DLM FORUM
FOUNDATION

MoReq2010®

MoReq2010®
Modular Requirements for Records Systems

# Volume 1
# Core Services & Plug-in Modules

Version 1.0

# Table of Contents

# Table of Figures


# PART ONE – CORE SERVICES

# 1. Fundamentals

## 1.1 Important Information

### 1.1.1 Intellectual property rights

The MoReq2010® specification is copyright © DLM Forum Foundation, 2010 & 2011, all rights reserved, including all text and original illustrations included with the work.

Some illustrations make use of royalty free clip art sourced from Microsoft Corporation at http://www.microsoft.com/.

Reproduction of this work is authorised except for commercial purposes, provided the source is acknowledged. All acknowledgements should be to the DLM Forum Foundation at http://www.dlmforum.eu/.

### 1.1.2 Authenticity

The latest updated version of this document is only available from http://moreq2010.eu/ and http://www.dlmforum.eu/.

The DLM Forum® does not update, support or endorse the MoReq2010® specification on any other websites, services or distribution mechanisms.

### 1.1.3 Citation

This publication should be formally cited as:

DLM Forum Foundation, *MoReq2010®: Modular Requirements for Records Systems – Volume 1: Core Services & Plug-in Modules*, 2011, published at http://moreq2010.eu/.

### 1.1.4 Translations

Permission must be obtained before any translation of MoReq2010® is published or otherwise distributed for any purpose. Translators should apply to the DLM Forum secretariat for permission by sending an email to secretariat@dlmforum.eu.

Permission to make a translation of MoReq2010® is subject to allowing the DLM Forum® and its members to freely copy, use and distribute the translation for non-commercial purposes and to host the translation on the MoReq2010® website.

### 1.1.5 Logos and trademarks

The DLM Forum Foundation logo, the MoReq Governance Board logo and the MoReq2010® logo are copyright © DLM Forum Foundation, 1996 to present.

The terms "DLM Forum", "MoReq", "MoReq2" and "MoReq2010" are registered community trademarks of the DLM Forum Foundation.

The symbol of the European Commission is used with permission.

### 1.1.6 Conventions used in this publication

Throughout this specification all formal requirements and all data type definitions are prefixed for easy identification, as follows:

- **D** – Data structure;
- **E** – Entity type;
- **F** – Function definition;
- **M** – Metadata element definition;
- **N** – Non-functional requirement; and
- **R** – Requirement (functional).

Note that any reference numbers prefixed in this way are for document look up purposes only, are relative to a specific minor version of MoReq2010®, and may be subject to change in any future major or minor version as requirements and definitions are added or changed.

Records systems and other applications implementing MoReq2010® should always use the universally unique identifiers provided by the information model.

Both functional and non-functional requirements in this specification may be accompanied by an explanatory rationale in *italics*. Where provided, the rationale is intended to provide clarity and amplification to the requirement.

## 1.2 Purpose

### 1.2.1 Objective

MoReq2010® aims to provide a comprehensive, but simple and easily understood set of requirements for a records system that is intended to be adaptable and applicable to divergent information and business activities, industry sectors and types of organisation. It avoids a "one size fits all" approach to implementing a records management solution by establishing instead a definition of a common set of core services that are shared by many different types of records system, but which are also modular and flexible, allowing them to be incorporated into highly specialised and dedicated applications that might not previously have been acknowledged as records systems.

The purpose of this document is to describe the minimum functionality required of a MoReq2010® compliant records system, to define common processes, such as export and disposal, and to establish and standardise on an underlying information model that includes entity types, data structures, metadata element definitions and function definitions. Where they are fully implemented, these will reliably support and underpin records system interoperability, including the successful transfer and migration of records in mid lifecycle, between differently implemented but compliant solutions from the same and different suppliers.

The functionality described by the MoReq2010® specification is purposefully intended to be built on and extended through a series of modules covering both generic and specific topics that will be developed in the coming months and years, overseen by the DLM Forum's MoReq Governance Board, to meet the needs and demands of different markets, industries, countries and regions.

Separate guidance will be issued by the MoReq Governance Board covering backwards compatibility, for those consumers wishing to upgrade from MoReq2® to MoReq2010®.

### 1.2.2 Audience

This specification may be used in many different ways, including:

**By businesses:**

- As an aid for the procurement of a records system;
- As a practical tool in helping organisations configure records systems to meet their business and legal obligations; and
- As a guide to the audit of an existing records system implementation.

**By experts:**

- As a reference document for training courses and the preparation of course material;
- As a teaching resource for academic institutions; and
- As an example of how traditional records management approaches and archival science can be applied to modern systems requirements.

**By industry:**

- To guide the development of records systems by suppliers;
- To integrate records systems with other business systems; and
- As the authoritative source when undertaking the testing and certification of compliant solutions by accredited test centres.

**By users:**

- As a user-centric and easily understandable resource and primer on implementing records systems;
- As the original for all translations; and
- As a reference glossary for guidance on records management terms and their meanings.

### 1.2.3 Best practice

MoReq2010® is best used within consumer organisations as part of an overarching records management policy within a well developed strategic framework. Educating users, encouraging adoption, fostering a corporate culture of good practice around records and information, raising awareness of information governance requirements, highlighting and briefing staff at all levels on important considerations such as security, privacy, data sensitivity, freedom of information and open data initiatives; as well as putting in place clear, practical manual procedures accompanied by quality assurance checks are all equally as important as automation and the integration of a records system into the business environment.

Implementing good records management requires forward planning, anticipation of the issues that will arise, and the development and implementation of organisational policies and procedures that cover what records should be kept, how records are created and captured; how records are held, managed and accessed throughout their active lifecycle; and all aspects of their eventual disposal. This forward planning needs to reach beyond the limited lifespan of any one technology or system solution and consider the question of how

records will be migrated to the next corporate records system with the same weight as is given to how to ensure their capture into the current records system.

Within such an environment the adoption and use of MoReq2010® compatible records and business systems can make a sound organisational investment.

## 1.3 Background

### 1.3.1 MoReq®

The first MoReq® specification was published in 2001 as a result of close cooperation between the DLM Forum and the European Commission. MoReq® provided a new pan-European specification for computer systems that manage electronic records. Prior to its publication there were only a few countries in Europe with their own national standards for records management.

Even from its earliest publication, MoReq® has always had the following characteristics:

- **Universal in scope and application** – MoReq® is an international specification and has been used and adopted across a large number of countries, including many outside Europe;
- **Available in many languages** – MoReq® and its successor MoReq2® have been translated in full into over a dozen European, and some non-European, languages; and
- ***De facto* standardisation** – although originally conceived as a specification and not a *de jure* standard, MoReq® is today widely recognised as a *de facto* industry standard because of its universal appeal, availability and adoption.

"MoReq" was first used as an abbreviation for "Model Requirements" and it was originally envisaged that the specification would provide a templated set of requirements that would then be further modifiable to meet local needs. The first edition, therefore, contained guidance on how to add, edit and delete chapters and requirements, and to manage issues such as cross-referencing within the specification while doing this.

### 1.3.2 MoReq2®

In 2005, the DLM Forum® completed a scoping study aimed at updating and extending the original MoReq® specification. The result of this review was the development of MoReq2® and its publication in early 2008.

A key feature of MoReq2® was the inclusion for the first time of a testing and certification regime. Suppliers could now have their solutions tested at a MoReq2® testing centre and receive independent certification that their products were compliant with the specification. In order to support testing and certification, MoReq2® introduced a metadata model into the specification, as well as an XML schema that was intended to define a common import/export format across different products and implementations.

The introduction of the testing and certification programme in MoReq2® was an extremely important and progressive step that introduced a necessary element of rigour and quality assurance into the adoption of the specification. Suppliers with high quality products could now obtain independent verification and evidence of conformance; while consumers could choose from a set of products which all met recognised quality standards.

There was also, however, an unintended consequence brought about by the testing regime and this lay with the very concept of "model" requirements. If products were to be pre-tested and certified against MoReq2®, it became more difficult for consumer organisations to then take the specification and alter it at the local level. How could the supplier of a common off-the-shelf software product, which had already been tested and certified as compliant, anticipate within the application later additions, changes or deletions to individual requirements at the organisational level? Another related issue was that as the complexity of the specification increased, how could organisations be sure of the subtle ramifications of adding, altering and deleting requirements, in their total effect on the integrity of the specification as a whole?

In December 2008, at its triennial conference in Toulouse, the DLM Forum® appointed a permanent sub-committee to be called the MoReq Governance Board. The role of the governance board was and remains, to manage all aspects of the MoReq® specification including to:

- Provide for its ongoing maintenance, publish a roadmap for MoReq®, and plan for future upgrade of the specification;
- Manage the translation programme, arrange for the validation of accepted translations, and give guidance to MoReq® translators;
- Grant accreditation to recognised test centres to undertake software testing against the specification;
- Oversee the testing and certification of software products against MoReq® by accredited test centres;
- Run a parallel education programme including providing workshops and training, issuing supplementary guidance and educational materials; and
- Actively market the specification, collect case studies, and encourage its adoption while simultaneously protecting the MoReq® brand.

### 1.3.3 MoReq® roadmap

In 2009, the MoReq Governance Board produced a roadmap for MoReq®, which was subsequently adopted by DLM Forum resolution at a members' meeting in Härnösand, Sweden in November of that year.

The roadmap, an image of which is shown in **Figure 1a**, identified that while the MoReq® specification was widely viewed by industry as addressing the requirements for mainstream records management in traditional, often office based or clerical, domains such as Electronic Document and Records Management Systems (EDRMS) and Enterprise Content Management (ECM), it was also seen as less applicable to adoption in areas like medical, pharmaceutical, legal and financial services where specialised applications that solved domain-specific problems were the norm. These industry sectors were typically governed by legislation and regulation particular to themselves and as a result tended to invent and adopt their own set of records management criteria.

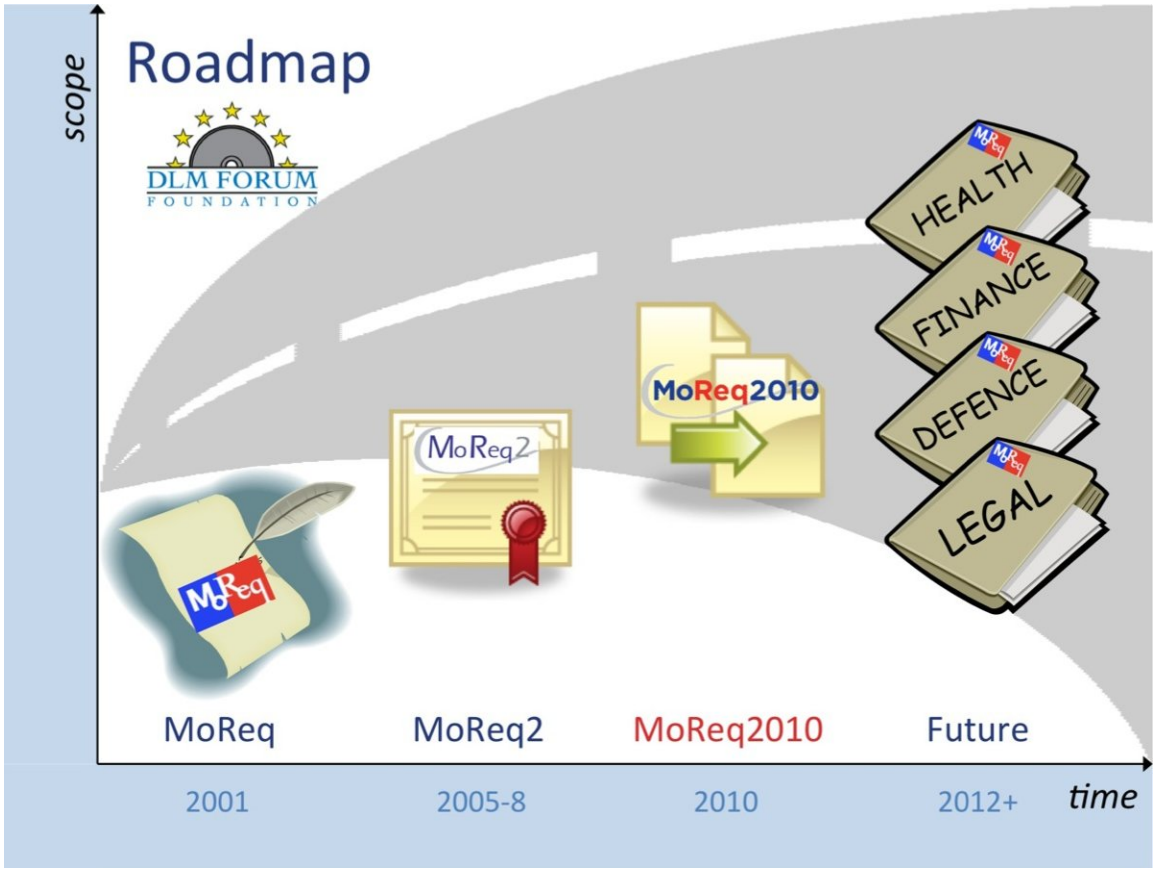


***Figure 1a*** *– The MoReq Governance Board's Roadmap (circa 2009)*

Another trend the MoReq Governance Board identified was the increasing heterogeneity within records system design. Conceptually, MoReq® was originally based on a single centralised repository model where an organisation's standalone records system would capture records into its own data store from a variety of external sources, including users and other business systems. This traditional architecture is shown in **Figure 1b**.

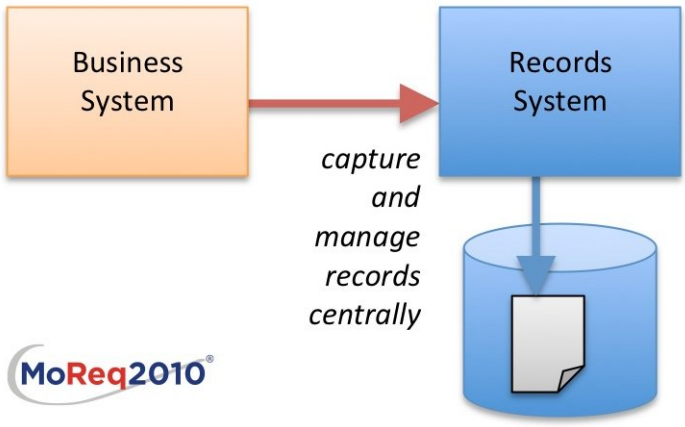


***Figure 1b*** *– The "traditional" architecture of a records system includes capture of records from other business systems and centralised management in a repository controlled by the records system*

The governance board's roadmap recognised an expanding adoption of alternative architectures. One emerging model, shown in **Figure 1c**, is that of the storage-less records system that manages records *in situ* within the business systems in which they originate, rather than duplicating them into its own centralised repository.

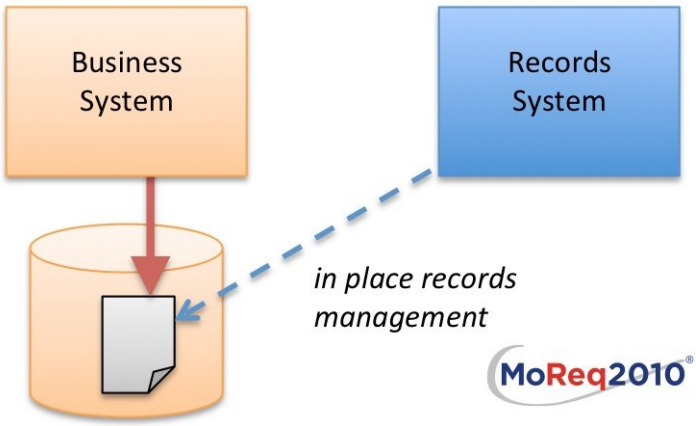


***Figure 1c** – An alternative architecture is to manage records in place by allowing the records system to apply controls and processes to records that have been declared in situ*

Another possible architecture is the adoption of records controls by the business system itself. Such a business system is, in effect, simultaneously a records system albeit one that manages only the specific set of records captured or generated by that particular business system. The business system as records system model is shown in **Figure 1d**.


***Figure 1d** – As records management controls and processes become simple, flexible and better understood, then it is increasingly possible that business systems will "become" records systems, at least for the business records they themselves produce*

The MoReq® roadmap also identified the need for flexible and scalable requirements that were equally applicable to both large and small records management solutions. MoReq2® had more than doubled the number of functional requirements and nearly tripled the number of pages of its predecessor. It was clear that each successive edition of MoReq® could not continue to similarly grow in size and complexity and remain as primarily a single block of requirements, or it would eventually in itself provide an insurmountable barrier to

adoption, particularly by smaller, local and niche suppliers. For these reasons the 2009 roadmap called for two future phases of MoReq® evolution:

- In the short-term, starting in 2010, to launch a refactoring project that would reorganise the specification along modular lines, simplify it where possible, and introduce support for alternative records systems architectures; and
- In the longer-term, from 2012, with the assistance of specialised industry experts, and based on the flexibility afforded by a more modular approach, to broaden the applicability of MoReq® into all fields of human endeavour where the sound management of records is an essential prerequisite.

The first of these objectives was realised by the DLM Forum's annual general meeting in Madrid in May 2010, where the MoReq2010® work programme was officially launched.

### 1.3.4 MoReq2010®

In addition to innovation within the specification itself, the DLM Forum® decided to incorporate two phases of public consultation into the development of the new specification, while the European Commission appointed an Experts' Review Group made up of a cross-section of world renowned industry experts to provide advice to the project. By opening up the consultation phases to the public, an increased aspect of collegial and social networking was introduced to the development programme. The resulting specification has thereby benefited from having been developed and discussed in an open way, with an unexpectedly high quantity and quality of collaborative input.

Records management today can be a complex undertaking and MoReq2010®, with its flexible and extensible architecture, provides one possible approach to navigating the pitfalls of specifying a records system implementation that is attractive and suitable for suppliers, practitioners and consumers alike.

With MoReq2010® the concept of "model" requirements has been replaced by that of "modular" requirements. The now established programme of pre-testing and certification of software products and the increased emphasis on interoperability negates the value of modifying and customising individual requirements. Appropriately, there is no longer any need for this as the new modular approach allows consumers to easily specify a flexible yet comprehensive and cohesive set of organisational requirements simply by choosing a suitable combination from a selection of modules that correspond to their organisational needs. Over time the number and variety of modules that build on the platform of the MoReq2010® core services will steadily increase and extend coverage into more and more industries, sectors and jurisdictions.

Suppliers too will benefit from the refactoring of MoReq2010®. While the set of additional extension modules will continue to grow and embody more and more specialised applications, the core set of requirements is correspondingly reduced by comparison. The core services are the only requirements that all records systems must have in common and show compliance with. The implementation of, and certification against, other modules then depends on the particular focus of a given product, its target sector and its degree of specialisation or generalisation. Suppliers are therefore free to pick and choose which functionality to implement to meet their target markets, in the same way as consumers may pick and choose the modular requirements that have significance for them.

The third group that will benefit directly from the approach taken by MoReq2010® is comprised of records management experts and practitioners. MoReq2010® seeks in every aspect to directly tie leading records management theory and best practice back into the specification. Professionals will find that concepts, terms and models adopted by MoReq2010® are closely linked to those used in other international standards, and propounded by leading experts. In addition to its practical application, the specification therefore forms a sound learning and educational platform.

2011 marks the 15th anniversary of the DLM Forum® and the 10th anniversary of MoReq®. Launched alongside these significant anniversaries, MoReq2010®, the next generation version of the "model requirements" points squarely to the future as both a catalyst and a springboard for the improved recognition, understanding and adoption of good records management throughout Europe and internationally.

## 1.4 Primer

### 1.4.1 Records and information

Every organisation and every citizen has and uses records.

Records are those pieces of information that have an intrinsic worth which makes them important enough to save, and keep secure, for their evidential value.

Most people, for example, have several records in their possession which they keep as proof of their identity. These may include:

- A birth certificate,
- A passport,
- A driver's licence, and/or
- An identity card.

In each of the examples listed above the importance of the record, and its evidential nature, is obvious. However, this may not always be the case. People do not usually regard all of the pieces of information they have in their possession as necessarily records.

For example, a shopping list might not be regarded as a record, but by comparison, the receipt from the shop where the goods on the shopping list were bought provides proof of purchase and may be considered a record if it is important to an individual or a business. Such a receipt could be used to claim back expenses from an employer or to receive a refund when returning damaged or spoilt goods. This example is illustrated by **Figure 1e**.


***Figure 1e** – One of these is probably not a record… the other may be?*

The distinction between information and records is the same for organisations as it is for individuals, so that for any organisation it can be stated that the set of all its records is a sub-set of the set of all its information assets, shown conceptually as a Venn diagram in **Figure 1f**.

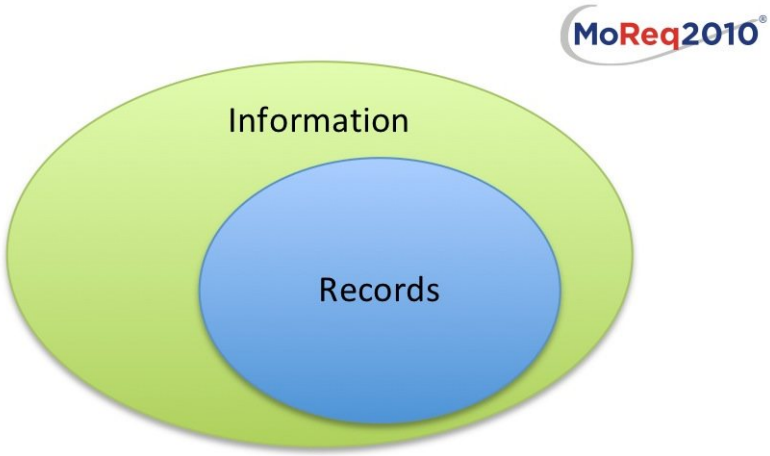


***Figure 1f** – Records are a sub-set of all information held by a person or organisation*

In order to decide whether a piece of information is a record or not, its business context must be understood as well as its relevance and significance to the organisation. An important task for any organisation, therefore, is to gain an understanding of its business and to be able to use this understanding to evaluate what information it requires to be retained and managed as its records.

Information assets that belong to an organisation, but are not regarded as records, such as the early drafts of a document, or an incomplete transaction, must be considered transient and should be routinely excised. For example, intermediate drafts might be erased when a document is published, and partial transactional information might be erased when the transaction is completed or cancelled.

Failure to excise transient information may, at best, waste the organisation's data storage capacity with redundant or incomplete copies and waste staff time sorting out which information is complete and correct and which is partial. At worst, it may be in breach of privacy or other regulations or lead to costly exercises compiling the information if it is called for as part of a legal action or freedom of information request, or similar.

### 1.4.2 Records management processes and systems

ISO 15489, published in 2001, is perhaps the most influential standard in records management internationally. Determining the information that must be managed as records is only the first of the records management processes it identifies. The full list of records management processes identified by ISO 15489 includes, additionally:

- Determining how long to retain records;
- Creating and registering records;
- Classification of records;
- Storage and handling of records;
- Controlling access to records;
- Tracking records;
- Disposing of records; and
- Documenting records management processes.

ISO 15489 proposes that an organisation should use a records system to implement these processes. It defines a records system as an "information system which captures, manages and provides access to records through time" (ISO 15489-1:2001, 3.17).

MoReq2010® is a specification for defining a records system expressed as a modular set of requirements. It goes beyond the broad description offered by ISO 15489 and adds a far greater level of specificity in how these processed should be carried out. Achieving Moreq2010® compliance requires a greater degree of rigour than can be achieved by simply building a records system that handles the records management processes described by ISO 15489 in its own proprietary way.

One of the advantages of this, and a design goal of MoReq2010®, is the potential for interoperability between MoReq2010® compliant records systems (MCRS). An MCRS does not only understand its own entities and its own processes, it can export them to a standardised format that can be understood by another MCRS.

Interoperability is essential to the management of records using a records system. Today's organisations typically refresh their technology every three to five years. Records are often held for much longer than that. If an organisation is required to keep a particular record for 75 years then, at the end of that period, it will typically have been transferred from one records system to another between 15 and 25 times. This is shown in **Figure 1g**. If each transfer results in some loss of contextual information about the record then this number of transfers may have a severe impact on the record's integrity.

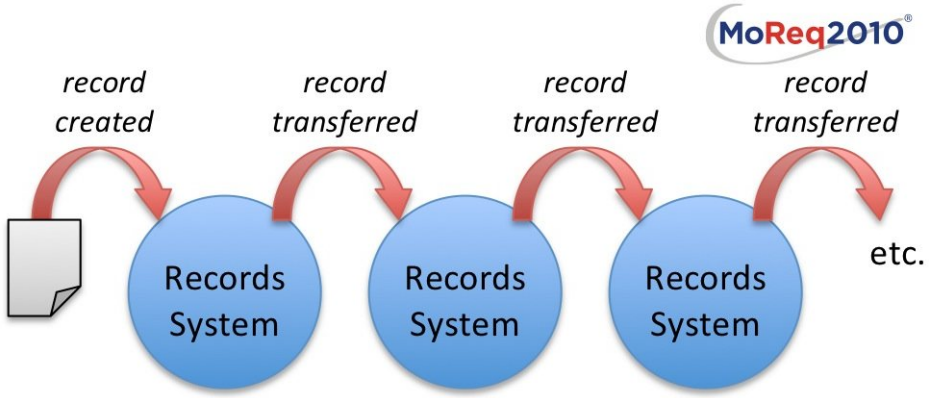


***Figure 1g** – Records may be transferred multiple times between records systems during their lifespan*

It should be noted that import is not part of the core services of an MCRS, but every MCRS must be able to export its information to the MoReq2010® common XML export format. Import requires a far higher level of application sophistication than export, and mandating it for all records systems would preclude many dedicated business systems from adopting MoReq2010®.

### 1.4.3 The nature of records

Almost anything that has informational of evidential value can be managed as a record. ISO 15489 formally defines a record as, "information created, received, and maintained as evidence and information by an organization or person, in pursuance of legal obligations or in the transaction of business" (ISO 15489-1:2001, 3.15).

In the past, records were mostly paper based but during the 21st century this is being relentlessly replaced by the exponential increase in electronic information. However, the need for systems that perform physical records management will always remain, both for paper records and for other physical records, for example, to keep track of bio-medical or forensic samples.

All records, including both physical and electronic, have certain characteristics. ISO 15489 lists the essential characteristics of a record as:

- Authenticity – the record is what it purports to be and was created by the person purported to have created it;
- Reliability – the information in the record is accurate and can be depended on;
- Integrity – the record is complete and unaltered; and
- Usability – the record can be located, retrieved, presented and interpreted.

An MCRS can only ensure these characteristics from the point at which the record is created in the records system. As stated by ISO 15489, "Records should be created at the time of the transaction or incident to which they relate, or soon afterwards, by individuals who have direct knowledge of the facts or by instruments routinely used within the business to conduct the transaction." (ISO 15489-1:2001, 7.2.3).

The statement indicates two ways in which records may be created, either:

- They are created by an individual; or
- They are created by an instrument.

In MoReq2010® both humans and business systems are regarded as possible "users" of a records system and may be authorised to create records. An MCRS may be developed to interface only to human users, or to other business systems, or both.

In addition to the characteristics listed previously all records in a records system must also have metadata associated with them. Metadata is defined by ISO 15489 as, "data describing the context, content and structure of records and their management through time" (ISO 15489-1:2001, 3.12).

### 1.4.4 Entities and services

A MoReq2010® compliant records system manages records as entities. Records are only one of the entity types defined by the specification. In addition to records, MoReq2010® also defines a number of other entities of different types. For example, MoReq2010® defines a user entity type that represents the users that access the records system, and a class entity type for each entry in the records system's classification scheme, and so on. A full list of entity types may be found in **14.2 Entity Types**.

Even though the entities managed by an MCRS are of different types, MoReq2010® attempts to make them as uniform as possible in the way that their metadata is represented and their event history is managed, in their access controls and in their entity lifecycle. Unlike the entities in other information systems, entities in an MCRS are destroyed, rather than deleted, leaving a residual entity that remains in the MCRS. Residual entities are an important concept in records systems as they indicate entities that were once present in the system. Without them it would not be possible to reconstruct the full context of an historic record.

Within an MCRS, entities of different types are nominally described as being managed by different services according to a "service based architecture" (see **2. System Services**):

- A user and group service manages user entities and group entities (see **3. User and Group Service**);
- A role service manages roles (see **4. Model Role Service**);
- A classification service manages classes (see **5. Classification Service**);
- A record service manages records and aggregations of records (see **6. Record Service**);
- A metadata service manages metadata and metadata templates (see **7. Model Metadata Service**);
- A disposal scheduling service manages disposal schedules (see **8. Disposal Scheduling Service**); and

- A disposal holding service manages disposal holds (see **9. Disposal Holding Service**).

Other services are purely process based and do not manage entities, including:

- A searching and reporting service (see **10. Searching and Reporting Service**), and
- An export service (see **11. Export Service**).

While it uses the language, and promotes the adoption, of a service based architecture, MoReq2010® acknowledges that historically records systems have not necessarily delivered functionality using a discrete service model. For this reason, MoReq2010® does not do more than merely bundle its functional requirements into logical services and test against each "service" (bundle) of functional requirements individually. An MCRS that does not provide discrete services in its implementation will still be certified as compliant to the MoReq2010® specification.

Nonetheless, the approach taken by MoReq2010® is deliberate and anticipates a future where interoperability is not confined to the transfer of records from one MCRS to another, but where different records systems can share the same services in common. Within an organisation of the future, it might be possible for all records systems to make use of a single user and group service, a single role service, a single classification service, a single metadata service, a single disposal scheduling service, a single disposal holding service and/or a single searching and reporting service.

Such an approach would allow, for example, a business classification scheme to be defined once across the whole organisation and managed centrally using a shared classification service, as shown in **Figure 1h**. The same applies equally to other services.

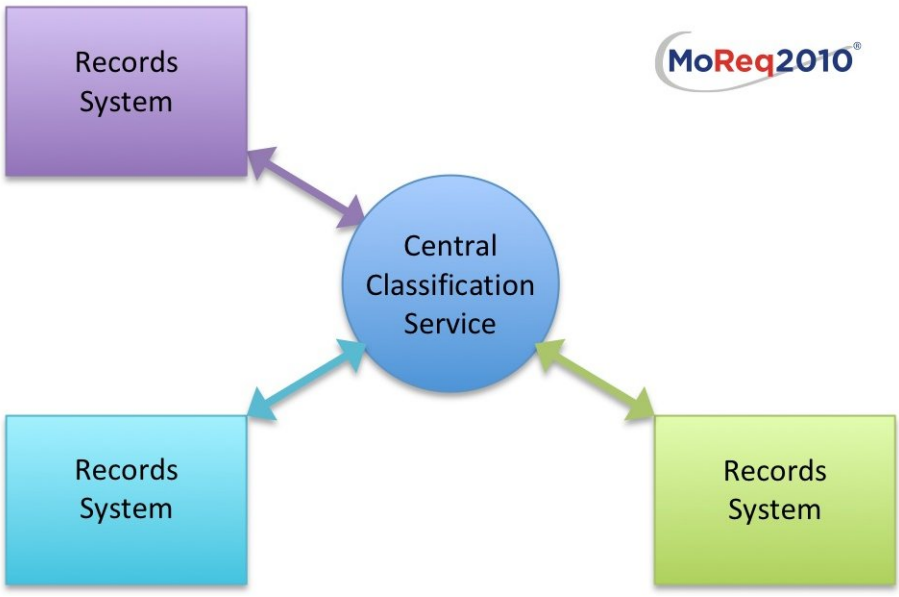


***Figure 1h** – In the future multiple records systems may be able to share a single centralised classification service*

Shared service utilisation will not necessarily be confined to within the boundaries of a single organisation. For example, a regulator could issue and maintain a standardised set of disposal schedules to be used by all organisations in a particular sector by hosting them as a

widely accessible disposal scheduling service, or a disposal holding service shared between litigants might manage the disposal holds issued by a particular legal court.

### 1.4.5 Classification and aggregation

There are several other concepts that are new in MoReq2010®. One is the distinction made between classification and aggregation. ISO 15489 defines classification as the "systematic identification and arrangement of business activities and/or records into categories according to logically structured conventions, methods, and procedural rules represented in a classification system" (ISO 15489-1:2001, 3.5).

While classification is concerned with providing the business context for a record and establishing the relationship between a record and the transactional activity by which it was created, aggregation describes the activity of assembling related records together. Unlike classification, aggregation may be based on any organisational requirement or criteria, not business context alone. Aggregation is layered, with higher level aggregations made up of an assembly of lower level aggregations. A whole record service might arguably be considered to represent one high level aggregation.

