ Instance of archive \ object type \ serial number	DRAMS_id	Yes
itm_number	Itm_number is used for ingest to a production and is defined as the last used number +1. Example: 266187	number	Yes
itm_title	Primary item title. Example: Prisoners in Iraq This field is compliant with Dublin Core.	title	Yes
itm_title_alt	Alternative item title. Example: Dogs in the sun This field is compliant with Dublin Core.	title	No
itm_description	A detailed description of the content. Example: Int. Christiansborg: Shootings of prime minister Anders Fogh arriving; Anders Fogh making a speech in the Folketinget. This field is compliant with Dublin Core.	description	No
itm_publisher	An entity responsible for making the production available. Examples: DR, CNN, BBC... Reference table is attached. This field is compliant with Dublin Core.	char100	No

itm_department	Department responsible for making the production. Example: News, DR Life Style... Reference table is attached. This field is compliant with Dublin Core.	char100	No
itm_office	Office responsible for making the production. Example: DR Morning Reference table is attached. This field is compliant with Dublin Core.	char100	No
itm_create_timestamp	Creation date and time. Example: 18-01-2004 20:33:37 For imported material the create date refer to the date for formal issuance (e.g. publication) This field is compliant with Dublin Core.	date_time	Yes
itm_update_timestamp	Update date and time. Example: 18-01-2004 20:33:37 This field is compliant with Dublin Core.	date_time	No
itm_digitized_timestamp	Date and time for digitization of material. Example: 18-01-2004 20:33:37 This field is compliant with Dublin Core.	date_time	No
itm_technical_comment	Technical comments. Example: Some shots on video pixelated	description	No
itm_archive_comment	Archive comments. Example: Names not verified	description	No
itm_rights_url	URL pointing to detailed rights information in Libra. Is not implemented yet (20041216).	URL	No
itm_rights_description	Short description of rights and restrictions. Example: Can only be used by Teh News. Copyright: Vesterbro TV. This field is compliant with Dublin Core.	description	No
itm_source_url	Unique source url for item imported from external system. Example: Dalet+\News\955208ad-5c4c	URL	No
itm_category	Category for Item. This field is default inherited from Production. Examples: News, Entertainment... This field is compliant with Dublin Core.	char100	No
itm_killdate	Date when item should be deleted. Example: 10-06-2004	date_time	No
itm_tx_type	Information of what media the item was produced for. Indicates the usability of the item. Example: TV Reference table is attached.	char100	No

itm_coverage_spatial	Location for the content's subject. Example: the subject 'danish foreign policy ' gets Denmark in Location Reference table is attached. This field is compliant with Dublin Core.	char100	No
itm_location	Location for recording. Example: France, Paris Reference table is attached.	char100	No
itm_email	Email adress on the person, which should be notified, when the item is going to be deleted.	email	No

Comments:

An Item is created as part of a production or an independent item (such as an ingested asset before a relation to a production has been established).

The Item serves as a container for media sharing the same descriptive metadata such as multiple versions of an edited story.

Relations between items that are based on other items are maintained using the Item-To-Item entity. This relationship is maintained upon item creation and deletion.

Item creation is described in section .

A language and a subject relation cover the Dubin Core fields Language and Subject.

Relations:

Refers to 0:M media_object_group

Refers to 0:M subjectrel

Refers to 0:M languagerel

Refers to 0:M namerel

Refers to 1:1 archive_status

Refers to 1:1 registration_level

Refers to 0:1 production

Refers to 1:1 rightscode

6.11.language

A list of valid codes and names for Language, which describes the primary spoken language of the program.

Attribute name	Comment	Data type	Mand.
lan_id	ID for Language.	id_number	Yes
lan_code	International code for language. Example: dan	char100	Yes
lan_name_dk	Danish name for language. Example: dansk	char100	Yes

lan_name_uk	English name for language. Example: Danish	char100	Yes
-------------	---	---------	-----

Comments:

Language is based on the ISO 639-2 standard.

The system administrator maintains the table of names.

Relations:

Refers to 1:M languagerel

6.12.languagerel

Relations between Language, Country and Item are stored in the Language Relation entity.

Attribute name	Comment	Data type	Mand.
lre_id	ID for Languagerel.	id_number	Yes
itm_id	ID for Item.	id_number	Yes
cou_id	ID for Country.	id_number	Yes
lan_id	ID for Language.	id_number	Yes

Relations:

Refers to 1:1 language

Refers to 1:1 country

Refers to 1:1 item

6.13.media_object

A media_object defines a piece of media in terms of media type and relations to zero or more instances.

All instances defined by the media_object should be semantically equivalent.

Attribute name	Comment	Data type	Mand.
mob_id	ID for media_object.	id_number	Yes
mog_id	ID for media_object_group.	id_number	Yes
mot_id	ID for mob_type.	id_number	Yes
mos_id	ID for mob_subtype.	id_number	No
mob_smpte_umid	Unique material identifier.	char100	Yes
mob_characteristics	Text describing the content. Example: Sound - cleansound.	title	No

mob_create_timestamp	Time for creation or the recording. Default Mog_create_timestamp. Example: 21-02-2004 17:10:32	date_time	Yes
mob_isapprox	The create timestamp is approximately.	boolean	No
mob_isready	Flag indicating the instance (s) is ready for publication.	boolean	No
mob_isready_review	Flag indicating the instance is ready for review.	boolean	No
mob_keyframes	A collection of automatic generated images (keyframes) associated to video media objects.	image	No
mob_physical_location_id	ID describing indentifier on physical media e.g. a tape number. This attribute is only used when storing essence on physcial media such as videotapes or optical discs. Example: BD029179	char100	No
mob_physical_location	Description of physical location. This attribute is only used when storing essence on physical media such as videotapes or optical discs. Example: 300/38	char100	No
mob_source_asp_id	Attribute describing the aspect_ratio. Reference table is attached.	id_number	No
mob_source_art_id	Attribute describing the audio recording type. Reference table is attached.	id_number	No

Comments:

Type can be one of (non-conclusive list): video, audio, still, graphics, data (abstract types) and text (ascii). These types are defined in the entity mob_type.

It is anticipated that the media object type is updated automatically by the asset management system.

For Text type Media objects, there will be associated an reference to the mob_subtype entity. The text object subtypes are used to qualify various text content types as for example intro, manuscript etc.

System implementations should support free-text search of relevant text media objects.

The DATA type is used to store publishable material such as web articles, teletext information, MHP programs etc.

mob_subtype entity.

A media_object can have more than one instance. This is relevant where material is stored in multiple formats and locations such as low- or high resolution formats, physical media or digitized files in an asset management system.

Relations:

Refers to 1:M mob_instance

Refers to 0:M timecode_description

Refers to 1:1 mob_type

Refers to 1:1 media_object_group

Refers to 0:1 mob_subtype

6.14.media_object_group

The media_object_group is a container for one or more semantically equivalent media_objects defined in time by start- and stop time code.

Attribute name	Comment	Data type	Mand.
mog_id	ID for media_object_group.	id_number	No
itm_id	ID for Item.	id_number	Yes
mog_number	The mog_number is used in combination with the mog_version to identify a particular version of Media Objects.	number	Yes
mog_version	Media object group version.	number	No
mog_version_name	Name for group like Camera 1, or scene 1 - used to describe the type of the mog_version. Afvigelse fra den fysiske model mog_version_name	title	No
mog_start_timecode	Start time code for the media_object_group. Example: 00:02:00	timecode	No
mog_stop_timecode	Stop time code for the media_object_group. Example: 02:30:05	timecode	No
mog_duration	Duration of the media_object_group. Example: 00:01:04	duration	No
mog_create_timestamp	Creation time stamp for media_object_group. Example: 28-05-2004 18:02:19	date_time	Yes
mog_update_timestamp	Update timestamp for media_obejct_group. Example: 27-05-2004 19:02:19	date_time	No
mog_classification	Classification of the essence type. The field can have none or one value. Examples: transmission copy, stock shots.... Reference table is attached. The classification is intended to be filled in automatically from the systems, which feed essence into the DRAMS.	char100	No

mog_isprotected	Flag indicating that the instance is boolean referenced by an element (program or publication). If this flag is set then it is anticipated that the Item must not be deleted. It is anticipated that flag IsProtected automatically is set when a relation between an Element and a Media Object Instance is created.	boolean	No
mog_1st_tx_timestamp	Timestamp for 1st transmission. Example: 08-08-2004.	date_time	No

Comments:

The media_object_group is used to provide a common frame of reference for related media objects and the time code constraint applied at this level enable implementation of features for example correlation between video, keyframes and audio.

Even though a media_object_group may contain many Media Objects representing content with derived formats, this content must be semantically equivalent. For example, if there are two media objects of type respectively audio and video, these objects must represent the same content.

It is anticipated that flag IsProtected automatically is set when a relation between an Element and a Media Object Instance is created.

There must be at most one media object per video content type. For example, a single video media object should be used to describe a high-res and a low-res video instance.

The mog_version attribute is used to support versioning of content. When used in combination with the mog_number attribute, mog_version can be used to identity a particular version of Media Objects.

It is anticipated that all fields for the media object group entity are maintained by the asset management system.

Relations:

Refers to 0:M media_object

Refers to 0:M media_tracking

Refers to 0:M media_tracking

Refers to 1:1 item

6.15.media_tracking

Media_tracking contains linkage between Media Object Groups. This Linkage is used to save inheritance and reference information between Media Object Groups.

Attribute name	Comment	Data type	Mand.
met_id	ID for media_tracking.	id_number	Yes
mog_id	ID for media_object_group.	id_number	Yes

mog_id_source	Reference ID to the source video Media Object - the parent.	id_number	Yes
met_in_offset	Offset for 'start location' for the actual video Media Object - the child	timecode	Yes
met_out_offset	Offset for stop location for the actual video Media Object - the child.	timecode	Yes
met_source_out_offset	Offset for 'stop location' for the source video Media Object - the parent.	timecode	Yes
met_source_in_offset	Offset for 'start location' for the source video Media Object - the parent.	timecode	Yes
met_source_url	Unique source system identifier for the source Media Object video. The ID could be a SMPTE UMID, a DRAMS_ID or an ID with the syntax: DR\system\database\id in the database.		No

Comments:

The Media Tracking Table contains a full reference table where a direct link between every pair of related video Media Objects is listed.

The primary motivation for the linkage and reference rules above is to enable efficient media tracking and reference rules above is to enable efficient media tracking and cascading updates (such as rights annotation) where it's possible.

Relations:

Refers to 1:1 media_object_group

Refers to 1:1 media_object_group

6.16.mob_instance

The Mob Instance entity represents a reference to a Media Object Instance. The instance description contains basic properties such as date, location, format, status and format-specific information.

Attribute name	Comment	Data type	Mand.
min_id	ID for mob_instance.	id_number	Yes
mob_id	ID for media_object.	id_number	Yes

min_drams_id	Unique identifier for a DRAMS Mob_instance. Example: DR\Mediearkivet\MIN\2661224 Syntax: Organisation \ Instance of archive \ object type \ serial number	char100	Yes
min_location_url	Locator for this essence file. This can be a reference to a file in a production area or on a remote server. This may also be used as a system-specific identifier for archive systems or similar.	URL	No
min_physical_format	Foreign Key to Media format entity. Audio/video format for physical media. Examples: Beta sp, Betadigital... The system administrator maintains the table.	char100	No
min_fmt_mime_type	MIME type describing the format (file extension). Examples: .doc, .tif	char100	No
min_fmt_codec	Foreign key to Media codex entity. Actual coding format. Examples: DVCPRO, BWF The system administrator maintains the table.	char100	No
min_fmt_audio_channels	Audio channel format (1,2,4,5.1)	char100	No
min_fmt_picture_size	Picture size pixels.	char100	No
min_fmt_bitrate	Average bit rate for this instance in Bytes/s.	number	No
min_fmt_size	Size of referenced file in Bytes.	number	No
min_create_timestamp	Creation time stamp. Example: 21-02-2004 17:10:32	date_time	Yes
min_update_timestamp	Update time stamp Example: 21-02-2004 17:10:32	date_time	No
min_data	If Text Media Object is supposed to be indexed then data are stored in min_data.	description	No
min_isblackandwhite	Flag for black and white video, still and graphics.	boolean	No
min_isstoredinline	Flag telling that text media object is stored in min_data.	boolean	No

Comments:

Instance formats for file based and non-physical media's are specified as described below:

	Video	Audio	Still	Graphic	Data	Text
MIN_FMT_MIME_TYPE	X	X	X	X	X	X
MIN_FMT_CODEC	X	X				

MIN_FMT_AUDIO_CHANNELS	X	X				
MIN_FMT_PICTURE_SIZE	X		X	X		
MIN_FMT_BITRATE	X	X				
MIN_FMT_SIZE	X	X	X	X	X	X

It is anticipated that fields describing instance formats, timestamps and Location URL are updated automatically by the asset management system.

It is recommended that searchable instances such as texts are kept online for indexing purposes.

Example

A media object (of type video) can contain two instances:

1. A DVCPRO file in an asset management system
2. A physical DVCPRO tape.

In this example, the file instance locator's point to the asset management system and the tape locator contains a physical tape ID.

Relations:

Refers to 0:M element

Refers to 1:1 media_object

6.17.mob_subtype

A list of valid values for the mob_subtype, which describes the subtypes of the mob_types data and text.

Attribute name	Comment	Data type	Mand.
mos_id	ID for mob_subtype.	id_number	Yes
mot_id	ID for mob_type.	id_number	Yes
mos_code	Code for Subtype. Examples: I, M ...	char100	Yes
mos_name_dk	Name for subtype in Danish. Examples: Oplæg, Nedlæg, Manus	char100	Yes
mos_name_uk	Name for subtype in English. Example: Into, Outro, Manuscript	char100	Yes
mos_isindexed	Flag indication that the text shall be indexed.	boolean	No

Comments:

Mob_subtype is referenced by the entities media_object and Element and related to the mob_types data and text.