Historically, some records management specifications conjoin a hierarchical classification scheme above a layer of aggregation so that each record always inherits its class via its aggregation. This approach, shown in **Figure 1i**, uses classes in place of higher level aggregations. Such an arrangement, while desirable for its simplicity if it can be achieved, is also inflexible and does not always lend itself to real world usage. The restrictions this approach imposes has led in many cases to organisational and subject based elements being mixed into a functional business classification scheme to create a localised hybrid.

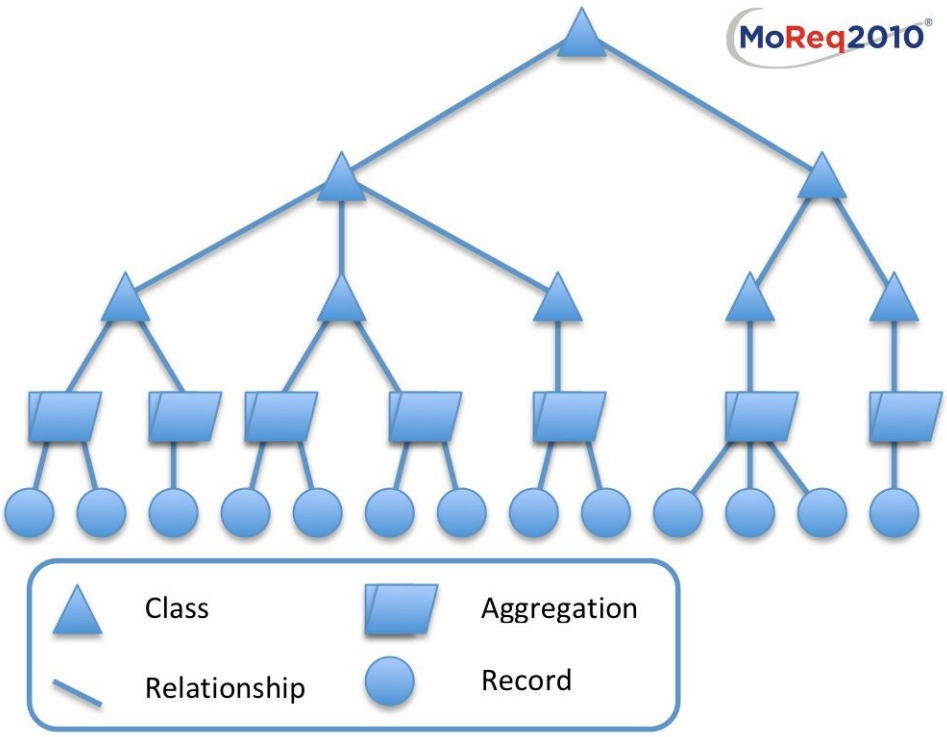


***Figure 1i** – A traditional hierarchical model of classification and aggregation, where both classes and aggregations are joined together into a single structure (this approach can be implemented by an MCRS, but MoReq2010® also allows for greater flexibility)*

Many organisations undertake collaborative teamwork, casework or project based work. Where these activities occur the natural tendency within organisations is to aggregate records based on the main topic, such as the particular project, and not based purely on the business activity or process that creates the record.

For example, a small association may keep records about each of its members in a single aggregation per member. Within each member aggregation there might be found:

- The member's original registration form, assessment and approval,
- Identification, banking and contact details including various change notifications,
- Annual subscription and membership renewals,
- Applications, appointments and offices held within the association,
- Correspondence and other information on various association activities undertaken by the member,
- Expenses and reimbursement claims,
- Contracts, certificates, legal documents and waivers,
- and so on.

In this example, the records in the member aggregation relate to different functions, activities and transactions and should therefore attract different business classifications. There may be legal, regulatory or good practice reasons based on these classifications why some records in each member aggregation must be retained for a relatively long statutory period, while other records in the aggregation should only be kept for a short time and then destroyed.

An arrangement like this, even though it is commonly found across organisations, is difficult to squeeze into a traditional hierarchical classification based structure as shown in Figure 1i. This is because in the traditional arrangement, the whole member aggregation must fit under a single class in the classification scheme.

As a result, there is inevitably a tension created between practical and operational efficiency and records management needs. Either the classification scheme is hybridised, for example by introducing omnibus classifications such as "casework", "clients", "projects", "staff" and "events"; or naturally occurring aggregations are split up so that, for example, the registration records for all members are to be found together under a single classification/aggregation such as "membership applications", while annual subscription records for all members are collected in an entirely separate classification/aggregation, such as "membership renewals 2011". Neither compromise is likely to be viewed as entirely satisfactory.

By providing a clear distinction between the related concepts of classification and aggregation, MoReq2010® allows for greater flexibility in making planning decisions about what records to keep together, combined with what classification scheme to use and how to apply it. This in turn makes MoReq2010® more adaptable to real world situations. The specification allows aggregation to be based on operational criteria while classification can be applied at any layer of aggregation, including even associating classes with records individually if required (this is shown in **Figure 5d** and discussed further in **5. Classification Service**). At the same time, backwards compatibility with the traditional approach shown in **Figure 1i** is maintained.

### 1.4.6 Retention and disposal

MoReq2010® closely associates business classification with retention and disposal, so that each class has an associated disposal schedule and each record inherits its disposal schedule, by default, from its class; adopting the principle that "classification determines destiny". This is different to some approaches where disposal schedules are inherited from a record's aggregation and only indirectly from its classification.

A significant feature of disposal scheduling is that MoReq2010® does not allow a record to be subject to more than one disposal schedule simultaneously. The specification permits the default disposal schedule, inherited from the record's class, to be overridden, but at any point in time only one disposal schedule can apply a particular record. There is therefore no chance for a disposal conflict to occur that requires direct user intervention to resolve.

As each record in an aggregation may have a different classification to other records, and each record also has its own disposal schedule inherited from its class, then it is possible that individual records within an aggregation will be due for disposal at different times. This is determined by the disposal schedules that the organisation uses and what disposal triggers are specified.

MoReq2010® uses the principle of bottom up destruction to dispose of an aggregation only when all of its contents have been destroyed and the aggregation is closed. Bottom up destruction is described in greater detail in 8.2.9. One of the advantages of bottom up destruction is that it does not require that aggregations have disposal schedules. There is only one type of disposal schedule in MoReq2010®, which is related to the record.

Even though MoReq2010 applies disposal individually to each record, it is possible to apply the same disposal action to many records simultaneously. For example, MoReq2010® enables a user to authorise the same disposal action once for an aggregation as a whole. This allows for ease of use while maintaining a simple, yet flexible, approach to retention and disposal.

### 1.4.7 Event histories and audit

ISO 15489 requires the use of either metadata or, alternatively, audit trails to be kept as "complete and accurate representations of all transactions that occur in relation to a particular record" (ISO 15489-1:2001, 8.3.2). MoReq2010® adopts this approach but extends it by adopting the concept of an event history for each record from ISO 23081.

ISO 23081, an international standard on metadata for records, describes an event history as follows, "The event history metadata group documents past records events and other management events on both the entity and its metadata. For each event it specifies the type of event, what happened, when it took place, why it occurred, and who carried it out. The metadata in this element are a sequence documenting a specific event."

In MoReq2010® every entity has an event history associated with it. This is particularly important in supporting interoperability, where entities are transferred from one records system to another. Each entity is transferred as a whole, including its metadata, event history, access controls, and so on. The event history is an integral part of the entity and this approach allows all MCRS to import and fully understand events that occurred to an entity while it was part of a previous records system.

Even though event histories are tied to entities, MoReq2010® still allows a "system audit trail" view across an MCRS by allowing users to search across all events for all entities and sort them by when they occurred. In this way, the cumulative events from all event histories make up the audit trail for an MCRS.

Under MoReq2010® event histories and metadata for entities are pruned when entities are destroyed in line with ISO 15489 which states that audit trails should be kept at least as long as the document [i.e. record content] to which they relate is retained" (ISO 15489-1:2001, 8.3.2).

### 1.4.8 Modular architecture

**Figure 1j** shows the modular architecture of MoReq2010®. Each box in the diagram represents a bundle of requirements that represents either a service or a module. The core services are defined in Volume 1 of the specification and provide the minimum set of functionality for compliance with MoReq2010®; or, in other words, they define the simplest possible MCRS.

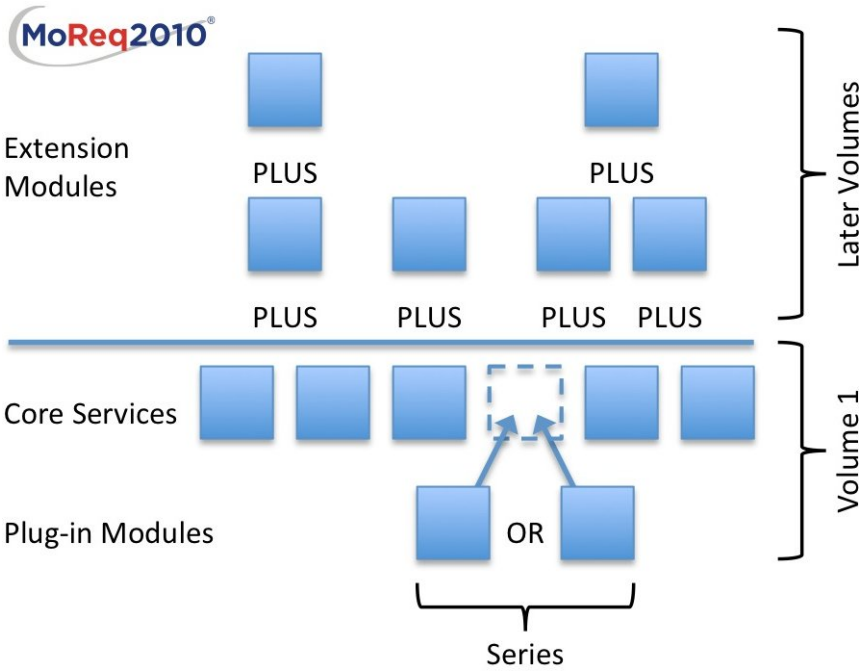


***Figure 1j** – The structure of the MoReq2010 specification*

Some of the functionality described by the core services can be implemented in alternative, but equally valid, ways. Where this occurs, MoReq2010® makes use of plug-in modules. Each plug-in module represents exactly equivalent functionality as any of the other plug-in modules in the same series. An MCRS must implement the functionality described by at least one of the plug-in modules in a series and may choose to implement more than one plug-in module.

Plug-in modules are used by the core services to allow flexibility around the following areas of functionality:

- Type of interface – including whether the MCRS is managed directly by users through a human computer interface or as a business support system through a machine interface;
- Type of classification – allowing an MCRS to adopt different approaches to classification, for example, a hierarchical classification scheme; and
- Type of record components – allowing an MCRS to support different types of records, such as electronic components rather than physical components.

The core services and plug-in modules described in Volume 1 of the MoReq2010® specification form the essential platform on which different extension modules can be hosted. The range and variety of extension modules to MoReq2010® will be extensive, and will cover:

- Additional services – such as import;
- Further concepts – such as vital records;
- Specific technologies – such as email; and
- Requirements for records systems for particular industries and jurisdictions.

As shown in **Figure 1j**, extension modules can build on each other as well as extending the core services. The preface to each module contains a list of its prerequisites and co-requisites.

Prerequisites define a dependency where the MCRS must implement the prerequisite module as well as the extension module. Co-requisites are modules which attract additional functional requirements, if they are implemented simultaneously with the extension module.

### 1.4.9 Model services

The development of MoReq2010® has required compromise between the state of the records management industry today and the vision of the future promoted by the specification and the MoReq Governance Board. This compromise is particularly apparent across two services: roles and metadata.

In the interest of interoperability, MoReq2010® seeks to codify both an entity's access controls and its metadata elements so that they can be transferred to a new records system where they will be successfully imported, understood and used. Metadata can describe valuable contextual information about the original entity, while access controls, even when they are superseded in the receiving system provide important knowledge about who had access to which records in the original records system.

Because no preceding specification has sought to address these issues through standardisation, suppliers have by necessity implemented their own individual and proprietary methods of applying metadata and access controls within their records system software. As a consequence, an unreasonably large amount of refactoring would be required within these existing records systems to adopt a metadata model or an access control model that was common to all MCRS.

For this reason, MoReq2010® specifies these two services as model services. A model service is an exemplar service, it is intended to be adopted as the appropriate way to develop new records management software in the future, without impeding existing products from seeking compliance with, and certification against, MoReq2010®.

MoReq2010® accepts either of two different methods for proving compliance with a model service. Method A is to implement the functional requirements for the model service and be tested against them. Method B, intended mainly for pre-existing software, is to demonstrate a proprietary solution that is as rich in functionality as the model service, and where the constructs and data are able to be converted and exported as if they had originated in a records system that implements the model service.

This requirement is essential for compliance with any model service: that the entities, their metadata and their access controls, are meaningfully exported to the standard MoReq2010® XML export format. If this can be achieved then another MCRS can import the entities and use them in conjunction with a model service.

### 1.4.10 Testing and certification

The DLM Forum® has initiated a testing and certification programme for MoReq2010®. The programme allows suppliers of common-off-the-shelf (COTS) records systems, as well as in house records systems, to be tested by a DLM Forum® accredited MoReq2010® test centre.

Suppliers must have their products tested against the core services, and may then optionally have any of the additional extension modules tested as well. Following the successful completion of testing using the MoReq2010® test framework a product or installation may become certified by the DLM Forum® as MoReq2010® compliant.

Suppliers will be able to show that their products are fully certified as MoReq2010® compliant; while members of the DLM Forum® will benefit by having access to the test reports, allowing them to undertake a preliminary analysis of different MoReq2010® compliant records systems.

The DLM Forum® will publish lists of accredited MoReq2010® test centres, and lists of certified MoReq2010® compliant records systems, on its website at www.dlmforum.eu.

# 2. System Services

## 2.1 Service Information

Each of the core services of MoReq2010® is defined by its service name (for example, "User and Group Service"), its service version (for example, "1.0") and its Implements Service Identifier (for example, "5e69596d-5fac-4017-b204-1aeb85b004b0"), or Implements Module Identifier for plug-in and extension modules.

These details can be found in the service information block (see, for example, **3.1**). The Implements Service Identifier is a universally unique identifier used internally by an MCRS to show which services it implements.

This section, **2. System Services**, contains functionality common to all MoReq2010® core services and, as such, does not have separate service information. Refer instead to the service information block for each service individually.

## 2.2 Key Concepts

### 2.2.1 Service based architecture

The functional requirements of the MoReq2010® core are bundled into nine service definitions, shown in **Figure 2a**. Taken together these services describe the functionality required by an MCRS. This initial module, entitled **2. System Services**, describes the common functionality required by every MoReq2010® core service.

The service based architecture of MoReq2010® is not intended to constrain software suppliers from developing fully compliant solutions that combine the functionality of many or even all core services together and deliver them from within a single application. However, by dividing the architecture of MoReq2010® into separate services, future consideration may also be given by suppliers to developing records systems where each of the services is decoupled from the others and can then be shared between more than one MCRS.

For example, in the future each records system within an organisation might be capable of sharing the same classification service or the same disposal holding service. It may also be possible in the future to build an MCRS by sourcing different services from different suppliers and integrating them together.

Whether provided as a single application, a tightly integrated or a loosely integrated collection of services, all MCRS solutions must be tested against the same compliance criteria.

At the heart of the service based architecture of an MCRS is its record service. The record service is the only core service that cannot be shared with another MCRS. Indeed it is literally only the record service that distinguishes one MCRS from another. All other services that support the record service may simultaneously support other record services, and may therefore be a part of several MCRS solutions simultaneously.

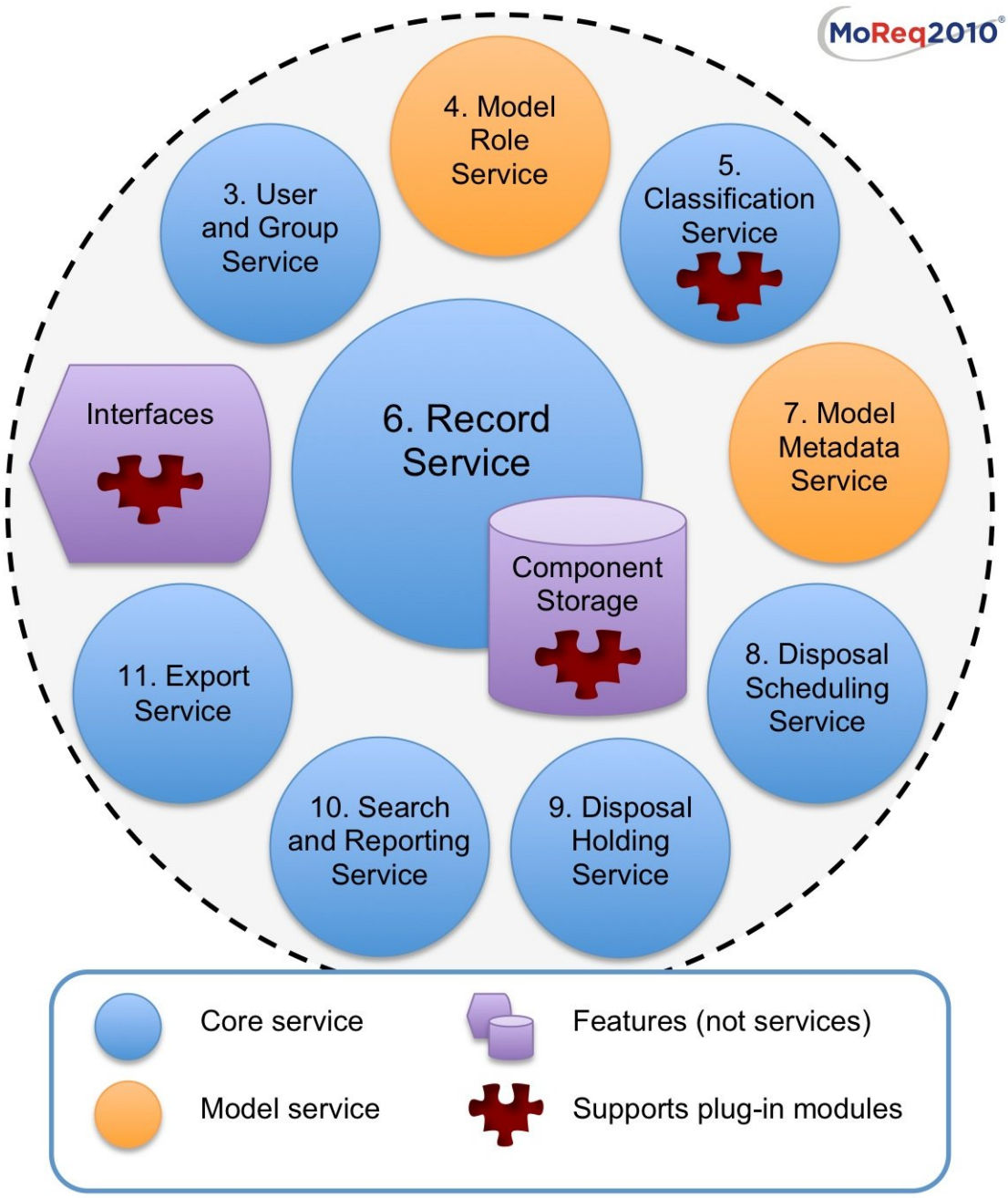


***Figure 2a** – A MoReq2010® compliant records system (MCRS) seen as a grouping of interrelated services with a service based architecture (each core service has its own numbered section of the specification)*

### 2.2.2 Model services and plug-in modules

Two of the core services of MoReq2010® are model services (these are **4. Model Role Service** and **7. Model Metadata Service**). This means that while the specification provides a default set of functional requirements for each of these services, MoReq2010® does not require suppliers to implement these services exactly as they are specified, except where the supplier wishes to support advanced modules, such as the import module.

There are also three areas within the core services where plug-in functionality is specified. Plug-in modules represent alternative, but equally valid, sets of functionality that achieve

the same goal but in different ways. Suppliers may choose which approach they adopt. Each MCRS must provide support for, and be certified against, at least one implementation of the functionality. It may also support, and be certified against, other alternative plug-in modules in the same series as well. The interface to the MCRS is an example of one area where plug-in functionality is specified.

### 2.2.3 Interfacing with Users

MoReq2010® does not require that users always interface with records systems directly.

In traditional records system implementations, users interact directly with the records system through a graphical user interface (GUI), shown conceptually in **Figure 2b**.

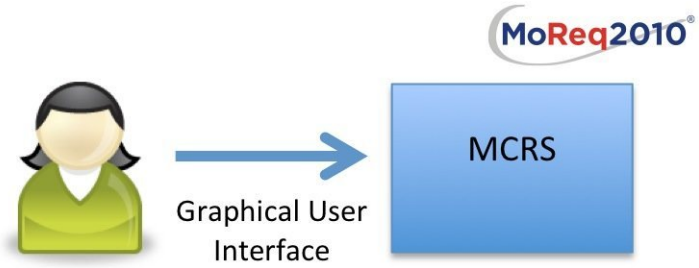


***Figure 2b** – User directly interacting with a records system through a GUI*

Increasingly, records systems are being developed that support one or several different business systems. These records systems do not provide a graphical user interface but interact directly with the business systems using an application programming interface (API). In this scenario, depicted in **Figure 2c**, the user performs functions in the records system only indirectly as a result of the user's actions in the business system. The user may not even realise the existence of a separate records system.

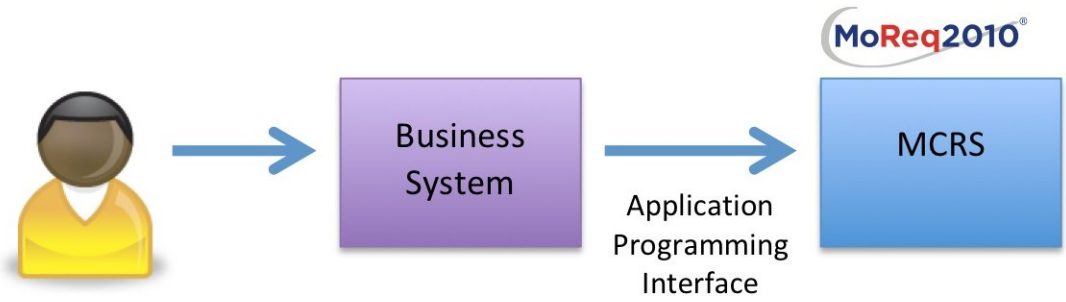


***Figure 2c** – User indirectly interacting with a records system through an API*

MoReq2010® provides support for both these types of records system by allowing an MCRS to specify which type of interface it supports. An MCRS may also support more than one type of interface simultaneously. Similarly, each of the core services of MoReq2010 may support a different interface option. For example, the user and group system may provide an API while the searching and reporting system offers a GUI.

***Throughout MoReq2010® it is important to keep in mind that the user of an MCRS is not necessarily a person. The authorised user could be a business system instead.***

### 2.2.4 Entity types and sub-types

Each MoReq2010® core service manages entities belonging to a specified number of entity types. The MoReq2010® core services refer to the following entity types:

- Access control lists, defined in **4. Model Role Service**;
- Aggregations, defined in **6. Record Service**;
- Classes, defined in **5. Classification Service**;
- Components, defined in **6. Record Service**;
- Disposal holds, define in **9. Disposal Holding Service**;
- Disposal schedules, defined in **8. Disposal Scheduling Service**;
- Entity types, defined in **2. System Services**;
- Events, defined in **2. System Services**;
- Function definitions, defined in **2. System Services**;
- Groups, defined in **3. User and Group Service**;
- Metadata element definitions, defined in **7. Model Metadata Service**;
- Records, defined in **6. Record Service**;
- Roles, defined in **4. Model Role Service**;
- Services, defined in **2. System Services**; and
- Users, defined in **3. User and Group Service**.

Tables giving the attributes of each of these entity types can be found in **14.2 Entity Types**.

Access control lists and events are listed with other entity types above, but these are not independent entities. All entities except access control lists and events will have an associated event history, consisting of a series of events, and an access control list, consisting of a series of access control entries.

The entity types managed by each service can be found listed under **1.4.4 Entities and services** and in the rationale to **R2.4.9**. For example, the record service manages aggregations, records and components, while the disposal scheduling service manages disposal schedules. Where services are not logically separated within an MCRS, then it must manage entities belonging to all of the MoReq2010® entity types collectively.

MoReq2010® allows for specialised sub-types of each entity type. For example, Metadata element definitions are divided into:

- System metadata element definitions, and
- Contextual metadata element definitions.

Each of these is a sub-type of the base entity type Metadata element definition. Sub-types are usually characterised by having extra system metadata elements, compared to the base entity type, and by additional rules of behaviour specified by the requirements.

Some entity types within the MoReq2010® core services are intentionally designed to be a base type for entity sub-types. In addition to metadata element definitions, these are:

- **Classes** – class sub-types are defined by different modules in the MoReq2010®, **200. Classification Series**; and
- **Components** – component sub-types are defined by different plug-in modules in the MoReq2010®, **300. Component Series**.

A Hierarchical Class, defined in the **201. Hierarchical Classification** plug-in module, is an example of a sub-type of the Class entity type. An Electronic Component, defined by the **301. Electronic Components** plug-in module, is an example of a sub-type of the Component entity type.

Extension modules to MoReq2010® may add new entity types and sub-types.

### 2.2.5 Anatomy of an entity

Most entities have three sets of information associated with them, shown in **Figure 2d**:

- **Metadata** – information that describes the entity, contained in metadata element definitions, and divided into system metadata (defined by MoReq2010®) and contextual metadata (defined by the supplier and/or the user);
- **Event History** – a set of events, associated with the entity, that store information about the different functions that have been performed on the entity; and
- **Access control list** – a list of access control entries specifying which users and groups can perform functions on the entity, where specific sets of functionality are collectively defined as roles.

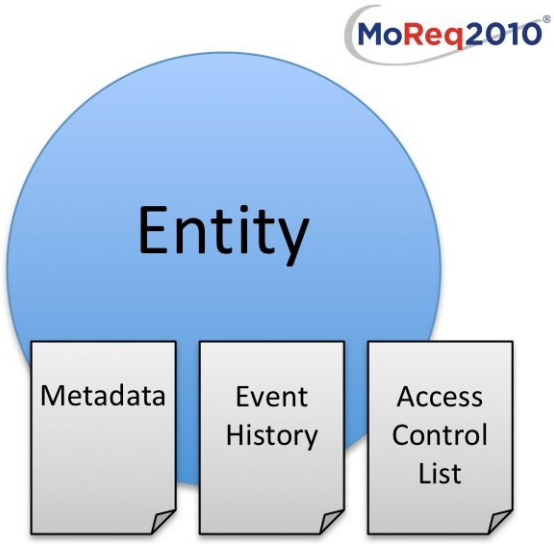


***Figure 2d** – Each entity has associated metadata, an event history and an access control list*

Further information on how metadata is managed in MoReq2010® can be found in **7. Model Metadata Service** while event histories are discussed below. Information on access control can be found in **4. Model Role Service**.

Events and access control entries share the event history and access control list of the entity to which they belong. Components share the access control list of the record entity to which they belong.

Each MoReq2010® core service, is not just a container for entities of particular entity types, but is itself regarded as an entity with its own metadata, event history and an access control list which is inherited by all the entities in that service. The composition of entities and their relationship with services is shown in **Figure 2e**.

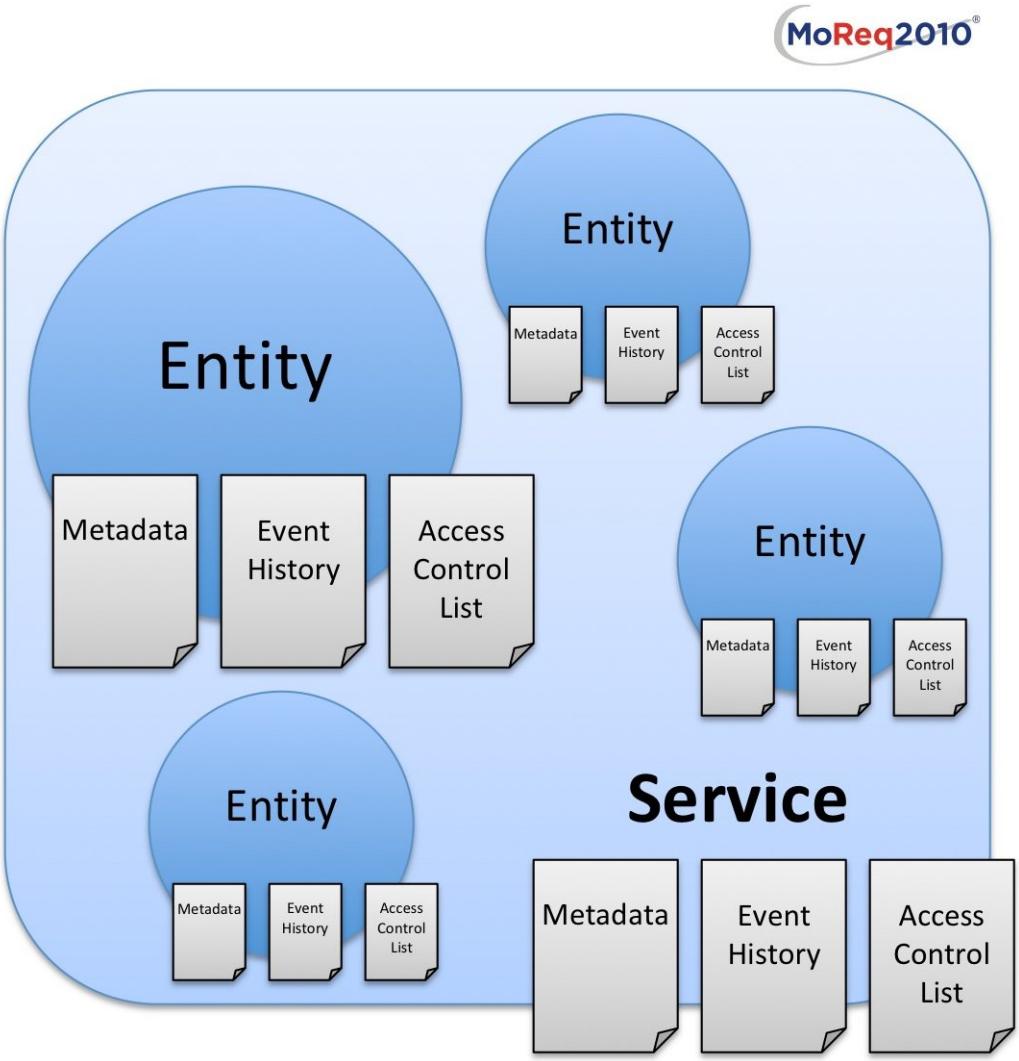


***Figure 2e** – A service contains entities with their own metadata, event history and ACL, but is itself considered an entity with metadata, event history and ACL*

### 2.2.6 Identifying entities

Perhaps the most important metadata element for any entity is its System Identifier. MoReq2010® requires universally unique identifiers (UUID) for each entity and each service within an MCRS. The use of a UUID is mandatory for compliance with the specification. This means that any entity can be exported from one MCRS and imported into another MCRS and will continue to be uniquely identified. The importing MCRS can even match up different copies of the original entity that were exported at different times or were transferred to it via an intermediate MCRS. All entities can be traced back to the specific instance of their originating service where they were created.

### 2.2.7 Performing functions

Users manipulate the entities in an MCRS by performing functions on them. Sometimes it is the MCRS itself that performs the function such as when it generates a new system identifier for a service at installation.

MoReq2010® is a requirements specification and each function that can be performed to an entity in an MCRS can be traced back to one or more of the functional requirements.

Users may only perform functions for entities where they have sufficient authority to do so.

In accordance with **4. Model Role Service**, the authority to perform a function comes from associating a role with a user or a group to which the user belongs. Roles are then assigned to an individual user or to a group using an access control entry which becomes part of either a service or an entity's access control list.

As discussed in 2.2.2, above, some MCRS solutions may deviate from the specific requirements of the model role service, but all MCRS solutions must be able to provide a similar and compatible degree of functionality. The meaning of a model role service is discussed further in **4. Model Role Service**.

### 2.2.8 Event histories

Each entity in an MCRS has an event history, made up of a series of events that have occurred to that entity. Whenever a function is performed, whether by a user or by the system, in which the entity is a participating entity, an event is generated and added to that entity's event history. Each event in an event history therefore correlates to a single function that has been performed in the MCRS.

So as to avoid event histories growing too big, or being filled with trivial events, MoReq2010® includes provision for an authorised user to turn off event generation for selected functions.

The metadata for an event is always set by the MCRS and must not be allowed to be altered by a user. Events do not have an event history.