The system administrator maintains the table.

Relations:

Refers to 0:M element

Refers to 0:M media_object

Refers to 1:1 mob_type

6.18.mob_type

A list of valid values for mob_type, which describes the type of material.

Attribute name	Comment	Data type	Mand.
mot_id	ID for mob_type.	id_number	Yes
mot_code	Code for type: Examples: A (Audio), V (Video), S (Still), G (Graphic), D (Data), T (Text)....	char100	Yes
mot_name_dk	Danish name for type: Examples: Lyd (Audio), Video (Video), Billede (Still), Grafik (Graphic), Data (Data) and Tekst (Text)...	char100	Yes
mot_name_uk	English name for type: Examples: Audio, Video, Still, Graphic, Data, Text...	char100	Yes

Comments:

Mob_type is referenced by the entity Media_object.

For all other mob_types than 'data' the mob_type=mob_subtype.

This list is maintained by the system provider and changes will reflect search and update procedures.

Relations:

Refers to 1:M mob_subtype

Refers to 0:M media_object

6.19.name

The Person entity stores basic person information such as name and characteristics.

Attribute name	Comment	Data type	Mand.
nam_id	ID for Name.	id_number	Yes
nam_name	Name for person, group, ensemble etc. Example: Anders Fogh Rasmussen This field is compliant to Dublin Core.	char100	Yes
nam_characteristics	Characteristics such as current title and salutation. Example: primeminister, Denmark.	char100	No
nam_source_url	A reference to an external catalogue URL system.		No
nam_isverified	Name has been verified by DR-ARC and is the authorized form, which must be used in DR for the peson/group etc.	boolean	No

Comments:

Names are specified in straight order. For a person this means from first to last name.

The Characteristics attribute is restricted to store current title etc, ie information that may not be subject to manual maintenance.

The Source URL attribute should be used to link the above name information to the central name and address catalog system where up-to-date and accurate information can be retrieved at any time.

If no characteristics and source_url there should only be one row for a specific name.

Relations:

Refers to 0:M namerel

6.20.namerel

Relation between Name and Role is stored in the Name Relation entity.

Attribute name	Comment	Data type	Mand.
nre_id	ID for Namerel.	id_number	Yes
itm_id	ID for Item.	id_number	No
prd_id	ID for Production.	id_number	No
pro_id	ID for Program.	id_number	No
pub_id	ID for Publication.	id_number	No
ele_id	ID for Element.	id_number	No

rol_id	ID for Role.	id_number	Yes
--------	--------------	-----------	-----

Relations:

Refers to 0:1 item
 Refers to 0:1 production
 Refers to 0:1 program
 Refers to 0:1 publication
 Refers to 0:1 element
 Refers to 1:1 role
 Refers to 1:1 name

6.21.object_type

A list of valid values for object _type, which describes the object in relation to the data model.

Attribute name	Comment	Data type	Mand.
obj_id	ID for the object _type.	id_number	Yes
obj_code	Contains value for object _type. Examples: Item, Mog, Mob_instance., Program...	char100	Yes
obj_description	Extended description of object_type.	description	No
obj_current	Flag indicates that the value is in use.	boolean	Yes

Comments:

Object_type is referenced by the entity folder_content.
 The system provider maintains the list and changes will reflect the workflow.

Relations:

Refers to 0:M folder_content

6.22.pro_publish

Publishing log containing information like channel, planned and actual start-, stop timestamp and some flags.

Attribute name	Comment	Data type	Mand.
ppu_id	ID for pro_publish.	id_number	Yes
pro_id	ID for Program.	id_number	Yes
ppu_source_url	Unique source system transmission slot identifier Example: whatsOn\553426859	URL	Yes
ppu_channel	Reference to Channel entity. The broadcast mother channel.	char100	Yes
ppu_start_timestamp_1	Planned date and time for start of transmission. This field is compliant to Dublin Core.	date_time	Yes
ppu_stop_timestamp_1	Planned date and time for end of transmission. This field is compliant to Dublin Core.	date_time	Yes
ppu_start_timestamp_2	Date and time for actual start of transmission. This field is compliant to Dublin Core.	date_time	No
ppu_stop_timestamp_2	Date and time for actual end of transmission. This field is compliant to Dublin Core.	date_time	No
ppu_isapprox_timestamp	Flag indicates that the start time for transmission is approximately.	boolean	No
ppu_isrerun	Flag is set for Rerun.	boolean	No
ppu_isquick_reprise	Flag is set for Quick reprise.	boolean	No
ppu_isnotpublished	Flag is set for 'Has not been transmitted'.	boolean	No
ppu_islive	Flag is set for Transmitted live.	boolean	No

Comments:

All data are intended automatic to be generated from the program planning system and from the playout system.

The entity pro_publish is related to continuous program slot in the program plan. Each time a program is broadcasted it will result in a pro_publish.

The flag ppu_isrerun or ppu_isquick_reprise are used to describe the type of the additional publications of the same Program (and material).

Ppu_isapprox_timestamp is used for old material where the exact start timestamp is an estimate.

Relations:

Refers to 0:1 running_order

Refers to 0:M simul_channel

Refers to 1:1 program

Refers to 1:1 program

6.23.production

A production may be independent or related to a main production (parent). A main production defines a group of productions such as a series. The entity description contains basic properties such as DR production number and title.

Attribute name	Comment	Data type	Mand.
prd_id	ID for Production.	id_number	Yes
prd_number	The DR production number. Example: 591144050 This field is compliant to Dublin Core.	char100	Yes
prd_parent_number	If this production number exist then the actual production is a child production. Parent number referring to the parent production (main production). Example: 590300200 This field is compliant to Dublin Core.	char100	No
prd_type	Indicates the type of the Production. Examples: Radio, TV, Web... Reference table is attached.	char100	Yes
prd_title	Primary production title. Example: Ørnen (The Eagle) This field is compliant to Dublin Core.	title	Yes
prd_title_alt	Alternate production title 'Working title'. This field is compliant to Dublin Core.	title	No
prd_description	Description of the production This field is compliant to Dublin Core.	description	No
prd_category	Category for associated program. Examples: News, Entertainment. Reference table is attached. This field is compliant to Dublin Core.	char100	Yes
prd_targetgroup	Subcategory to Category - Targetgroup for the production. Example: Women, Adults. Reference table is attached. This field is compliant to Dublin Core.	char100	Yes

prd_intention	Subcategory for Category - Intention for production. Examples: Inform, Educate. Reference table is attached This field is compliant to Dublin Core.	char100	Yes
prd_department	Department (descriptive name) responsible for the production. Examples: News, DR Life Style... Reference table is attached.	char100	Yes
prd_create_timestamp	Creation timestamp. Example: 02-09-2003 09:05:00 This field is compliant to Dublin Core.	date_time	Yes
prd_update_timestamp	Update timestamp. Example: 22-06-2004 10:48:20 This field is compliant to Dublin Core.	date_time	No
prd_is_series	If the actual production is part of a series a flag is set.	boolean	No
prd_series_title	The title of the series the actual production is a part of.	title	Yes

Comments:

All data are intended automatic to be generated from the production planning system.

Type is an important attributes that implicit gives meaning to the production number.

The prd_type and prd_number is the unique key (URL) to the DR DataWarehouse where economy data and other administrative and production related metadata are hosted.

Relations:

Refers to 0:M namerel

Refers to 0:M program

Refers to 0:M project

Refers to 0:M item

Refers to 0:M publication

6.24.program

Material sequences from productions assembled to one sequence and transmitted as a program. Has metadata such as title, description etc. The Program is guaranteed to represent a single and unique sequence of material (list of Running Orders and Elements).

Attribute name	Comment	Data type	Mand.
pro_id	ID for Program.	id_number	Yes
prd_id	ID for Production.	id_number	No
rle_id	ID for registration_level.	id_number	Yes

pro_drams_id	Unique identifier for a DRAMS program. Example: DR\Mediearkivet\PRO\17529	DRAMS_id	Yes
pro_title	The program title. Example: Ørnen (The Eagle) This field is compliant to Dublin Core.	title	Yes
pro_description	The program description. This field is compliant to Dublin Core.	description	No
pro_category	Category for associated program. Examples: News, Entertainment. Reference table is attached. This field is compliant to Dublin Core.	char100	No
pro_targetgroup	Subcategory to Category - Targetgroup for the production. Example: Women, Adults. Reference table is attached. This field is compliant to Dublin Core.	char100	No
pro_intention	Subcategory for Category - Intention for production. Examples: Inform, Educate. Reference table is attached This field is compliant to Dublin Core.	char100	No
pro_department	Department (descriptive name) responsible for the programme. Examples: News, DR Life Style... Reference table is attached.	char100	No
pro_create_timestamp	Creation timestamp. Example: 02-09-2003 09:05:00 This field is compliant to Dublin Core.	date_time	Yes
pro_update_timestamp	Update timestamp. Example: 22-06-2004 10:48:20 This field is compliant to Dublin Core.	date_time	No
pro_archive_comment	Archive comment. Example: cataloguing not finished	description	No
pro_tx_type	Indicates the type of the Program. Examples: Radio, TV, Web... Reference table is attached.	char100	Yes
pro_islocked	Locked for automatic update.	boolean	No

Comments:

Normally there will be exactly one reference to pro_publish for each Program. If there are several transmissions for a program then the flag PPU_ISRERUN og PPU_ISQUICK_REPRISE is set for all but the first occasion.

All data except for Pro_archive_comment data are intended automatic to be generated from the transmission planning system.

Category. Targetgroup, Intention and Department are normally replicated from the Production. For old material these information are explicit given.

The flag pr_istv should be set depending on the value of prd_type. For an old material this information should be explicit given. If the flag is not set then the program is made for radio.

Refer to section 12.3 for information regarding use and best practice of the pro_islocked attribute.

Relations:

Refers to 0:M subjectrel

Refers to 0:M namerel

Refers to 0:M running_order

Refers to 1:M pro_publish

Refers to 1:M pro_publish

Refers to 1:M pub_pro_rel

Refers to 0:M subjectrel

Refers to 0:1 production

Refers to 1:1 registration_level

6.25.project

Project reference to a production.

Attribute name	Comment	Data type	Mand.
prj_id	ID for Project.	id_number	Yes
prd_id	ID for Production.	id_number	Yes
fol_id	ID for Folder.	id_number	Yes
prj_title	Project title.	title	Yes
prj_drams_id	Unique indentifier for a DRAMS project. Example: DR\Mediearkivet\PRJ\45678	DRAMS_id	Yes

Relations:

Refers to 1:1 folder

Refers to 0:M project_history

Refers to 1:1 production

6.26.project_history

The Project History is meant as a help for the user. If for instance a user wants to export a Project to an NLE it can be crucial to know if the Project already is in use and has been checked out 2 hours earlier. It is also important to know that a NLE Project never has been backed up in the Media Archive.

When a Project is imported into or exported from the Media Archive the Project History is automatically updated. The user are not allowed to manipulate data in Project History.

Attribute name	Comment	Data type	Mand.
phi_id	ID for project _history.	id_number	Yes
phi_action_type	Valid values for action_type are 'import' and 'export' ,	char100	Yes
phi_destination	Destination contain the name for the workspace / machine from where the Project has been imported or exported. Example: 'AVID WS01'	char100	Yes
phi_timestamp	Timestamp for the action 'import' or 'export'. Example: 22-06-2004 10:48:20	date_time	Yes
phi_userid	Userid for the person responsible for the action 'import' or 'export'. Example: oh	char100	Yes

Relations:

Refers to 1:1 project

6.27.pub_pro_rel

Relation between Publication and Program is stored in the pub_pro_rel entity.

Attribute name	Comment	Data type	Mand.
ppr_id	ID for Publication to program relation.	id_number	Yes
pub_id	ID for Publication.	id_number	Yes
pro_id	ID for Program.	id_number	Yes

Comments:

The relations are expected to be maintained both manually and automatic.

Relations:

Refers to 1:1 publication

Refers to 1:1 program

Refers to 1:1 publication

6.28.pub_publish

Publishing log that contains start-, stop timestamp and location.

Attribute name	Comment	Data type	Mand.
puu_id	ID for pub_publish.	id_number	No
pub_id	ID for Publication.	id_number	Yes
puu_start_timestamp	Timestamp for start of publishing. Example: 22-06-2004 10:48:20 This field is compliant to Dublin Core.	date_time	No
puu_stop_timestamp	Timestamp for end of publishing. Example: 22-08-2004 24:00:00 This field is compliant to Dublin Core.	date_time	No
puu_site	Site name or low-level address for publication	char100	No
puu_portal	Foreign Key to Portal entity. Name of portal or high-level address for publication. Examples: dr.dk, education ... Reference table is attached.	char100	No

Comments:

All data are intended automatic to be generated from the web CMS system.

Relations:

Refers to 1:1 publication

6.29.pub_to_pub

Link between a Publication extracted from a Publication and mapping to pub_id for internal navigation.

Attribute name	Comment	Data type	Mand.
ptp_id	ID for pub_pro_rel.	id_number	Yes
pub_id	ID for Publication.	id_number	Yes
pub_id	ID for related Publication.	id_number	Yes

ptp_href_url	Original link extracted from Publication.	URL	Yes
ptp_title_link	Name or description related to link. title		No

Comments:

All data are intended automatic to be generated from the web CMS system.

Relations:

Refers to 1:1 publication

Refers to 1:1 publication

6.30.publication

Material from productions packaged for publishing such as print, web, teletext etc.

Attribute name	Comment	Data type	Mand.
pub_id	ID for Publication.	id_number	Yes
prd_id	ID for Production.	id_number	No
rle_id	ID for registration_level.	id_number	Yes
pub_drams_id	Unique identifier for a DRAMS publication. Example: DR\Mediearkivet\PUB\24466187 Syntax: Organisation \ Instance of archive \ object type \ serial number This field is compliant to Dublin Core.	DRAMS_id	Yes
pub_source_url	Unique article source system identifier. Example: dr.dk\Science\34758697	URL	Yes
pub_title	Primary article title. Example: DR Nyheder Online - Udland - Italien vil trække sig ud af Irak (DR News Online - Foreign Affairs - Italy wants retreat from Irak). This field is compliant to Dublin Core.	title	Yes
pub_title_alt	Alternate article title. This field is compliant to Dublin Core.	title	No
pub_description	Description of the publication. This field is compliant to Dublin Core.	description	No
pub_category	Category for associated program. Examples: News, Entertainment. Reference table is attached. This field is compliant to Dublin Core.	char100	No

pub_targetgroup	Subcategory to Category - Targetgroup for the production. Example: Women, Adults. Reference table is attached. This field is compliant to Dublin Core.	char100	No
pub_intention	Subcategory for Category - Intention for production. Examples: Inform, Educate. Reference table is attached This field is compliant to Dublin Core.	char100	No
pub_department	Department (descriptive name) responsible for the production. Examples: News, DR Life Style... Reference table is attached.	char100	No
pub_archive_comment	Archive comment. Example: cataloguing not finished.	description	No
pub_create_timestamp	Creation timestamp. Example: 02-09-2003 09:05:00 This field is compliant to Dublin Core.	date_time	Yes
pub_update_timestamp	Update timestamp. Example: 22-08-2004 10:48:20 This field is compliant to Dublin Core.	date_time	No
pub_islocked	Locked for automatic update	boolean	No

Comments:

The pub_description attribute is used to store meta-information for the publication. The full text content, such as an article source, may be stored in a Media Object.

Content text are treated as essence and stored in a Media Object / Instance hierarchy and referenced using Elements. Publication references to content (instances) managed by DRAMS are also translated to Elements.

All data except for pub_archive_comment are intended automatic to be generated from the web CMS and similar systems.

Category, Targetgroup, Intention and Department should be taken from a DR controlled lists.

Relations:

Refers to 0:M pub_publish

Refers to 0:M pub_to_pub

Refers to 0:M pub_to_pub

Refers to 0:M pub_pro_rel

Refers to 0:M namerel

Refers to 0:M running_order

Refers to 0:M pub_pro_rel

Refers to 0:M subjectrel

Refers to 0:1 production

Refers to 1:1 registration_level

6.31.registration_level

The Registration Level describes the level of the registration in relation to applied metadata and method of viewing/listning of content file.

Attribute name	Comment	Data type	Mand.
rle_id	ID for registration_level.	id_number	Yes
rle_code	Code for registration_level.	char100	Yes
rle_text	Text decribing the level of registration. Examples: Fully described, Level 4, Viewed ...	description	No

Comments:

Registration level is referenced by the entities Item, Program and Publication.

The system administrator maintains the list.

Relations:

Refers to 0:M item

Refers to 0:M program

Refers to 0:M publication

6.32.rights

Rights contains information relevant for reuse of the material.

Attribute name	Comment	Data type	Mand.
rig_id	ID for Rights.	id_number	Yes
rig_create_timestamp	Create timestamp for Rights information. Example: Example: 02-09-2003 09:00:00	date_time	Yes
rig_create_user	User id for the person, who creates the Rights information.	char100	Yes
rig_update_timestamp	Update timestamp for Rights information. Example: Example: 02-09-2003 09:05:00	date_time	Yes
rig_update_user	User ID for the person, who updates Rights information.	char100	Yes
rig_comment	Comment to the Rights information	description	No

rig_use_start_timestamp	Start timestamp for the DR usage of the material. Example: 02-09-2004 00:00:00	date_time	No
rig_use_end_timestamp	End timestamp for the DR usage of the material. Example: 03-09-2003 00:00:00	date_time	No
rig_use_counter	Counter that count how many times the material has been used.	number	No
rig_use_counter_initial	The maximum number for usage.	number	No
rig_cleared_for_ondemand	Flag indicates that the rights have been cleared for Ondemand usage.	boolean	Yes
rig_cleared_for_broadcast	Flag indicates that the rights have been cleared for broadcast usage.	boolean	Yes
rig_cleared_for_download	Flag indicates that the rights have been cleared for download usage.	boolean	Yes
rig_cleared_for_clips	Flag indicates that the rights have been cleared for clip usage.	boolean	Yes

Relations:

Refers to 0:M document

Refers to 1:1 rightscode

Refers to 1:1 rights_target

6.33.rights_target

Rights_target is used to connect rights information to files in DR Asset Management Systems.

Attribute name	Comment	Data type	Mand.
rta_id	ID for rights_target.	id_number	Yes
rta_untreated	Flag indicates that the rights has not been cleared.	boolean	Yes
rta_source_url	Rta_source_url point to the production, program, item or publication, for which the rights information is valid through the DRAMS ID Example: DR\Mediearkivet\PRO\17529	URL	Yes

Relations:

Refers to 1:1 rights

6.34.rightscore

A list of valid values for Rightscode, which describes rights and restriction in rough categories.

Attribute name	Comment	Data type	Mand.
ric_id	ID for Rightscode.	id_number	Yes
ric_code	Code for rights. Examples: A (red light), B (yellow light), C (green light) ...	char100	Yes
ric_text	Description of the meaning of the RightsCode. Examples: Free broadcast (except for collecting societies for music and litterature), Broadcast against extra payments Broadcast requires new contract(-s) or cannot be broadcasted...	description	Yes

Comments:

Rights_code is referned by the entities Item and Rights.

The system administrator maintains the list.

Relations:

Refers to 1:M rights

Refers to 1:M item

6.35.role

A list of valid codes and names for roles.

Attribute name	Comment	Data type	Mand.
rol_id	ID for Role.	id_number	Yes
rol_role_dk	Role name in Danish. Example: Fotograf	char100	Yes
rol_role_dc	Role name in English. Example: Cameraman	char100	No
rol_iscreator	Role only applicable for creator attributes. Flag is set when role is creator.	boolean	No

Comments:

Role is referenced through Namerel.

The roles addressed by the role identifier are specified in Dublin Core (EBU Tech 3293). In additions, Danish role name translations are provided.

The system administrator shall manually be able to insert and update names.

Relations:

Refers to 0:M namerel

6.36.running_order

A Running Order defines a schedule for a Program (Program slot) or a part of a Program and consists of an ordered sequence of Element Groups and Elements.

Attribute name	Comment	Data type	Mand.
ruo_id	ID for running_order.	id_number	Yes
pub_id	ID for Publication.	id_number	No
pro_id	ID for Program.	id_number	No
ppu_id	ID for pro_publish.	id_number	Yes
ruo_drams_id	Unique identifier for a DRAMS running order. Example: DR\Mediearkivet\RUO\453215 Syntax: Organisation \ Instance of archive \ object type \ serial number This field is compliant to Dublin core.	DRAMS_id	Yes
ruo_source_url	Unique source system identifier for a Running Order. Example: NEWSTAR\TVA0700_02/11/2004_22:35:22 This field is compliant to Dublin Core.	URL	No
ruo_title	Title for Running Order. Example: Deadline 17:00 This field is compliant to Dublin Core.	title	Yes
ruo_previous	Indicates situations where one Running order follows another. Content is the unique ruo_drams_id for the previous Running order.	DRAMS_id	No
ruo_next	Indicates situations where one Running Order is followed by another Running Order. Content is the unique RUO_DRAMS_ID for the next Running Order.	DRAMS_id	No
ruo_create_timestamp	Creation date and time. Example: 18-01-2004 20:33:37	date_time	Yes
ruo_update_timestamp	Update date and time. Example: 18-01-2004 20:33:37 This field is compliant with Dublin Core.	date_time	No

ruo_type	Type of Running Order. Examples: Transmitted running order, Archived running order, Consolidated running order.	char100	Yes
ruo_tx_timestamp	Timestamp for transmission.	date_time	No

Comments:

A RunningOrder is typically imported from a NCS (News Computer System), from a Transmission system or from a Computer Aided Radio or TV System but a RunningOrder may also be created and maintained in DRAMS.

The fields RUO_NEXT and RUO_PREVIOUS are used to support navigation between chained RunningOrders.

A RunningOrder or a chained list of RunningOrders describe one continuous timeline.

Example:

The News Hour may include 3 RunningOrders.

One Running Order covers the whole program and consists of one ElementGroup and one Element. This Element refers to one Media Object Instance ingested when the Program was transmitted (transmission log).

```
ruo_drams_id: DR\Mediearkivet\RUO\142541
ruo_source_url: HARRIS\TXlog\34523145
ruo_title: TVA 18:30
ruo_type: Transmissionlog
ruo_next
ruo_previous
```

The other two RunningOrders cover parts of the program respectively News and Sport. Both Running Orders are imported from the NCS system and consists of many ElementGroups and Elements and some or all of the Elements may refer to a Media Object Instance.

```
ruo_drams_id: DR\Mediearkivet\RUO\142533
ruo_source_url: Dalet+\News\23145
ruo_title: News 18:30
ruo_type: Archived Running Order
ruo_next: DR\Mediearkivet\RUO\142534
ruo_previous
```

```
ruo_drams_id: DR\Mediearkivet\RUO\142534
ruo_source_url: Dalet+\News\23147
ruo_title: Sport 18:30
ruo_type: Archived Running Order
ruo_next
ruo_previous: DR\Mediearkivet\RUO\142534
```

Relations:

Refers to 0:M element_group

Refers to 0:1 publication

Refers to 0:1 program

Refers to 0:1 pro_publish

6.37.simul_channel

The Simultaneously Channel entity describes child channels associated to one transmission of a Program.

Attribute name	Comment	Data type	Mand.
sic_id	ID for simul_channel.	id_number	Yes
sic_channel	Short name for transmission channel.	char100	Yes

Comments:

All data are intended automatic to be generated from the transmission planning system.

Relations:

Refers to 1:1 pro_publish

6.38.subject

The Subject entity contains flat list of keywords that constitutes a controlled vocabulary.

Attribute name	Comment	Data type	Mand.
sub_id	Id for Subject.	id_number	Yes
sub_type	Valid keyword types D, T, U D: Keyword is a Danish librarian code (DK5) T: Keyword is part of the thesaurus U: Keyword is not a part of the thesaurus	char100	Yes
sub_keyword	Contains the keyword.	char100	Yes

Comments:

Depending on the sub_type the Subject table will contain a flat version of the centrally managed DR thesaurus or lists of uncontrolled keywords or classifications (DK5).

The system administrator maintains the thesaurus. The other lists U (Uncontrolled) and D (DK5 classification) are maintained by the user.

Relations:

Refers to 0:M subjectrel

6.39.subjectrel

Relations between Subject and Item, Program, Publication and Keyframe are stored in the Subject Relation entity.

Attribute name	Comment	Data type	Mand.
sre_id	ID for subjectrel.	id_number	Yes
sub_id	Id for Subject.	id_number	Yes
itm_id	ID for Item.	id_number	No
pro_id	ID for program.	id_number	No
tid_id	ID for timecode_description.	id_number	No
pub_id	ID for Publication.	id_number	No

Relations:

Refers to 1:1 subject

Refers to 0:1 item

Refers to 0:1 program

Refers to 0:1 program

Refers to 0:1 timecode_description

Refers to 0:1 publication

6.40.timecode_description

Defines a description primarily related to a video Media Object. The entity description contains basic properties such as time code and maybe an image covering the actual keyframe related to timecode.

Attribute name	Comment	Data type	Mand.
tid_id	ID for timecode_description.	id_number	Yes
mob_id	ID for media_object.	id_number	Yes
tid_timecode	Time code for the capture of the keyframe.	timecode	Yes
tid_duration	Duration for clip.	duration	No
tid_title	Timecode title.	title	Yes

tid_description	Text describing the clip. This field is compliant to Dublin Core.	description	No
tid_keyframe	Actual keyframe (image).	image	No

Comments:

Keyframes related to a video Media objects are intended automatic to be generated when a video Media Object is created or updated. The set of automatic generated images are stored in the field Mob_keyframes.

A Video image and description can also be added manual. In this case the image is stored in the Tid_keyframe field.

Relations:

Refers to 0:M subjectrel

Refers to 1:1 media_object

7. Taxonomies and thesaurus

7.1. Reference tables based on scheduled configuration update jobs

In contrast to hardcore reference tables for department etc. there should automatically be generated reference tables for Channel, Portal, Category, Targetgroup, Intention, Department and Production type. Data are maintained in systems out side DRAMS environment but DRAMS subscribe on these data and the reference tables are based on already imported data. For few of the reference tables it shall be possible to add new values e.g. Department.

These lists are primarily intended as a help function for search.

Category

A list of valid names for Category used in prd_category, pro_category, itm_category and pub_category. The user may add values.

Examples: News, Entertainment etc.

Channel

A list of valid names for Channel used in ppu_channel and sic_channel. The user can't add values.

Examples: P1, DR2 etc.

Department

A list of valid names for Department used in prd_department, pro_department, itm_department and pub_department. The user may add values.

Examples: B&U, News

Intention

A list of valid names for Intention used in prd_intention, pro_intention and pub_intention. The user can't add values.

Examples: Inform, Educate ect.

Office

A list of valid names for Office used in itm_office. The user may add values.

Examples: Science, Trailer office etc.

Portal

A list of valid names for Portal used in puu_portal. The user can't add values.

Examples: dr.dk, education etc.

Targetgroup

A list of valid names for Targetgroup used in prd_targetgroup, pro_targetgroup and pub_targetgroup.

The user can't add values.

Examples: Women, Adults etc.

7.2. Reference tables based on International standards

Country

A list of valid codes and names for countries based on the ISO 3166-1 standard.

The system administrator shall manually be able to insert and update names in Danish.

Language

A list of valid codes and names for languages based on the ISO 639-2 standard.