Different events will have different metadata depending on the function that has been performed to generate the event. This makes it possible for an event to appear in the event history of more than one participating entity. For example,

- If an authorised user changes the name of an aggregation, under **R6.5.3**, then there is only one participating entity (the aggregation). The event (**F14.5.17 Aggregation – Modify Metadata**) will appear only in the event history of the aggregation.
- If an authorised user creates a record in an aggregation, under **R6.5.10**, then there are two participating entities (both the aggregation and the record) and the same event (**F14.5.121 Record – Create**) will appear in the event histories of both.
- If an authorised user moves a record from one aggregation to another, under **R6.5.13**, then there are three participating entities (the previous parent aggregation, the new parent aggregation and the record). The event (**F14.5.3 Aggregation – Add Record**) will have three participating entities and the single event will appear in three event histories simultaneously.
- Under **R6.5.21**, a record is always a participating entity in functions that are performed on its components so that the events generated always appear in the event histories of both the component and the record to which it belongs.

**Figure 2f** shows how an event may belong to more than one entity's event history.

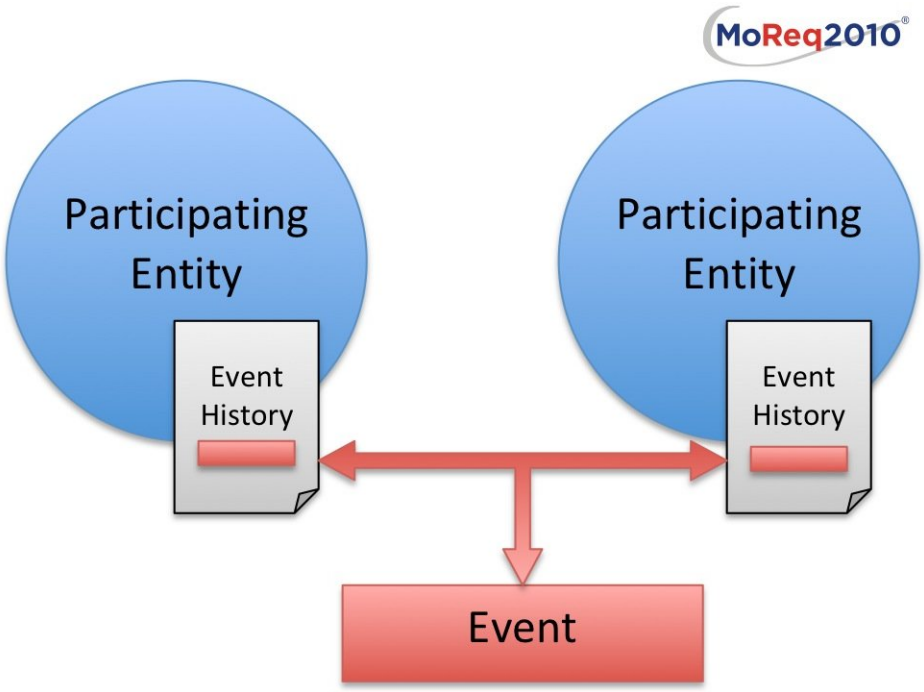


***Figure 2f** – The same event entity can appear in more than one event history*

A traditional audit trail may be conceptualised as a view of all events from the event histories of all entities across the whole MCRS (in timestamp order).

### 2.2.9 Timestamps

MoReq2010® has particular features that allow records systems to interoperate at a universal level. One of these is the use of timestamps.

The specification requires that timestamps must be applied by the MCRS as metadata to accompany every event it generates. For example, each entity has a created timestamp that indicates when the entity was created.

Timestamps must contain complete and accurate date and time data, including time zone information, that allows events to be ordered in the sequence in which they occurred. If an MCRS is capable of performing many events in a second then the MCRS should provide millisecond or better precision so that events remain in order when sorted by timestamp.

Timestamps support interoperability by allowing entities to be successfully transferred to another MCRS in another time zone.

### 2.2.10 Universal language support

Another universal feature of MoReq2010® is its support for Unicode. All textual metadata elements must be in Unicode format and must be accompanied by a language identifier. An MCRS may support only one or a limited number of languages. Nevertheless, so as to support interoperability, a language identifier must be captured for all textual metadata.

### 2.2.11 Lifecycle of an entity

Regardless of its entity type, all entities in an MCRS have a similar lifecycle. An outline of an entity's lifecycle is shown using the timeline in **Figure 1g**.

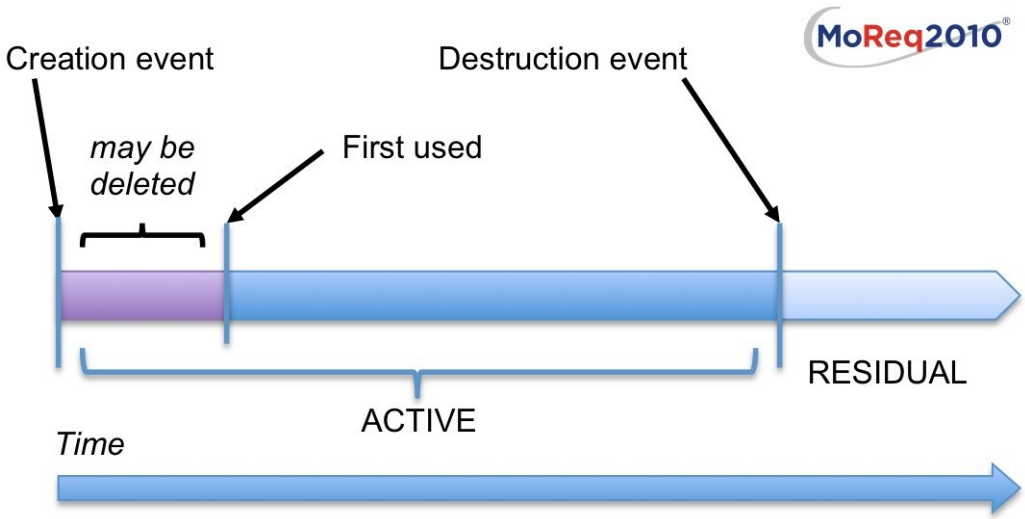


***Figure 2g** – Each entity in an MCRS follows a similar lifecycle*

Each entity will be created in the MCRS, so that its first event is always its creation event. The entity will then remain active until it is destroyed, generating a destruction event. Following its destruction the MCRS will still keep a residual entity to indicate that the entity previously existed in the MCRS.

All MCRS solutions must retain residual entities. Destruction is different to deletion, where all trace of the entity is erased. It is not possible to delete entities from an MCRS as if they had never been created unless they are deleted before they are used. Entities that have been used may not be deleted.

Some entities, most importantly records and their components, (but also entities such as events, access control entries, system metadata element definitions, and so on) do not have a first used timestamp and may never be deleted. Once they have been created, entities of these entity types may never be erased without trace from an MCRS. The lifecycle of a record is explained in more detail in **6. Record Service** and **8. Disposal Scheduling Service**.

## 2.4 Functional Requirements

### R2.4.1

An MCRS must implement the functionality of:

- A user and group service,
- A role service,
- A classification service,
- A record service,
- A metadata service,
- A disposal scheduling service,
- A disposal holding service,
- A searching and reporting service, and
- An export service.

Each service may be implemented individually or several services may be bundled together.

*Functional requirements for each of these services may be found in the MoReq2010® core services. The MCRS may also implement additional functionality as defined in the extension modules to MoReq2010®.*

*The MCRS may share any of its services, except its record service, with other records systems. MoReq2010® does not specify how services are shared between records systems.*

*The MCRS must implement the functionality described by these services, however there is no requirement to implement them as discrete services. The MCRS may create each service separately or it may combine the functionality of several services together into a bundle of services. The whole MCRS may represent a single bundle of services.*

#### R2.4.2

On installation, the MCRS must initialise the following metadata for each service (**E14.2.14**), or bundle of services under **R2.4.1**:

- System Identifier (**M14.4.100**),
- Implements Service Identifier (**M14.4.42**),
- Implements Module Identifier (**M14.4.41**),
- MCRS Certification Identifier (**M14.4.54**),
- Supplier Information (**M14.4.99**),
- Default Language Identifier (**M14.4.12**),
- Title (**M14.4.104**),
- Description (**M14.4.16**), and
- Owner Information (**M14.4.62**).

Each service, or bundle of services, also has:

- Entity types for each entity managed by the service or bundle,
- Entities that are managed by the service or bundle,
- Event history for the service or bundle,
- Access control list (***or equivalent***, see **4. Model Role Service**),

And may have:

- Contextual metadata (***or equivalent***, see **7. Model Metadata Service**)

*Each service or bundle of services has its own metadata, event history and access control list. Each value of the metadata element Implements Service Identifier must match a service published with the MoReq2010® specification (the value of the identifier is given in the service information block at the head of each section from **3. User and Group Service** onwards). Each Implements Module Identifier must match a module published with the MoReq2010® specification. Each MCRS Certification Identifier must match a certificate issued by the DLM Forum® to an MCRS that has passed compatibility testing with an accredited test centre and been certified.*

*Supplier Information should describe the supplier, the product and the version of the product installed. It may also contain other useful information about the supplier such as contact information and the URI for the product's support site.*

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, a MoReq2010® access control list may not be present during system operation and may only be added to a service at export.*

*Depending on the approach taken by the MCRS in implementing **7. Model Metadata Service**, the mechanism by which contextual metadata is added to services may vary.*

#### R2.4.3

The MCRS must allow an authorised user to browse across its services, or bundles of services under **R2.4.1**, and inspect the metadata of each as listed under **R2.4.2**.

*The terms "browse" and "inspect" are defined in **13. Glossary of Terms**.*

*The user may browse each service or bundle of services. If the MCRS combines all functionality into a single bundle then there will be only one set of metadata to inspect for the system as a whole.*

*Function reference: **F14.5.158***

#### R2.4.4

The MCRS must allow an authorised user to modify the metadata for each service, or bundle of services under **R2.4.1**, including:

- Title,
- Description,
- Owner Information, and
- Contextual metadata (if any).

*The Owner Information gives information about the organisation or organisations using the MCRS, and may include help desk or contact information. The Title and Description should provide the local name of the MCRS and additional descriptive information.*

*Function reference: **F14.5.162***

#### R2.4.5

The MCRS must allow an authorised user to generate a MoReq2010® compliance report listing each of its active services, or bundles of services under **R2.4.1**, and for each service or bundle listing the metadata for that service or bundle, under **R2.4.2**.

*The report should indicate which services are implemented individually, and where several services are bundled together.*

*This requirement is intended to provide authorised users with assurance that an installed MCRS is MoReq2010® compliant when implemented at a particular site for a particular organisation. For this reason the values of the Implements Service Identifier, the Implements Module Identifier and the MCRS Certification Identifier metadata elements listed against each service, must accurately report the compliance status of the particular MCRS installation.*

*The report must do more than indicate whether the MCRS has historically passed MoReq2010® compliance testing, the report must also indicate if the MCRS has been installed and configured correctly and is compliant with these services and modules at the time when the report is generated. MoReq2010® does not specify how each individual MCRS should check its internal consistency and configuration after installation, however this is an important requirement for operating an MCRS in a regulatory compliance environment.*

*Function reference: **F14.5.163***

#### R2.4.6

The MCRS must ensure that each service, or bundle of services under **R2.4.1**, has an interface that implements one of the modules of the MoReq2010®, **100. Interface Series**.

*An MCRS, or different services within the MCRS, may implement more than one of the 100 series interface modules, but each service or bundle of services, must be fully compliant with and tested against at least one interface module in the 100 series.*

#### R2.4.7

Whenever the MCRS fails to complete a function requested by itself or an authorised user the MCRS must write at least the following error information to an external log:

- The date/time of the failure;
- The system identifier of the function that was attempted;
- The system identifier of the authorised user who initiated the function;
- The system identifier(s) of any participating entities; and
- Extended error information explaining the failure.

*This requirement refers to an error log that is not maintained within the MCRS itself. The purpose of keeping a log external to the system is to make it accessible even when the MCRS is not available.*

*Extended error information may be generally defined as diagnostic information including error codes, detail about the system state at the time the error occurred, and software exceptions that were encountered. MoReq2010® does not specify what extended error information should be provided by an MCRS*

*Note that functions must be performed atomically. MoReq2010® does not permit functions to be partially successful as this would leave the MCRS in an undefined state. If a function fails at any point then the MCRS must roll back all changes to its internal state made by the function before they are committed. MoReq2010® does not specify how this is done.*

#### R2.4.8

Following the failure of a function requested by a user, under **R2.4.7**, the MCRS must provide a means by which the user can retrieve extended error information about the failed function without accessing the external log.

*MoReq2010® does not specify the mechanism by which an MCRS should provide extended error information following a failed function by a user. (This functionality is not included in the function definitions in **14.5 Function Definitions**.)*

*The way in which the MCRS communicates a failed function back to the user will be dependent on the type of interface the MCRS provides, see **R2.4.6**.*

#### R2.4.9

The MCRS must allow an authorised user to browse the entity types associated with each service, or bundle of services under **R2.4.1**, and inspect their metadata.

*The following entity types are associated with each service:*

- *The user and group service manages user entities and groups;*
- *The role service manages roles;*
- *The classification service manages classes;*

- *The record service manages aggregations, records and components;*
- *The metadata service manages metadata element definitions and templates;*
- *The disposal scheduling service manages disposal schedules; and*
- *The disposal holding service manages disposal holds.*

*Function definitions and events may be found in every service listed above, that manages one or more entity types.*

*For more information on each of these entity types, see **14.2 Entity Types**.*

*Function reference: **F14.5.83***

#### R2.4.10

Each entity type (**E14.2.7**) under **R2.4.9**, must have the following metadata:

- System Identifier (**M14.4.100**),
- Title (**M14.4.104**), and
- Description (**M14.4.16**).

Each entity type, also has:

- System metadata element definitions for that entity type,
- Function definitions for that entity type,
- Event history, and
- Access control list (***or equivalent***, see **4. Model Role Service**),

*System identifiers for every entity type in MoReq2010®, along with default titles and descriptions can be found in **14.2 Entity Types**. The MCRS must always use the MoReq2010® provided system identifiers and must not generate its own system identifiers for entity types. However, the default titles and descriptions may be replaced by the MCRS with localised values.*

#### R2.4.11

For each entity type, under **R2.4.10**, the MCRS must allow an authorised user to browse the function definition entities associated with that entity type and inspect their metadata.

*For comprehensive lists of the function definitions associated with each entity type, see **14.5 Function Definitions**.*

*Function reference: **F14.5.87***

#### R2.4.12

Each function definition (**E14.2.9**) under **R2.4.11**, must have the following metadata:

- System Identifier (**M14.4.100**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),
- Generate Event Flag (**M14.4.34**), and
- Retain On Destruction Flag (**M14.4.88**).

Each function definition, also has:

- System metadata elements to be added to events,
- Event history, and
- Access control list (***or equivalent***, see **4. Model Role Service**),

*System Identifiers for every function definition in MoReq2010®, along with default Titles, Descriptions and the entity types they apply to, can be found in **14.5 Function Definitions**.*

*The MCRS must always use the MoReq2010® System Identifiers and must not generate its own System Identifiers for function definitions. However, the default Titles and Descriptions of function definitions may be replaced by the MCRS with localised values.*

#### R2.4.13

For each function definition, under **R2.4.11**, the MCRS must allow an authorised user to specify whether or not an event should be generated by the MCRS when the function is performed.

*The value is stored in the Generate Event Flag, see **R2.4.12**.*

*Performing the function described by this requirement, and changing the value of the Generate Event Flag, must always result in an event being generated, as explained by **R2.4.14** below. Events are generated and added to event histories under, **R2.4.15** below.*

*In addition, performing the functions described in **R2.4.21**, **R3.4.4** and **R7.5.7** will always result in an event being generated, regardless of the value of the Generate Event Flag.*

*Function reference: **F14.5.91***

#### R2.4.14

An event must always be generated by the MCRS when the function described by **R2.4.13** is performed.

*This requirement is an exception to **R2.4.13** as the generation of an event for this function must not be able to be suppressed.*

*The purpose of this requirement is to ensure that a history is kept by the MCRS of all changes to the settings for whether or not particular functions are captured as events and the authorised users who made those changes.*

#### R2.4.15

Whenever a function described in MoReq2010® is performed for any entity in the MCRS, and subject to the Generate Event Flag described under **R2.4.13**, then the MCRS must automatically create a new event that describes the function that was performed and include it in the event history of all participating entities.

*The MCRS must maintain an event history as part of the metadata of every entity in the MCRS.*

#### R2.4.16

For each event (**E14.2.8**) created under **R2.4.15**, by a function being performed, the MCRS must include the following metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Event Timestamp (**M14.4.27**), and
- Event Function Identifier (**M14.4.26**).

Where the function is performed by a user, and not the MCRS itself, the event will also include:

- Performed By User Identifier (**M14.4.83**),

And may include, under **R2.4.18**:

- Event Comment (**M14.4.25**).

Where the event has been duplicated under **R6.5.16**, it will also include:

- Duplicate Identifier (**M14.4.23)**.

Where the event modifies the metadata of the entity, under **R2.4.17**, it will include:

- Metadata Change Entry (**D14.3.3**).

Events must also have one or more of the following additional system metadata elements depending on the function that they represent, as specified by the description of each function given in **14.5 Function Definitions**:

- Applied Template Identifier (**M14.4.2**),
- Deleted Event Function Definition Identifier (**M14.4.14**),
- Deleted Metadata Function Definition Identifier (**M14.4.15**),
- Export Commencing Timestamp (**M14.4.28**),
- Export Completed Timestamp (**M14.4.29**),
- Export Identifier (**M14.4.30**),
- Exported In Full Flag (**M14.4.31**),
- Granted Role Identifier (**M14.4.35**),
- Historical Date/Time (**M14.4.40**),
- Overdue Disposal Action Code (**M14.4.58**),
- Overdue Disposal Action Due Date (**M14.4.59**),
- Overdue Disposal Confirmation Due Date (**M14.4.60**),
- Participating Aggregation Identifier (**M14.4.64**),
- Participating Class Identifier (**M14.4.65**),
- Participating Component Identifier (**M14.4.66**),
- Participating Disposal Hold Identifier (**M14.4.67**),
- Participating Disposal Schedule Identifier (**M14.4.68**),
- Participating Duplicate Identifier (**M14.4.69**),
- Participating Entity Type Identifier (**M14.4.70**),
- Participating Event Identifier (**M14.4.71**),
- Participating Function Definition Identifier (**M14.4.72**),
- Participating Group Identifier (**M14.4.73**),
- Participating Metadata Element Definition Identifier (**M14.4.74**),
- Participating New Parent Identifier (**M14.4.75**),
- Participating Previous Parent Identifier (**M14.4.76**),
- Participating Record Identifier (**M14.4.77**),
- Participating Role Identifier (**M14.4.78**),
- Participating Service Identifier (**M14.4.79**),
- Participating Template Identifier (**M14.4.80**),
- Participating User Identifier (**M14.4.81**),
- Participating User or Group Identifier (**M14.4.82**),

- Rescinded Role Identifier (**M14.4.87**),
- Search Query (**M14.4.98**), and
- Total Entities (**M14.4.105**).

*The Event Function Identifier must be a reference to the function definition of the function which was performed by the user, referenced by the Performed By User Identifier. The Event Timestamp reflects when the function was performed; this will be set by the MCRS, and will usually be the same as the Created Timestamp, unless the event has been imported from another MCRS.*

*The additional metadata elements to be added to an event depend on the function that was performed and are listed as part of the function definition (see **R2.4.12**) and are provided in full in **14.5 Function Definitions**. Depending on the function that was performed the event may require additional metadata. For example, if the function was to export the entity, then the MCRS must include the Export Identifier and the Exported In Full Flag in the event, under **R11.4.8**.*

*Any entity referenced by an identifier that takes the form, "Participating … Identifier" is a participating entity in the event. All functions have at least one participating entity and some functions have several participating entities. The event must appear in the event history of all participating entities.*

#### R2.4.17

Whenever the metadata of an entity is modified as a result of performing a function and an event is generated for the function, under **R2.4.15**, the MCRS must include a metadata change entry (**D14.3.3**) in the event for each metadata value that is modified, with the following metadata:

- Metadata Element Definition Identifier (**M14.4.55**),
- Previous Value (**M14.4.85**), and
- New Value (**M14.4.57**).

*A metadata change entry is a data structure defined in **D14.3.3**, and captures the before and after states of a metadata element belonging to the entity. The Metadata Element Definition Identifier contains a reference to the metadata element that was modified. The Previous Value contains the value of the metadata element before the function was performed. The New Value contains the value of the metadata element after the function was performed. There will be no Previous Value if the value is newly applied to the entity and no New Value if the previous value is deleted.*

*For example, if a user changes the title of an entity then the corresponding event would have a metadata change entry that referenced the Title metadata element definition (**M14.4.104**), its old value before the function was performed, and the new value it was given by the user.*

*Every metadata element that is changed by a function will result in a metadata change entry included in the event. If multiple metadata elements are changed simultaneously by one function then the event will include a metadata change entry for every metadata element that is changed.*

#### R2.4.18

Whenever the MCRS performs a function requested by a user, and not by itself, which changes the metadata of an entity, the MCRS must allow the user to provide a comment explaining why the function was performed. If provided, the comment must be included in the event as the Event Comment, under **R2.4.16**.

*The MCRS is not required to make provision for the user to enter an Event Comment for functions which do not change the metadata of the entity, such as browsing the entity and inspecting its metadata. Note that changing the metadata of an entity will result in the generation of one or more metadata change entries, under* ***R2.4.17****.*

*Providing a comment is optional in most cases. However, some functions require a comment by the user (for example, see* ***R2.4.21****,* ***R2.4.26****,* ***R8.4.17****,* ***R8.4.18****,* ***R7.5.7*** *and* ***R11.4.10****).*

*While the MCRS is not required to provide an Event Comment for functions it performs itself, under* ***R2.4.16****, the MCRS may be implemented in such a way that it automatically generates an Event Comment while performing such functions and includes it in the event.*

#### R2.4.19

The MCRS must allow an authorised user to browse the event history of an entity in order of Event Timestamp, and inspect the metadata of each event.

*The terms "browse" and "inspect" are defined in* ***13. Glossary of Terms****. Event histories are most usually browsed from the most recent events to the earliest events in descending order of Event Timestamp. The MCRS may also provide other ways of browsing the event history.*

*Function references:* ***F14.5.14****,* ***F14.5.32****,* ***F14.5.45****,* ***F14.5.65****,* ***F14.5.79****,* ***F14.5.85****,* ***F14.5.89****,* ***F14.5.103****,* ***F14.5.111****,* ***F14.5.133****,* ***F14.5.151****,* ***F14.5.160****,* ***F14.5.173****,* ***F14.5.189***

#### R2.4.20

For each function definition, under **R2.4.11**, the MCRS must allow an authorised user to specify whether or not an event generated by the MCRS, when the function was performed, should be retained when the entity it belongs to is destroyed.

*The value is stored in the Retain On Destruction Flag, see* ***R2.4.12****.*

*Events for functions where the Retain On Destruction Flag is not set will be pruned from an entity's event history when the entity is destroyed. They will not be part of the event history of the residual entity.*

*Automatic pruning of events for residual entities may be done for a number of reasons, including but not limited to:*

- *Ensuring destruction – to remove events whose metadata, especially as a result of the information stored in metadata change entries, under* ***R2.4.17****, may make it possible to partially or fully reconstitute the entity,*
- *Privacy – to remove events that may potentially store personal information, and*
- *Storage - to reduce the footprint of a residual entity.*

*Note that it must not be possible to make a residual entity active once it has been destroyed.*

*Events are only automatically pruned on the destruction of the participating entity against which the function for the event was originally run (see the rationale to* ***R2.4.21****). This is only applicable for events where there there is more than one participating entity. See* ***14.5 Function Definitions****.*

*Function reference:* ***F14.5.92***

### R2.4.21

The MCRS must allow an authorised user to delete an event from the event history of a residual entity, provided the user gives a reason for the deletion and an event is generated.

*This requirement describes a special circumstance where it becomes necessary to remove metadata or events from individual entities, for example, where this has been ordered by a court of law. It should not be necessary to exercise this requirement as part of routine records management processes.*

*This requirement may only be applied to a residual entity which has already been destroyed, and is in addition to the automatic pruning of metadata and events which occurs on destruction.*

*Note that a new event must **always** be generated for this function, superseding requirement **R2.4.13**. The authorised user must give a reason for deleting the deleted event which is stored as the event comment in the new event.*

*The new event must also contain a Deleted Event Function Definition Identifier (see **M14.4.14**) which indicates the function identifier of the deleted event. This gives an indication of which type of event was deleted without retaining any of the metadata of the deleted event.*

*Where the event to be deleted has more than one participating entity, the event must be deleted from the entity against which the function was originally performed. This is also the participating entity for the event notification of the deletion. For example, to delete the event where a record is moved from one aggregation to another, it must be deleted from the record's new parent aggregation, because this is the entity against which the function was performed and the event generated, see **F14.4.3 Aggregation – New Record**.*

*See also **R7.5.7**.*

*Function references: **F14.5.7**, **F14.5.26**, **F14.5.39**, **F14.5.50**, **F14.5.59**, **F14.5.73**, **F14.5.97**, **F14.5.122**, **F14.5.145**, **F14.5.167**, **F14.5.181***

### R2.4.22

When an authorised user is browsing a set of entities, the MCRS must by default, limit the set to active entities only, unless the user specifically chooses to browse both active and residual entities.

*The term "browse" is defined in **13. Glossary of Terms**.*

*By default, entities that have been destroyed will not be browsed. See also **R10.4.17** and **R11.4.2**.*

### R2.4.23

The MCRS must always use the system identifiers that accompany the MoReq2010® specification, where provided.

*MoReq2010® provides system identifiers that have been previously generated for:*

- *Services and modules (see **2.1 Service Information**);*
- *Entity types (see **R2.4.10** and **14.2 Entity Types**);*
- *Function definitions (see **R2.4.12** and **14.5 Function Definitions**); and*
- *System metadata element definitions (see **R7.5.1** and **14.4 System Metadata Element Definitions**).*

*The system identifiers provided with MoReq2010® must be used to ensure interoperability with other records systems.*

**R2.4.24**

The MCRS must generate System Identifiers for new entities as universally unique identifiers (UUID) and must not allow these identifiers to be modified.

*MoReq2010® does not specify which algorithm the MCRS should use to generate System Identifiers. The approaches listed in RFC4122 are recommended and can support "high allocation rates of up to 10 million per second per machine" (RFC4122:2005, 2.).*

**R2.4.25**

The MCRS must automatically set the Created Timestamp and, where it exists, the Originated Date/Time for all new entities on creation.

*By default, both the Created Timestamp and the Originated Date/Time will reflect the date and time of creation of the entity.*

*Function references:* ***F14.5.5***, ***F14.5.24***, ***F14.5.38***, ***F14.5.48***, ***F14.5.57***, ***F14.5.71***, ***F14.5.95***, ***F14.5.121***, ***F14.5.143***, ***F14.5.165***, ***F14.5.179***

**R2.4.26**

Where it exists, the MCRS must allow an authorised user to change the Originated Date/Time for an active entity to an earlier date and time than the Created Timestamp, provided a the user gives a reason for the change.

*By default the Originated Date/Time is set to the same date and time as the Created Timestamp, under* ***R2.4.25****. An authorised user can subsequently change this value for an active entity to reflect an earlier date and time, but never a later date and time. Whenever a user changes the Originated Date/Time to an earlier date and time a comment must be provided which is stored in the event generated for the function.*

*Note that the Originated Date/Time is a possible retention trigger, under* ***R8.4.4****, and resetting it for an aggregation or record may affect the calculation of one or more retention start dates.*

*Function references:* ***F14.5.18***, ***F14.5.36***, ***F14.5.47***, ***F14.5.54***, ***F14.5.68***, ***F14.5.82***, ***F14.5.106***, ***F14.5.136***, ***F14.5.154***, ***F14.5.176***, ***F14.5.192***

**R2.4.27**

The MCRS must generate timestamps that are compatible with W3C XML dateTimeStamp formats and must always include time zone information in timestamps.

*The required format is defined in W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. The dateTimeStamp format is a variant of the XML date/time format which is in turn a subset of ISO 8601 that does not allow truncated or reduced formats.*

*Time zone information is a necessary inclusion in all timestamps to determine when events occur relative to other events, and to ensure interoperability with records systems in different time zones.*

**R2.4.28**

The MCRS must store all textual metadata elements in Unicode accompanied by a language identifier that is compliant with RFC5646 and the IANA Language Subtag Registry.

*A metadata element is considered textual if the Is Textual Flag is set in the corresponding metadata element definition, see* ***R7.5.1****.*

*MoReq2010® does not specify how an MCRS will identify the language used for textual metadata. Options may include:*

- *In monolingual contexts it may be derived from the default language identifier for the service or bundle of services (see* ***R2.4.2****),*
- *It may be specified in operational procedures or by the nature of the metadata,*
- *It may be derived from a user's client operating system or session settings, or*
- *It may be supplied as additional information by the user.*

*Use of Unicode ensures that all character sets can be represented and is necessary for interoperability. The latest version of the Unicode Standard is 6.0.*

# 3. User and Group Service

## 3.1 Service Information

| | |
|---|---|
| **Service Name** | **User and Group Service** |
| **Service Version** | **1.0** |
| **Implements Service Identifier *(see M14.4.42)*** | cd532472-85b0-4c1c-82b4-5c8370b7d0e6 |

## 3.2 Key Concepts

### 3.2.1 MoReq2010® Approach to User and Group Management

Good management of users and groups is essential to the successful operation of a records system. All business systems share this same need. As a consequence there are many system tools available that manage users and groups and this functionality is often built directly into computer operating systems. There are also well established standards for management of users and groups and related directory services, such as X.500, and for user authentication, such as OpenID.

For this reason MoReq2010® does not mandate the protocols that MCRS solutions should use for user authentication and user and group management. The user and group service in MoReq2010® is a set of requirements that act as a wrapper allowing the use of either an external corporate directory service or a custom directory service built into the MCRS. Other than the basic concepts of a user and a group, described below, MoReq2010® is not prescriptive about how users and groups are managed.

### 3.2.2 Requirements for Records Management

Traditional directory services do not, by themselves, provide all the functionality required for the discipline of records management. The data held within a directory service is often transient in nature. There is usually little or no ability to be able to trace past users of a system. The system identifiers for users and groups may not be universal or they may be subject to change, for example, if an external directory service is duplicated or rebuilt. Importantly the data in a proprietary directory service may not be in a common format that can be easily understood by another records system when records are transferred. These factors may make it difficult from an historical perspective to determine which users performed particular actions, who they were and what groups they belonged to.

MoReq2010® therefore requires that the MCRS keep additional and stable data about users and groups including historical information. This includes creating entities to represent all users and groups within the MCRS, using universal MoReq2010® system identifiers, tracking changes to the metadata of entities, and retaining information by destroying users and groups to leave residual entities, rather than deleting them entirely from the system when they are no longer active.

If the MCRS uses an external directory for user and group management then it will need to synchronise or otherwise integrate with it to ensure that this information is captured, updated regularly and retained following its removal from the external system. MoReq2010® does not specify how this should be done.

### 3.2.3 How Users and Groups Work

**Figure 3a** shows the many-to-many relationship between users and groups in an MCRS. Each user may belong to many groups and many users may belong to the same group.

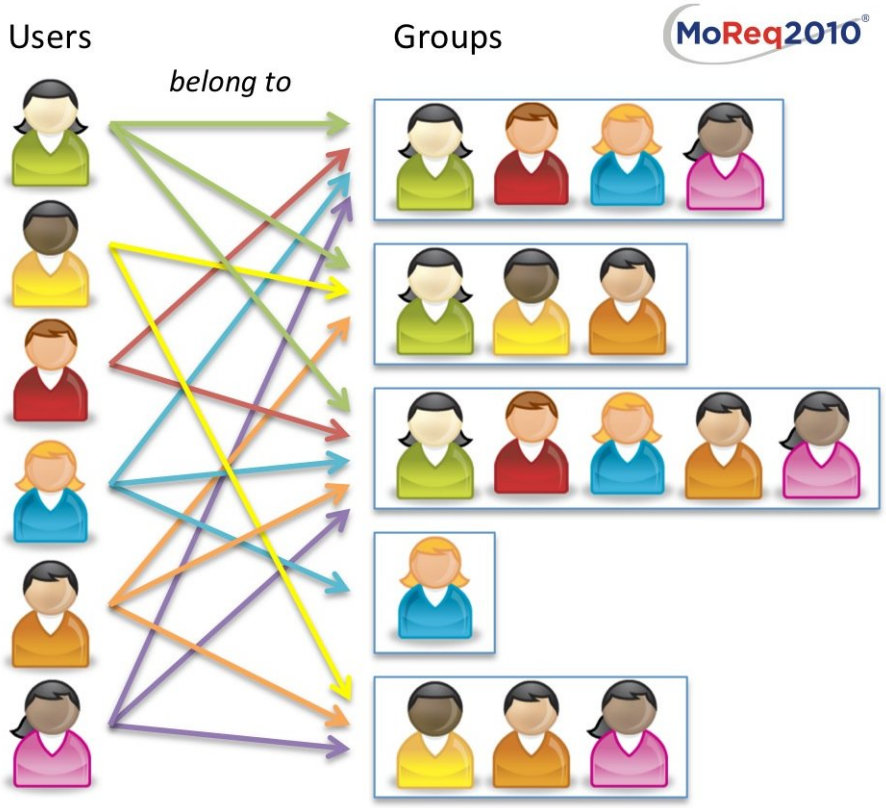


***Figure 3a** – In an MCRS, users and groups have a many-to-many relationship*

This flattened structure represents the extent of the information that MoReq2010® requires about the relationship between users and groups. There is no requirement to track the relationships between groups even though, for example, several of the groups shown in **Figure 3a** may in practice be sub-groups of another group.