The system administrator shall manually be able to insert and update names in Danish.

Role

A list of valid codes and names for roles based on the Dublin Core (EBU Tech 3293) standard. The system administrator shall manually be able to insert and update names in Danish.

7.3.Reference tables managed by the System administrator

Aspect ratio

A list of valid values for Aspect_ratio used in mob_source_asp_id. Data are maintained by the system administrator.

Examples: 16:9, letterbox etc.

Audio recording type

A list of valid values for Audio_recoding_type used in mob_source_art_id. Data are maintained by the system administrator.

Examples: Right, Left etc.

Classification

A list of valid values for Classification used in mog_classification. Data are maintained by the system administrator.

Examples: Transmission copy with logo, footage etc.

Document type

A list of valid values for Document_type referenced by the entity Document. Data are maintained by the system administrator.

Examples: Contract, Music list etc.

Location

A list of often used values for location used for itm_location and itm_coverage_spatial. Data are maintained by the system administrator.

Examples: Copenhagen, DR Studio 3 etc.

MOB subtype

A list of valid values for MOB_subtype referenced by the entities Media_object and Element. Data are maintained by the system administrator and is related to the MOB_types Data and Text.

Example: Manus, Intro, MHP application etc.

PhysicalFormat

A list of valid names for physical formats used in min_physical_format. The user can't add values.

Examples: Video tape, Disk etc.

Publisher

A list of valid values for Publisher used in itm_publisher. Data are maintained by the system administrator.

Examples: DR, CNN, BBC etc.

Registration level

A list of valid values for Registration_level referenced by the entities Item, Program and Publication. Data are maintained by the system administrator and reflects the archivist's needs.

Examples: with Archive comments, not finished etc.

Rights code

A list of valid values for Rights_code referenced by the entities Item and Rights. Data are maintained by the system administrator.

Examples: may be used, may not be used etc.

Subject

Keywords (sub_type='T') are the words that are covered by the DR thesaurus. We have not described this in the model because the actual implementation of the thesaurus depends on the database tools and utilities.

Data in DR thesaurus are maintained by the system administrator and reflects the archivist's needs other data (sub_type='D' or 'U') are maintained by user.

Tx_type

A list of valid values for Tx_type used in itm_txt_id, pro_txt_id, pub_txt_id and prd_type. Data are maintained by the system administrator.

Examples: Radio, TV etc.

7.4.Reference tables based on system provider update**Archive status**

A list of valid values for Archive_status referenced by the entity Item. Data are maintained by the system provider.

Examples: archive candidate, archived etc.

Media codec

A list of valid values for Media_codec used in min_fmt_codec. Data are maintained by the system provider.

Examples: DVCPRO25, BWF etc.

MOB type

A list of valid values for MOB_type referenced by the entity media_object. The system provider maintains data and changes will reflect search and update procedures.

Examples: Video, Audio etc.

Object type

A list of valid values for Object_type referenced by the entity folder_content. Data are maintained by the system provider and changes will reflect the workflow.

Examples: Item, Mog, Mob-instance, Program etc.

7.5.Maintenance for lookup dataUser Definitions:

User User with appropriate rights

SysA System administrator

Sys System user

Modul definitions:

WFE Web Front End

SA System Administration modul

SP System Provider modul

Delete only permitted if no references

	Create	Update	Delete	Current	Join	mass	Modul
Archive_status	Sys	Sys	Sys			X	SP

Aspect_ratio	SysA	SysA		SysA		X	SA
Audio_recording_type	SysA	SysA		SysA			SA
Category	User	SysA		SysA		X	WFE/SA
Channel							inherited
Classification	SysA	SysA		SysA		X	SA
Country	SysA	SysA	SysA			X	SA
Department	User	SysA		SysA		X	WFE/SA
Intention		SysA		SysA		X	SA
Language	SysA	SysA	SysA			X	SA
Location	SysA	SysA		SysA		X	SA
Media_codec	Sys	Sys	Sys			X	SP
Mob_subtype	SysA	SysA		SysA		X	SA
Mob_type	Sys	Sys	Sys			X	SP
Name	User	User	User		SysA	X	WFE/SA
Object_type	Sys	Sys	Sys			X	SP
Office	User	SysA		SysA		X	WFE/SA
Physical_format	SysA	SysA		SysA		X	SA
Portal							inherited
Publisher	User	SysA		SysA		X	WFE/SA
Registration_level	SysA	SysA		SysA		X	SA
Rightscode	SysA	SysA		SysA		X	SA
Role	SysA	SysA	SysA			X	SA
Subject							
- Issue	User	SysA	SysA		SysA	X	WFE/SA
- DK5	User	SysA	SysA		SysA	X	WFE/SA
- DR Thesaurus	SysA	SysA	SysA		SysA	X	SA (DBA script)
Targetgroup		SysA		SysA		X	WFE/SA
Tx_type		SysA		SysA		X	SA

8. Search

This section provides specifications for search and sort operations of information described by the DR Metadata standard. These specifications consist of:

- Search targets - type of information to search for
- Search criteria - specifies what to search for using free text or field data
- Search filters - limit the amount of search results by specifying additional criteria
- Sorting criteria - output order for search results

This purpose of this section is to provide support for system designers and implementers using the DR Metadata standard.

8.1. Search Targets

The DR Metadata standard supports four distinct search targets.

- Production
- Items
- Programs
- Publication

Implementations using the DR Metadata standard should support search operations for each relevant target managed by the implemented system. These search function should provide free text and parametric search using the search criteria and filtering criteria specified in this chapter.

8.2. Primary search criteria

Production	Item	Program	Publication	in-index	free text	Comment
prd_title, prd_title_alt	itm_title, itm_title_alt	pro_title ruo_title	pub_title, pub_title_alt	yes	yes	title
-	itm_create_ timestamp, mob_create_ timestamp	ppu_start_ timestamp_2 (default ppu_start_ timestamp_1)	puu_start_ timestamp	yes	no	timestamp Date and Time for publishing (creation). Interval search shall be possible.
-	-	ppu_channel, sic_channel	-	yes	no	channel Chose from list.
-	-	-	puu_portal	yes	no	portal Chose from list
sub_keyword type=T or U	sub_keyword type=T or U	sub_keyword type=T or U	sub_keyword type=T or U	yes	yes	subject 1 Controlled vocabulary.
sub_keyword type=D	sub_keyword type=D	sub_keyword type=D	sub_keyword type=D	yes	no	subject 2
prd_category	itm_category	pro_category	pub_category	yes	no	subject 3 Controlled vocabulary.
prd_ targetgroup	prd_ targetgroup	pro_ targetgroup	pub_ targetgroup	yes	no	subject 4 Controlled vocabulary.

prd_intention	prd_intention	pro_intention	pub_intention	yes	no	subject 5 Controlled vocabulary.
prd_department		pro_department	pub_department	yes	no	department Controlled vocabulary.
prd_description	itm_description kef_description	pro_description	pub_description	no	yes	description 1
	itm_archive_comment	pro_archive_comment	pub_archive_comment	no	yes	comment
	mos_isindexed			no	yes	description 3 MOB types where mot_name=TEXT and mos_isIndexed is set are intended to be indexed.
	item_coverage_spatial			no	yes	coverage (spatial) Controlled vocabulary
	itm_location			no	yes	location
prd_number	itm_drams_id, mob_smpte_umid	pro_drams_id	pub_drams_id	yes	no	identifier
nam_name where rol_iscreator	nam_name where rol_iscreator	nam_name where rol_iscreator	nam_name where rol_iscreator	yes	yes	creator
nam_name where not rol_iscreator	nam_name where not rol_iscreator	nam_name where not rol_iscreator	nam_name where not rol_iscreator	yes	yes	contributor
nam_char-acteristics	nam_char-acteristics	pro_char-acteristics	pub_char-acteristics	yes	yes	name comment
	mob_char-acteristics			no	yes	MOB comment
	itm_publisher			yes	yes	publisher
	mog_duration			yes	no	format (extent)

8.3.Search filters

Production	Item (Element)	Program	Publication	Comment
------------	-------------------	---------	-------------	---------

prd_type converted to a code	prd_type converted to a code if item is related to production	pro_istv converted to a code	code for publication	code for Type of material Controlled vocabulary based on prd_type.
	min_fmt_bitrate			bitrate range
	min_fmt-size			filesize
	mob_type			objectType Controlled vocabulary.
	mob_isready_review			is ready for review
	mob_isready			is ready for Broadcast
	if mob_location_url exist			isOnline
	itm_language (primary language if more)			language Controlled vocabulary.
	rights_code based on relation			rights Controlled vocabulary.
	min_physical_format			format Controlled vocabulary.
	rle_code	rle_code	rle_code	registration level Controlled vocabulary.
	mog_classification			Classification

Search Engine

In the digital archive the search engine is in the center of the digital workflow - the search engine will be used not only for research purpose - but also for workflow management.

The search engine shall work with Danish character set and support search neutrality for special characters

The search engine shall provide the basic facilities for searching in the metadata and in the course of time new facilities for searching in 'essence' - audio and video.

The basic search facilities are:

- free-text search in a single field as well as in several fields at the same time
- index search
- masking and left/right truncation and simultaneous use of these options
- Boolean and proximity operators

- interval search
- thesaurus search
- usage of 'search history' where the search results are displayed and can be edited and reused ad hoc in new searches
- usage of 'search profile' which are a predefined search used for continuous or ad hoc monitoring e.g. a particularly subject area.

In addition to the above list there will be a demand for more advanced search functionality within special areas and furthermore the search engine must have the ability to evolve in line with new developments within the area of search facilities such as search for sounds, image recognition etc.

The search engine will be used in connection with different work tasks and by different categories of employees. For that reason the search engine must have flexible user interface as well as assistance and guidance functionality.

It shall be possible to navigate between the main targets Production, Program, Publication, Item and person without initiating a new search and in general to navigate using cross references between all entities.

If the search target is an Item it should easily be possible to navigate directly to matching Media Object or video keyframe. E.g. if a search hit on a description for a key frame in a video the user should directly be navigated to the key frame.

9. Standards for content exchange in general.

9.1. Information Exchange Contexts

Three distinct contexts for information exchange have been identified in the DR asset management execution environment.

- System-to-System exchange (S2S) between DR systems.
- Business-to-Business exchange (B2B) between DR and business parties.
- Business-to-Consumer exchange (B2C), which refers to viewers, listeners and readers.

All data exchange will be conducted using relevant standards and wrapped in XML containers where applicable. The following general guidelines for exchange apply.

S2S exchange is based on DR specified exchange schemes.

B2B exchange is based on international exchange standards as specified below.

B2C exchange is currently not addressed in this edition of the DR Metadata standard.

10. System-to-System Exchange

Efficient exchange between DR systems is the major enabler of new applications and use cases. Most system-to-system exchange will be automatic and unattended. To support a growing number of applications while maintaining a consistent data set, each exchange point must perform careful validation of the delivered information.

10.1. Systems Involved

The following system categories are subject to S2S exchange.

- DR production planning system
- DR transmission planning system
- DR scheduling systems (NCS, CAR etc.)
- DR playout and publishing systems
- DR editing systems for postproduction (NLE, audio, graphics etc.)

10.2. Scope

The system-to-system exchange specifications cover interfaces and high-level operational requirements in an import/export context. However, the intention with specifications of import/export operations is to indicate the required level of services to support the specified exchange. How this is achieved is not in scope for the DR Metadata standard. However, it is anticipated that metadata exchange will be conducted in XML format using Web Services when appropriate in order to provide an open interface to DRAMS systems.

10.3. Operations

S2S exchange consists of the following high-level operations:

- **Import** (SaveService) of individual entities or entity hierarchies using unattended quality control mechanisms.
- **Export** (GetService) of individual entities or entity hierarchies. The export operations may either be activated from the DRAMS system dialog or as a result of an external request (GetService), or it could be a system-initiated publication targeted to other systems subscribing to the particular information.
- **Subscription** to events triggering import and in some cases, export operations.

Note that exchange of essence as part of the operations above should be performed *by reference*. It is up to the implementation of each individual system to develop efficient methods to transfer essence files where applicable.

10.4. Exchange and Interface Specifications in general

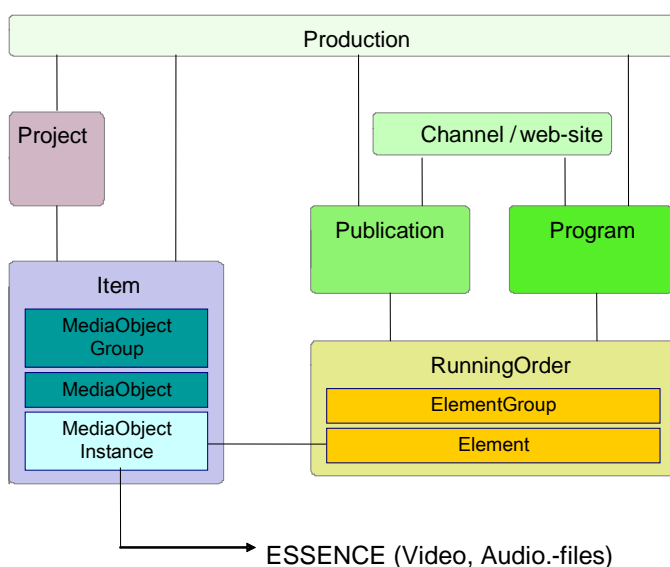
This section contains requirements and interfaces for import, subscribe and export operations for the complex data types in the DR Metadata standard that are subject to S2S exchange. Complex data types that can be exchanged individually or as master nodes in hierarchies. For example, exchange of a Program including one or more RunningOrders.

10.5. Metadata and perspectives for information exchange

An Item will include metadata and essence file of the type video, audio, graphic (artificially-produced objects), stills (photographs), data (subtitles for programmes) and text (ASCII texts such as menus, intros etc.). Examples include: a short sound clip, MHP application, news script or an episode of a drama programme.

Items can be associated with a project. A Project can be compared with a container for all raw recordings and new fully produced transmission files.

The related metadata will be in the form of Production and Programme data, including any detailed information on the programme. Detailed information is represented as a RunningOrder. The Media Archive will eventually contain information on DR's publications on the web, in print etc.



10.5.1. Item

An Item is primarily identified by its unique id (itm_drams_id) which also includes a name for the actual instance of DRAMS (*see specification for DRAMS instance in appendix 1*). This is important in an environment where there exist more instances of the DRAMS system. An Item may also be unique identified by the SMPTE UMID for the Media Object (mob_smpte_umid) or a specific instance of a Media Object (min_drams_id).

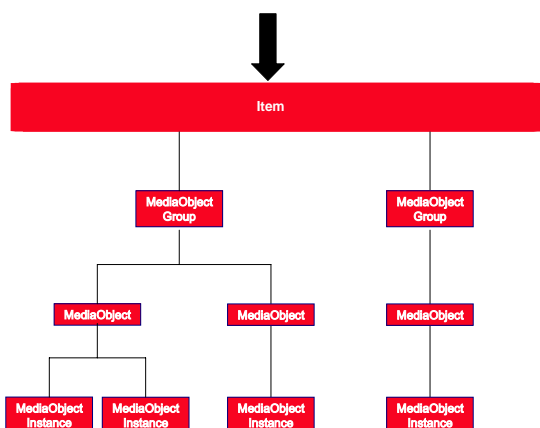
Depending on the actual use case you may focus on the whole Item structure or just a part of it. It makes sense to view the Item structure from the different perspectives listed above. The special case where an Item include exactly one Media Object Group, one Media Object and one Media Object Instance the different perspectives would include the same metadata.

Metadata can be seen from the following three perspectives ITEM, MOG and MOB-INSTANCE.

ITEM utilises metadata seen from the ITEM perspective (top - down).

The Item target is mainly used for material exchange between different instances of DRAMS systems or DRAMS-compliant systems.

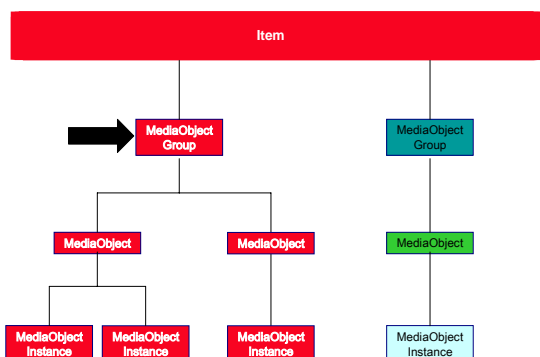
Identifiers: itm_drams_id is the identifier but mob_smpte_umid or min_drams_id may also be used as unique identifiers.



MOG utilises metadata seen from a MOG perspective (bottom - up).

This target could be useful when a whole 'package' of materials describing a TV program with subtitles, graphics etc are exchanged.

Identifiers: min_drams_id or mob_smpte_umid are unique identifiers but also a combination of itm_drams_id, mog_number and mog_version may be used.

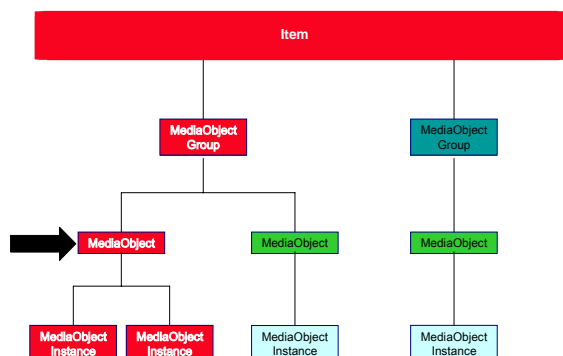


MOB-INSTANCE utilises metadata seen from a MOB-Instance perspective (bottom - up).

This perspective is actually seen from a Media Object (MOB) view, but because import/export usually involves only one instance (essence file) with the integration, it is called MOB-INSTANCE.

The essence file exported/imported is situation-dependent and the user will not have to be concerned with Media Object Instance level as far as possible.

Identifiers: `min_drams_id` is the unique identifier but a combination of `mob_smpte_umid` and additional data like coding format (`min_fmt_codec`) or bit rate (`min_fmt_bitrate`) may also be used.



Item data include all metadata referenced by the complex data types *Item* and *Name and Role* in accordance with the DR Metadata standard.

- Item perspective
- MOG perspective
- MOB-Instance perspective

Additional fields for Item.

Attribute name	Comment	Mand.
source_system	Identifier for the sending system. e.g. 'DR/MediaArchive' where 'MediaArchive' is the name of the actual DRAMS instance in DR	Y
item_perspective	'A', 'B' or 'C' for Item, MOG or MOB-Instance perspective	Y
prd_number	Reference to a Production in a DRAMS system	N

10.5.2.Project

Project uses metadata represented in all entities related to the complex types *Project_and_folder*, *Item* and *Name and Role* in accordance with the DR Metadata standard. Metadata related to items is seen in a MOB-Instance perspective.

Identifiers: `prj_drams_id` is the unique identifier.

Additional fields for the Project.

Attribute name	Comment	Mand.
source_system	Identifier for the sending system. e.g. 'DR/MediaArchive' where 'MediaArchive' is the name of the actual DRAMS instance in DR	Y
prd_number	Reference to a Production in a DRAMS system	N

A sequence of referenced Items (MOB-Instance perspective).

Attribute name	Comment	Mand.
source_system	Identifier for the sending system. e.g. 'DR/MediaArchive' where 'MediaArchive' is the name of the actual DRAMS instance in DR	Y
item_perspective	'C' for MOB-Instance perspective	Y
prd_number	Reference to a Production in a DRAMS system	N

10.5.3. RunningOrder

RunningOrder uses metadata represented in all entities related to the complex types *RunningOrder* and *Name and Role* in accordance with the DR Metadata standard.

Identifiers: ruo_drams_id is the unique identifier.

A RunningOrder may include a sequence of associated Items seen in a MOB-Instance perspective.

- RunningOrder
- RunningOrder + MOB-Instances

Additional fields for RunningOrder.

Attribute name	Comment	Mand.
source_system	Identifier for the sending system. e.g. 'DR/MediaArchive' where 'MediaArchive' is the name of the actual DRAMS instance in DR	Y
pro_drams_id	The unique id for a specific DRAMS Program	N
ppu_source_url	The unique id for a specific publication of a Program - timeslot on the transmission plan	Y
pub_drams_id	The unique id for a specific DRAMS Publication	N
pub_source_url	The unique id for a specific publication of a Publication - timeslot on the internet/intranet	Y

10.5.4. Production

Production uses metadata represented in all entities related to the complex type *Production* in accordance with the DR Metadata standard.

Identifiers: prd_number is the unique identifier.

Additional fields for Production.

Attribute name	Comment	Mand.
source_system	Identifier for the sending system. e.g. 'DR/MediaArchive' where 'MediaArchive' is the name of the actual DRAMS instance in DR	Y

10.5.5.Program

Program uses metadata represented in all entities related to the complex types *Program*, *Name and Role*, *RunningOrder* and *Item* in accordance with the DR Metadata standard. Metadata related to items is seen in a MOB-INSTANCE perspective.

Identifiers: *pro_drams_id* is the unique identifier but also *ppu_source_url* will do.

A DRAMS Program is identified by its unique id (*pro_drams_id*). The unique identifier for a publication of a Program is the Unique TX Slot ID, created in the DR transmission planning system. In the absence of a DRAMS Transmission Number, either the Unique TX Slot ID (*se specification for the field ppu_source_url in the data dictionary*) or the three-tuple (Date,Start Time,Channel) may be used as an alternative method to identify program data.

The interface is divided in two

- Program interface (for program information related to planning and exchange)
- Playout interface (for additional program information related to playout)

The Playout interface only include few basic metadata described below

Program interface

The interface consists of all metadata in the complex data types *Program* and *Name and Role* plus the following additional fields. A Program may include a sequence of RunningOrders and associated Items seen in a MOB-Instance perspective.

- Program
- Program + RunningOrders
- Program + RunningOrders + MOB-Instances

Additional fields for Program.

Attribute name	Comment	Mand.
source_system	Identifier for the sending system. e.g. 'DR/MediaArchive' where 'MediaArchive' is the name of the actual DRAMS instance in DR	Y
prd_number	Reference to a Production in a DRAMS system	N

A sequence of referenced Publications.

Attribute name	Comment	Mand.
pub_drams_id	The unique id for a specific DRAMS Publication	Y

Playout Interface

The interface consists only of the following fields.

Attribute name	Comment	Mand.
----------------	---------	-------

source_system	Identifier for the sending system. e.g. 'DR/MediaArchive' where 'MediaArchive' is the name of the actual DRAMS instance in DR	Y
unique_tx_slot_id	The unique id for a specific timeslot on the transmission plan (ppu_source_url)	Y
channel	Transmission channel or publishing service	Y
start_timestamp	Date and time for start of the program	Y
stop_timestamp	Date and time for end of the program	Y

10.5.6.Publication

Publication utilises metadata represented in all entities related to the complex types *Publication* and *Name and Role* in accordance with the DR Metadata standard.

Identifiers: pub_drams_id is the unique identifier but also pub_source_url will do.

A Publication may include a sequence of RunningOrders and associated Items seen in a MOB-Instance perspective.

- Publication
- Publication + RunningOrders
- Publication + RunningOrders + MOB-Instances

Additional fields for Publication.

Attribute name	Comment	Mand.
source_system	Identifier for the sending system. e.g. 'DR/MediaArchive' where 'MediaArchive' is the name of the actual DRAMS instance in DR	Y
prd_number	Reference to a Production in a DRAMS system	N

A sequence of referenced Programs.

Attribute name	Comment	Mand.
pro_drams_id	The unique id for a specific DRAMS Program	Y

Note that entities subject to change can be exchanged more than once. For example, a Program entity may be exported every time a new Program Publication event (Pro_Publish) has occurred and a new child entity has been created.

Formal Specifications

The formal specification for all complex data type is provided as an XML schema, which is listed in *Appendix 2, Formal specifications for System-to-System Exchange*.

10.6.Methods for integration

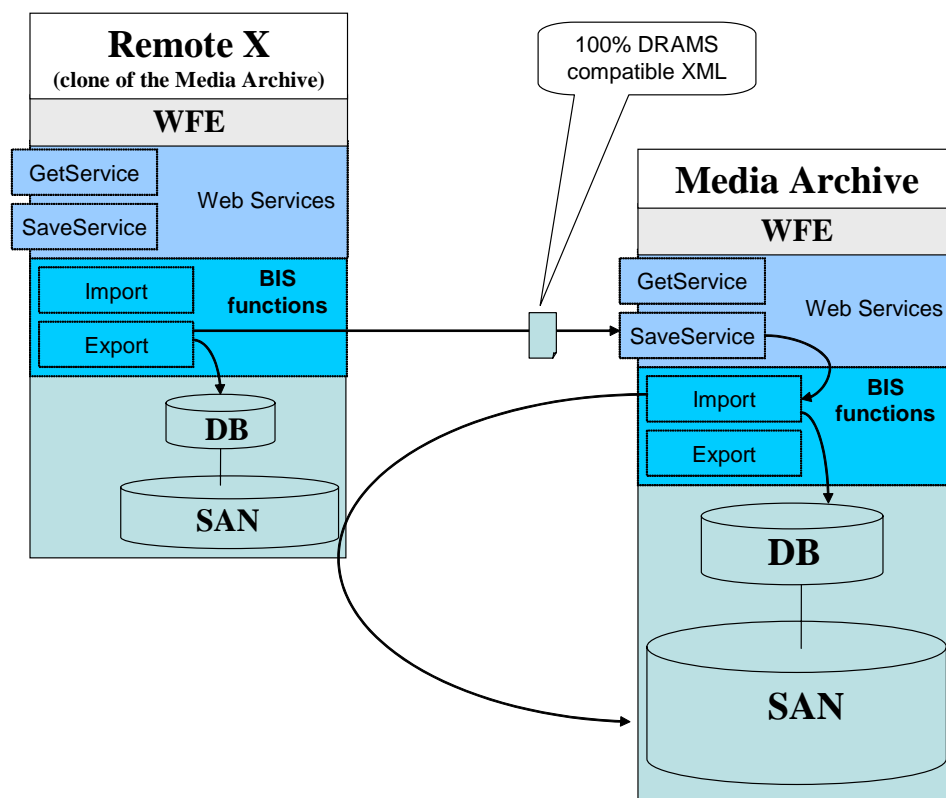
10.6.1.WebServices (Save- and GetService)

These integrations are based on the idea of Service Oriented Architecture (SOA), and the physical implementation is facilitated by WebServices.

10.6.2.SaveService

The SaveService is a Web Service that can be called up directly or indirectly from the SaveDialog dialogue.

SaveService can be called up directly from systems that are 100% DRAMS compatible, i.e. those that are genuine clones of the DRAMS system. But they can also be systems that have based their data model on the DR Metadata Standard. SaveDialog will always conclude by a call from SaveService. SaveService supports in principle all complex data types (business objects).