### 3.2.4 Destroying users and groups

The user and group service must adopt the approach common throughout the MoReq2010® specification of destroying entities to leave a residual entity. This is so that a record is kept of these entities even when they are no longer functioning. MoReq2010® requires that user and group entities must not be deleted from the MCRS once they have been used, and must instead be retained to provide context to the records managed by the system. See the previous discussion under **2.2.10 Lifecycle of an entity**.

A residual entity does not exercise the same functionality as an active entity. In the case of a user entity that has been destroyed, a residual user must not be allowed to access the MCRS.

In the case of residual group entities, any roles they have been granted no longer apply to the members of that group. Therefore, an active user that is a member of the residual group, will no longer benefit by inheriting any roles allocated to that group. For further details, see **3. Model Role Service**.

## 3.4 Functional Requirements

### R3.4.1

The MCRS must only be accessed by users who have been authenticated and for whom there exists an active user entity (**E14.2.16**) with at least the following system metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- First Used Timestamp (**M14.4.32**),
- Group Identifier (**M14.4.36**),
- Title (**M14.4.104**),
- Description (**M14.4.16**), and
- Destroyed Timestamp (**M14.4.17**).

Each user entity also has:

- Event history (see **2. System Services**),
- Access control list (***or equivalent***, see **4. Model Role Service**),

And may have:

- Contextual metadata (***or equivalent***, see **7. Model Metadata Service**).

*Authentication describes the process of establishing the identity of the user, so that the MCRS can allow an appropriate level of access to perform functions, and can associate every function that is performed with a particular user entity. MoReq2010® does not define how authentication should occur. A simple form of authentication is to require that the user provide a name and password, and this may be suitable for many records systems. Other more complex methods include two factor authentication, where more than one item of identification is required.*

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, a MoReq2010® access control list may not be present during system operation and may only be added to a user entity at export.*

*Depending on the approach taken by the MCRS in implementing **7. Model Metadata Service**, the mechanism by which contextual metadata is added to user entities may vary.*

### R3.4.2

The MCRS must support a process for creating new user entities in the MCRS with the metadata and other properties listed under **R3.4.1**.

*MoReq2010® does not define how this process should operate and, depending on the specific implementation, new user entities may not be created by MCRS users. Organisations must assess whether the supplier's process meets their security and operational needs.*

*Function reference: **F14.5.179***

#### R3.4.3

The MCRS must support a process for updating the Title and Description of an active user entity, and any contextual metadata, to reflect changes to the user's details.

*MoReq2010® does not define how this mechanism should be implemented and it may be performed externally to the MCRS and be later synchronised to it, however changes to user metadata must generate an event in the event history of the user, under* ***R2.4.15****, subject to* ***R2.4.13****.*

*Function reference:* ***F14.5.191***

#### R3.4.4

The MCRS must support a process for adding and removing active users to and from active groups and whenever this occurs an event must be generated.

*MoReq2010® does not define how this mechanism should be implemented and it may be performed externally to the MCRS and be later synchronised to it. Only active users may change their group membership, they may only be removed from or added to active groups. In this way a residual user will retain its group membership as it was at destruction; and a residual group will retain its user membership as it was when the group was destroyed.*

*Note that a new event must* ***always*** *be generated for this function, superseding requirement* ***R2.4.13****. Events are always generated so as to provide accurate data for reporting under* ***R3.4.7*** *and* ***R3.4.13****.*

*Function references:* ***F14.5.94****,* ***F14.5.107***

#### R3.4.5

The MCRS must support a process for deleting a user that has never used the MCRS to perform any function.

*MoReq2010® does not define how this mechanism should be implemented and it may be performed externally to the MCRS and be later synchronised to it. When a user first uses the MCRS and performs a function, the MCRS must set the First Used Timestamp.*

*Function reference:* ***F14.5.180***

#### R3.4.6

The MCRS must support a process for destroying a user that has used the MCRS to perform functions.

*MoReq2010® does not define how this mechanism should be implemented and it may be performed externally to the MCRS and be later synchronised to it. Once a user has used the MCRS, the corresponding user entity may not be deleted, it may only be destroyed. Destroying a user will set the Destroyed Timestamp and will leave a residual user entity.*

*Function reference:* ***F14.5.183***

#### R3.4.7

The MCRS must be able to generate a report for an authorised user listing the active groups that a nominated user entity belonged to at a specified historical date and time.

*The report must not include groups that the user requesting the report is not authorised to inspect.*

*The report must indicate if the date and time provided were before the user entity was created or after it was destroyed.*

*Function reference:* ***F14.5.194***

#### R3.4.8

The MCRS must be able to maintain groups (**E14.2.10**) with at least the following system metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- First Used Timestamp (**M14.4.32**),
- Title (**M14.4.104**),
- Description (**M14.4.16**), and
- Destroyed Timestamp (**M14.4.17**).

Each group also has:

- Users that belong to the group,
- Event history (see **2. System Services**),
- Access control list (***or equivalent***, see **4. Model Role Service**),

And may have:

- Contextual metadata (***or equivalent***, see **7. Model Metadata Service**).

*Depending on the approach taken by the MCRS in implementing* ***4. Model Role Service****, a MoReq2010® access control list may not be present during system operation and may only be added to a group entity at export.*

*Depending on the approach taken by the MCRS in implementing* ***7. Model Metadata Service****, the mechanism by which contextual metadata is added to group entities may vary.*

#### R3.4.9

The MCRS must support a process for creating new groups in the MCRS with the metadata and other properties listed under **R3.4.8**.

*MoReq2010® does not define how this process should operate and, depending on the specific implementation, new group entities may not be created by MCRS users.*

*Function reference:* ***F14.5.95***

#### R3.4.10

The MCRS must support a process for updating the Title and Description of an active group, and any contextual metadata, to reflect changes to the group's details.

*MoReq2010® does not define how this mechanism should be implemented and it may be performed externally to the MCRS and be later synchronised to it, however changes to group metadata must generate an event in the event history of the group, under* ***R2.4.15****, subject to* ***R2.4.13****.*

*Function reference:* ***F14.5.105***

#### R3.4.11

The MCRS must support a process for deleting a group that has never had users added to it.

*MoReq2010® does not define how this mechanism should be implemented and it may be performed externally to the MCRS and be later synchronised to it. Users are added to groups under* ***R3.4.4****.*

*Function reference:* ***F14.5.96***

#### R3.4.12

The MCRS must support a process for destroying a group that has had users as members.

*MoReq2010® does not define how this mechanism should be implemented and it may be performed externally to the MCRS and be later synchronised to it. Any group to which users belong, or have previously belonged, may not be deleted, it may only be destroyed. Destroying a group will leave a residual group entity.*

*Under the model role service a user may not inherit roles granted to a residual group to which that user belongs, even if the user is active, see* ***R4.5.10****.*

*Function reference:* ***F14.5.99***

#### R3.4.13

The MCRS must be able to generate a report for an authorised user listing the users that were active and belonged to a nominated group at a specified historical date and time.

*The report must not include user entities that the user requesting the report is not authorised to inspect. The users must have been active on the date and time specified, but do not need to be active users when the report is generated.*

*The report must indicate if the date and time provided were before the group entity was created or after it was destroyed.*

*Function reference:* ***F14.5.108***

#### R3.4.14

Subject to **R2.4.22**, the MCRS must allow an authorised user to browse and inspect users and groups, in at least the following ways:

- Browse across the users in the user and group service and inspect their metadata,
- Browse across the groups in the user and group service and inspect their metadata,
- Browse from a user to the groups to which that user belongs and inspect their metadata,
- Browse from a group to the users that belong to that group and inspect their metadata.

*The terms "browse" and "inspect" are defined in* ***13. Glossary of Terms****.*

*Function references:* ***F14.5.101***, ***F14.5.187***

# 4. Model Role Service

## 4.1 Service Information

| | |
|---|---|
| **Service Name** | **Model Role Service** |
| **Service Version** | **1.0** |
| **Implements Service Identifier *(see M14.4.42)*** | ***For an MCRS that implements the MoReq2010® model role service use:***<br>2f6d05c6-51e6-4a32-a7fc-c0a6883eb85b<br>***For an MCRS that implements its own native permissions model use:***<br>d945dcd9-dc2d-491d-965a-11ce936d044b |

## 4.2 Complying with the Model Role Service

### 4.2.1 Lack of an industry standard for roles and permissions

This module of MoReq2010® contains a definition of a model role service with functional requirements covering how users are authorised to perform functions in the MCRS.

At the time of publication of MoReq2010® there remains no industry-wide standard for applying a permissions model to entities in an information system. Basic permissions models based on create, read, update and delete (CRUD) are too simple for direct application to records systems. For example, CRUD makes no distinction between deletion and destruction, and no allowance for retaining the residual entities, necessary in records systems.

Because there is no industry standard, suppliers have understandably developed their own approaches to controlling user access to records systems. These proprietary methods, while often highly effective, are almost always application specific and do not lend themselves to interoperability between records systems, as one supplier's permissions model may not readily map to another's.

### 4.2.2 A model role service

The MoReq2010® model role service is a simple standardised model that addresses the specific requirements of records systems within the context of the specification. Care has been taken to ensure that the model role service outlined here is both neutral and based around common concepts to be found across many modern information systems, such as access control lists and role definitions.

Nevertheless, the MoReq2010® model role service can only represent one possible approach to access control and it is recognised that it may vary sufficiently from the approach taken by some pre-existing records systems to render them unable to be tested for compliance against MoReq2010® without major redevelopment to their own embedded access control methods.

This introduces a dichotomy: while new records systems have the opportunity to adopt the model role service, it may not prove cost effective for established products in the market.

### 4.2.3 Approaches to testing and certification against the model role service

For the reasons given above, the DLM Forum® allows two possible approaches to the testing and certification of a records system for compliance with the MoReq2010® model role service.

**EITHER**

**A.** The records system implements the MoReq2010® model role service in full, and is tested and certified against the requirements in this module.

**OR**

**B.** The records system implements its own native permissions model, in which case the application must satisfy the following criteria:

- It must demonstrate that its native permissions model is equivalent in flexibility and functionality to the MoReq2010® model role service, and
- It must support interoperability by being able, on export, to convert its native permissions into the XML format used by the model role service, so that users and groups retain equivalent levels of access to entities and the same authority to perform functions when the data is subsequently transferred to another MCRS.

### 4.2.4 How to meet the alternative (type B) requirements

To show that the MCRS's native permissions model is equivalent in flexibility and functionality to the MoReq2010® model role service, the records system must be able to demonstrate all of the following:

- That a user cannot access any entity in the MCRS until that user has been granted the authority to access the entity either individually or as a member of a group;
- That there exists multiple, user configurable, levels of access to entities, including discretionary ability to set the permissions of users so that they can:
  - Discover some entities and not others, and
  - Perform some functions and not others;
- That the authority to access entities and perform functions may be set at the level of an individual entity or once across a whole collection of entities, such as an aggregation of records;
- That the authority to access entities and perform functions may be set differently in different parts of the MCRS, for example, so a the user's authority to perform functions to some classes can be set so as to be different from the same user's authority over other classes within the same classification service;
- That when a new entity is created it is given an appropriate set of default permissions, for example, a new record captured into an aggregation should automatically receive a similar level of permissions as the other records in the aggregation; and

- There exist some types of roles or permissions that may not be blocked by the owners of particular entities, so that users with those permissions can properly administer all or part of an MCRS.

When converting metadata for export from the records system's native permissions model to the MoReq2010® model role service metadata format, the following must be observed:

- No user may be granted authority to access more entities than the user was able to in the original environment,
- No user may be granted authority to perform functions to entities that that user was unable perform to the same entities in the original environment,
- Authority granted through group membership must not be converted into multiple instances of authority granted to individual users,
- The records system must utilise the inheritance features of the MoReq2010® model role service where possible and avoid adding repetitive sets of access control entries to the access control list of every child entity it exports if it is possible to attribute a single set of access control entries to the child's parent entity instead,
- The supplier must describe the conversion algorithm used by the records system and provide information and examples of how the exported data for roles and access control lists has been made to imitate the model role service as accurately as possible, and
- The supplier must make its permissions mapping schema available for inclusion in the full test report of its product.

Further advice and guidance to suppliers can be requested from the MoReq Governance Board through the DLM Forum® secretariat.

The remainder of this module describes the concepts and requirements of the MoReq2010® model role service.

## 4.3 Key Concepts

### 4.3.1 Defining Roles

Before performing any function on any entity, the MCRS must always conduct a preliminary check to determine whether the user requesting the function has the authority to carry it out. The authority to perform functions is given to users when they are granted roles.

A properly authenticated user (see **R3.4.1**) that is accessing the MCRS with the authority to perform a function is defined throughout this specification as an "authorised user".

Because the same role may be granted for any entity in the MCRS, or utilised across several records systems, all role definitions are managed by a role service. A role definition represents a set of function definitions, as shown in **Figure 4a**.

Role definitions and function definitions have a many-to-many relationship; any role may have many function definitions associated with it and the same function definition may be associated with more than one role.

The term role indicates that the set of functions it defines are selected logically so as to collectively describe the authority required by a particular member of staff or a particular position within an organisation, for example, a "Local Records Officer".

Constructing roles in this way, based on coherent sets of functionality is integral to managing a records system. There are many functions defined by the MoReq2010® specification, and it would be impractical to assign them to users and groups individually.

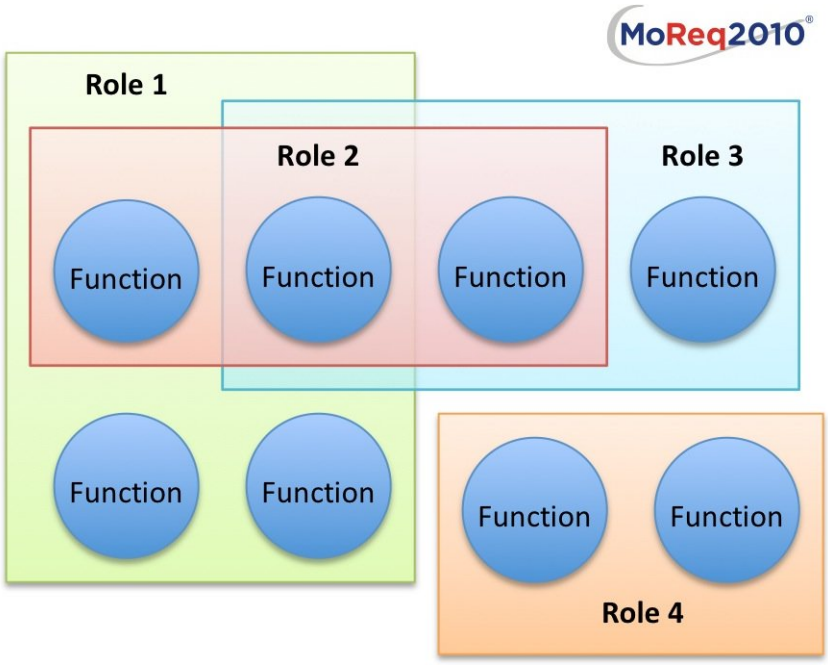


***Figure 4a** – Functions are associated with roles (all functions should be included in at least one role)*

### 4.3.2 Granting Roles

A role may be granted to either a user or to a group in relation to any entity in the MCRS, including a service. When a role is granted, an access control entry nominating the user or group and listing the roles to be granted is created and added to the entity's access control list.

Granting a role to a user in relation to an entity allows that user to perform any of the functions listed in the role definition that are applicable to that entity.

Granting a role to a group in relation to an entity has the effect of granting the role to every user that is a member of that group (see **3. User and Group Service**). New users joining the group will automatically inherit the role, while users that leave the group automatically lose their access to the role, without the role having to be granted or rescinded individually for each user.

Granting roles to groups rather than to individual users is recommended good practice as it makes it easy to manage users' access to entities when they join or leave the organisation, and change jobs, without having to update access control lists for entities in the MCRS. Group management is invariably simpler and less error prone that the micromanagement of continually granting and rescinding roles to individual users.

**Figure 4b** shows how a user or group is granted one or more roles in relation to an entity by adding a new access control entry to the access control list for that entity. All entities in the MCRS, as well as services, have an access control list.

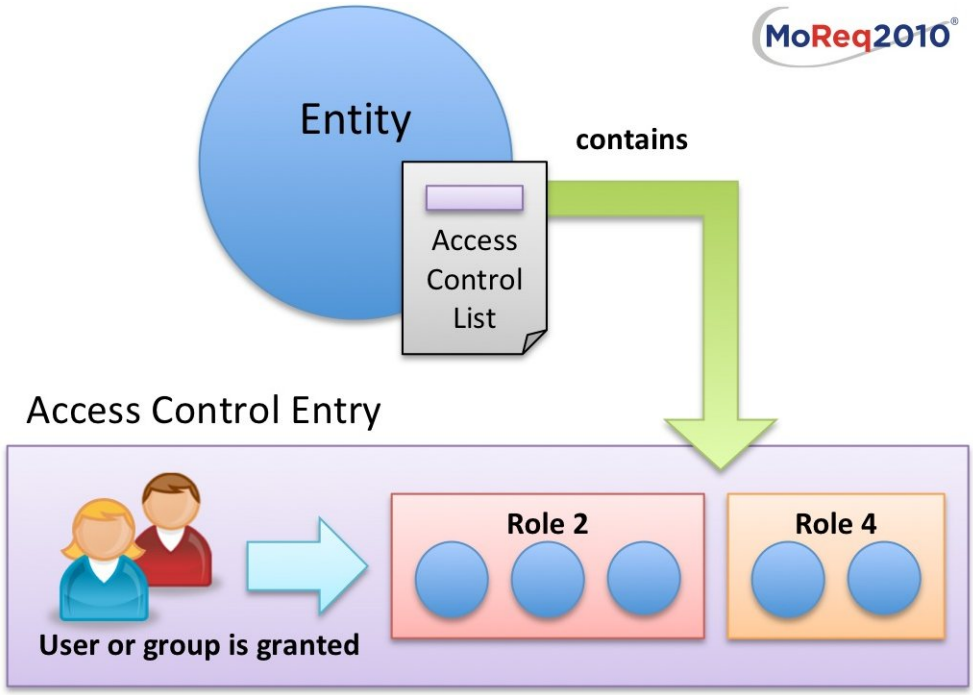


***Figure 4b*** *– An access control list is made up of access control entries that link a user or a group to a role*

### 4.3.3 Inheriting Roles

In addition to the access control entries in its own access control list, an entity may also inherit the roles granted to users and groups from other entities. MoReq2010® indicates in specific functional requirements where such inheritance applies. As a general rule, throughout the specification if a parent/child relationship exists between entities then the child entity will inherit its parent's access control list. For example, a child aggregation inherits from its parent aggregation, as does a record. There are also some circumstances under which entities can inherit from multiple sources, as described below.

Inheritance is a very important mechanism for managing large records systems where granting roles against single entities is impractical.

The usual pattern of inheritance of access control entries can also be broken for some roles if required. MoReq2010® makes provision within each access control list for an Include Inherited Roles Flag that specifies whether roles assigned to parent entities are inherited by the child entity or not.

If the Include Inherited Roles Flag is switched off in an entity's access control list then only administrative roles will be automatically inherited from the parent entity.

### 4.3.4 Administrative Roles

There are two different types of role in MoReq2010®:

- Administrative roles, and
- Non-administrative roles.

The type of role is specified as part of the role definition.

An administrative role, once granted across a service as a whole, or for any parent entity, always applies to all the entities that inherit from that service or entity. In this way, administrative roles override the setting of the Include Inherited Roles Flag for child entities.

By comparison, non-administrative roles will only be inherited by a child entity if the child's access control list is set to include inherited roles.

An example of this is shown in **Figure 4c**. In this example, Role 2 is an administrative role, while Role 4 is set as a non-administrative role. The entity Child 1 is set to include inherited roles and duly inherits both Role 2 and Role 4. However, the entity Child 2 does not include inherited roles and will therefore only inherit the administrative Role 2.

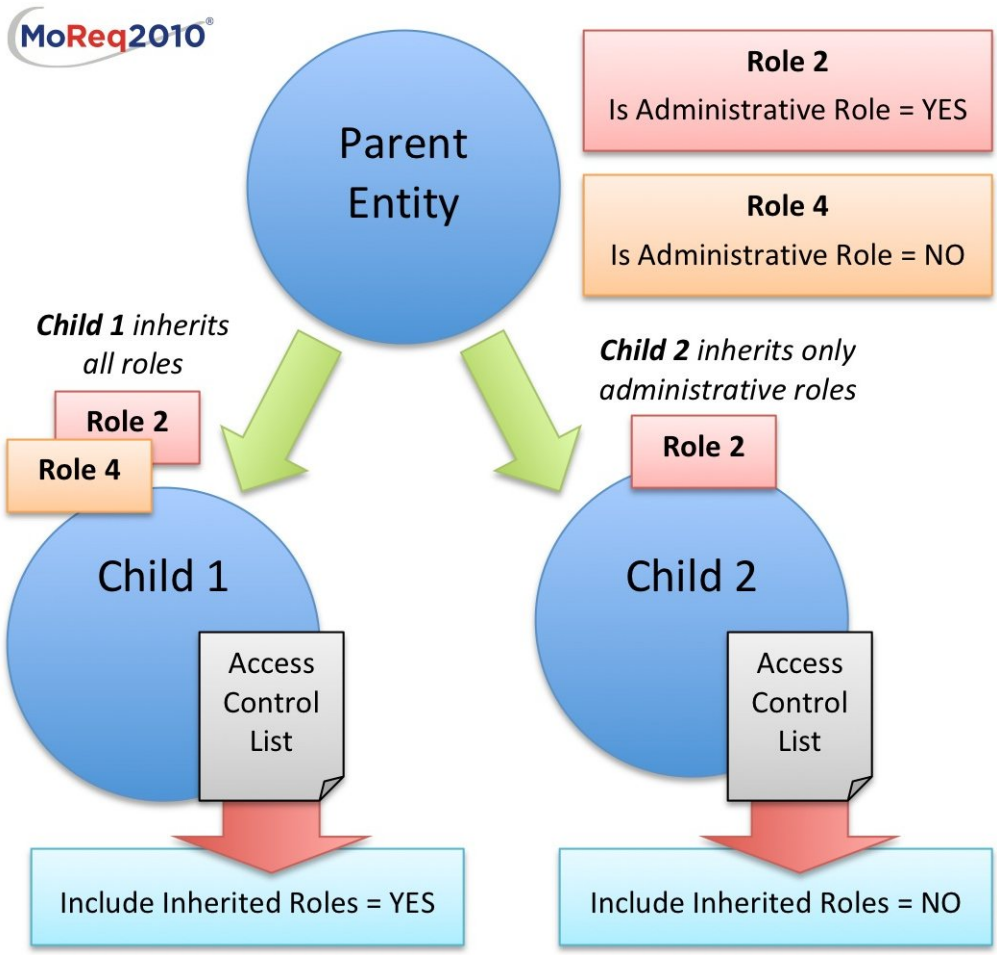


***Figure 4c** – Administrative roles override the operation of the include inherited roles flag and are always inherited from parent entities*

### 4.3.5 Multiple inheritance

As mentioned previously there are some parts of the MoReq1010® specification where an entity may inherit roles from more than one entity.

For example, aggregations and records maintained by the record service will inherit access control settings from both their parent aggregation and from their classification. This is shown in **Figure 4d**. Where this occurs, the entity will inherit the roles from both its parent

and its class. Of course, if the inherit all roles flag is turned off then the child entity will inherit only the administrative roles from these entities.

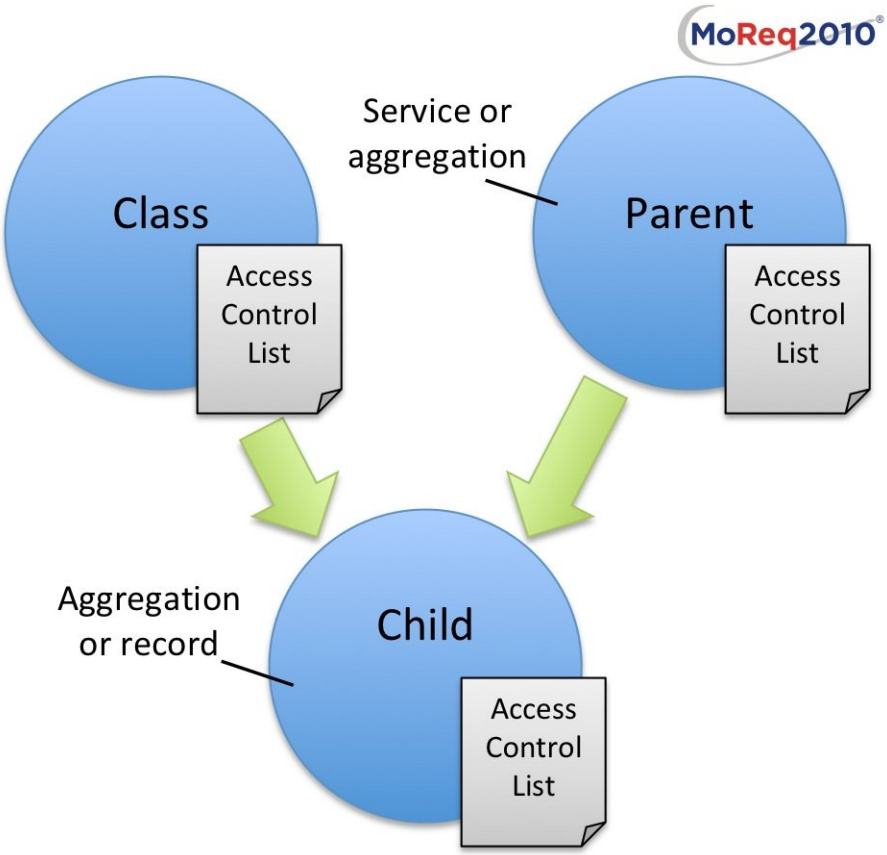


***Figure 4d** – Sometimes access control lists may be inherited from more than one source*

This arrangement allows different organisations to configure access to records both at the classification level and at the aggregation level.

### 4.3.6 Preconfigured Roles

It is common practice for records systems to be installed with a default set of roles that have been preconfigured by the supplier for the convenience of the consumer organisation and to allow immediate use of the system. Where this occurs, some of these preconfigured roles may be able to be modified and even deleted (if they have not been used) or destroyed by an authorised user, while other preconfigured roles may be fixed by the supplier and rendered unalterable within that particular MCRS solution.

MoReq2010® permits suppliers to provide one or more preconfigured roles within their products, including fixed roles, provided they are documented as part of the test verification report, and the MCRS also includes functionality for users to create, modify and destroy their own custom role definitions in accordance with the requirements below and the records management policies of the consumer organisation.

Note that while some preconfigured roles may be fixed by a supplier within its own MCRS solution, such roles will not necessarily continue to be fixed when entities and their associated roles are transferred to another MCRS solution.

## 4.5 Functional Requirements

### R4.5.1

The MCRS must allow an authorised user to create roles (**E14.2.13**) with the following metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- First Used Timestamp (**M14.4.32**),
- Is Administrative Role Flag (**M14.4.44**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),
- Scope Notes (**M14.4.97**),
- Function Definition Identifier (**M14.4.33**), and
- Destroyed Timestamp (**M14.4.17**).

Each role also has:

- Event history (see **2. System Services**),
- Access control list for the role,

And may have:

- Contextual metadata (***or equivalent***, see **7. Model Metadata Service**).

*Depending on the approach taken by the MCRS in implementing **7. Model Metadata Service**, the mechanism by which contextual metadata is added to user entities may vary.*

*Note that fixed roles are provided as part of an MCRS solution and are not created by users, however they have the same metadata as created roles (see **4.3.6 Preconfigured Roles**).*

*Function reference: **F14.5.143***

### R4.5.2

The MCRS must allow an authorised user to modify the Title, Description and Scope Notes of an active role, and any of its contextual metadata.

*Note that this function does not apply to fixed roles provided in an MCRS solution (see **4.3.6 Preconfigured Roles**).*

*Function reference: **F14.5.153***

### R4.5.3

The MCRS must allow an authorised user to make a role an administrative role or a non-administrative role, but only if the role has never been included in any access control entries.

*Once the role has been used it cannot be changed from an administrative role to a non-administrative role, or vice versa, as this may have unintended consequences for entities where the role has been granted, and their descendants.*

*The value is stored in the Is Administrative Role Flag, see **R4.5.1**.*

*Note that this function does not apply to fixed roles provided in an MCRS solution (see **4.3.6 Preconfigured Roles**).*

*Function reference:* ***F14.5.153***

#### R4.5.4

The MCRS must allow function definitions to be added to, and removed from, active roles, ensuring that every function definition is at all times associated with at least one active role.

*Otherwise there will be functions that can never be performed. Function definitions can be added and removed while active roles are in use.*

*Function definitions cannot be added to, or removed from, residual roles. Residual roles retain their references to the function definitions that belonged to them at the moment they were destroyed.*

*Note that this function does not apply to fixed roles provided in an MCRS solution where the function definitions are already assigned (see* ***4.3.6 Preconfigured Roles****).*

*Function references:* ***F14.5.142***, ***F14.5.155***

#### R4.5.5

The MCRS must allow an authorised user to delete a role that has never been included in any access control entries, provided every function definition is at all times associated with at least one active role.

*When a role is first included in an access control entry the MCRS must set the First Used Timestamp.*

*Note that this function does not apply to fixed roles provided in an MCRS solution (see* ***4.3.6 Preconfigured Roles****).*

*Function reference:* ***F14.5.144***

#### R4.5.6

The MCRS must allow an authorised user to destroy a role that has previously been included in an access control entry, provided every function definition is at all times associated with at least one active role.

*Once a role has been used it may no longer be deleted under* ***R4.5.5****, it may only be destroyed. Destroying a role will leave a residual role. Residual roles must never provide users of the MCRS with the authority to perform functions.*

*Note that this function does not apply to fixed roles provided in an MCRS solution (see* ***4.3.6 Preconfigured Roles****).*

*Function reference:* ***F14.5.147***

#### R4.5.7

Subject to **R2.4.22**, and in addition to **R2.4.11**, the MCRS must allow an authorised user to browse and inspect roles and function definitions, in at least the following ways:

- Browse across the roles in the role service and inspect their metadata,
- Browse across the function definitions in the role service and inspect their metadata,
- Browse from a role to the function definitions included in that role and inspect their metadata,

- Browse from a function definition to all the roles that include that function definition inspect their metadata.

*The terms "browse" and "inspect" are defined in **13. Glossary of Terms**.*

*Function references: **F14.5.87**, **F14.5.131***

#### R4.5.8

The MCRS must automatically create an access control list (**D14.3.2**) for each service, or bundle of services under **R2.4.1**, and for each entity in the MCRS where so specified, with the following metadata:

- Include Inherited Roles Flag (**M14.4.43**).

Each access control list also has:

- Access control entries for that entity.

*An access control list is a data structure defined in **D14.3.2** that contains the access control entities that determine which users and groups may access an entity using which roles. Each access control list is an integral part of the entity to which it belongs and each entity has a single access control list.*

*Access control lists are applied to services in MoReq2010® as well as entities, so that they may be inherited by all entities in the service. The access control list for the record service is only inherited by root aggregations. This provides an easy way to give a group, for example, managerial access to the classification scheme within the classification service. If the organisation uses two different classification services then different groups may manage each service independently, as each service will have its own access control list.*

*Access control lists are mandated for most types of entities by other requirements. For example, within the system services module by **R2.4.2** (for services), **R2.4.10** (for entity types) and **R2.4.12** (for function definitions); and similarly within the user and group service by **R3.4.1** (for users) and **R3.4.8** (for groups); etc.*

*Within the MoReq2010® core services only access control lists, access control entries, events and components do not have an access control list. Note that the metadata of an access control list does not include a separate system identifier because the access control list belongs to, and is inseparable from, an entity. The Include Inherited Roles Flag is therefore included in the metadata of every entity with an access control list.*

*The value of the Include Inherited Roles Flag is not relevant to a service because services do not inherit their access control lists.*

#### R4.5.9

The MCRS must allow an authorised user to browse an entity's access control list and inspect the access control entries it contains.