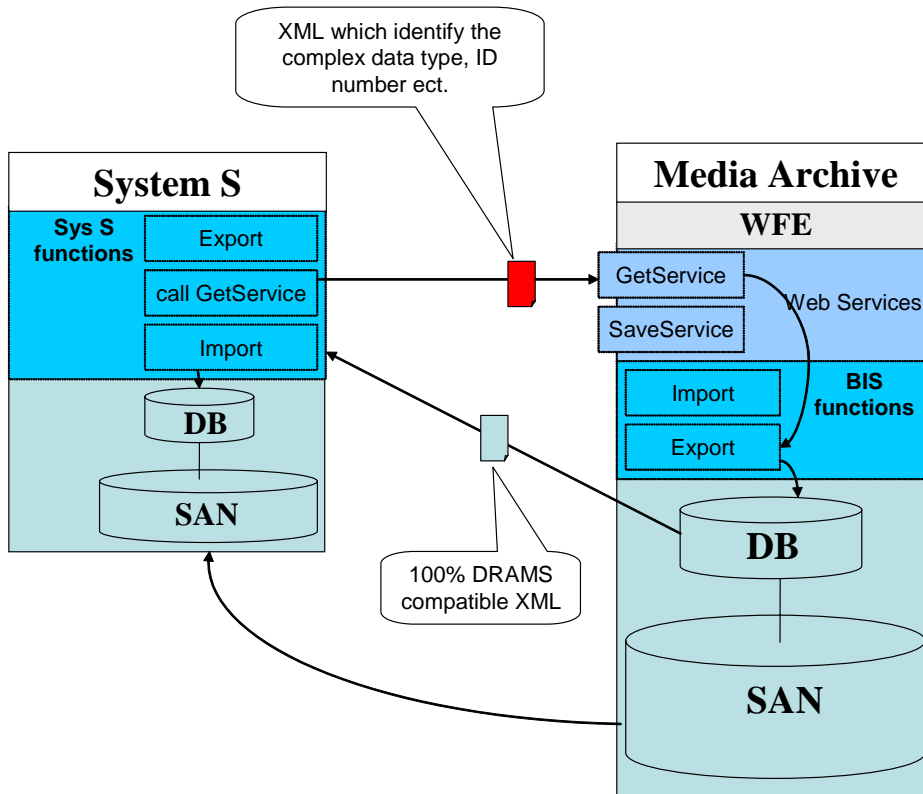
Metadata only receives XML forms which are 100 % DRAMS compatible. XML data does not contain essence, but can contain locators for essence.

SaveService must be able to activate a TransferAgent that can import copies of essence from an external location and copy essence to the DRAMS system.

SaveService supports in principle all complex data types defined in the chapter 'Matrix for information exchange'.

10.6.3. GetService

The DRAMS GetService is a Web Service that can be called up direct from other systems. GetService facilitates the export of metadata and essence and supports in principal all complex data types. Calling GetService up requires a key (DRAMS_ID, Production number or SMPTE_UMID).



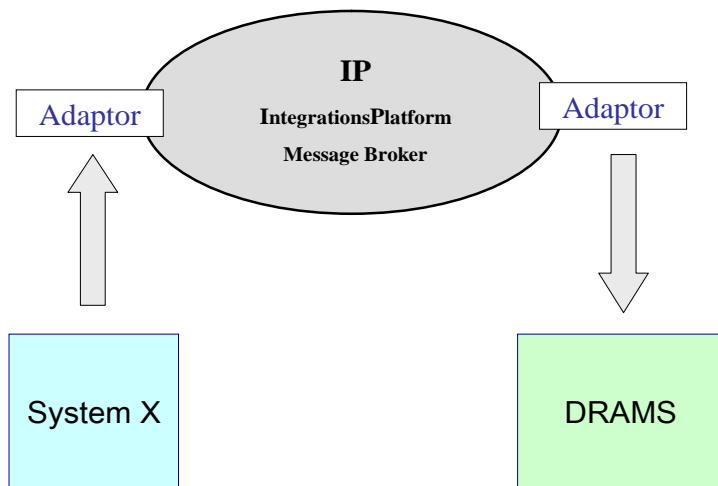
Metadata is received and delivered in an XML format, and is supplied in 100 % DRAMS compatible XML. There may however be reasons for the supply of XML formats aimed at particular types of recipients, such as editing systems. XML data does not contain essence, but may contain a UNC to essence.

GetService must be able to activate the TransferAgent that can supply a copy of one or more essence files from the Media Archive to a specified external location and in addition the related metadata. The essence copy shall be deliverable in full or part (partial retrieve) and the copy shall be deliverable in the coding format requested.

GetService supports all complex data types defined in the chapter 'Matrix for information exchange'.

10.6.4. Import/export via Message Broker

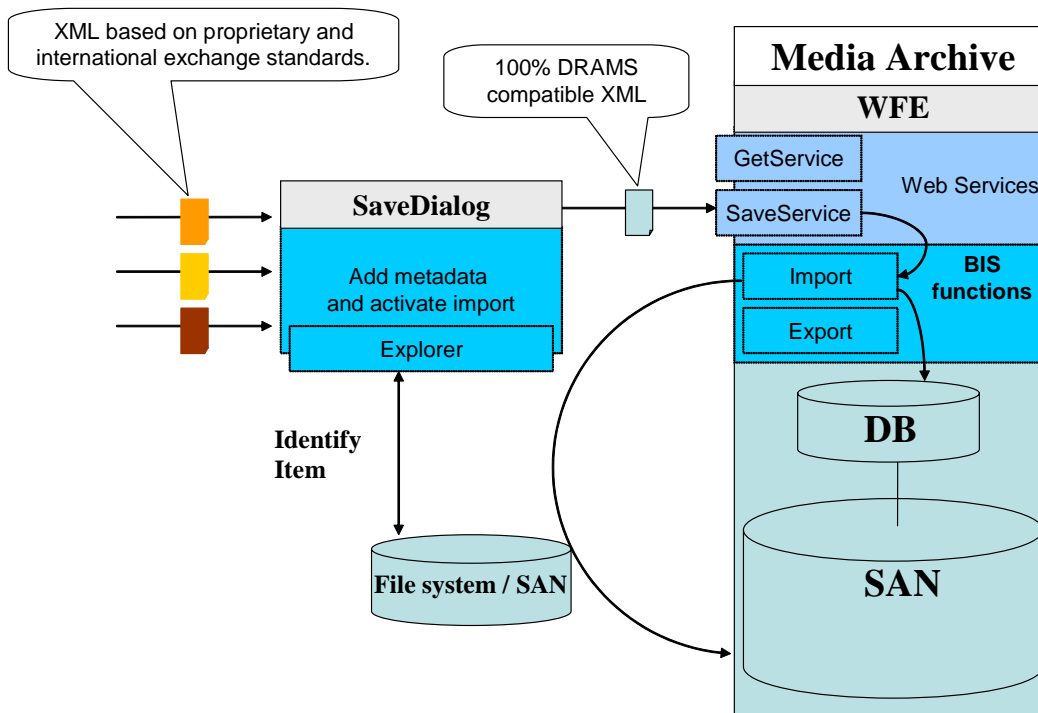
These integrations are based on the idea of Message Oriented Middleware (MOM), where a Message Broker functions as middleware (Message Broker).



For integrations to the DRAMS system, the system will subscribe to information from other systems and deliver it in a queue. There will thus be an import queue for each of the covered integrations and defined complex data types. An import may be concluded by a call to SaveService.

10.6.5.Import via SaveDialog

SaveDialog supports import of Items, Projects incl. Items and Running Order with or without items. This component also facilitates import of an Item from any source (memory stick, server drive etc.) accessible via Windows explorer. The user has to be able to activate the web-based SaveDialog from any standard PC and to save items in the DRAMS system without having to open it first.



Well-defined XML schemes (proprietary and 100% DRAMS compatible) are always to be used when SaveDialog is called up, but it should also be activate able without any accompanying XML scheme. If the Dialog is activated without an XML scheme, it is because the user wants to transfer one or more Items from a source that can be identified via explorer.

SaveDialog is concluded by a call from SaveService, and therefore, all communication between SaveDialog and SaveService is to be based on 100 % DRAMS compatible XML schemes.

XML data does not contain essence, but can contain locators for essence.

Header information

The Item, Project or RunningOrder to be saved are to be shown and selected in the top of the dialog box. The relevant information will be shown according to selection. The list below shows the necessary information:

Items: Display of production number

- Production number can be stated and is to be validated
- Whether Items are to be saved in a private folder can be stated

Project: Display of project title and production number.

- A project can be changed to a folder
- If the production number is not stated, it must be entered and validated

RunningOrder:

Display of title of RunningOrder and an production number/program title

- It shall be possible to link a RunningOrder to an existing Program via production number and publication date

- It shall be possible to enter RunningOrder and a list of RunningOrders for a given Program
- It shall be possible to see a list of all related Items (list information)

Material list

The following main data shall be amendable or added in SaveDialog:

- Marking of new Items (metadata and essence) to be saved
- Marking of in-/out time codes if only a part of essence is to be saved (partial retrieve)
- Marking of Items if they are to inherit a production number
(will only apply to Items not already linked to a production number)
- New Items shall be saved as follows:
 - New Item (default)
 - New MOG
 - New MOG version
- Known Items shall be saved as follows:
 - New MOG version

It shall also be possible to enter only selected data such as title, description etc.

Media Archive fields	New Item	New MOG/New MOG version
itm_title	m	
itm_description	o	o (update)
itm_publisher	o	
itm_department	o	
itm_rights_code	m	
itm_rights_description	o	
itm_archive_comment	o	
itm_technical_comment	o	o (update)
itm_location	o	o (update)
itm_killdate	m	
itm_email	o	o
ars_code	o	
Sequence of Name&role	o	o (update)
mog_name	o	o
mog_classification	m	m
mob_create_timestamp	o	o
mob_ready_review	o	o
mob_ready	o	o

mob_physical_location_id	o	o
mob_source_asp_id	o	o
mob_source_art_id	o	o

10.6.6.Export

It shall be possible to export metadata and essence via functions in the DRAMS system. This function should use the same methods as included in the GetService. Export function supports in principle all complex data types (business objects).

10.7.Overview for information exchange

Item perspectives

- Item perspective
- MOG perspective
- MOB-Instance perspective

Project

- Project + MOB-Instances

RunningOrder

- RunningOrder
- RunningOrder + MOB-Instances

Production

- Production

Program

- Program
- Program + RunningOrders
- Program + RunningOrders + MOB-Instances

Publication

- Publication
- Publication + RunningOrders
- Publication + RunningOrders + MOB-Instances

10.8.Child Entities

This section contains interface specifications for those 'child' entities where additional attributes exist. These entities may only be exchanged as part of a complex data type. Refer to the complex data type diagrams in Chapter 4 for an overview of the hierarchies implied by the data model.

The exchange interfaces for these entities are generally identical to the entity data dictionaries, although exceptions like renormalizations and additional attributes exist.

10.8.1.Country

This entity may only be exported or imported as a child related to an Item entity. The interface specification covers all attributes in the entity. Refer to the corresponding entry in the entity dictionary for the full set of attributes that constitute the import/export interface.

Related Entities

None

10.8.2.Element

This entity may only be exported or imported as a child entity to the complex data type Running Order. The interface specification covers all attributes in the entity plus a number of additional attributed listed below. Refer to the corresponding entry in the entity dictionary for the full set of attributes that constitute the base of import/export interface for this entity.

Additional Attributes

Attribute name	Comment	Mand.
min_drams_id	Unique identifier for a media object instance in DRAM	S
min_location_url	The locator for an internal essence file	N
external_essence_url	The locator for an external essence file	N

Related Entities

None

10.8.3.Language

This entity may only be exported or imported as a child related to an Item entity. The interface specification covers all attributes in the entity. Refer to the corresponding entry in the entity dictionary for the full set of attributes that constitute the import/export interface.

Related Entities

None

10.8.4.MOB Type

This entity may only be exported or imported as a child related to a Media Object entity. The interface specification covers all attributes in the entity. Refer to the corresponding entry in the entity dictionary for the full set of attributes that constitute the import/export interface.

Related Entities

MOB Subtype

10.8.5.MOB Subtype

This entity may only be exported or imported as a child related to a MOB Type entity. The interface specification covers all attributes in the entity. Refer to the corresponding entry in the entity dictionary for the full set of attributes that constitute the import/export interface.

Related Entities

None

10.8.6.Name

This entity may only be exported or imported as a child related to an Item, Program, Production, Publication or RunningOrder. The interface specification covers all attributes in the entity. Refer to the corresponding entry in the entity dictionary for the full set of attributes that constitute the import/export interface.

Related Entities

Role

10.8.7.Rights Code

This entity may only be exported or imported as a child or value related to an Item entity. The interface specification covers all attributes in the entity. Refer to the corresponding entry in the entity dictionary for the full set of attributes that constitute the import/export interface.

Related Entities

None

10.8.8.Role

This entity may only be exported or imported as a child related to an Item, Program, Production, Publication or RunningOrder. The interface specification covers all attributes in the entity. Refer to the corresponding entry in the entity dictionary for the full set of attributes that constitute the import/export interface.

Related Entities

Name

10.8.9.Subject

This entity may only be exported or imported as a child or value related to an Item, Program, Production or Publication entity. The interface specification covers all attributes in the entity. Refer to the corresponding entry in the entity dictionary for the full set of attributes that constitute the import/export interface.

Related Entities

None

11. Business-to-Business Exchange

Business-to-business (B2B) exchange will be implemented as a set of import and export functions for metadata and essence. Each exchangeable object is described using one dedicated XML schema and associated XML file for metadata. Essence associated with an object is exchanged as physical files referenced from within the XML document.

Metadata objects will be exchanged as stand-alone XML files with or without essence. Exchange of object hierarchies should be implemented using the support of external facilities such as XML wrapper documents or directory structures.