*This is a separate function to inspecting the general metadata of an entity.*

*Function references: **F14.5.13**, **F14.5.31**, **F14.5.64**, **F14.5.78**, **F14.5.84**, **F14.5.88**, **F14.5.102**, **F14.5.110**, **F14.5.132**, **F14.5.150**, **F14.5.159**, **F14.5.172**, **F14.5.188***

#### R4.5.10

The MCRS must allow an authorised user to modify an entity's access control list to change the value of the Include Inherited Roles Flag and to add, delete and modify access control entries (**D14.3.1**) with the following metadata:

- User Or Group Identifier (**M14.4.107**), and
- Role Identifier (**M14.4.96**).

*An access control entry is a data structure defined in **D14.3.1** that is contained by an access control list. Access control access control entries are always part of an access control list.*

*Each entity can have only one access control entry for each identified user or group. Each access control entry must have a User Or Group Identifier, and at least one Role Identifier. By adding an access control entry to an entity's access control list, the authorised user is granting to the specified user or group the authority to perform functions on the entity and its descendants. By removing an access control entry from an entity's access control list, the authorised user is rescinding this authority from the user or group. By modifying an access control entry the authorised user can increase or decrease the number of roles and, as a result, the amount of functionality the user or group can exercise over the entity and its descendants.*

*Under **R2.4.15**, subject to **R2.4.13**, whenever an access control entry is added, modified or deleted an event will be generated in the entity's event history.*

*Setting the Include Inherited Roles Flag will mean that the entity inherits the full access control list of its parent entity or service in conjunction with any access control entries in its own access control list. Clearing the Include Inherited Roles Flag will mean that the entity will only inherit access control entries for administrative roles. It is not possible to block the inheritance of administrative roles.*

*Function references:* ***F14.5.15***, ***F14.5.33***, ***F14.5.66***, ***F14.5.80***, ***F14.5.86***, ***F14.5.90***, ***F14.5.104***, ***F14.5.112***, ***F14.5.134***, ***F14.5.152***, ***F14.5.161***, ***F14.5.174***, ***F14.5.190***

#### R4.5.11

The MCRS must authorise any active user to perform any function on any entity provided the function to be performed is included in an active role that has been granted to that user or to any active groups to which the user belongs, specifically in relation to that entity, including those roles it inherits from its service, parent entity or class (if any).

*The specific rules of inheritance used by the model role service are as follows:*

- *Root aggregations inherit from the record service and their class;*
- *Child aggregations inherit from their parent aggregation and their class;*
- *Components use the access control list of the record they belong to (they do not have their own access control lists);*
- *Disposal holds inherit from the disposal holding service;*
- *Disposal schedules inherit from the disposal scheduling service;*
- *Entity types inherit from their respective service;*
- *Function definitions inherit from the service of their entity type;*
- *Groups inherit from the user and group service;*
- *Metadata element definitions inherit from the metadata service;*
- *Records inherit from their parent aggregation and their class;*

- *Roles inherit from the role service;*
- *Templates inherit from the metadata service; and*
- *Users inherit from the user and group service.*

*The roles an entity inherits from its service, parent entity or class will be influenced by the setting of the Include Inherited Roles Flag, under* ***R4.5.10****. Administrative roles will always be inherited, while non-administrative roles will only be inherited if the Include Inherited Roles Flag is set.*

*When a group entity is granted a role then all of the active users that are members of that group will inherit that role while they remain group members and provided the group is an active group. Users do not inherit the roles granted to residual groups.*

*This requirement explains the meaning of the term "authorised user". It does not matter by which path a user becomes authorised or by how many alternative paths a user may be authorised.*

#### R4.5.12

The MCRS must allow an active user to discover which functions the user is authorised to perform in relation to any entity.

*The method by which the MCRS provides this feedback to the user will be dependent on the type of interface implemented by the MCRS, see* ***R2.4.6****.*

#### R4.5.13

The MCRS must allow an authorised user to generate a report showing which functions a nominated user is authorised to perform in relation to any entity, and how they have been determined.

*In addition to requiring authorisation to perform the function for the nominated user entity, the user generating the report must also have authorisation to inspect the entity and its access control list.*

*The report should include:*

- *All the functions the user is authorised to perform on the entity;*
- *For each function the active roles that include that function that the user may exercise; and*
- *For each active role the different paths by which that role has been granted to the user, including by membership of a active group and roles inherited from the entity's parent.*

*Function reference:* ***F14.5.193***

#### R4.5.14

The MCRS must allow an authorised user to generate a report listing the function definitions that belonged to a nominated role at a specified historical date and time.

*The report must indicate if the date and time provided were before the role entity was created or after it was destroyed.*

*Function reference:* ***F14.5.156***

#### R4.5.15

The MCRS must allow an authorised user to search for and find:

- Entities for which an access control entry contains a nominated role, and
- Entities for which an access control entry contains a nominated user or group.

*Users may only search for and find entities for which they are authorised to browse and inspect the access control list under **R4.5.9**. An authorised user may either search entities in the MCRS where a particular role has been assigned or, alternatively, for entities in the MCRS where a particular user or group is assigned a role. Though explicitly specified here, these search options should be part of general searching and reporting in **10. Searching and Reporting Service**.*

*Function reference: **F14.5.195***

# 5. Classification Service

## 5.1 Service Information

| | |
|---|---|
| **Service Name** | **Classification Service** |
| **Service Version** | **1.0** |
| **Implements Service Identifier *(see M14.4.42)*** | 10fea10e-9c2f-4760-9095-f4f9295f4b19 |

## 5.2 Key Concepts

### 5.2.1 Classifying records

Every record in an MCRS must be classified. In MoReq2010® this means that, from its creation, every record must always be associated with a class entity. Classes represent business functions, activities and transactions, and associating a class with a record provides it with a definitive business context that continues to link the record with the business process that generated it.

In MoReq2010®, records are also placed within aggregations. Unlike classes, aggregations may be created for many different purposes. For example, an aggregation may represent a traditional "file" or folder of records. Or it might represent an "online library" of records made available for viewing at a particular website. Records can be arranged in aggregations for operational convenience, to allow them to be managed as a single entity, or to allow them to share the same set of access controls. Aggregations are described in **6. Record Service**.

Where the records in an aggregation share the same business context then they may inherit their class directly from their parent aggregation. So as to facilitate this, MoReq2010® requires that aggregations, like records, are classified. This approach to classification by inheritance is recommended for managing large numbers of records as it avoids the need to individually classify each record. However, it is only possible where the records contained within a particular aggregation are homogenous.

MoReq2010® also allows for heterogeneous aggregations that contain records that fall under different business classifications. These may be aggregated together for operational reasons, for example because they all relate to a particular person, place, project, event, case, client or incident (see **6.2.1 Purposeful Aggregation**). In the example given previously, the records in the online library may not have all been generated by the same business process. In this circumstance, each of the aggregated records would be classified individually, overriding the default class they inherit from their parent aggregation.

Classifying a record gives it a business context, which in turn gives it a default disposal schedule. Disposal schedules manage the retention and eventual disposal of a record. Initially this always comes from the record's class. More information on disposal schedules is available in **8. Disposal Scheduling Service**.

### 5.2.2 Inheriting classification

All root level aggregations (see **6.2.2 Root Aggregations**) must be classified. By default, each child aggregation and each record will then inherit its parent aggregation's class. The inherited classification can be overridden by directly assigning a class to an aggregation or record.


***Figure 5a** – **Explanatory note** – for illustrative purposes each of the classes appearing in diagrams in this module are depicted using a different shape and colour (all are labelled as "Class"); in diagrams accompanying other modules, such as in **Figure 1i**, all entities of the same type, such as classes, are given a uniform shape and colour*

**Figure 5b** shows how a class, from the classification service, is used to classify an aggregation. The same class is then automatically inherited by all the descendants of the aggregation, including both child aggregations and records.

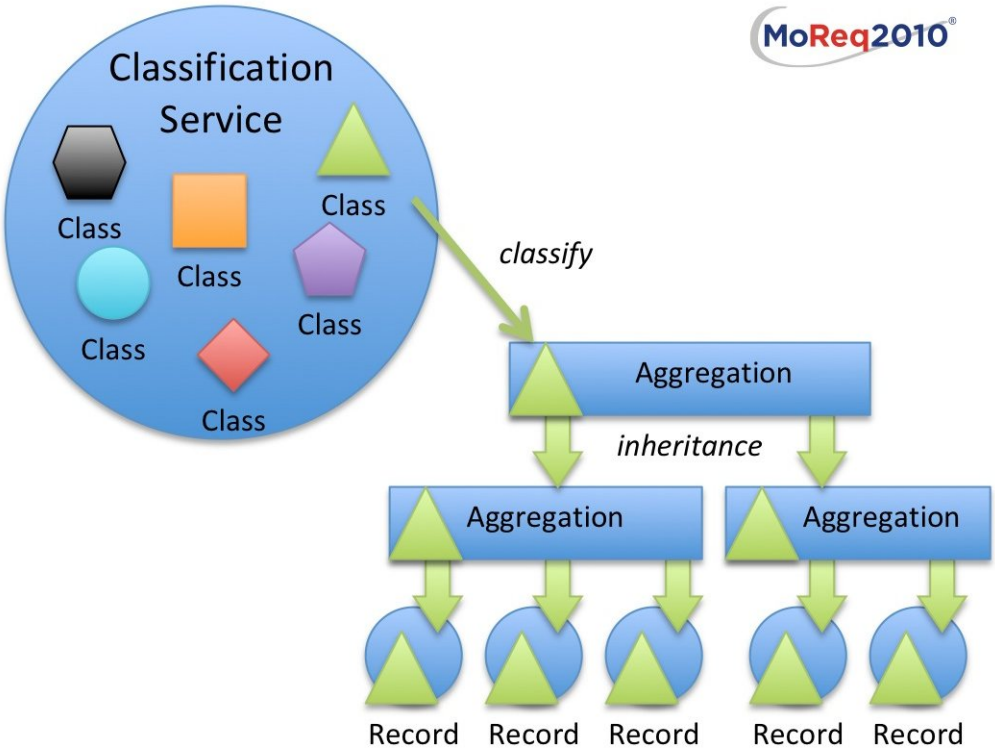


***Figure 5b** – By default, all child aggregations and records will inherit their class from their parent aggregation*

**Figure 5c** shows how class inheritance can be overridden by classifying a child aggregation. In this example the class inherited from the parent aggregation is replaced by a class directly assigned to the child aggregation. This replacement class will then be the class that is inherited by the child aggregation's descendants.

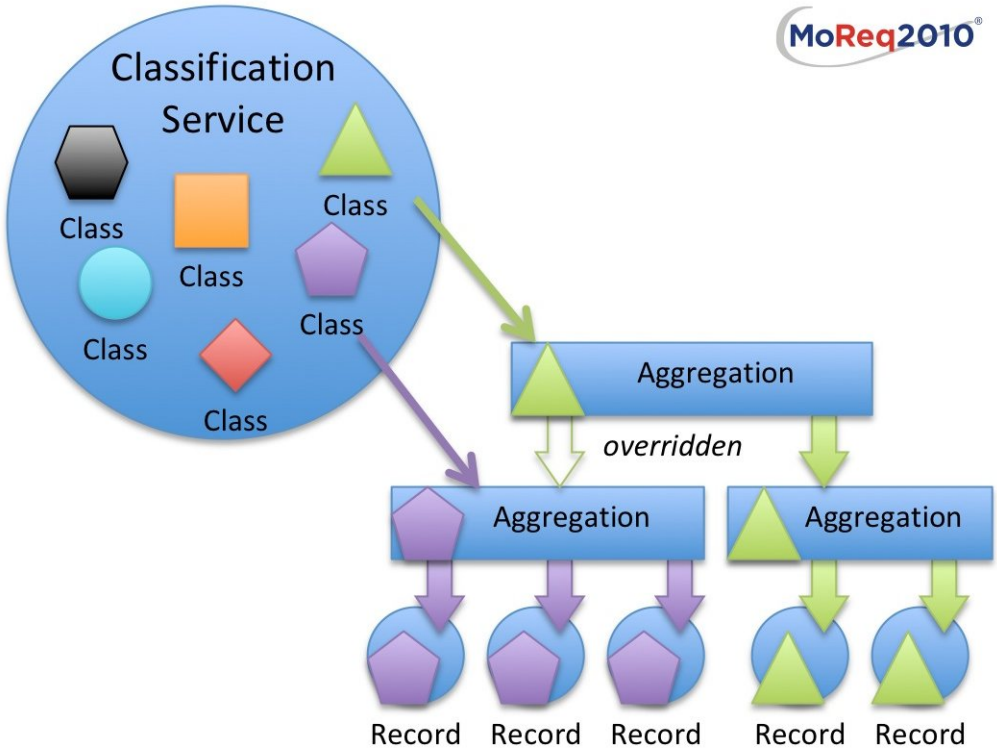


***Figure 5c*** *– Classifying a child aggregation overrides the default class it inherits from its parent aggregation*

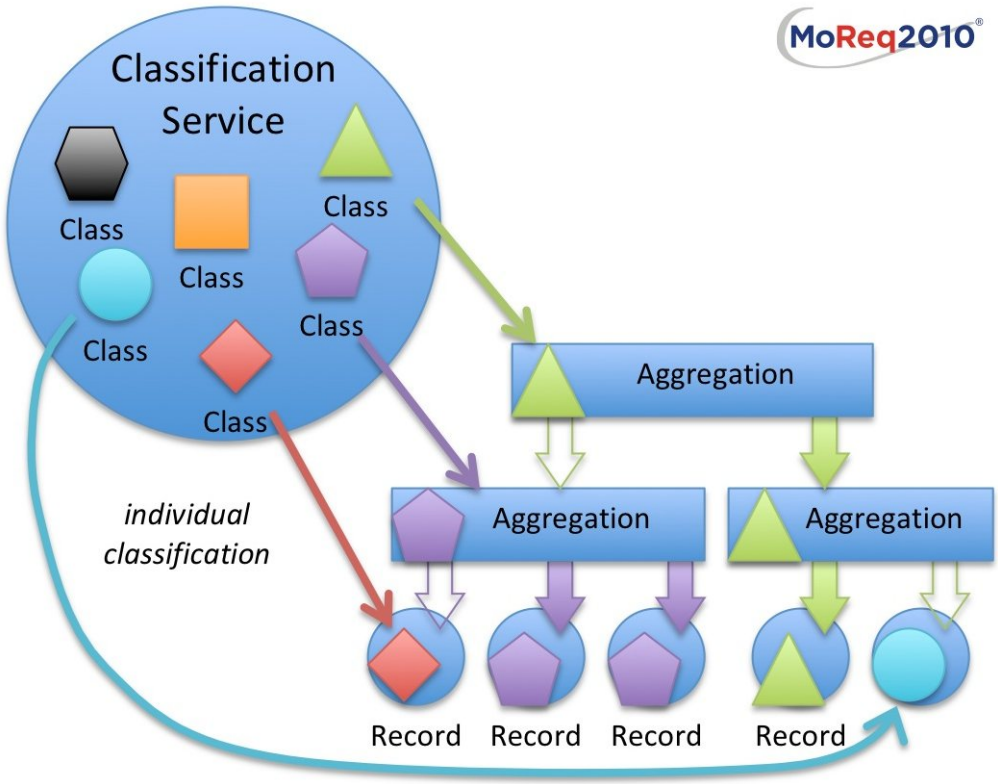


***Figure 5d*** *– Individually classifying a record overrides the default class it inherits from its parent aggregation*

**Figure 5d** shows how inheritance from a record's parent aggregation can be overridden by directly classifying individual records.

### 5.2.3 Reclassification

From time to time it becomes necessary to reclassify records in a records system. This may happen in any records system as business classifications change over time and are updated. However, it is particularly true when bringing records together that come from different sources, such as when two different business units merge.

MoReq2010® requires that each MCRS provide at least minimal support for reclassification by allowing an authorised user to replace one nominated class with another, wherever it is applied within the MCRS.

### 5.2.4 Classification services and classification schemes

Many organisations create their own, or use industry-wide, classification schemes that are applicable across the whole of their business. The service based architecture of MoReq2010® makes it possible for different records systems across the enterprise to share a common centralised classification service. A single MCRS may also have more than one classification service.

Whether centralised, or built into an MCRS, each classification service must support a particular classification scheme. Classification schemes represent different ways of arranging the classes within a classification service.

For example, one of the simplest and most popular classification schemes is hierarchical classification where classes are arranged in a simple tree structure, as shown in **Figure 5e**.

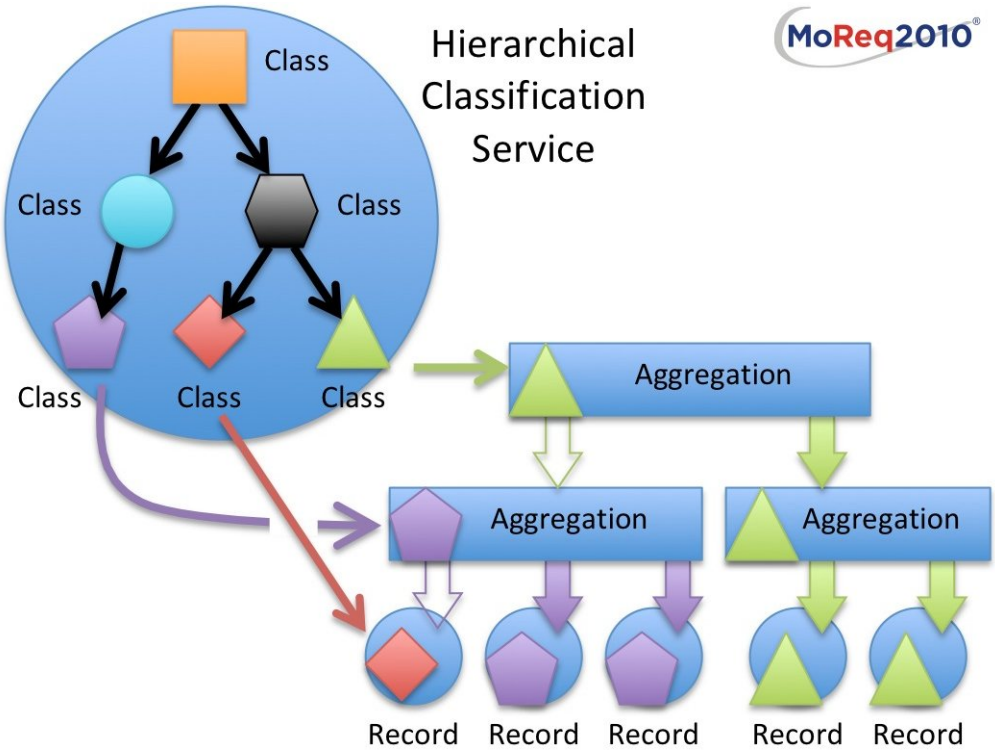


***Figure 5e** – An example of a classification service that adopts a hierarchical classification scheme*

**Figure 5e** shows an example of a classification service that implements hierarchical classification. Under this type of classification scheme only the classes at the bottom of the hierarchical structure are used to classify aggregations and records.

Other types of business classification are also possible, such as Keyword AAA, a widely used functional classification scheme expressed in a polyhierarchical structure derived from an ISO 2788 compliant monolingual thesaurus. Keyword AAA makes use of additional inter-class linkages such as related terms.

So as to facilitate the adoption of different classification schemes by different industries and organisations in different market sectors, MoReq2010® allows suppliers choice over the particular classification scheme, or schemes, that their MCRS solutions implement. Requirements specific to each type of classification scheme are defined in the different classification modules. Each MCRS must therefore implement the requirements below, as well as at least one of the MoReq2010®, **200. Classification Series**, plug-in modules.

While, for internal consistency, each classification service must implement the structure of only one classification scheme, MoReq2010® does not prevent an MCRS from having more than one classification service each with its own distinct classification scheme.

## 5.4 Functional Requirements

### R5.4.1

The MCRS must incorporate the functionality of a classification service that manages classes within a classification scheme in accordance with one of the modules in the MoReq2010®, **200. Classification Series**.

*The MCRS may implement and be tested against more than one of the 200 series classification modules, but each separate classification module must be implemented in its own separate classification service.*

### R5.4.2

The MCRS must allow an authorised user to create new classes (**E14.2.2**) with at least the following system metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- First Used Timestamp (**M14.4.32**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),
- Scope Notes (**M14.4.97**),
- Default Disposal Schedule Identifier (**M14.4.11**), and
- Destroyed Timestamp (**M14.4.17**).

Each class also has:

- Disposal holds associated with the class (see **9. Disposal Holding Service**),
- Event history (see **2. System Services**),
- Access control list (***or equivalent***, see **4. Model Role Service**),

And may have:

- Contextual metadata (***or equivalent***, see **7. Model Metadata Service**).

*Note that additional class metadata may be specified by the series 200 classification module the MCRS implements.*

*The mandatory disposal schedule that must be associated with a new class on creation must be an active disposal schedule.*

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, a MoReq2010® access control list may not be present during system operation and may only be added to a class at export.*

*Depending on the approach taken by the MCRS in implementing **7. Model Metadata Service**, the mechanism by which contextual metadata are added to classes may vary.*

*Function reference: **F14.5.24***

#### R5.4.3

The MCRS must allow an authorised user to modify the Title, Description and Scope Notes of an active class, and any of its contextual metadata.

*Function reference: **F14.5.35***

#### R5.4.4

The MCRS must allow an authorised user to change the default disposal schedule for an active class, provided the new disposal schedule is active, and subject to **R8.4.4**. Whenever this occurs, the MCRS must replace the previous default disposal schedule with the new default disposal schedule for all active records classified by that class.

*The disposal schedule must only be changed for active records that inherit their disposal schedule from their class. If the default disposal schedule has been overridden, for example under **R6.5.15**, then this function has no effect on the record. Note that the disposal schedule is never changed on residual records as these have already been destroyed under the previous schedule.*

*Function reference: **F14.5.34***

#### R5.4.5

The MCRS must allow an authorised user to delete any class that has never been used for classification.

*The Last Used Timestamp should be updated automatically by the MCRS once the class has been used to classify an aggregation or a record. A class cannot be deleted once it has been used as it has then become part of the history of aggregations and records in the MCRS. However, the class can be destroyed and made residual, under **R5.4.6**, to prevent it being used again.*

*Function reference: **F14.5.25***

#### R5.4.6

The MCRS must allow an authorised user to destroy any active class, provided it is not associated with any active aggregation or record.

*All active classes must have an active disposal schedule associated with them and cannot be destroyed if they are being used to classify active aggregations or records.*

*Function reference:* ***F14.5.28***

#### R5.4.7

Subject to **R2.4.22**, the MCRS must allow an authorised user to browse classes and their associated entities in the following ways:

- Browse across the classes in the classification service and inspect their metadata,
- Browse from a class to its default disposal schedule and inspect its metadata, and
- Browse from a class to any associated disposal holds and inspect their metadata.

*The terms "browse" and "inspect" are defined in* ***13. Glossary of Terms****.*

*Function references:* ***F14.5.30***, ***F14.5.63***, ***F14.5.77***

#### R5.4.8

The MCRS must allow an authorised user to replace a nominated class with another active class, for all aggregations and records classified by the class.

*All aggregations and records classified by the original class will now be classified by the replacement class. The default disposal schedule associated with the original class will be replaced by the default disposal schedule associated with the replacement class, similar to* ***R5.4.4****, for all active records that are reclassified under this requirement.*

*This function may result in a large amount of processing, depending on the number of records to be reclassified. For this reason, it should be used sparingly and only to support necessary changes to the business classification scheme.*

*Function references:* ***F14.5.20***, ***F14.5.137***

# 6. Record Service

## 6.1 Service Information

| | |
|---|---|
| **Service Name** | **Record Service** |
| **Service Version** | **1.0** |
| **Implements Service Identifier *(see M14.4.42)*** | ced3d0df-3f9f-4807-9e96-b5b790adad4a |

## 6.2 Key Concepts

### 6.2.1 Purposeful aggregation

A record service manages records within an MCRS under different levels of aggregation. Each aggregation represents a grouping of records or a grouping of aggregations. Records are placed into aggregations for all or some of the following reasons:

- Because they relate to the same business transaction or process,
- Because they have the same business classification,
- Because they share the same subject area or topic;
- Because they relate to the same person, place, project, case, client, event or incident,
- Because they share common metadata,
- Because they have the same source or format,
- Because they are managed by the same business unit,
- Because they are intended for the same audience,
- Because they have the same level of security access control, or
- Because they are held under the same retention and disposal conditions.

Aggregation often enhances the semantic understanding of records by purposefully placing them into a meaningful context alongside other like records. The aggregation as a whole may thereby collectively provide a vivid descriptive narrative of its subject.

Aggregations may also be gathered into higher levels of aggregation for similar reasons to those above. An aggregation containing records may belong to a parent aggregation which represents a grouping of like aggregations.

The highest level of aggregation available to any MCRS is the record service itself, which taken as a whole, represents a single aggregation of all of the records it contains. ISO 23081 envisages a tiered model of aggregation that goes beyond a single records system or archive, such that taken collectively all of the records systems within an organisation might abstractly be considered as a further level of aggregation even beyond that.

There is also a practical side to aggregation. Aggregations may be determined by operational considerations or by the technical limitations of the records being managed. This is particularly so if the records are being managed in place in another business system. Where this occurs the nature of an aggregation may be determined by the nature of the business system. If the business system offers, for example, workspaces for collaborative teamwork, then these workspaces will by necessity form the basis of the aggregations to be

managed by the records system. Similarly if the records system is designed to be carried on a mobile device for personal use, it may have a flatter structure than a cloud based records system intended for sharing across multiple government departments. System specialisation, storage location, capacity and volume, secure access restrictions, and other considerations may all play a part in shaping how aggregations are implemented, sized and managed.

### 6.2.2 Root aggregations

Within a record service, aggregations that are not the children of other aggregations are called root aggregations. There is no limit to the number of root aggregations in a record service. This allows an MCRS to be multi-tenanted, if necessary, where different organisations, or different parts of an organisation manage different aggregations and records without reference to other aggregations and records in the same record service.

**Figure 6a** shows a typical arrangement of aggregations and records within a record service where multiple aggregations can be at the root level.

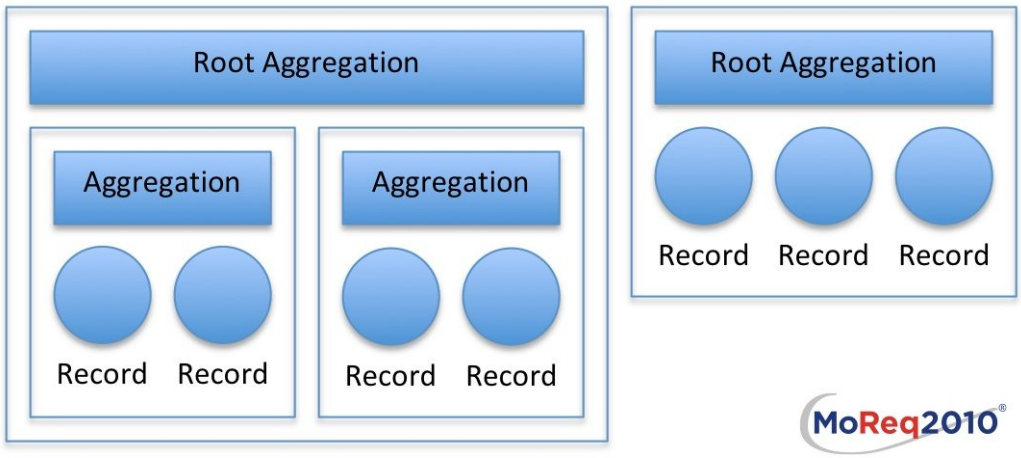


***Figure 6a** – Showing different levels of aggregation within a record service where there is no single root aggregation*

### 6.2.3 Restrictions on aggregation

In an MCRS, records and aggregations cannot be placed together at the same level of aggregation as one another. This is shown in **Figure 6b**. This arrangement preserves the integrity and separate identity of each level of aggregation, it allows consistent management policies to be uniformly applied to each level of aggregation, and ensures that there is no ambiguity over where records should be created.

A necessary consequence of this restriction is that records cannot be created outside an aggregation. This is also shown in **Figure 6b**. Creating records directly within the record service would place them at the same level of aggregation as any existing root aggregations or block the creation of any future root aggregations in the MCRS.

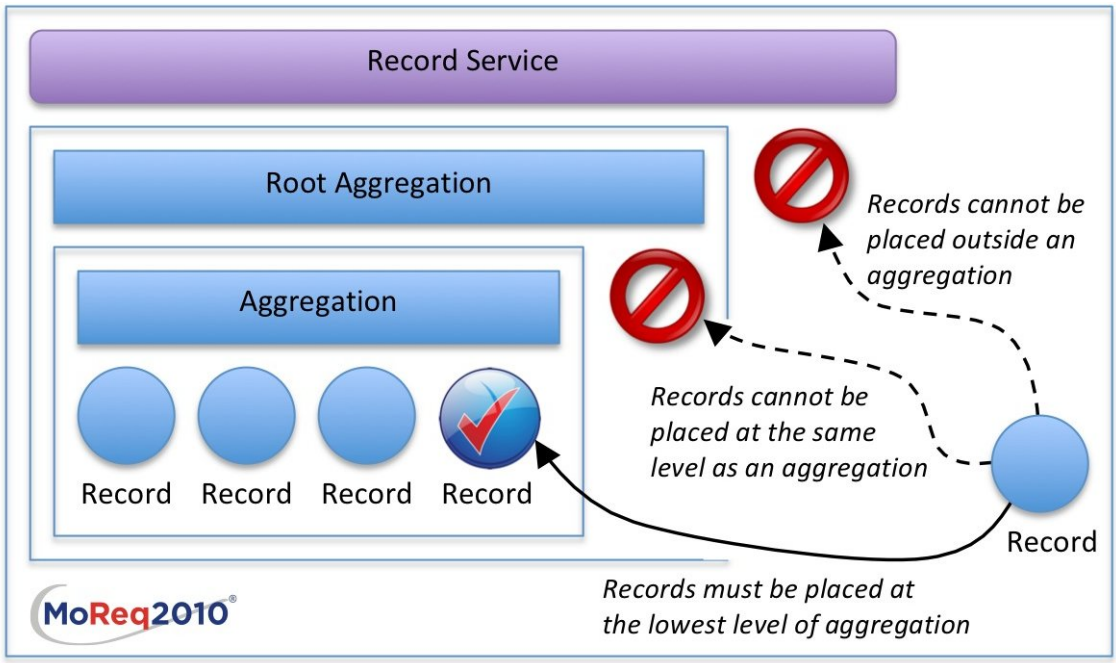


***Figure 6b** – A record cannot be stored at the same level as an aggregation*

### 6.2.4 Aggregation and inheritance

One of the primary advantages of aggregation is to support inheritance. Some of the characteristics that child entities can inherit from their parent aggregations include:

1. **Classification** – an aggregation or record can inherit its class from its parent aggregation;
2. **Access controls** – an aggregation or record can inherit its parent aggregation's access control list (or equivalent, see **4. Model Role Service**); and
3. **Metadata** – a user can search for aggregations and records based on metadata assigned to their parent aggregations.

### 6.2.5 Aggregation and classification

Classification can be applied at different levels of aggregation. By default each child aggregation or record inherits its class from its parent aggregation. However, this may be overridden by applying a separate classification to each level of aggregation, and even directly to individual records, as described in **5.2.2. Inheriting classification**.

Aggregation and classification may interact together in different ways. So as to provide insight into how this may occur, various scenarios are described under **6.3 Aggregation and Classification Examples**. It should be noted that these scenarios do not necessarily reflect the recommended best practice in records management, however they are intended to represent commonly encountered real world situations.

### 6.2.6 Constraining sprawl

The aggregation feature of MoReq2010® is powerful, allowing multiple levels of aggregation within a record service. While this is extremely useful in particular circumstances, it is

important that this feature not be overused, in particular it is desirable to avoid deep structures consisting of many levels of aggregation.

As a general rule flatter structures of only one or sometimes two levels of aggregation are preferable, as these are easier to manage and the aggregation feature must not be employed to create pseudo-classification hierarchies.

For this reason, MoReq2010® allows users to set, for any root aggregation, the maximum number of levels of aggregation that may be added below it. By setting this value users can set limits over the depth to which any aggregation can grow.

Planned extensions to MoReq2010® will also introduce specialised types of aggregation which feature a fixed number of levels of aggregation.

### 6.2.7 Original order

One of the important benefits of aggregating records is that the process of adding records over time places them into a linear sequence that supports browsing. This natural sequencing provides an historical timeline and adds an important narrative dimension to the aggregation, shown in **Figure 6c**.

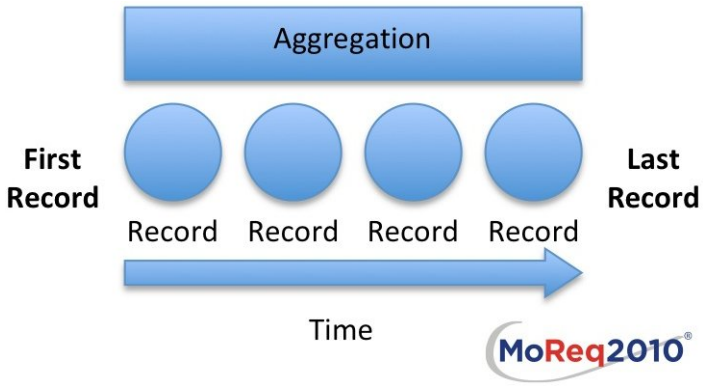


***Figure 6c** – By ordering on originated date and time the records in an aggregation can be browsed in a logical historical sequence*

By default the originated date and time will reflect the timestamp applied when the record is created in the MCRS. However, where records are created out of order or the record is not immediately added to the MCRS, the specification supports reordering, allowing the originated date and time to be modified by a user to indicate an earlier moment of origin.

Time based ordering within aggregations is particularly useful when capturing records generated by dynamic processes such as workflow. As a workflow progresses, each major stage or transition can produce a snapshot of the current workflow status that can be sequentially captured into the aggregation. The aggregation can then reveal how, historically, the end-to-end process was conducted.

Preservation of the original order within an aggregation is another reason why MoReq2010® does not allow the same level of aggregation to contain both records and child aggregations (see **6.2.3 Restrictions on aggregation**). Placing records at different levels of the aggregation breaks the natural linear sequence of records within the aggregation and thus interrupts its narrative. This conundrum is shown in **Figure 6d**.

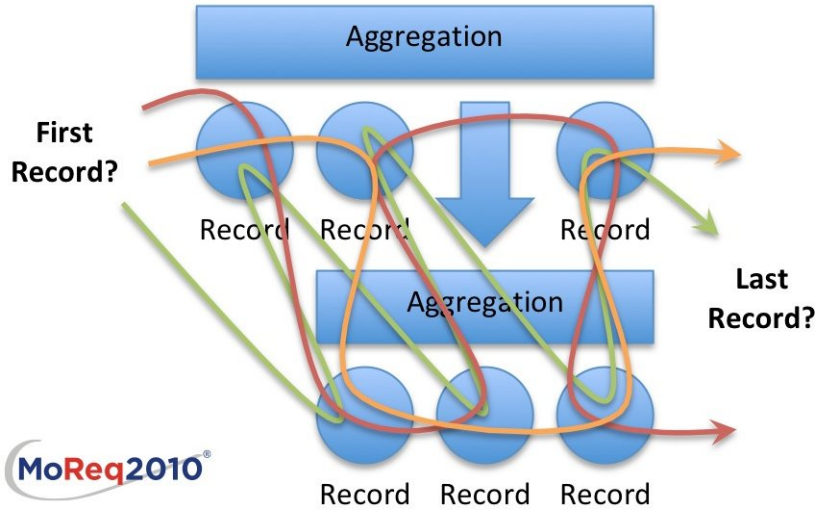


***Figure 6d*** *– An aggregation cannot retain its linear narrative if it contains both records and aggregations at different levels*

### 6.2.8 Record atomicity and duplication

Each entity in MoReq2010® is considered to be self-contained or atomic. This characteristic supports interoperability and allows the transfer of entities between different records systems. This is particularly true of records. Each record is made up of its metadata, its event history, its components and its access control list (or equivalent). In particular, a record's event history confirms the key events that have occurred since the record was first created in the MCRS. This provides the provenance of the record, an important ingredient in establishing its authenticity.

If, within the MCRS, one record is copied from another without also duplicating the event history of the first record, then the second record cannot represent a complete record. Even though they are identical copies of one another, there will be a gap in the event history of the second record up to point at which the copy was created. This is shown in **Figure 6e**.

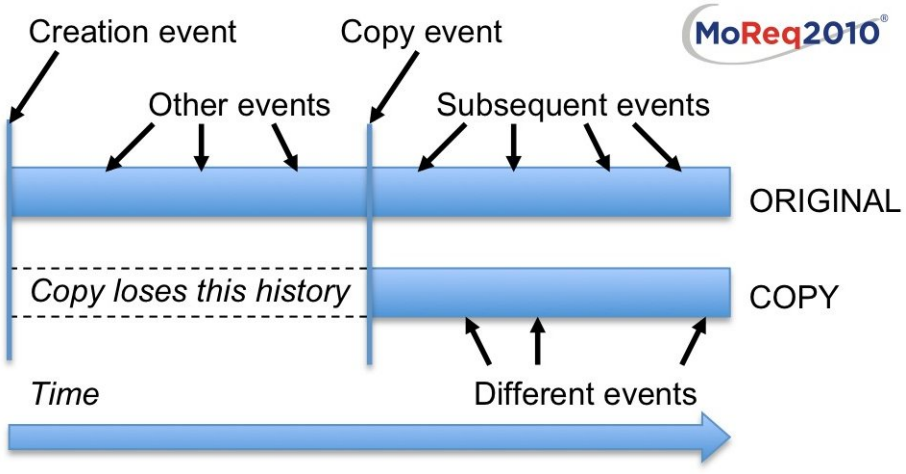


***Figure 6e*** *– When a copy is made of a record it loses part of its event history and does not have parity with the original*

For this reason, MoReq2010® makes no provision for the copying of records and requires instead a process of duplication. Duplication does not simply create a new record with the same metadata as the source but also appends to it a copy of the source record's event history. Following the duplication process neither record can or should be regarded as the original and the other as the copy. Both records are original, as shown in **Figure 6f**.

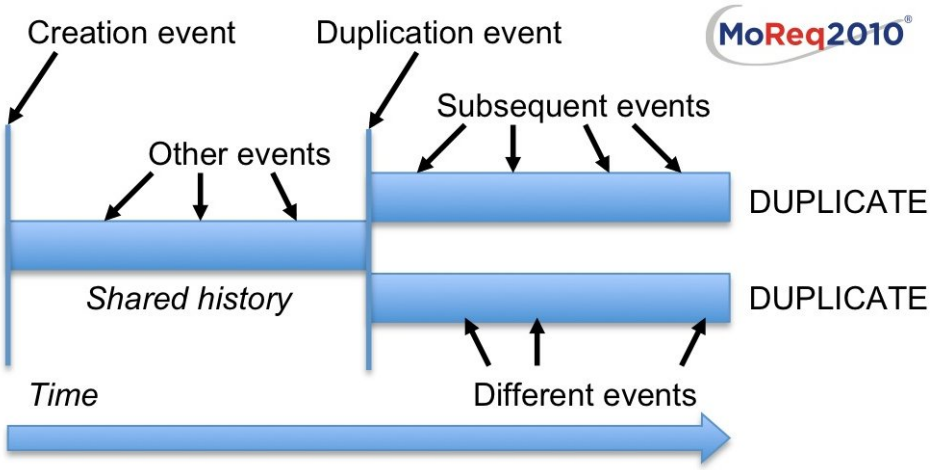


***Figure 6f** – When a record is duplicated then the result is the equivalent of having two original records with the same history up to the moment of duplication*

The primary purpose for duplicating a record in an MCRS is to allow the same record to appear in more than one aggregation. Where this occurs, each separate appearance of the record must be represented by a duplicate, each a record in its own right, with its own classification, disposal schedule, parent aggregation, component entities, access controls, event history, title, description and other metadata.

When a record is duplicated, the MCRS may choose to duplicate the content of the record or it may maintain a system of pointers that utilise the same content for each duplicate. If the MCRS does make two separate instances of the content of the record then the content is said to be "physically discrete". If the MCRS uses pointers then the content is said to be "logically discrete" (see the principle of discreteness, explained in **6.2.10 Principles of managing component content**).

In either circumstance, irrespective of whether the content is physically or logically discrete, when one record is destroyed, then the remaining duplicates, and their content, must be entirely unaffected. If the MCRS only logically separates content through the use of pointers then the pointers linking the destroyed record to the content should be deleted but the content itself should not be deleted until the last duplicate record is destroyed.

### 6.2.9 Record components

What MoReq2010® refers to as a "record" is really an "entity"; meaning it is a set of metadata stored by the MCRS that describes the record content. The term "record" is therefore an abstraction. The actual content or data of the record is separate to the record entity in the MCRS and may also be stored in a different database or location.

The content of each record may take many forms and MoReq2010® makes provision for plug-in modules that describe different types of record content. These forms will usually fall

into one of two categories. They will either be electronic, referring to a digital resource such as a datafile, or physical, referring to a real world object.

The following list contains different examples of the types of content that might comprise a record that is managed by an MCRS. This is by no means an exhaustive list.

**Electronic examples**

- A digital datafile, such as an electronic document produced by a word processing application;
- Several digital datafiles, collectively, such as the various types of datafiles required to render a web page in a web browser (for example, HTML, CSS, JavaScript, JPEG/GIF/PNG, etc.);
- A row in a database table or more likely a set of interrelated rows across different tables in a database; or
- A reference to an entity in a business system (for example, a link to a customer entity in a CRM system).

**Physical examples**

- A paper document;
- A physical object (for example, a CD, DVD, microfilm, etc.); or
- A biomedical sample.

In MoReq2010® a record can have more than one discrete resource making up its content. These different resources may even be stored in different locations. The association between a record and its content is provided by component entities. Each record can have one or more components. Each component is a reference to a single item of content. This is shown in **Figure 6g**.

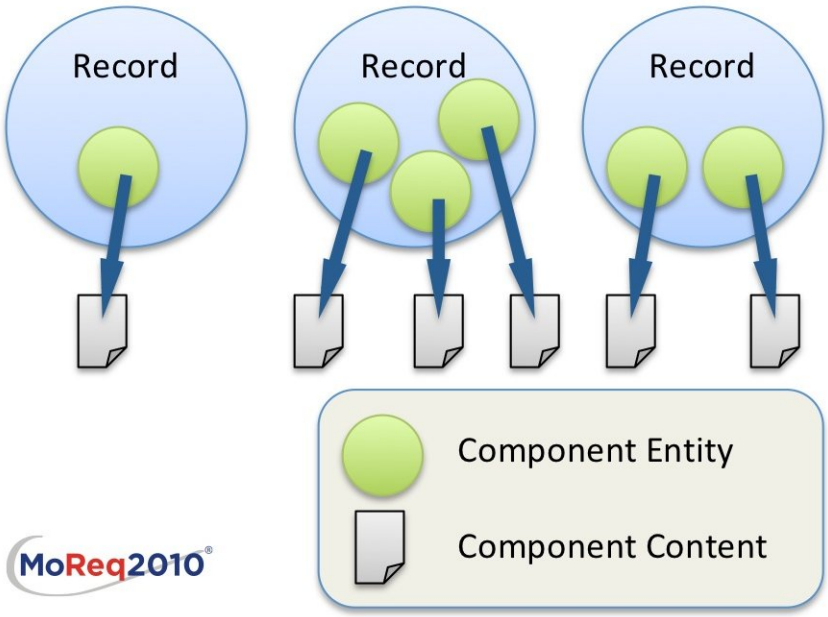


***Figure 6g** – Each record has one or more components each of which refers to a single item of content of a particular type*

Note that in **Figure 6g**, each record is shown as inclusive of its components. Component entities are an integral part of the record to which they belong. They do not, for example, have separate access controls, but have the same level of access as the record and its metadata.

The types of record components which an MCRS can support are determined by which of the MoReq2010® **300. Component Series** component plug-in modules they implement. Each of these modules refers to a different type of component content, such as an "electronic component".

An MCRS may provide support for more than one type of component. Where this occurs, records with different component types may be freely mixed together, for example, within the same aggregation.

### 6.2.10 Principles of managing component content

Not every resource is necessarily suitable for consideration as the content of a component. MoReq2010® defines specific characteristics that distinguish component content from other general information. These characteristics apply regardless of whether the content is electronic or physical, or what particular form it takes.

In order to be considered as compliant with the specification, the content referred to by components managed by the records system must have these characteristics:

- Discreteness,
- Completeness,
- Immutability, and
- Destructibility.

The **principle of discreteness** states that the content of each component must be separate and distinguishable from the content of other components. Records cannot share components and components cannot share their content.

Where a component is created that refers to a shared resource, then the records system must ensure that the resource is managed as discrete content belonging only to that component. MoReq2010® does not specify how this is done but it may mean physically separating the content, for example by taking and keeping a separate copy of the resource, or it may mean logically separating the content, for example by the use of pointers. The principle of discreteness is illustrated in **Figure 6h**.

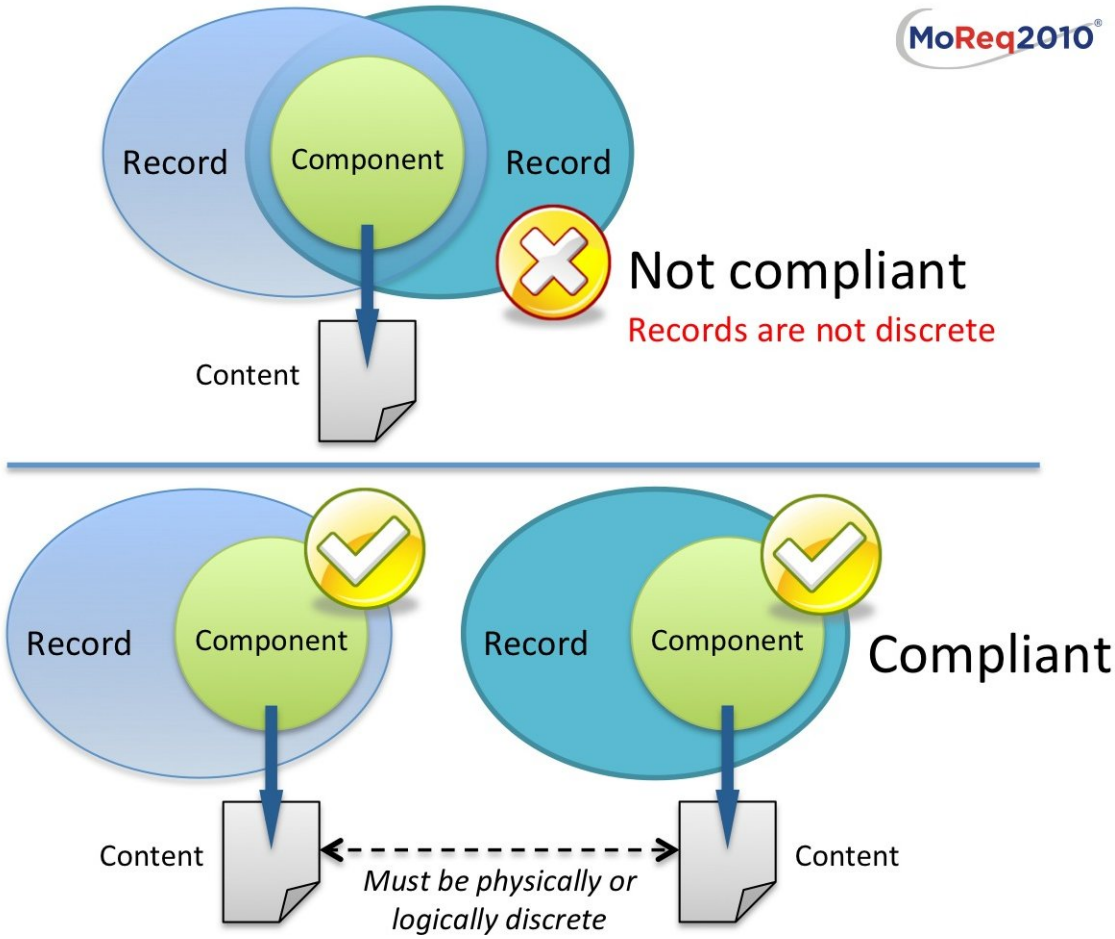


***Figure 6h** – The principle of discreteness means that each component must belong to only one record and its content must be separate and distinct*

It is important for MCRS solutions to be able to manage the process of duplication of a record, described in **6.2.8 Record atomicity and duplication**, while observing the principle of discreteness. When a record is duplicated the record entity and its metadata and access controls are all duplicated, the events in its event history are duplicated, its components are duplicated and, most importantly, the content of its components must be duplicated (as explained above, either physically or logically).

The **principle of completeness** states that taken together, the content referred to by the components of a record make up the entire record and there is no dependency on any other resource. The components and their content must not be reliant on components or content stored outside the record and not contained within the record and its components. Components within the same record may have interdependencies between each other but must not have external dependencies. The principle of completeness is shown in **Figure 6i**.

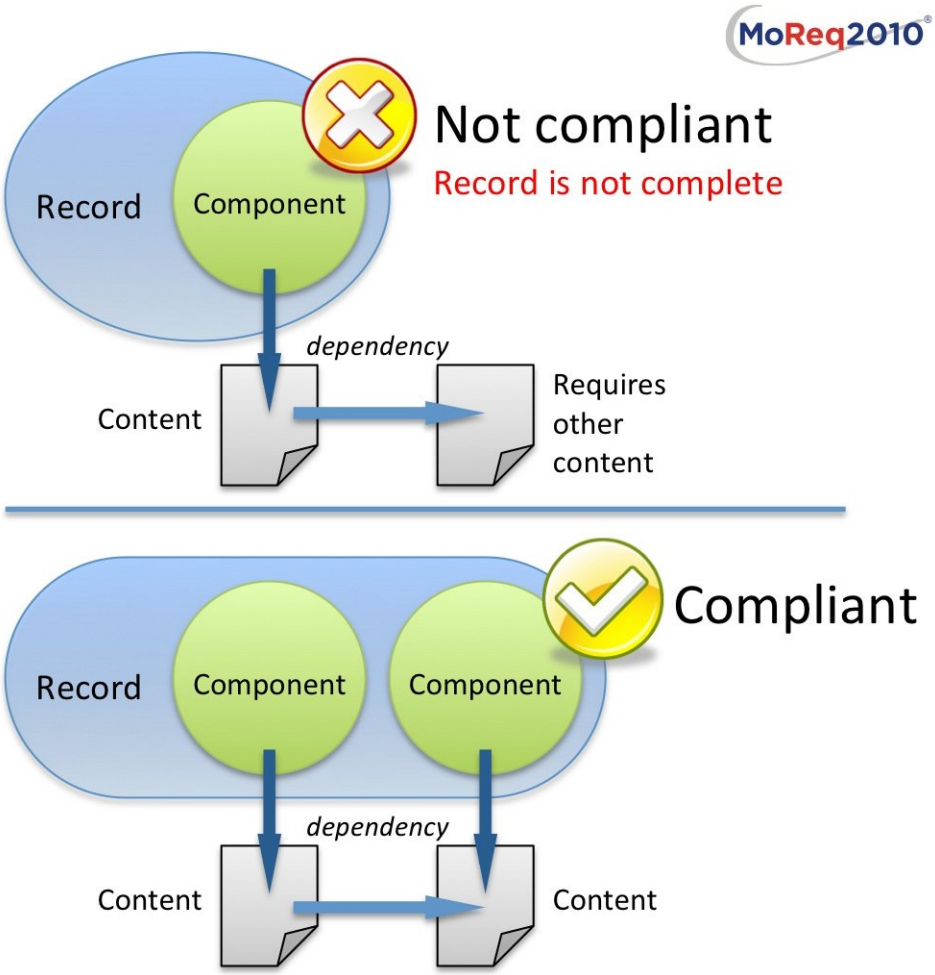


***Figure 6i** – The principle of completeness means that each record must be fully self-contained and manage all dependent content within its own components*

The **principle of immutability** states that the content of a record cannot be altered once the record has been created. The metadata of a record or its components may be modified but not their content. The content must be secure from alteration, replacement and deletion.

Achieving compliance with the principle of immutability is the reason why many records systems manage content within their own content store. However, it may be neither possible nor practical to capture all of an organisation's records into a single content store. Therefore under MoReq2010®, it is possible for a compliant records system to manage content in external content stores provided appropriate measures are taken to ensure immutability. The principle of immutability is shown in **Figure 6j**.

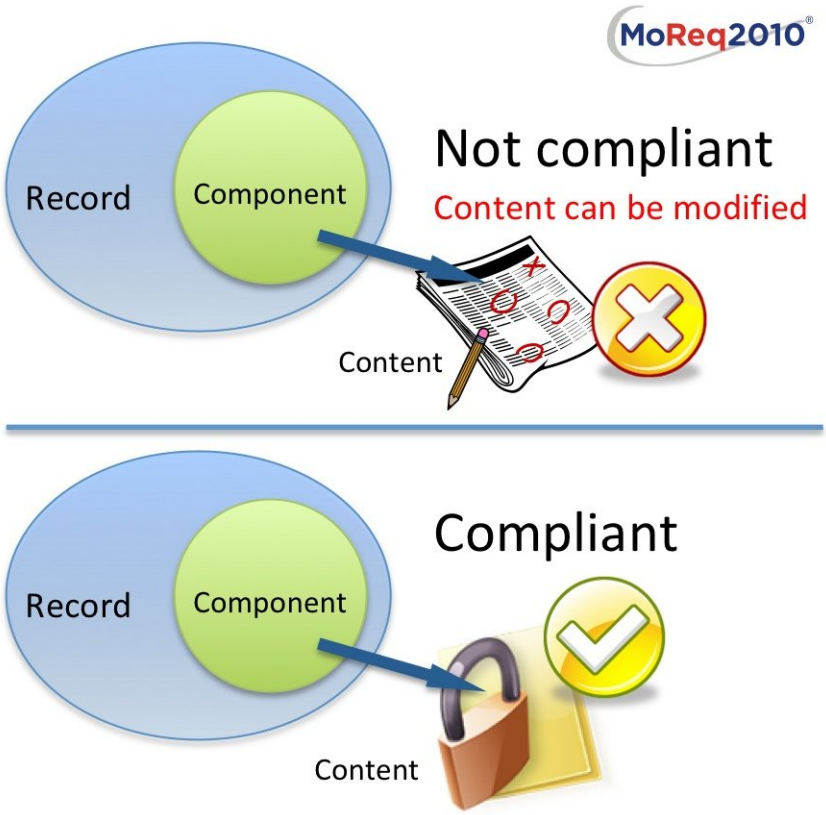


***Figure 6j** – Under the principle of immutability the content of a component must not be able to be altered after record creation*

The **principle of destructibility** states that a record cannot be destroyed in the records system, unless the content of its components is previously or simultaneously deleted from the content store(s) in which it is managed. In secure environments the principle of destructibility extends to such activities as, for example, shredding or incinerating paper records and, in electronic environments, overwriting magnetic or other storage media to ensure it cannot be read. The principle of destructibility is illustrated at **Figure 6k**.

Implementing the principle of destructibility and ensuring the proper planned destruction of records, is very important to good records management practice and ensures that records really are destroyed in accordance with their disposal schedule and at the appropriate time.

Depending on the type of content belonging to a component, the MCRS may not be able to automatically initiate its destruction. Particularly for physical components, but also on occasion for electronic components, some manual intervention is required to perform the destruction.

For this reason, MoReq2010® allows for components where either:

- The content of the component can be deleted automatically by the records system;
- The content of the components requires user confirmation that it has been deleted before destruction of the record and its component entities can continue.

This information must be kept as part of the metadata of each component individually.

**8. Disposal Scheduling Service** describes how the disposal scheduling service must respond to each of these types of component content when records are scheduled for destruction.

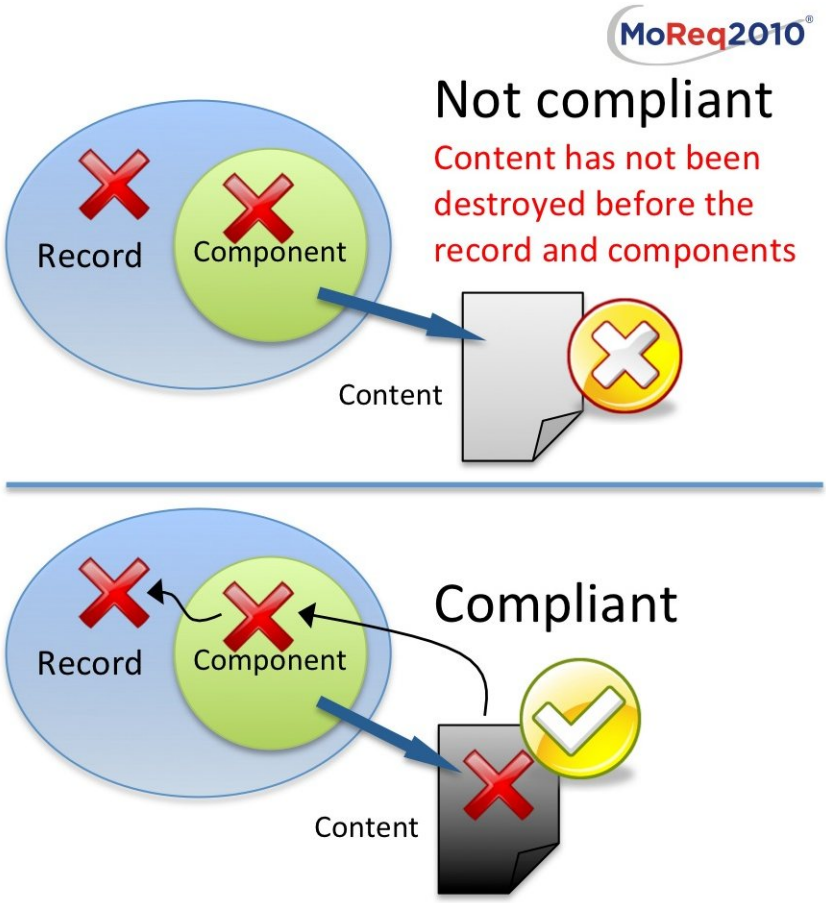


***Figure 6k** – Before records can be destroyed, under the principle of destructibility, the corresponding component content must erased from all content stores*

### 6.2.11 Design approaches to managing component content

The way content is managed and the different types of content that can be managed by an MCRS will vary depending on the purpose of the records system and its intended usage. When developing a records system, suppliers must give consideration to the following factors:

- The number of components each record will have – one, several or variable;
- The type of content belonging to each component;
- Where the content will be stored – in an internal or external content store, or across many content stores;
- How the records system will ensure that the content it manages is made immutable – especially if records are managed in place or in an external content store;
- How the records system will allow users to access the content of components;
- Whether, the records system will automatically delete content when records are destroyed or whether confirmation of deletion will be required, and for which component content types.

MoReq2010® compatible records systems may be specialised and dedicated systems and are not required, necessarily, to be able to manage all forms of component content. However, irrespective of its other design goals, each MCRS must be able to manage at least one type of component content, be it electronic or physical, in either an internal or external content store as an integral part of the record service it provides.

## 6.3 Aggregation and Classification Examples

### 6.3.1 Casework

Casework and other related activities, such as project work, are commonplace and particularly prevalent in service industries, such as health, insurance, banking and legal work. (A previous example related to casework was explained in the discussion of **1.4.5 Classification and aggregation**.)

The aggregations used for casework focus on keeping together all records that are germane to the main subject of the case. For example, hospital records will invariably be organised by patient, with records from each subsequent visit by the same patient being captured into the same patient aggregation, even though they may occur at different times, be for different ailments, check-ups and treatments by different medical staff from different departments within the hospital.

Under these circumstances it is common to find records within the same casework aggregation that were created by separate business functions, activities and transactions and should therefore have separate classifications. Sometimes casework aggregations will be subdivided into several secondary aggregations that allow different default classifications to be applied at the level of these secondary aggregations. For example, within a patient aggregation, the patient's medical insurance claims might be separated from the patient's treatment records by placing them into different aggregations within a secondary level of aggregation. On other occasions, it may not be desirable to subdivide casework aggregations, and records within the same aggregation must be individually classified to reflect their different contexts.

### 6.3.2 Parallel business activities

Many large organisations have different divisions or business units that work in parallel, performing the same business functions as each other. For example, a large corporation may divide its sales activities between three regional sales teams: the Americas, Asia and EMEA (Europe, the Middle East and Africa). Each of these sales teams performs the same functions and activities but within their own designated area of operations.

Where this occurs, organisations will often create root aggregations for each of their teams. In the example above there would be three root aggregations. Each team would then create its records within its own separate root aggregation. The arrangement of aggregations within each root aggregation would be mirrored. Classification of records would also be applied identically within each root aggregation, using the same classification service.

Note how in this example the classification scheme is not altered to accommodate the organisational structure. The distinction between EMEA, Asia and the Americas is not introduced as an element of the classification structure or used to differentiate between

different classes, instead it is provided independently of the classification service by means of separate aggregation.

### 6.3.3 Security considerations

The example given in the previous section **6.3.2 Parallel business activities**, was of three business units that managed their records independently under different root aggregations. It may be that the organisation also requires that users from one business unit be restricted from accessing the records of another business unit. For example, users might be only able to create records under their own team's aggregation and be granted only browse and inspect access to aggregations and records in the other teams' aggregations. Or it may be a requirement that users cannot access the aggregations of other business units at all. This is more like a multi-tenancy arrangement where the different business units share the same records system but their users do not interact together.

Because aggregations are managed separately from the classification structure, MoReq2010® allows access controls to be more easily applied to sensitive aggregations. Each business unit may set the level of access by users to its records while continuing to share classification in common with other business units. Most organisations have secure or sensitive aggregations where a different level of access permission is needed and MoReq2010® provides the capability to secure these aggregations within one or several secure areas formed by higher level aggregations, with access controls set at that level of aggregation without compromise to the classification scheme.

### 6.3.4 Managing records in place

An MCRS may manage records created and stored on other business systems. This is also known as managing records in place. Where this occurs the MCRS will usually construct aggregations that mirror the naturally occurring aggregations of the other business system.

The business system will usually create its own form of aggregation that is suited to its specialised purpose. For example, the business system may be organised as a Wiki, it may take the form of a database, or a content management system, it may be an email archiving system, or it may support the setting up of areas for collaborative teamwork on documents. There are many different possibilities.

Rarely will a business system be organised so as to directly support a business classification scheme, however, depending on the specific nature of the business system and the level of integration with the MCRS it may be possible to classify records automatically based on their type and other attributes held by the business system.

In this scenario, it is likely that each record will need to be classified individually while being placed an aggregation that mirrors its equivalent location within the business system. The other records in the same aggregation may well have different classifications depending on how this is determined from the attributes held in the business system.

## 6.5 Functional Requirements

#### R6.5.1

The MCRS must allow an authorised user to create active aggregations (**E14.2.1**) with the following system metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- First Used Timestamp (**M14.4.32**),
- Last Addition Timestamp (**M14.4.48**),
- Class Identifier (**M14.4.4**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),
- Scope Notes (**M14.4.97**),
- Closed Timestamp (**M14.4.5**), and
- Destroyed Timestamp (**M14.4.17**).

Each root aggregation also has the following system metadata:

- Max Levels Of Aggregation (**M14.4.52**).

Each child aggregation also has the following system metadata:

- Parent Aggregation Identifier (**M14.4.63**), and
- Aggregated Timestamp (**M14.4.1**).

All aggregations also have:

- Child entities (either aggregations or records),
- Disposal holds associated with the aggregation (see **9. Disposal Holding Service**),
- Event history (see **2. System Services**),
- Access control list (***or equivalent***, see **4. Model Role Service**),

And may have:

- Contextual metadata (from a template ***or equivalent***, see **7. Model Metadata Service**).

*Aggregations must always be created as active aggregations, however, they may be created in either the open or closed state.*

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, a MoReq2010® access control list may not be present during system operation and may only be added to an aggregation at export.*

*Depending on the approach taken by the MCRS in implementing **7. Model Metadata Service**, the mechanism by which contextual metadata is added to aggregations may vary and may not utilise templates.*

*Function reference: **F14.5.5***

#### R6.5.2

The MCRS must allow aggregations to be created under R6.5.1 that are either:

- Root aggregations, without a parent aggregation, provided they are also assigned an active class on creation; or
- Child aggregations, under a parent aggregation, provided they are placed into an open aggregation that does not already contain records, either active or residual, are either assigned an active class or inherit their parent aggregation's class, and do not

exceed the Max Levels Of Aggregation set for the root aggregation under which they are placed.

*An aggregation without a parent is a root aggregation. A record service may manage many root aggregations. Each root aggregation must be directly classified with an active class. Each root aggregation also has a maximum number of levels of aggregation which limits the depth of the tree of child aggregations that can be placed under the root aggregation.*

*Child aggregations may only be placed into parent aggregations that do not contain records of any kind. Child aggregations may be directly classified with an active class or inherit their parent's classification. Child aggregations cannot be created in contravention of the depth limitation imposed by Max Levels Of Aggregation for the root aggregation that is their ancestor.*

*Function reference:* ***F14.5.5***

#### R6.5.3

The MCRS must allow an authorised user to modify the Title, Description and Scope Notes of an active aggregation, and any of its contextual metadata.