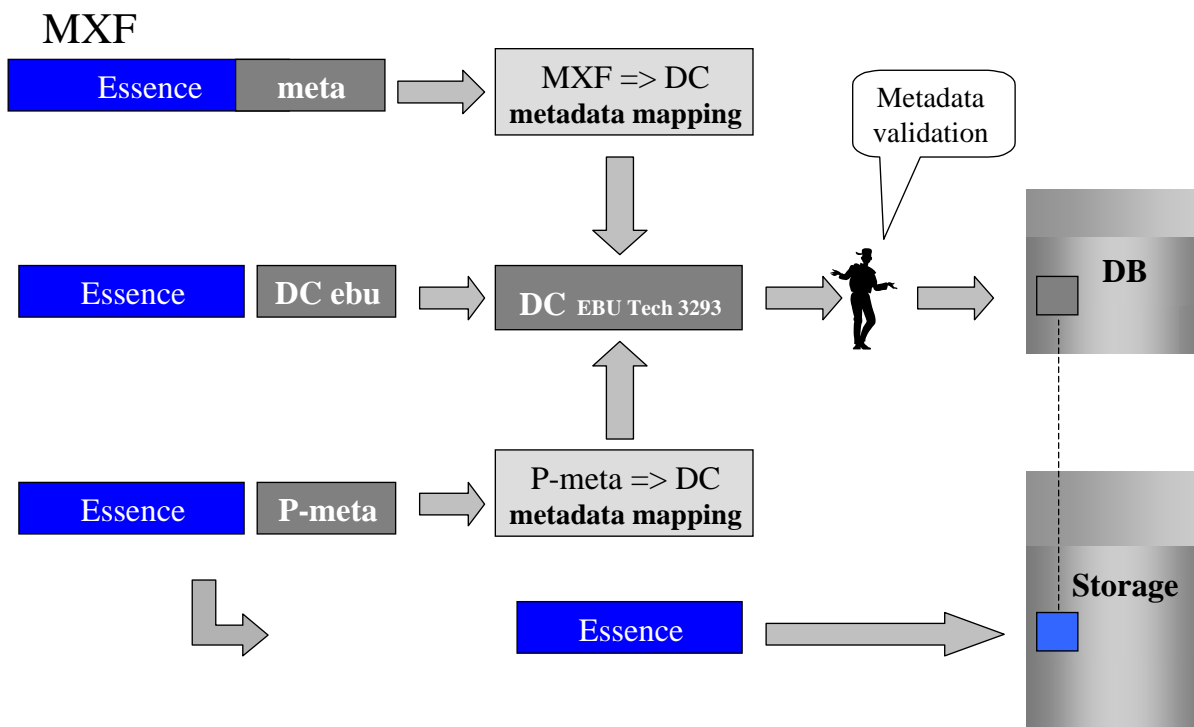
B2B import is restricted to Item (mob_instance perspective) and export is restricted to Item (mob_instance perspective), Program and Publication.

Export from DRAMS can be automatic and unattended as well as manual and request-based. One XML schema for each target plus a set of referenced essence will be used to avoid conflicts when supporting a potentially large set of export formats.

All importing from external parties must be validated and corrected to ensure the quality of the inserted information. This is likely to be a manual procedure for a long period of time. Import using new and existing exchange standards and formats will be supported through the use of adapters that provide mappings between particular standards and the internal DR exchange format, which is primarily based on Dublin Core EBU Tech 3293.

The figure below illustrates the use of adapters, mappings and validation for content import.

Content import (B2B) using adapters, mapping and validation



Validation and approval of imported metadata and essence should be supported by DRAMS using a staging area or equivalent where imported material can be validated before it is made available in the central repository.

Essence stored in formats that carry metadata (such as MXF) will be preserved without modifications. Wherever there is a conflict between extracted metadata and embedded, the extracted (and processed) metadata will have priority.

At the time of writing, the anticipated adapter implementation sequence is estimated to start with Dublin Core and MXF(dms-1) , later followed by P/META when industry acceptance has been established.

11.1.Use Cases

Import and export of material can be performed to support the following business processes:

- Intake of material produced by external companies.
- Delivery of material to companies supporting IT-based exchange.
- Contribution to material exchange networks outside DR.
- Delivery of program information to related authorities.
- Delivery of publication information to related authorities.

All of these processes are likely to require use of adapters for mapping of Metadata standards to the DR exchange interface.

11.2.Import

Data describing an Item may be exported and imported. Typically, the Item will be associated with a production and zero or more related Items.

The unique identifier for an Item in mob-instance perspective is the mob_smpte_umid.

Imported material may be associated with a DR production number. This association should be performed in the staging area in DRAMS as part of the validation process.

Item descriptions should be de-normalized as part of the import /export process. Only one essence file should be associated with an item at this stage.

The mapping between the exchange format and DRAMS is listed in the table below.

DC to DRAMS conversion table

DC element name	DC qualifier	DRAMS field name	Comment
Title		itm_title	
Title	Alternative	itm_title_alt	
Creator		nam_name	
Creator	Role	rol_role_dc	
Subject		sub_keyword	
Description		itm_description	
Publisher		itm_publisher	
Contributor		nam_name	
Contributor	Role	rol_role_dc	
Date	Created	itm_create_timestamp	
Type			The encoding list for DC.TYPE is still under construction and DR awaits results of the discussion concerning adjustment of the EBU Tech 3293 standard.
Format			Format information may be generated manually or automatically.
Format	Extent	mog_start_timecode mog_stop_timecode	

Format	Medium	min_physical_format min_fmt_mime_type	min_physical_format is used to represent type of medium, tape/disc type etc or file. min_fmt_mime_type should contain the file format for digital content
Identifier			The identifier will used to locate the essence during import. This may be a SmppteUMID or file reference
Source		itm_external_source_id	
Language		lan_code	ISO-639-2
Relation	HasPart	None	This identifier is used to describe relations between Items and Media Object Groups.
Coverage	Spatial	itm_coverage_spatial	DCMI Point / ISO 3166 / DCMI Box / TGN
Rights		itm_rights_code itm_rights_description itm_rights_url	

Formal Specifications

The formal specification for this interface is provided as an DC XML schema, which is listed in *Appendix 3, Formal specifications for Business-to-Business Exchange*.

XML schema name: drams_import_item.xsd

11.3.Export

Business to business export consists of packaging metadata and optional essence into a DR export format container in Dublin Core (EBU Tech 3293) XML-wrapped format. This export format can be delivered as it is or it may be subject to further translations and packaging into other formats such as MXF (DMS-1) and subsequently, P/META.

Export operations will be supported for the entities listed below.

1. Item (mob-instance perspective)

An exported Item contains metadata and optionally includes related essence describing entities and essence files.

2. Program

An exported program consists of metadata for that program. The exported program may also contain a set of Items.

3. Publication

An exported publication consists of metadata for that publication. The exported publication may also contain a set of Items.

The selection of Items to export should be available to users in the export function in DRAMS. For Programs and Publications, the set of selectable Items should be restricted to what is related to the primary export entity. For example, only Items that are referenced in an element in a runningOrder should be presented as export candidates for a Program. Be aware that the export function does not include the actual runningOrder.

11.3.1.Item

An Item is the primary exportable entity and as such, it can contain metadata and essence. The export mappings between DR metadata and Dublin Core (EBU Tech 3293) are listed in the tables below. Note that the Item metadata will be exported once for each related media object and essence file. This is done to ensure that the metadata stays attached to the essence, for example in the case of export to MXF tape.

DRAMS to DC conversion table for Items and related entities:

DRAMS field name	DC element	DC qualifier	Comment
itm_title	Title		
itm_title_alt	Title	Alternative	
nam_name	Creator		rol_iscreator should be TRUE
rol_role_dc	Creator	Role	
sub_keyword	Subject		
itm_description	Description		
itm_publisher	Publisher		
nam_name	Contributor		rol_iscreator should be FALSE
rol_role_dc	Contributor	Role	
itm_create_timestamp	Date	Created	
None	Type		The encoding list for DC.Type is still under construction and DR awaits results of the discussion concerning adjustment of the EBU Tech 3293 standard.
mog_start_timecode mog_stop_timecode	Format	Extent	
min_physical_format min_fmt_mime_type	Format	Medium	
	Identifier		This is either a UMID or a path to the essence file, depending on export medium.
mob_smpte_umid	Source		UMID from Media Object.
lan_code	Language		Language code. There may be zero or more language codes per Item.

prd_type prd_number	Relation	IsPartOf	The first digit in the DR production number decides the type.
itm_coverage_spatial	Coverage	Spatial	
itm_rights_code itm_rights_description itm_rights_url	Rights		

Formal Specifications

The formal specification for this interface is provided as an XML schema, which is listed in *Appendix 3, Formal specifications for Business-to-Business Exchange*.

11.3.2.Program

Data describing a Program are exported as a high-level entity describing the properties of a particular program, including references to items addressed in the programs through its running orders and elements.

The B2B export data structure containing only a Program and associated Items is simplified and does not contain all information stored in DRAMS. Therefore, an exported program cannot be imported back into DRAMS. However, the individual items can be imported.

DRAMS to DC conversion table for Programs and related entities:

DRAMS field name	DC element	DC qualifier	Comment
pro_title	Title		
prd_title_alt	Title	Alternative	Alternative title taken from Production.
nam_name	Creator		rol_iscreator should be TRUE
rol_role_dc	Creator	Role	
sub_keyword	Subject		
pro_description	Description		
DR	Publisher		Will be set to DR.
nam_name	Contributor		rol_iscreator should be FALSE
rol_role_dc	Contributor	Role	
ppu_start_timestamp_1	Date	Issued	This element may be specified once for each publish occasion. If only one date is exported, the first available date should be selected.
None	Type		The encoding list for DC.TYPE is still under construction and DR awaits results of the discussion concerning adjustment of the EBU Tech 3293 standard.

ppu_start_timestamp_2 ppu_stop_timestamp_2 or ppu_start_timestamp_1 ppu_stop_timestamp_1	Format	Extent	Duration calculated from program start and stop times. The actual transmitted duration will be used by default (TIMESTAMP_2) with fallback to planned duration (TIMESTAMP_1).
pro_drams_id	Source		The unique DRAMS id for this program.
prd_type prd_number	Relation	IsPartOf	The first digit in the DR production number decides the type.
Item Description <i>Can be one of :</i> <i>File path</i> <i>(IT-based media)</i> <i>SMPTE UMID</i> <i>(physical media)</i>	Relation	HasPart	Relation to Item that is exported together with this Program. This relation is specified once for each Item that it part of the export set. The contents of this element must uniquely identify the description file for the exported Item, using a file path or SMPTE UMID. The order of the specified relations must reflect the intended running order of the referenced Items.

Formal Specifications

The formal specification for this interface is provided as an XML schema, which is listed in *Appendix 3, Formal specifications for Business-to-Business Exchange*.

11.3.3.Publication

Data describing a Publication is exported as a high-level entity describing the properties of a particular Publication including references to items addressed in the publications through its elements.

The B2B export data structure containing only a Publication and associated Items is simplified and does not contain all information stored in DRAMS. Therefore, an exported publication cannot be imported back into DRAMS. However, the individual items can be imported.

DRAMS to DC conversion table for Publications and related entities:

DRAMS field name	DC element	DC qualifier	Comment
pub_title	Title		
pub_title_alt	Title	Alternative	
nam_name	Creator		rol_iscreator should be TRUE
rol_role_dc	Creator	Role	
sub_keyword	Subject		
pub_description	Description		
None	Publisher		Will be set to DR
nam_name	Contributor		rol_iscreator should be FALSE

rol_role_dc	Contributor	Role	
puu_start_timestamp	Date	Issued	This element may be specified once for each publish occasion. If only one date is presented, the first available date should be selected.
None	Type		The encoding list for DC.TYPE is still under construction and DR awaits results of the discussion concerning adjustment of the EBU Tech 3293 standard.
pub_drams_id	Source		Unique identifier in DRAMS for this Publication.
prd_type prd_number	Relation	IsPartOf	The first digit in the DR production number decides the type.
Item Description <i>Can be one of :</i> <i>File path</i> <i>(IT-based media)</i> <i>SMPTE UMID</i> <i>(physical media)</i>	Relation	HasPart	Relation to Item that is exported together with this Publication. This relation is specified once for each Item that it part of the export set. The contents of this element must uniquely identify the description file for the exported Item, using a file path or SMPTE_UMID.

Formal Specifications

The formal specification for this interface is provided as an XML schema, which is listed in *Appendix 3, Formal specifications for Business-to-Business Exchange*.

12. Business-to-Consumer Exchange

At the time of writing, no requirements for automated B2C exchange have been identified.

Mappings and mechanisms for business-to-consumer (B2C) exchange are likely to be required as soon as DR starts to deliver future interactive- and data distribution services to viewers and users.

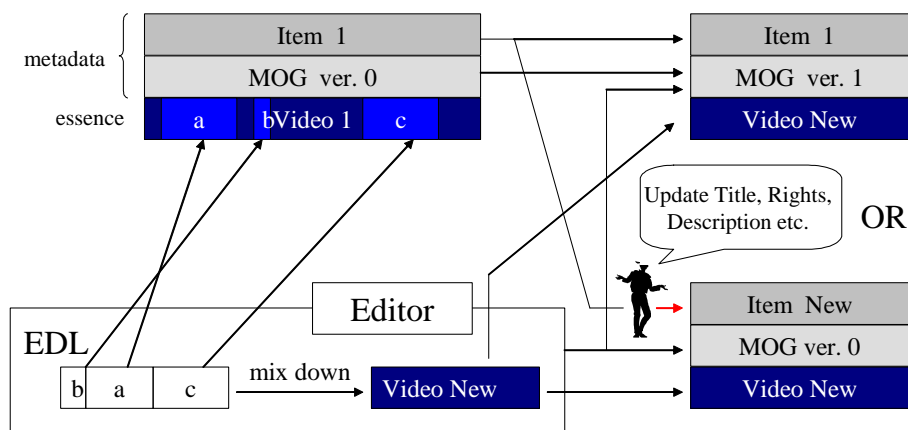
DR should consider B2C exchange and prepare for new data-driven services as part of the strategic program development.