*Function reference:* ***F14.5.17***

#### R6.5.4

The MCRS must allow an authorised user to reclassify an aggregation by:

- Removing the class directly assigned to any child aggregation and instead inheriting the class assigned to the aggregation's parent, or
- Assigning an active class directly to an aggregation replacing its previous classification and overriding any inherited class.

Whenever an aggregation is reclassified, the MCRS must ensure its children and descendants are also reclassified if they inherit their class from their parent, and the disposal schedule for records must also be updated in accordance with **R6.5.15**.

*Reclassifying an aggregation will have a recursively cascading impact on any children of that aggregation that inherit their parent's class, and their descendants, and so on.*

*Function references:* ***F14.5.11***, ***F14.5.20***

#### R6.5.5

The MCRS must allow an authorised user to add, modify and remove a value for Max Levels Of Aggregation allowed under a root aggregation, provided the value is not set lower that the number of levels of aggregation that already exist under the root aggregation.

*The maximum number of levels of aggregation allowed under a root aggregation is not a mandatory metadata element. Where the value is not set it indicates that the root aggregation can have an unlimited number of levels of aggregation under it.*

*Where Max Levels Of Aggregation is set it must contain a positive number or zero. If the root aggregation already has one or more levels of aggregation below it and a user modifies Max Levels Of Aggregation then the MCRS must ensure that its value is set to a value equal to or greater than the number of existing levels of aggregation.*

*Function reference:* ***F14.5.16***

#### R6.5.6

The MCRS must allow authorised users to close and open active aggregations, ensuring that an aggregation cannot be closed unless all of its child aggregations are closed, and that aggregations that are closed are immediately destroyed if the following conditions apply:

- The aggregation has been used for aggregating records or aggregations, and
- All of the child entities of the aggregation (either child aggregations or records) have already been destroyed.

*Closing an aggregation prevents users from placing any more entities into the aggregation. It does not prevent users from removing entities from a closed aggregation.*

*Closing an aggregation that contains aggregations must not be possible unless all child aggregations are also closed beforehand, or simultaneously as part of the same operation.*

*Aggregations that are closed can be reopened to accept additional entities, provided they are still active.*

*An aggregation with only residual child entities is automatically destroyed when it is closed, see* ***R8.4.22****.*

*Note that closing an aggregation will update the Closed Timestamp and opening an aggregation will clear it.*

*Function references:* ***F14.5.4****,* ***F14.5.9****,* ***F14.5.19***

#### R6.5.7

The MCRS must allow an authorised user to delete any aggregation that has never been used for aggregating entities.

*This is the purpose of the First Used Timestamp, listed under* ***R6.5.1****.*

*Function reference:* ***F14.5.6***

#### R6.5.8

For any aggregation, including a root aggregation, the MCRS must allow an authorised user to move it, either:

- To a new parent aggregation so that it retains its original classification, provided the parent aggregation is active, open, does not already contain records, either active or residual, and adding the aggregation will not exceed the maximum number of levels of aggregation set for the root aggregation under which it is being placed;
- To a new parent aggregation so that it adopts the classification of its new parent, provided the parent aggregation is active, open, does not already contain records, either active or residual, and adding the aggregation will not exceed the maximum number of levels of aggregation set for the root aggregation under which it is being placed; or
- So that it becomes a root aggregation while retaining its original classification.

*To retain the aggregation's classification, the MCRS must change the metadata of the aggregation during the move operation, to indicate that it does not inherit its parent's class.*

*To adopt the classification of its new parent, the MCRS must change the metadata of the aggregation during the move operation, to indicate that it does inherit its parent aggregation's class. This will have a recursively cascading impact on the descendants of the class, similar to that described for* ***R6.5.4****.*

*Note that when an aggregation is moved to a new parent its Aggregated Timestamp and the new parent aggregation's Last Addition Timestamp will be updated. When a root aggregation is moved to a parent then the metadata element Max Levels Of Aggregation will be deleted, if it exists for the aggregation.*

*The user requires two forms of authorisation to move a child aggregation. The user must be authorised to remove the child aggregation from its previous parent as well as being authorised to add the child aggregation to its new parent. Although there are two functions,* ***F14.5.21 Aggregation – Remove Aggregation*** *for removing the child aggregation from its previous parent and* ***F14.5.1 Aggregation – Add Aggregation*** *for adding the child aggregation to its new aggregation, these functions must be performed together and a single event is generated, under* ***R2.5.15****, subject to* ***R2.5.13****.*

*When moving a child aggregation to become a root aggregation or moving a root aggregation to become a child aggregation, the user must be authorised to add or remove the aggregation at the level of the record service.*

*Function references:* ***F14.5.1****,* ***F14.5.21***

#### R6.5.9

Subject to **R2.4.22**, the MCRS must allow an authorised user to browse and inspect aggregations in at least the following ways:

- Browse across all root aggregations in the record service and inspect their metadata,
- Browse from a parent aggregation to its children and inspect their metadata,
- Browse from a child aggregation to its parent aggregation and inspect its metadata,
- Browse from an aggregation to its class in the classification service and inspect its metadata, and
- Browse from an aggregation to any associated disposal holds and inspect their metadata.

*The terms "browse" and "inspect" are defined in* ***13. Glossary of Terms****.*

*Function references:* ***F14.5.12****,* ***F14.5.30****,* ***F14.5.63****,* ***F14.5.131***

#### R6.5.10

The MCRS must allow an authorised user to create active records (**E14.2.12**) in an active and open aggregation, that does not contain any aggregations, either active or residual, with the following system metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),
- Duplicate Identifier (**M14.4.23**),

- Parent Aggregation Identifier (**M14.4.63**),
- Aggregated Timestamp (**M14.4.1**),
- Class Identifier (**M14.4.4**),
- Disposal Schedule Identifier (**M14.4.22**),
- Retention Start Date (**M14.4.93**),
- Disposal Action Code (**M14.4.18**),
- Disposal Action Due Date (**M14.4.19**),
- Disposal Confirmation Due Date (**M14.4.20**),
- Disposal Overdue Alert Timestamp (**M14.4.21**),
- Last Review Comment (**M14.4.49**),
- Last Reviewed Timestamp (**M14.4.50**),
- Transferred Timestamp (**M14.4.106**), and
- Destroyed Timestamp (**M14.4.17**).

Each record also has:

- One or more components (see **7. Component Storage**),
- Disposal holds associated with the record (see **9. Disposal Holding Service**),
- An event history (see **2. System Services**),
- An access control list (***or equivalent***, see **4. Model Role Service**),

And may have:

- Contextual metadata (from a template ***or equivalent***, see **7. Model Metadata Service**).

*Records may only be placed into aggregations that do not contain child aggregations of any kind.*

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, a MoReq2010® access control list may not be present during system operation and may only be added to a record at export.*

*Depending on the approach taken by the MCRS in implementing **7. Model Metadata Service**, the mechanism by which contextual metadata is added to records may vary and may not utilise templates.*

*Function reference: **F14.5.121***

#### R6.5.11

The MCRS must allow an authorised user to modify the Title and Description of an active record, and any of its contextual metadata.

*Function reference: **F14.5.135***

#### R6.5.12

The MCRS must ensure that each record created under **R6.5.10** inherits its parent aggregation's class and allow an authorised user to reclassify a record at creation or at any other time by:

- Assigning an active class directly to a record replacing its previous classification and overriding inheritance from its parent aggregation, or
- Removing the class directly assigned to a record so that the record inherits its parent aggregation's class instead.

*All records must be classified. Records that inherit their parent aggregation's classification may also be reclassified indirectly under **R6.5.4** and **R6.5.8**.*

*Function references: **F14.5.129**, **F14.5.137***

### R6.5.13

The MCRS must allow an authorised user to move a record from its parent aggregation to any active and open aggregation that does not contain any aggregations, either active or residual, and either:

- Retain the record's previous classification by applying its class directly to the record; or
- Replace the record's previous classification with the classification of its new parent aggregation by removing any class directly applied to the record.

*Records may be moved between aggregations so that they retain their previous classification or so that their previous classification is replaced by the classification of the new parent aggregation. Both types of move must be supported by the MCRS.*

*Records must not be moved into an aggregation that contains any child aggregations, even residual child aggregations.*

*Note that the record's Aggregated Timestamp and the new parent aggregation's Last Addition Timestamp will be updated.*

*Function references: **F14.5.3**, **F14.5.22***

### R6.5.14

Whenever an active record is first created and classified, or whenever it is reclassified under **R6.5.4**, **R6.5.8**, **R6.5.12** or **R6.5.13**, the MCRS must ensure that the record always inherits the disposal schedule associated with its class, unless that disposal schedule has been overridden, under **R6.5.15**.

*By default, an active record will always inherit its disposal schedule from its associated class.*

*Function references: **F14.5.1**, **F14.5.3**, **F14.5.11**, **F14.5.20**, **F14.5.121**, **F14.5.129**, **F14.5.137***

### R6.5.15

The MCRS must allow an authorised user to change the disposal schedule for an active record, either by:

- Applying an active disposal schedule directly to the record and overriding the default disposal schedule it inherits from its class, or
- Removing a disposal schedule that has been directly applied to the record and inheriting instead the default disposal schedule of the class.

*If a disposal schedule is not directly applied to a record, then it must always use the default disposal schedule it inherits from its class.*

*Note that the disposal schedule of a residual record cannot be changed, as it represents the disposal schedule under which the record was destroyed.*

*Function references: **F14.5.130**, **F14.5.138***

#### R6.5.16

The MCRS must allow an authorised user to make a duplicate of a record including its:

- System metadata,
- Contextual metadata (or equivalent, see **R6.5.10**),
- Access control list (or equivalent, see **R6.5.10**),
- Event history,
- Components, and
- Component content.

*Whenever events are duplicated the MCRS must update the Duplicate Identifier for each event record (see **R2.4.16**) to allow them to be seen as a single shared event.*

*Whenever components are duplicated the MCRS must update the Duplicate Identifier for each component (see **R6.5.19**). Depending on the nature of a record's components, the MCRS may duplicate the whole of the content of the components or only appear to duplicate the content of the components by using pointers.*

*Function references: **F14.5.42**, **F14.5.126***

#### R6.5.17

Subject to **R2.4.22**, the MCRS must allow an authorised user to browse and inspect records in at least the following ways:

- Browse the records in an aggregation in order of their Originated Date/Time,
- Browse from a record to its parent aggregation and inspect its metadata,
- Browse from a record to its class, whether inherited or directly applied, and inspect its metadata,
- Browse from a record to its disposal schedule, whether inherited or directly applied, and inspect its metadata,
- Browse from a record to any associated disposal holds and inspect their metadata,
- Browse from a record to its component(s) and inspect the component metadata, and
- Browse from any component to the record it belongs to and inspect its metadata.

*The terms "browse" and "inspect" are defined in **13. Glossary of Terms**.*

*Function references: **F14.5.12**, **F14.5.30**, **F14.5.44**, **F14.5.63**, **F14.5.77**, **F14.5.131***

#### R6.5.18

The MCRS must allow an authorised user to search for and find:

- Aggregations and/or records classified by a nominated class, and
- Records with a nominated disposal schedule.

*The MCRS must find an aggregation or record with the nominated class regardless of whether the class is inherited from a parent aggregation or directly applied to the entity. Similarly, the MCRS must find a record with the nominated disposal schedule regardless of whether the disposal schedule is inherited from the record's class or directly applied to the record. Though explicitly specified here, these search options should be a part of general searching and reporting in **10. Searching and Reporting Service**.*

*Function reference: **F14.5.195***

#### R6.5.19

The MCRS must ensure that all records are created, under **R6.5.10**, with one or more components (**E14.2.3**) that must implement the functionality of one of the MoReq2010® **300. Component Series** modules, and are created with at least the following system metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- Record Identifier (**M14.4.86**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),
- Duplicate Identifier (**M14.4.23**),
- Automatic Deletion Flag (**M14.4.3**), and
- Destroyed Timestamp (**M14.4.17**).

Each component also has:

- Content (held in a content store),
- Event History,

And may have:

- Contextual metadata (***or equivalent***, see **7. Model Metadata Service**).

*Components must always be created as active components within the active record specified by the record identifier. They are destroyed as part of destroying the record that they belong to, see **8. Disposal Scheduling Service**.*

*The Originated Date/Time of a component may be used to indicate when a particular component was created in an external system. For example, the operating system date/time for a datafile.*

*The Duplicate Identifier is used to indicate other components whose content is a duplicate of the content of the component. This value is automatically populated when records and their components are duplicated, as described in **R6.5.16**.*

*The Automatic Deletion Flag indicates whether or not components are automatically destroyed by the MCRS or whether their destruction must be requested and later confirmed, see **8.2.8 Destruction lifecycle**.*

*Note that under MoReq2010®, components do not have their own access control lists but are accessible under the same access control list as the record. It is important that individual components are not conceptualised as independent entities that can be separated from the whole of the record to which they belong.*

*It is possible that components will have their own individual contextual metadata. Where this occurs it is often metadata that is extracted from the content of the component by the MCRS on capture. For example, if the managed content is a digital photograph then the MCRS may extract Exif (Exchange image file format) metadata from the photograph and store it with the component entity as contextual metadata to enable better searching and discovery of the record. MoReq2010® does not include requirements for this in the core specification.*

*Also, depending on the approach taken by the MCRS in implementing **7. Model Metadata Service**, the mechanism by which contextual metadata is added to components may vary.*

*Function reference:* ***F14.5.38***

**R6.5.20**

The MCRS must allow an authorised user to modify the title and description of an active component, and any of its contextual metadata.

*Function reference:* ***F14.5.46***

**R6.5.21**

Whenever the MCRS generates an event for the component, under **R2.4.15**, it must include a record identifier in the metadata of the event, so that the event also appears in the event history of the component's record as well as in the event history of the component.

*Each component has its own event history, however all events for the component will also appear in the event history of the record that the component belongs to. In this way, a user can browse the event history of the record and see any events that have occurred for any of its components.*

*Function references:* ***F14.5.37***, ***F14.5.38***, ***F14.5.39***, ***F14.5.40***, ***F14.5.41***, ***F14.5.42***, ***F14.5.43***, ***F14.5.44***, ***F14.5.45***, ***F14.5.46***, ***F14.5.47***

# 7. Model Metadata Service

## 7.1 Service Information

| | |
|---|---|
| **Service Name** | **Model Metadata Service** |
| **Service Version** | **1.0** |
| **Implements Service Identifier *(see M14.4.42)*** | ***For an MCRS that implements the MoReq2010® model metadata service use:***<br>a600f8d0-2d58-418e-bb41-211d1fd42350<br>***For an MCRS that implements its own native contextual metadata model use:***<br>66bf4419-d92f-4358-8506-7ee9c06abdcd |

## 7.2 Complying with the Model Metadata Service

### 7.2.1 Interoperability and metadata

A significant objective of MoReq2010® is to facilitate interoperability between compliant records systems. Interoperability means that entities can be exported from one MCRS and imported directly into another. The transmission of data between the two records systems must preserve the integrity and the context of the entities that are transferred.

This goal of interoperability can only be achieved if each entity and each metadata element belonging to an entity is recognisable and able to be interpreted universally. It must mean the same thing everywhere. The World Wide Web is a good example of interoperability between different applications and systems. There are many different web browser software products, from different suppliers, in use today. Each one has its own features and benefits. However, web pages, which hold the data that web browsers interpret and visually present, are formatted according to a well-known industry standard called HTML. All compliant web browsers can read, interpret and render standardised HTML.

In the same way, in the future, there may be many different MCRS solutions, with different features from different suppliers, but they will all be able to exchange records, and other entities related to records management, using the standardised MoReq2010® metadata model and schema.

To achieve this, MoReq2010® is more prescriptive that previous specifications, especially over the metadata elements that each compliant records system must keep as well as how particular processes, such as disposal, should be implemented. This standardisation allows MoReq2010® to specify a universally understood metadata schema for use by all MCRS solutions.

### 7.2.2 A model metadata service

Throughout MoReq2010® the metadata required by each entity and each service is specified as part of the functional requirements and expanded on in detail in **14. Information Model**.

This required metadata is called "system metadata". MoReq2010® takes the same technique for defining system metadata and allows it to be used to create additional metadata elements specific to a particular implementation. These are called "contextual metadata". So as to allow a pre-defined set of contextual metadata elements to be simultaneously added to an entity, MoReq2010® allows the definition of metadata templates which contain a list of metadata element definitions. Metadata templates can be applied to entities on creation as a result of their being created under a particular service or classification.

The MoReq2010® approach to managing metadata is therefore simple, easily understood and able to be expressed with a handful of different entity types. It builds up from the same approach used to define system metadata, and it should fit well with most existing records systems, many of which take far more sophisticated approaches to metadata management than is required by MoReq2010®.

At the same time, MoReq2010® recognises that some existing records systems may manage metadata using techniques that are internally incompatible with the model metadata service as specified. They may not use metadata templates to enable contextual metadata to be applied to entities, for example, but some other method. It is important in the interests of increasing interoperability, that these records systems are still able to transfer entities to other records systems that do implement the model metadata service, and that they are therefore able to be certified as MoReq2010® compliant records systems.

In the future, new records systems may fully incorporate the MoReq2010® model metadata service while the suppliers of existing records systems may gradually adopt and progressively support aspects of the model metadata service in newer versions of their software.

### 7.2.3 Approaches to testing and certification against the model metadata service

The DLM Forum® allows two possible approaches to the testing and certification of a records system for compliance with the MoReq2010® model metadata service.

**EITHER**

**A.** The records system implements the MoReq2010® model metadata service in full, and is tested and certified against the requirements in this module.

**OR**

**B.** The records system implements its own native metadata model, in which case the application must satisfy the following criteria:

- It must demonstrate that its native metadata model is equivalent in flexibility, functionality and the consistency of information it captures to the MoReq2010® model metadata service; and
- It must support interoperability by being able, on export, to convert its native metadata into the same XML format used by the model metadata service, so that metadata elements are populated with the same data, with the same semantic interpretation, and using the same standardised identifiers and codes, when the data is subsequently transferred to another MCRS.

### 7.2.4 How to meet the alternative (type B) requirements

To show that the MCRS's native metadata service is equivalent in flexibility, functionality and the information it captures to the MoReq2010® model metadata service, the records system must be able to demonstrate all of the following:

- That it has the equivalent of each of the entity types defined by MoReq2010®, these must be represented within the records system with the same purpose as given by the specification;
- For each entity type, that it has the equivalent of the system metadata elements defined by MoReq2010®, these must have the same datatypes, meaning and range of values as defined by the specification;
- It must be possible to create instance specific or contextual metadata elements, including both contextual metadata elements that store data values and contextual metadata elements that store references to entities;
- Different entity types, especially aggregations and records, must be able to include contextual metadata in their definitions, in addition to the system metadata defined by MoReq2010®,
- The supplier must be able to demonstrate how contextual metadata elements are defined and applied;
- The supplier must be able to demonstrate how values for contextual metadata elements are entered, modified and deleted;
- The metadata service must be able to support a theoretically unlimited number of additional contextual metadata element definitions;
- Contextual metadata elements must be able to be applied collectively (rather than individually) to entities in ordered sets (MoReq2010® describes these as templates) when those entities are first created and possibly later;
- The records system must allow these sets of contextual metadata elements to be selectively applied to entities for different business reasons but at least one reason for applying a set of contextual metadata elements to an aggregation or a record must be as a consequence of its classification (in other words, different records or aggregations created with different classifications must be allowed to have different sets of metadata associated with them depending on their class);
- Each metadata element should only ever be applied to an entity once;
- When a metadata element is applied to an entity, and requires that a mandatory value be supplied, the entity must not be created in the MCRS until a valid value has been provided by one or a combination of: a default value for the element, a value automatically calculated by the records system and/or a user supplied value;
- Textual metadata elements must always be accompanied by a language identifier indicating the language of the metadata element; and
- The record service must have provision for metadata elements to be selectively deleted from an entity when it is destroyed, this to be specified universally for each metadata element, and able to be reconfigured by an authorised user.

When converting its native metadata for export in accordance with the MoReq2010® model metadata service format, the following must be observed:

- The records system must use UUIDs for the System Identifiers of all entities and services;
- The records system must always use the MoReq2010® standard identifiers and codes where they are provided;
- The records system must not add additional entity types and services that are not defined by the MoReq2010® specification;
- MoReq2010® system metadata elements may not be used to store data values that are not in accordance with the stated purpose of the metadata element;
- The supplier must not expand upon the system metadata of MoReq2010® but may add additional information to entities and services using MoReq2010® contextual metadata elements where required;
- These contextual metadata elements may be added at export, so as to provide additional information about the internal state of the records system (for example, if the records system has its own internal identifier for each entity);
- All contextual metadata elements added to entities and services must have associated MoReq2010® compliant metadata element definitions that are included in the same export so that entities with contextual metadata elements can be imported and interpreted by the receiving system;
- The records system must not alter either the meaning or the value of the contents of a metadata element during export;
- The supplier must describe the metadata and template mapping used by its product and provide a complete mapping table showing which internal elements correspond to which MoReq2010® elements and showing how the exported metadata has been made to imitate the model metadata service as accurately as possible; and
- The supplier must make its metadata mapping schema available for inclusion in the full test report of its product.

Further advice and guidance to suppliers can be requested from the MoReq Governance Board through the DLM Forum® secretariat.

The remainder of this module describes the concepts and requirements of the MoReq2010® model metadata service.

## 7.3 Key Concepts

### 7.3.1 System metadata and contextual metadata

The MoReq2010® core requirements specify a simple set of metadata elements, arguably a minimal set, required to successfully manage records within an MCRS, and to transfer them using the MoReq2010® export data format, to other records systems. This essential working set of metadata specified by MoReq2010®, is referred to throughout the specification as "system metadata" and must be supported by every MoReq2010® compliant records system. System metadata represents only those metadata elements required to perform the functional requirements of MoReq2010®.

In addition to implementing the required records management functionality described by MoReq2010® using system metadata, records systems can also add metadata elements to records and related entities that enrich the historical and operational context of these entities.

Metadata that can provide this additional benefit when managing records, includes but is not limited to:

- Additional descriptive information not conveyed by the Title or Description of an entity – for example, GPS coordinates for records referring to particular geographic features or locations;
- External identifiers or reference numbers – for example, an accession number for an archive or a court reference number for a legal summons;
- Encoded information extracted from the contents of a record – for example, the name and address of a supplier for a purchase order formatted so that it can be used by an address verification application;
- Markings used privately by an organisation or within a specific legal or regulatory context – for example, a marking indicating whether a record has been assessed for release to the public under European Directive 2003/4/CE (Environmental Information);
- Workflow status – for example, if a record has been reviewed, the reviewer's name, review outcomes and date of review;
- Metadata captured from other business systems – for example, for email records the contents of the From, To, CC, Subject and other metadata in the email header;
- Metadata extracted from the components of a record – for example, for images, details such as their dimensions, pixel density, numbers of colours used, and compression format; and
- For users and groups – additional information extracted from the corporate directory, such as a person's position in the organisation, physical office location, email address and telephone number.

Throughout MoReq2010® the additional metadata listed above is referred to as "contextual metadata", because it is applied within a particular localised context, in an individual records system, and not for all records systems universally.

### 7.3.2 The MoReq2010® model metadata service

MoReq2010® provides for a metadata service that manages entity types and their related metadata element definitions. The metadata service may be shared, and used by several records systems and even by business systems as well, or it may be entirely built into a specific records system so that it is not distinguishable from the MCRS as a whole.

### 7.3.3 Metadata entity relationships

The relationships between the main entities in the model metadata service are shown in **Figure 7a**. This illustration captures all the relationships between entities. This is further broken down into individual relationships between entities, which are explained by reference to simpler diagrams below.

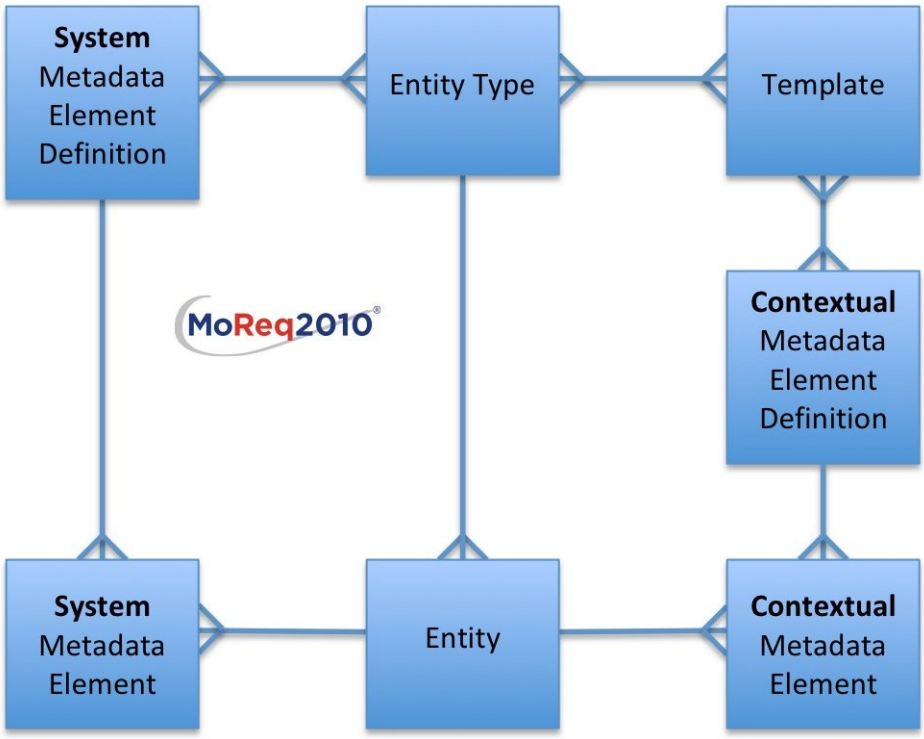


***Figure 7a** – Entity-relationships in the model metadata service*

At the centre of this diagram is the relationship between entities and entity types. Each entity must belong to one and only one of the entity types specified by MoReq2010®, this is shown in **Figure 7b**. Entities and entity types are managed under their appropriate service (see **R2.4.9**).

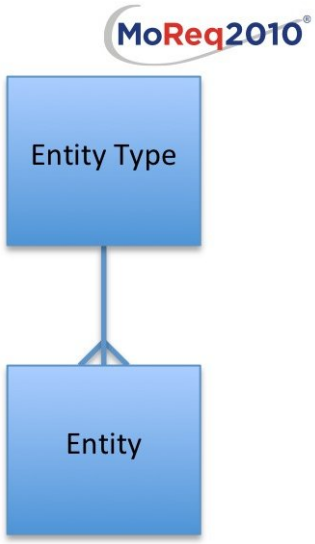


***Figure 7b** – Each entity has an entity type*

Each entity is comprised of metadata elements. The metadata elements mandated by MoReq2010® are the entity's system metadata elements, while additional descriptive metadata elements not referred to by MoReq2010® may also be added to the entity, either at creation or later. The relationship between an entity and its metadata elements is shown in

**Figure 7c**. The entity has many metadata elements, each of which belongs exclusively to that entity.

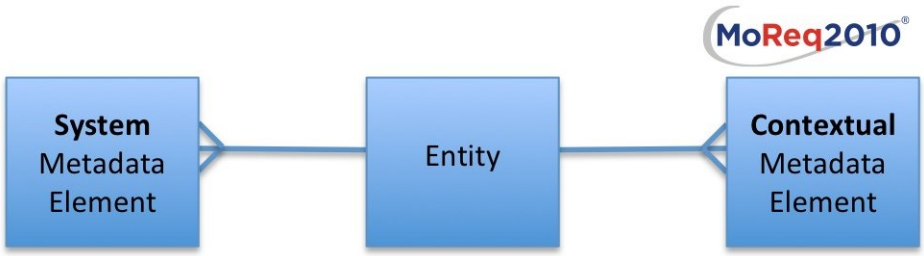


***Figure 7c** – Each entity has system metadata elements and may also be given contextual metadata elements*

In the same way as an entity is defined by an entity type definition, metadata elements are always linked to a metadata element definition. There must be either a system metadata element definition or a contextual metadata element definition that corresponds to every metadata element that is included an entity. This is shown in **Figure 7d**.

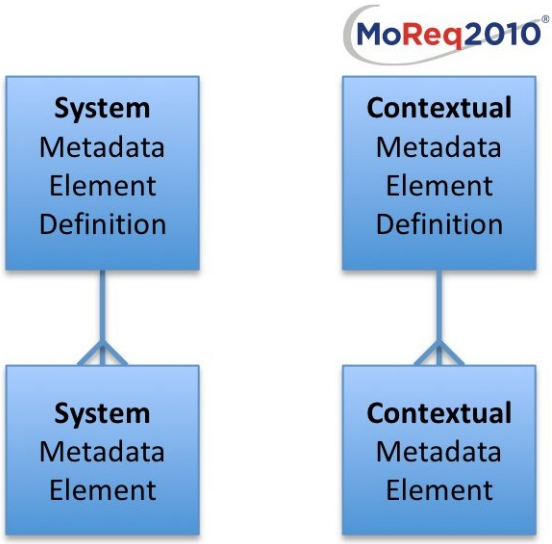


***Figure 7d** – All metadata elements are associated with a metadata element definition*

System metadata element definitions are defined and provided by MoReq2010® and are listed in **14.4 System Metadata Element Definitions**. These are associated with different entity types, as shown in **Figure 7e**, strictly in accordance with the functional requirements of MoReq2010®. Records systems are not able to define additional system metadata elements or additional entity types.

Contextual metadata element definitions are created by authorised users to meet local needs. These locally defined metadata element definitions are added to entities by being included in templates that are then applied to entities of a particular entity type.

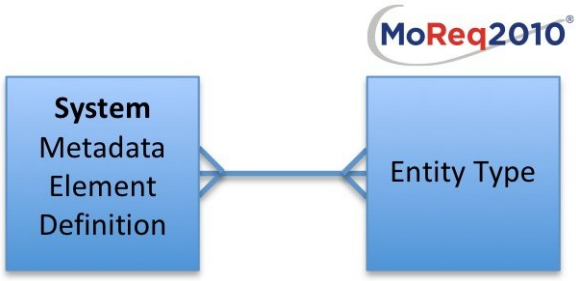


***Figure 7e** – System metadata element definitions are associated with an entity type*

Each template contains a collection of contextual metadata element definitions and may be applied to one or more entity types. If the template is specified as a service template for an entity type then it will be automatically applied by the MCRS whenever an entity of that type is created in that service. Otherwise, templates are applied to entities selectively by an authorised user or, for records and aggregations only, by classifying them with a particular class that is associated with a record type.

The relationship between templates and entity types, and templates and contextual metadata element definitions is shown in **Figure 7f**.

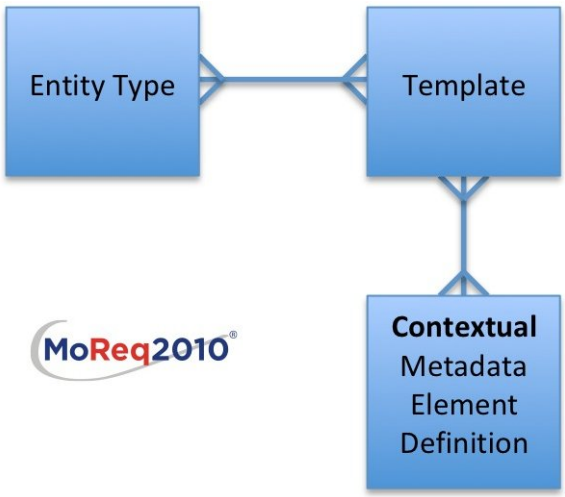


***Figure 7f** – Contextual metadata element definitions are associated with templates, which are in turn associated with an entity type*

Note that taken collectively, **Figures 7b** through **7f**, make up the entity relationship diagram given above in **Figure 7a**.

## 7.5 Functional Requirements

### R7.5.1

Each metadata service must manage metadata element definitions (**E14.2.11**) with the following system metadata:

- System Identifier (**M14.4.100**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),

- Scope Notes (**M14.4.97**),
- Presentation Order (**M14.4.84**),
- Min Occurs (**M14.4.56**),
- Max Occurs (**M14.4.53**),
- Is Modifiable Flag (**M14.4.46**),
- Is Entity Reference Flag (**M14.4.45**),
- Entity Reference Type Identifier (**M14.4.24**),
- Datatype (**M14.4.10**),
- Is Textual Flag (**M14.4.47**),
- Default Value (**M14.4.13**),
- Default Language Identifier (**M14.4.12**), and
- Retain On Destruction Flag (**M14.4.88**).