13. Best Practice for use of the DR data model

13.1. Creation of new Item or new MOG version

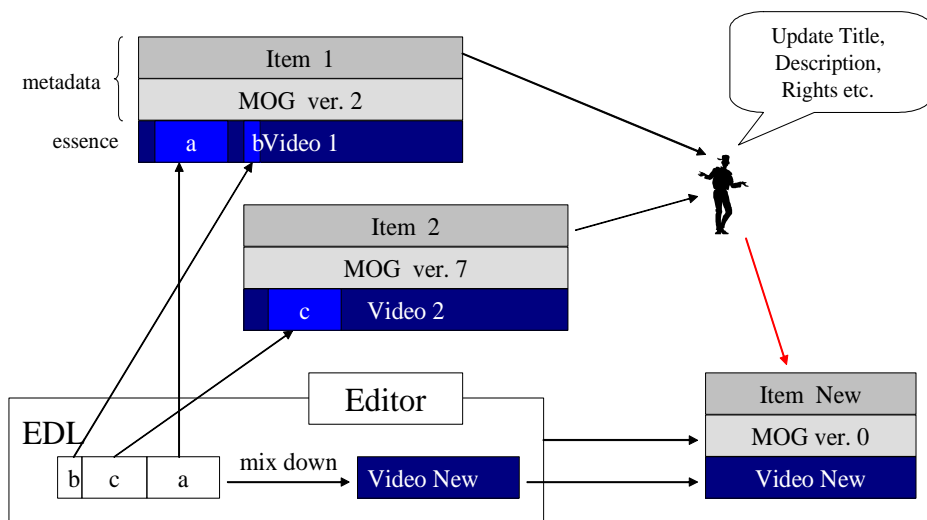
A new Essence instance may be created as an edited version of one existing Essence instance.

Two ways to create new Essence



The new material could either be understood as a new version of the referenced material or as version 0 for a new Item. When a new Item is created a metadata form has manually to be filled in.

Creation of new Essence on basis of two Items



A new Essence instance may also be created as an edited version of two or more existing Essence instances. In this case the new Essence file will implicate the creation of a new Item. The new material will then be understood as a version 0 for a new Item and a metadata form has manually to be filled in. Nevertheless - in some situations it would be more correct to create a new version of an already existing Item and the user should have this choice.

13.2.Guidelines for implementation of version control

The primary goal for implementation of version control is to make it possible to versioning of clips and to make it possible to role back a media production. This will typically include changes to the actual project file (descriptive Essence file) which is associated to the production.

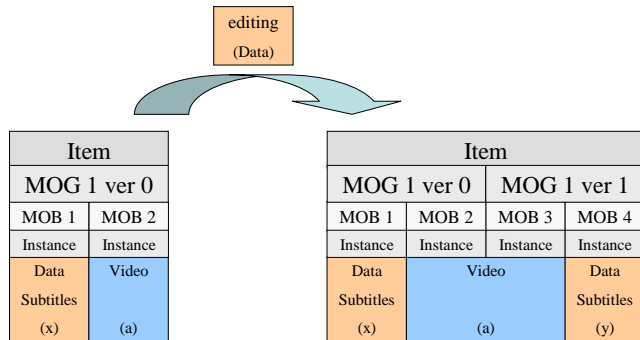
It's also important to minimize the creation of metadata and secure no redundancy of Essence files.

The version control must secure that a MOG including all associated media objects will remain consistent but version control doesn't mean that all previous versions have to be kept. It must be possible to delete previous versions.

13.2.1.Automated versioning of a Media Object Group

Condition: MOB type = video, audio, still, graphic or data

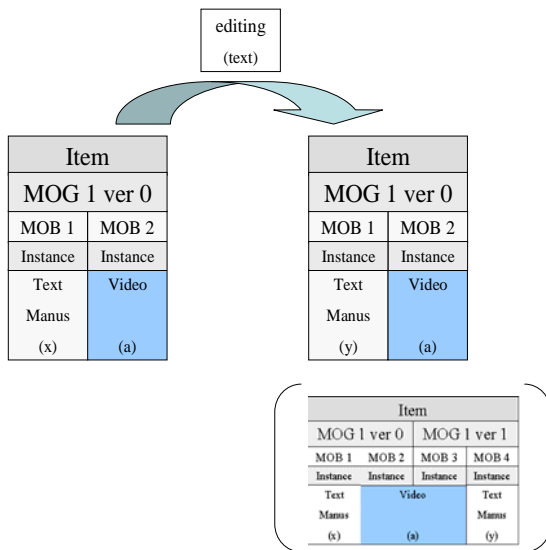
MOB2 and MOB3 and all underlying Instances are initially identical.



13.2.2. Not automated but optional versioning of a Media Object Group

Condition: MOB type = text

Updating of a Text MOB will default not imply the creation of a new version, but implementation should allow the user to choose versioning.



13.3. Asset Tracking.

The data model should be designed to support asset tracking as follows.

When new Item is created on the basis on material related to existing Items.

- A reference (Media_tracking) containing relations between Items (Media Object Groups) should be maintained by all asset manipulation procedures. The parent-child relation can be established between two Items in DRAMS but a parent Item in DRAMS may also refer to an external child Item.
- Essence may be deleted arbitrarily.
- Items for an asset may be deleted when there is no corresponding essence and *there are no remaining references to this asset.*

For changes in metadata, system implementations using can choose to support cascading updates or not depending on data type and feasibility of such cascades. The reference table Media Tracking is used to track an asset and it's descendants.

One benefit with this approach is the ability to choose cascading updates or not depending on feasibility and context. Another benefit is that availability of information remains event in the event of deletion of information in the asset chain (¿broken links¿).

The obvious drawbacks with the ¿copy¿ approach are redundancy, storage requirements and increased complexity for system implementers to determine proper update operations.

However, these drawbacks are considered of minor importance compared to the additional flexibility, especially when it comes to merging of metadata from multiple assets.

13.4.Control of Automatic Updates

Entities subject to automatic as well as manual updates have an attribute indicating that the content is locked for automatic update (pub_islocked, pro_islocked). The recommended use of this attribute is to set the islocked flag whenever a user update occurs.

In addition, this flag may also be set when administrative procedures determine the content is not subject to further automatic updates, such as after grace period expiry or content removal in source systems etc.

13.5.Consolidation of a transmitted Program - prepare for long term archiving

When a program has been transmitted the archivists have a job to do. Information associated to Elements in the RunningOrder may be copied to Item. It's typically non-volatile information such as manuscript, CG texts etc. The reason why is to make the information searchable. The process can not be fully automated.

Many programs are fully or partly live transmissions. The program will be recorded during transmission and the archivist must after transmission copy essence from the transmission log and create new items.

Parts of this process can be automated.

13.6.Integration architecture to support System-to-System exchange

DR's master integration architecture is designed to support a model for integration of business-oriented services, using Service Oriented Architecture (SOA), and Message Oriented Middleware - MOM. The area of focus for the two integration models will be different.

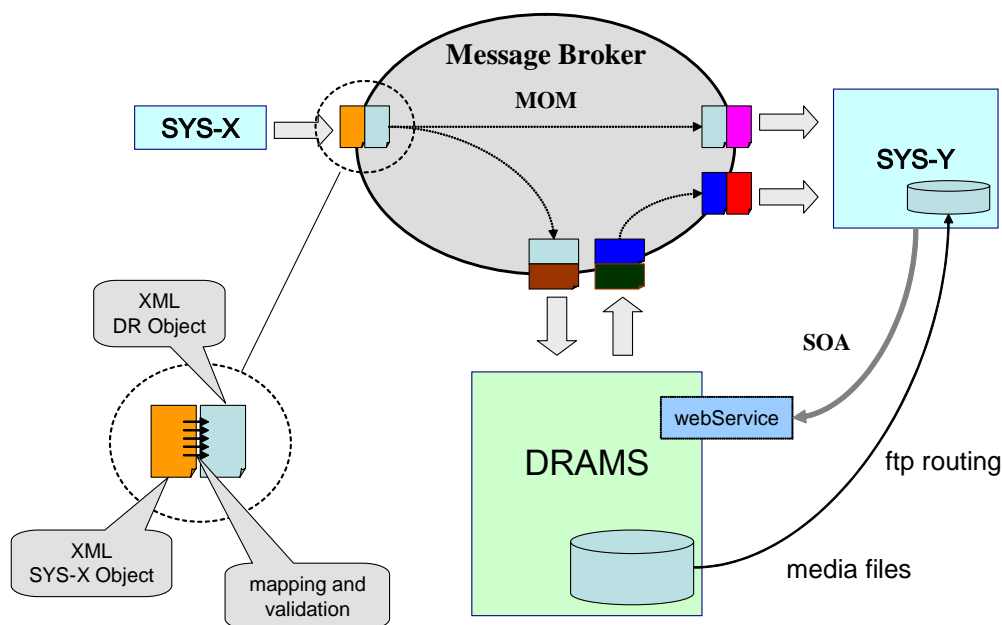
In the MOM model, focus will primarily be on data and business events - i.e. when a business event occurs in a system, such as the changing of a programme broadcast time, the change will be publicised on the integration platform via the MOM model, so that any systems subscribing to this type of event can be advised. The role of a standardized message broker is considered of vital importance to achieve successful integration between multiple systems in the future.

Focus for the SOA is a model for linking applications to each other via a network. The theory behind the model is that developers can perceive applications as services that can be linked to create a coherent business system. For SOA to be an effective model for the integration of business logic, the individual services must use a common, platform-independent form of communication. This means XML-based protocols, including WebServices based on SOAP or XML-RPC.

To ensure relevant and uniform use of the integration architecture, guidelines have been draw up for systems to be introduced at DR. Their intention is to ensure that new systems can:

- Integrate into the integration architecture
- Provide all relevant information and functionality to other systems
- Can make use of existing Superservices
- Comply with DR's requirement for a service oriented architecture

Refer to the figure below for a visualization of the role of the message broker, content management system interfaces and related information exchange contexts.



14. Appendixes

Appendix 1 - Relation between Metadata and Essence

This appendix is provided in the external file *DR Metadata Standard - Appendix 1*

Appendix 2 - Formal specifications for System-to-System Exchange

This appendix is provided in the external file *DR Metadata Standard - Appendix 2*

Appendix 3 - Formal specifications for Business-to-Business Exchange

This appendix is provided in the external file *DR Metadata Standard - Appendix 3*

Contents

1.Introduction	2
1.1.Standard development procedure	2
1.2.Contacts	2
2.Intended Audience	3
3.Scope	3
3.1.In scope	3
3.2.Not in scope	3
3.2.1. Access restrictions and user management	3
3.2.2. System-specific information	4
4.The information model	4
4.1.The main compositions	4
4.2.The Media Object types and subtypes	5
5.Logical models	6
5.1.Item	6
5.2.Name and Role	7
5.3.Production	7
5.4.Program	8
5.5.Project_and_folder	8
5.6.Publication	9
5.7.Rights	9
5.8.RunningOrder	10
6.Data dictionary	11
6.1.archive_status	11
6.2.country	11
6.3.document	12
6.4.document_page	12
6.5.document_type	13
6.6.element	13
6.7.element_group	15
6.8.folder	16
6.9.folder_content	16
6.10.item	17
6.11.language	19
6.12.languagerel	20
6.13.media_object	20
6.14.media_object_group	22
6.15.media_tracking	23
6.16.mob_instance	24
6.17.mob_subtype	26
6.18.mob_type	27
6.19.name	28

6.20.namerel	28
6.21.object_type	29
6.22.pro_publish	30
6.23.production	31
6.24.program	32
6.25.project	34
6.26.project_history	35
6.27.pub_pro_rel	35
6.28.pub_publish	36
6.29.pub_to_pub	36
6.30.publication	37
6.31.registration_level	39
6.32.rights	39
6.33.rights_target	40
6.34.rightscode	41
6.35.role	41
6.36.running_order	42
6.37.simul_channel	44
6.38.subject	44
6.39.subjectrel	45
6.40.timecode_description	45
7.Taxonomies and thesaurus	48
7.1.Reference tables based on scheduled configuration update jobs	48
7.2.Reference tables based on International standards	48
7.3.Reference tables managed by the System administrator	49
7.4.Reference tables based on system provider update	50
7.5.Maintenance for lookup data	50
8.Search	52
8.1.Search Targets	52
8.2.Primary search criteria	52
8.3.Search filters	53
9.Standards for content exchange in general.	55
9.1.Information Exchange Contexts	55
10.System-to-System Exchange	55
10.1.Systems Involved	56
10.2.Scope	56
10.3.Operations	56
10.4.Exchange and Interface Specifications in general	56
10.5.Metadata and perspectives for information exchange	57
10.5.1.Item	57
10.5.2.Project	59
10.5.3. RunningOrder	60
10.5.4.Production	60
10.5.5.Program	61

10.5.6.Publication	62
10.6.Methods for integration	63
10.6.1.WebServices (Save- and GetService)	63
10.6.2.SaveService	63
10.6.3.GetService	64
10.6.4.Import/export via Message Broker	64
10.6.5.Import via SaveDialog	65
10.6.6.Export	68
10.7.Overview for information exchange	68
10.8.Child Entities	68
10.8.1.Country	69
10.8.2.Element	69
10.8.3.Language	69
10.8.4.MOB Type	69
10.8.5.MOB Subtype	70
10.8.6.Name	70
10.8.7.Rights Code	70
10.8.8.Role	70
10.8.9.Subject	70
11.Business-to-Business Exchange	71
11.1.Use Cases	72
11.2.Import	73
11.3.Export	74
11.3.1.Item	75
11.3.2.Program	76
11.3.3.Publication	77
12.Business-to-Consumer Exchange	78
13.Best Practice for use of the DR data model	79
13.1.Creation of new Item or new MOG version	79
13.2.Guidelines for implementation of version control	80
13.2.1.Automated versioning of a Media Object Group	80
13.2.2.Not automated but optional versioning of a Media Object Group	81
13.3.Asset Tracking.	81
13.4.Control of Automatic Updates	82
13.5.Consolidation of a transmitted Program - prepare for long term archiving	82
13.6.Integration architecture to support System-to-System exchange	82
14.Appendixes	83