Metadata element definitions also have:

- Event history (see **2. System Services**),
- Access control list (***or equivalent***, see **4. Model Role Service**),

*These metadata apply to both system and contextual metadata element definitions.*

*System identifiers for every system metadata element definition in MoReq2010®, along with default titles, descriptions and the entity types they apply to, can be found in **14.4 System Metadata Element Definitions**. The MCRS must always use the MoReq2010® system identifiers and must not generate its own system identifiers for system metadata element definitions.*

*System metadata element definitions are specified by MoReq2010® and are not created by users and can never be destroyed. Their default Title, Description and Scope Notes may be replaced by the MCRS with localised values, however.*

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, a MoReq2010® access control list may not be present during system operation and may only be added to a metadata element definition at export.*

#### R7.5.2

The MCRS must allow an authorised user to create contextual metadata element definitions (**E14.2.4**) with the metadata specified under **R7.5.1**, that also include the following additional system metadata:

- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- First Used Timestamp (**M14.4.32**), and
- Destroyed Timestamp (**M14.4.17**).

*System metadata element definitions use only the base metadata element definition given in **R7.5.1.** Contextual metadata element definitions have additional metadata and more functions may be performed on them. Unlike system metadata element definitions, contextual metadata element definitions may be created, exported and destroyed. They are also included in templates.*

*Function reference: **F14.5.48***

#### R7.5.3

The MCRS must ensure that for each new contextual metadata element definition created under **R7.5.2**, the authorised user indicates whether the intent of the metadata element is to:

- Hold a reference to an entity by storing its System Identifier, or
- Hold a valid data value specified by a W3C XML datatype.

*Whether the contextual metadata element contains a System Identifier or a data value is determined by whether or not the Is Entity Reference Flag is set.*

*If the contextual metadata element definition is for an element that will hold a reference to an entity then the allowable entity types for the reference must be included in the Entity Reference Type Identifier element.*

*If the contextual metadata element definition is for an element that will hold a data value then the Datatype element must contain a value that conforms with the W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. The datatype definition must be self-contained within the element definition held by the MCRS, or long term preservation of the metadata cannot be assured. MoReq2010® does not allow XML definitions for datatypes that contain a URI to an external XSD schema.*

*Where relevant the Is Textual Flag must also be set to show that the metadata element will contain textual information. To set this value it is necessary to know the purpose of the field. Some datatypes may contain text but are not themselves textual elements; metadata with these datatypes do not require an indicator of which language they are written in. When a value is added to a textual metadata element it must have an accompanying language identifier under **R2.4.28**. For example, the datatype definition for a UUID is text based but not written in any particular language.*

*These values may be changed until the metadata element definition is first used, under **R7.5.9**.*

*Function reference: **F14.5.48***

#### R7.5.4

The MCRS must ensure that for each new contextual metadata element definition created under **R7.5.2**, the authorised user specifies the minimum number of occurrences of the metadata element as well as the maximum number of occurrences of the metadata element, if required.

*These values are stored in the Min Occurs and Max Occurs metadata elements, listed under **R7.5.1**, respectively. These settings may only be specified for new contextual metadata element definitions. For system metadata element definitions and for contextual metadata element definitions after they have been used, the values of Min Occurs and Max Occurs may not be changed (see **R7.5.9**).*

*Min Occurs may be set to zero, denoting that having a value is optional, or it may be set to one, indicating that a value is mandatory, or higher.*

*Max Occurs cannot be set to a number lower than the minimum number of values and may be set to one, meaning that only one value may be associated with the entity, any number higher than one, indicates a list of values.*

*If Max Occurs is not set then an unlimited number of values can be stored in the element, representing a list of any length.*

*Function reference:* ***F14.5.48***

**R7.5.5**

The MCRS must ensure that each metadata element definition is given a unique presentation order, under **R7.5.1**.

*The Presentation Order value provides a simple way of listing the metadata elements for any given entity in a chosen order that is neither random nor sorted on one particular attribute, such as title. Metadata is often presented in a list with the most significant elements towards the top and related elements grouped together.*

*For example, the desired order of metadata elements within an address might be:*

- *Addressee*
- *Street Address 1*
- *Street Address 2*
- *City*
- *Region*
- *Postal Code*
- *Country*

*This order would be observed if the contextual metadata element definition for "Addressee" had the lowest value for Presentation Order, and "Country" had the highest value for Presentation Order, with the various metadata elements arranged in sequence. The actual value of the Presentation Order element is not significant – more important is the relative difference to other elements' values.*

*An authorised user may change the Presentation Order of a metadata element definition, including for both system and contextual metadata elements, under* ***R7.5.8****, provided it remains unique.*

*Although records systems may also implement more sophisticated approaches to presenting metadata within their interfaces, Presentation Order remains valuable because it supports interoperability and gives a records system that imports entities important clues to how those entities were presented in their original environment.*

**R7.5.6**

For each active metadata element definition, under **R7.5.1**, except for system metadata elements that hold system identifiers or timestamps, the MCRS must allow an authorised user to specify whether or not metadata elements with that metadata element definition should be retained when the entity they belong to is destroyed.

*This is value is stored in the Retain On Destruction Flag and may only be changed while the metadata element definition remains active.*

*If this flag is not set then corresponding metadata element will be pruned from an entity when it is destroyed and will not be part of the resulting residual entity. Once it is pruned from a residual entity, the value of this element cannot be recovered. Note that changing the value of this flag will not have the effect of pruning the elements for entities that have already been destroyed, but only for entities that are destroyed in the future.*

*Pruning of some metadata on destruction is required to ensure that the residual entity is destroyed and remaining metadata cannot be used to reconstruct the original entity. Metadata values may also hold personal or sensitive information that should not be retained once the entity is destroyed.*

*System metadata elements that hold system identifiers (references to the entity and other entities) and timestamps are necessary to the integrity of the entity and must not be deleted from any entity. For example, the presence of the Destroyed Timestamp indicates that an entity is a residual entity.*

*Function reference:* ***F14.5.114***

#### R7.5.7

The MCRS must allow an authorised user to delete any metadata element, except a system metadata element that holds a system identifier or a timestamp, from any residual entity, provided the user gives a reason for the deletion and an event is generated.

*This requirement describes a special circumstance where it becomes necessary to remove metadata or events from individual entities, for example, where this has been ordered by a court of law. It should not be necessary to exercise this requirement as part of routine records management processes.*

*This requirement may only be applied to a residual entity which has already been destroyed, and is in addition to the automatic pruning of metadata and events which occurs on destruction.*

*Note that a new event must* ***always*** *be generated for this function, superseding requirement* ***R2.4.13****. The authorised user must give a reason for deleting the deleted metadata elements which must be stored as the Comment in the new event.*

*The new event must also contain a Deleted Metadata Element Definition Identifier (see* ***M14.4.15****) which indicates the system identifiers of the definitions of the deleted metadata elements. This shows that metadata elements were deleted from the entity without retaining their values.*

*See also* ***R2.4.21****.*

*Function references:* ***F14.5.8****,* ***F14.5.27****,* ***F14.5.40****,* ***F14.5.60****,* ***F14.5.74****,* ***F14.5.98****,* ***F14.5.123****,* ***F14.5.146****,* ***F14.5.168****,* ***F14.5.182***

#### R7.5.8

The MCRS must allow an authorised user to modify the following metadata for any active metadata element definition, including system metadata element definitions:

- Title,
- Description,
- Scope Notes,
- Presentation Order,
- Default Value, and
- Default Language Identifier.

The MCRS must ensure that the default value, if modified, is always consistent with the datatype of the metadata element definition.

*These values may be replaced for any system metadata element definition, including replacing the English titles and descriptions for system metadata definitions, provided by MoReq2010®, with alternatives in any language in accordance with the organisation's operational and business needs.*

*As described in the rationale to* ***R7.5.5****, the meaning of Presentation Order may be interpreted differently by different MCRS solutions. Each MCRS may make its own provision for how Presentation Order is determined and maintained. It may not be possible for authorised users to directly modify the Presentation Order numeric value if instead, for example, the MCRS allows them*

*to manipulate the positioning of elements on a display which has the equivalent effect of changing the underlying presentation order.*

*Function reference:* ***F14.5.113***

#### R7.5.9

In addition to **R7.5.8**, the MCRS must allow an authorised user to modify the following metadata for any active contextual metadata element definition that has never been applied to an entity:

- Min Occurs,
- Max Occurs,
- Is Modifiable Flag,
- Is Entity Reference Flag,
- Entity Reference Type Identifier,
- Datatype, and
- Is Textual Flag.

*Once the contextual metadata element definition has been used these values may no longer be changed, see* ***R7.5.3*** *and* ***R7.5.4****.*

*Function reference:* ***F14.5.53***

#### R7.5.10

The MCRS must allow an authorised user to delete a contextual metadata element definition that has never been applied to an entity.

*When a contextual metadata element definition is first applied to an entity, the MCRS must set the First Used Timestamp.*

*Note that this function does not apply to system metadata element definitions that may never be deleted or destroyed.*

*Function reference:* ***F14.5.49***

#### R7.5.11

The MCRS must allow an authorised user to destroy a contextual metadata element definition once it has been applied to an entity. The MCRS must ensure that when a contextual metadata element definition is destroyed, then:

- New metadata elements with this contextual metadata element definition are no longer created and applied to entities, and
- Existing metadata elements with this contextual metadata element definition remain associated with entities, but their values may no longer be modified.

*Once a contextual metadata element definition has been used it may no longer be deleted under* ***R7.5.10****, it may only be destroyed. Destroying a contextual metadata element definition will leave a residual entity. New metadata elements cannot be created from residual contextual metadata elements, but existing metadata elements remain as "read only" elements (note that this is the same effect as clearing the Is Modifiable Flag in the metadata element definition).*

*Note that this function does not apply to fixed roles provided in an MCRS solution (see* ***4.3.6 Preconfigured Roles****).*

*Function reference:* ***F14.5.51***

#### R7.5.12

Subject to **R2.4.22**, the MCRS must allow an authorised user to browse the metadata element definitions and templates in the metadata service, and entity types in their respective services under **R2.4.9**, and inspect their metadata in the following ways:

- Browse across all metadata element definitions in the metadata service in presentation order and inspect their metadata,
- Browse across all templates in the metadata service and inspect their metadata,
- Browse from a system metadata element definition to any of the entity types that contain that system metadata element definition and inspect their metadata,
- For each entity type, under **R2.4.10**, browse from the entity type to the system metadata element definitions associated with that entity type,
- Browse from a contextual metadata element definition to any of the templates that contain that contextual metadata element definition,
- Browse from a template to any of the contextual metadata element definitions associated with that template,
- Browse from a template to any of the entity types associated with that template, and
- For each entity type, under **R2.4.10**, browse from the entity type to the templates associated with that entity type.

*The terms "browse" and "inspect" are defined in* ***13. Glossary of Terms****. Further information on the MoReq2010® entity types and system metadata element definitions is provided in* ***14. Information Model****.*

*Function references:* ***F14.5.83***, ***F14.5.109***, ***F14.5.171***

#### R7.5.13

Whenever metadata elements are added to entities, upon the creation of the entity, or when a template is applied to an existing entity, or at any other time, and subsequently whenever the value of a metadata element is modified, the MCRS must ensure that the following rules are observed:

- The metadata element must be initialised with the Default Value in its metadata element definition, if one has been provided;
- The metadata element must never be given a value that is inconsistent with the Datatype of its metadata element definition;
- If the metadata element contains a reference to an entity, as indicated by the Is Entity Reference Flag in its metadata element definition, then its value must represent a valid System Identifier belonging to an entity of the type indicated by the Entity Reference Type Identifier;
- If the metadata element is textual, as indicated by the Is Textual Flag in its metadata element definition, then it must always have an accompanying language identifier which by default will be derived from the Default Language Identifier for the metadata element definition, if one has been provided;
- The metadata element must never have fewer values provided for it than allowed by the value of Min Occurs in its metadata element definition;

- The metadata element must never have more values provided for it than allowed by the value of Max Occurs in its metadata element definition;
- Once the metadata element has been created and given a value, if it is not modifiable under its metadata element definition as indicated by the Is Modifiable Flag, then the MCRS must prevent the value of the metadata element from being altered by a user; and
- If the metadata element definition for the metadata element has been destroyed then the MCRS must prevent the value of the metadata element from being altered by a user, under **R7.5.11**.

#### R7.5.14

The MCRS must allow an authorised user to create active templates (**E14.2.15**) with at least the following system metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- First Used Timestamp (**M14.4.32**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),
- Template Entity Type Identifier (**M14.4.102**),
- Template Service Identifier (**M14.4.103**),
- Template Class Identifier (**M14.4.101**),
- Contextual Metadata Element Definition Identifier (**M14.4.8**), and
- Destroyed Timestamp (**M14.4.17**).

Each template must also have:

- Event history (see **2. System Services**),
- Access control list (***or equivalent***, see **4. Model Role Service**),

And may have:

- Contextual metadata.

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, a MoReq2010® access control list may not be present during system operation and may only be added to a template at export.*

*Function reference: **F14.5.165***

#### R7.5.15

The MCRS must allow an authorised user to modify the following metadata for any active template:

- Title,
- Description,
- Template Entity Type Identifier,
- Template Service Identifier,
- Template Class Identifier,
- Contextual Metadata Element Definition Identifier, and

- Any contextual metadata elements.

*Note that changes to metadata will not affect or reverse any previous applications of the template.*

*The Template Entity Type Identifier indicates which types of entities the template can be applied to. If the template is a service template, as indicated by the presence of a Template Service Identifier, then it will be applied automatically to all new entities of these types that are created by the service, or bundle of services under* ***R2.4.1****, whose System Identifier matches the Template Service Identifier, under* ***R7.5.18****.*

*The Template Class Identifier indicates whether the template is associated with one or more classes in the classification service. If this is the case then the template will be automatically applied to all new records or aggregations (depending on the Template Entity Type Identifier) created with this classification, or changed to this classification.*

*The Contextual Metadata Element Definition Identifier is used to include contextual metadata element definitions in the template. Each template may include multiple contextual metadata element definitions, each specifying a contextual metadata element which is added to the metadata of the target entity whenever the template is applied. System Metadata Element Definitions cannot be included in templates.*

*Function reference:* ***F14.5.175***

#### R7.5.16

The MCRS must allow an authorised user to delete any template that has never been applied to an entity.

*This is the purpose of the First Used Timestamp, listed under* ***R7.5.15****.*

*Function reference:* ***F14.5.166***

#### R7.5.17

The MCRS must allow an authorised user to destroy an active template, preventing it from being applied to entities, provided the template has previously been applied to an entity.

*This will leave a residual template. If the template has never been used then it should be deleted, under* ***R7.5.16****.*

*Function reference:* ***F14.5.169***

#### R7.5.18

The MCRS must automatically apply an active template to an entity on its creation in all cases when:

- The entity is created in a service, or bundle of services under **R2.4.1**, for which there are one or more service templates of the same entity type, indicated by a combination of the Template Service Identifier and the Template Entity Type Identifier;
- The entity is an aggregation and is created and classified with a class for which there are one or more aggregation templates, indicated by a combination of the Template Class Identifier and the Template Entity Type Identifier; or

- The entity is a record and is created and classified with a class for which there are one or more record templates, indicated by a combination of the Template Class Identifier and the Template Entity Type Identifier.

*Applying a template to an entity means that the MCRS adds contextual metadata elements to the entity for all active metadata element definitions included in the template, that have not previously been applied to the entity.*

*New metadata elements are only added to the entity where it does not already have an existing metadata element with the same contextual metadata element definition. Each entity may only have one metadata element corresponding to a particular contextual metadata element definition (although the one metadata element may have multiple values under **R7.5.4**).*

*Function references: **F14.5.5**, **F14.5.24**, **F14.5.38**, **F14.5.57**, **F14.5.71**, **F14.5.95**, **F14.5.121**, **F14.5.143**, **F14.5.165**, **F14.5.179***

#### R7.5.19

The MCRS must allow an authorised user to apply an active template to an active entity at any time, provided the entity's type matches the Template Entity Type Identifier.

*See the rationale to **R7.5.18**. The MCRS may also automatically apply additional contextual metadata element definitions and values to an entity at any time, including on export, in accordance with its own design purpose and internal logic. For example, the MCRS may be capable of extracting additional metadata automatically from certain types of electronic content, such as the properties of a Microsoft Word document. As new records with these types of content are created, the MCRS might analyse the content of each component, automatically extract any properties it identified, and then add them as additional metadata elements to the component.*

*Function references: **F14.5.2**, **F14.5.23**, **F14.5.37**, **F14.5.55**, **F14.5.70**, **F14.5.93**, **F14.5.115**, **F14.5.141**, **F14.5.157**, **F14.5.164**, **F14.5.177***

#### R7.5.20

The MCRS must ensure that once metadata elements have been added to an entity under either **R7.5.18** or **R7.5.19**, they are never deleted while the entity remains active.

*Note that the metadata elements may be automatically pruned from the entity on its destruction, depending on the setting of the Retain On Destruction Flag in the metadata element definition under **R7.5.6**, or deleted from a residual entity, under **R7.5.7**.*

# 8. Disposal Scheduling Service

## 8.1 Service Information

| | |
|---|---|
| **Service Name** | **Disposal Scheduling Service** |
| **Service Version** | **1.0** |
| **Implements Service Identifier *(see M14.4.42)*** | fd05e284-181f-4f5d-bd8c-4bed835c8931 |

## 8.2 Key Concepts

### 8.2.1 The MoReq2010® record lifecycle

Disposal schedules are used to manage the lifecycles of records in all MCRS solutions.

A record, once it has been created in an MCRS is never be deleted in full, as if it had never existed. This concept of accountability is important to good records management: although the complete record and its content no longer exist, there remains a residual record to show that it was once held by the MCRS. The residual record, which remains with the MCRS for the life of the system, proves not only that a record was once active but also, and possibly more importantly, that the record was properly disposed of under an appropriate disposal schedule.

The transition from an active entity with complete metadata, entity history and content to a residual entity is termed "destruction" to distinguish it from "deletion" (where all trace of an entity is removed). MoReq2010® also applies this concept to entities other than records, such as aggregations, classes, disposal schedules, etc. Destruction is an irreversible process because parts of the entity are erased so that the entity cannot be returned to an active state. **Figure 8a** gives a simple overview of the MoReq2010® record lifecycle which considers only two events: the record's creation in the MCRS as an active entity and its subsequent destruction.

Note that unlike the generic entity lifecycle shown previously in **Figure 2g**, with records in particular there is no concept of "first use" and no period immediately after creation where the record may be deleted. The simple view shown in **Figure 8a** will be expanded on below where more detailed record lifecycles show how each disposal action has its own variation of the record lifecycle.

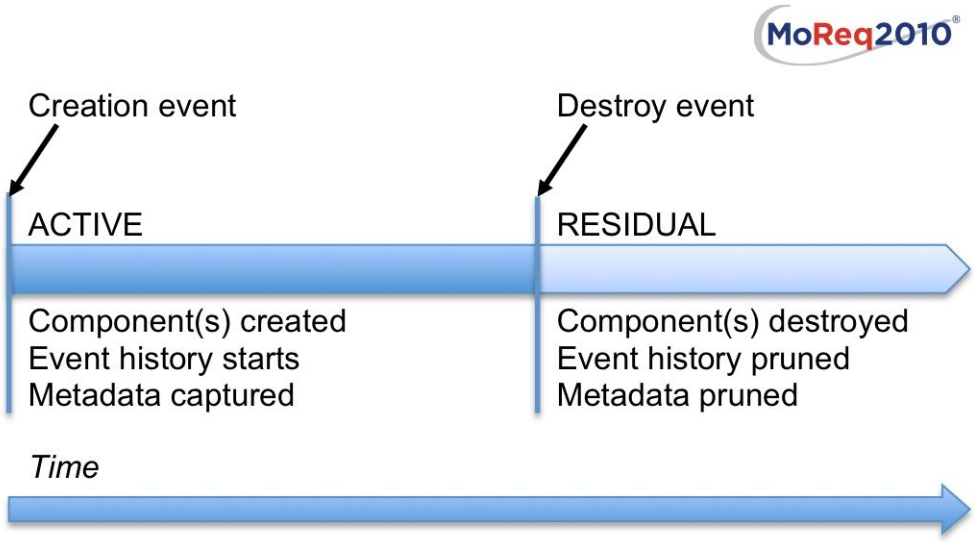


***Figure 8a** – Simple view of a record's lifecycle*

As **Figure 8a** shows, when an active record is created its components and their component content are simultaneously created, metadata is captured for the record and its components, and the record's event history begins with the creation event. When an active record is subsequently destroyed then its metadata and event history are pruned, along with the metadata and event histories of its components, and most importantly the record's component content is deleted.

The events and the metadata elements to be automatically pruned on destruction of an entity are configurable by an authorised user (see **R2.4.20** and **R7.5.6**). The remaining metadata of the record, along with their remaining event history, make up the residual record and its residual components.

Pruning is an important process in ensuring the proper destruction of the content of records, especially in sensitive environments where these events and metadata may reveal information about the original content of the record and may be able to be used to partially (or fully) reconstruct the destroyed content. For example, the metadata or event history of a hospital record may include the name of a patient, or a particular incident in that patient's medical history, in perpetuity, in breach of privacy considerations.

Pruning of events and metadata from a residual record that has previously been destroyed, may also be done by an authorised user, usually in response to a court order or judgement (see **R2.4.21** and **R7.5.7**). From time to time a particular record, including its metadata and events, must be expunged because a court feels their continued existence is problematic or unjust.

A secondary consideration in pruning metadata and event histories on destruction is storage capacity, since residual records are retained for the life of the MCRS.

### 8.2.2 Disposal schedules and disposal actions

Disposal schedules are critical to managing records because MoReq2010® specifies that a record in an MCRS may only be destroyed as part of a disposal process that is governed by the disposal schedule that has been assigned to that record. It is the record's disposal

schedule that determines how long the record is retained and how it is subsequently disposed of at the end of its retention period. For this reason, all records must have a disposal schedule.

While all disposal schedules must conform with the MoReq2010® disposal process, they may specify very different behaviours. The disposal schedule for one record may state that it is to be retained permanently and must never be destroyed. A different disposal schedule may state that a record is to be destroyed immediately. Yet another disposal schedule may state that a record is to be reviewed at the end of its retention period, and so on.

Each of these examples results in a different disposal action. MoReq2010® requires that all disposal schedules must specify one of four different possible outcomes:

- Retain permanently
- Review at the end of the retention period
- Transfer at the end of the retention period
- Destroy at the end of the retention period

### 8.2.3 Calculating the retention period

Each disposal schedule will have a retention period defined as a number of days, weeks, months or years. The retention period begins on a particular retention start date, which is defined by a retention trigger event. Retention trigger events can relate to the individual record or to the record's parent aggregation.

Organisations that prefer to manage whole aggregations, and dispose of them collectively, should specify aggregation-related retention triggers for their disposal schedules. They should also avoid classifying records individually within aggregations. The following retention triggers relate to a record's parent aggregation:

- From the aggregation's originated date,
- From the date of the most recent addition to the aggregation,
- From the date when the aggregation was closed, and
- From a date specified by a contextual metadata element associated with the aggregation.

By contrast, the retention triggers below relate to individual records. When these retention triggers are used then records will be selectively disposed of at different times from within their parent aggregations, while other records in the same aggregations remain active:

- From the record's Originated Date/Time,
- From the date record's was added to parent aggregation, and
- From a date specified by a contextual metadata element associated with the record.

The following retention triggers can be applied equally to both individual records or to aggregations:

- Retain Permanently (there is no retention trigger),
- From now (meaning with immediate effect), and
- From the date of the last review.

The retention start date for a record must be recalculated by the MCRS if the record's disposal schedule changes or if the value referred to by the retention trigger is updated, for example, whenever:

- A record's default disposal schedule is replaced through change (see **R5.4.4**) or reclassifying the record (see **R5.4.8**, **R6.5.4** and **R6.5.12**),
- A record's default disposal schedule is replaced by moving it, or its parent, to a new aggregation (see **R6.5.8** and **R6.5.13**), or
- A record's default disposal schedule is overridden by one directly applied to the record (see **R6.5.15**).

In addition to the above, the following are all possible retention triggers and changes to them may also result in recalculation of retention start dates:

- A record's Originated Date/Time is changed (see **R2.4.26**),
- A record's parent aggregation is closed (see **R6.3.7**), or
- A contextual metadata element associated with an aggregation is modified (see **R8.4.4**).

It should be noted that MoReq2010® views disposal as most likely to be a scheduled activity and therefore it does not require that retention triggers be checked and retention start dates be recalculated more than once per day. It is assumed that new batches of records will become due for disposal on a daily basis and records management routines will be planned around this. However, under **R8.4.14**, it is possible for an authorised user to individually request the immediate update of a record's disposal status by the MCRS.

It should also be noted that MoReq2010® requires that the Originated Date/Time for records and aggregations be used for calculating retention start dates, and not the Created Timestamp for these entities. This allows interoperability between records systems while retaining continuity over the record lifecycle. Each time a record or aggregation is imported into a new records system it will be created again in that system. It will have a different Created Timestamp that reflects when it was imported into the new MCRS, but by comparison the value set in the previous MCRS for the Originated Date/Time will not be updated when the record or aggregation is imported.

One consequence of this is that the retention start date for a record may predate the record's creation event. For example, if a new record is created in an aggregation and given a disposal schedule with the retention trigger based on its parent aggregation's Originated Date/Time then the retention period for the record will have already effectively commenced prior the record being created. This circumstance is not shown in any of the accompanying illustrations, such as **Figures 8c**, **8d** and **8e**, which for ease of understanding always show the retention start date occurring after the record creation date.

### 8.2.4 Confirming a record's disposal

In addition to the disposal action, the retention trigger and the retention period, a disposal schedule must also specify a confirmation period. The confirmation period represents the allowed period of time for carrying out the disposal action. The length of the confirmation period will vary from organisation to organisation and may also be different on different disposal schedules.

The precise meaning of the confirmation period depends on the disposal action chosen. For review actions, the confirmation period represents the time allocated for completing the review and applying the review decision. For destroy actions, the confirmation period represents the time allocated for deleting the content of the record and confirming that it has been deleted.

Transfer actions have two confirmation periods. First the records must be transferred to their new location outside the management of the MCRS. This will usually involve exporting the records from the MCRS. Upon confirmation that the transfer is complete the disposal action changes from transfer to destroy and the disposal process enters a second confirmation period to ensure that the content of the record held by the MCRS is destroyed.

If any confirmation period is exceeded without the MCRS receiving the necessary confirmation that the disposal action has been completed then the MCRS raises an alert that is sent to authorised users indicating that the relevant records are overdue for disposal. This measure is intended to ensure that records are disposed of in a timely manner in the shortest possible period following the disposal due date.

### 8.2.5 Permanent retention lifecycle

An important aspect of records management is the preservation of important records for very long periods of time, including the ability to designate some records that are never to be disposed of. In MoReq2010®, this is done by applying a disposal schedule with a retention trigger that specifies permanent retention. This type of disposal schedule, with no retention trigger, has the effect of preventing the calculation of a retention start date and a subsequent retention period. The effect of permanent retention is shown by **Figure 8b**, which may be contrasted with the simple lifecycle illustrated by **Figure 8a**.

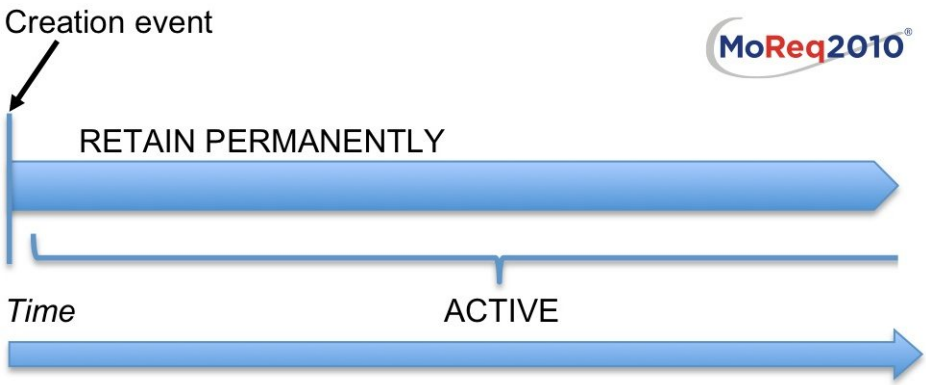


***Figure 8b** – If its disposal schedule specifies permanent retention then no retention start date will be set for a record and, without its disposal schedule being changed, it will be remain active for the life of the MCRS*

### 8.2.6 Review lifecycle

MoReq2010® encourages disposal to follow, and be determined by, classification wherever possible. This means that the period of time a record is kept and its method of disposal is based on its business classification. This principle of good management practice with records is sometimes referred to as their "sentencing on creation".

There may be some occasions, however, when the importance of a record and the length of time it should be retained are not known at the time the record is created, and cannot be calculated simply from subsequent events, such as the closing of the aggregation in which the record is placed. It may also be that in some jurisdictions the retention period is so long that it is felt that the guidance for their retention may change in the intervening period. For example, if it is required that a particular class of record be destroyed after 40 years, a records manager may argue that there is some likelihood that in 40 years time the 40 year disposal rule may have been updated. Under these circumstances, where there is a reasonable doubt about their final destiny, records can be scheduled for later review, rather than for permanent retention, transfer or destruction.

When a record's disposal action is set to review, it is not immediately subject to transfer or destruction. Instead the outcome of the review must include the application of a disposal schedule to the record based on the review decision. The new disposal schedule will replace the previous disposal schedule associated with the record and will then specify the ultimate fate of the record, or it may be used to schedule another later review, or to retain the record permanently.

**Figure 8c** shows how the completion of a review results in the application of a new disposal schedule to the record, which will in turn force the calculation of a new retention start date, disposal action and confirmation period. The disposal process will therefore keep going and may even result in further review periods. Note that disposal holds do not interrupt the review process, they only prevent the eventual destruction of a record.

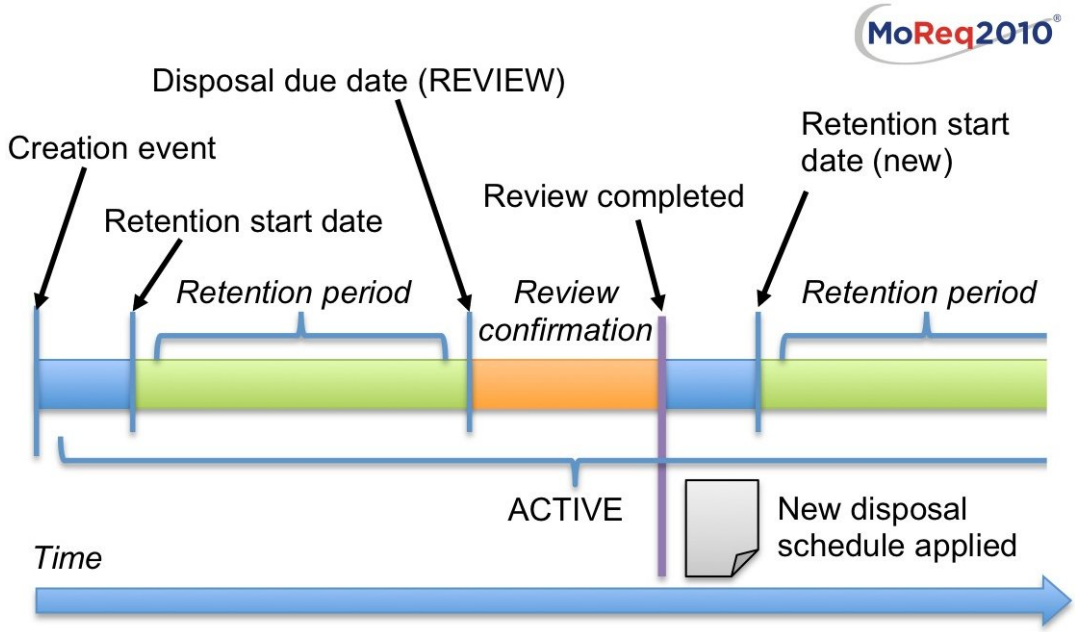


***Figure 8c** – If its disposal schedule specifies that a record be reviewed then a new disposal schedule must be applied to the record as part of completing and implementing the review decision*

### 8.2.7 Transfer lifecycle

A disposal schedule may state that a record is to be transferred to the control of another records system, for example, a more centralised corporate records system, a secondary

storage facility, or an archive. Transfer occurs in two phases. The records must firstly be exported from the MCRS and imported into the other records system. On confirmation that the transfer has taken place and the destination system has successfully received the MCRS changes the record's disposal action from transfer to destruction, and they are subsequently destroyed.

**Figure 8d** shows the record lifecycle as it applies to a transfer disposal action. The record is not destroyed until the successful transfer has been confirmed by an authorised user. Where the record is under a disposal hold, the first phase of transfer may continue, however the disposal process must be halted prior to the second phase destruction of the record.

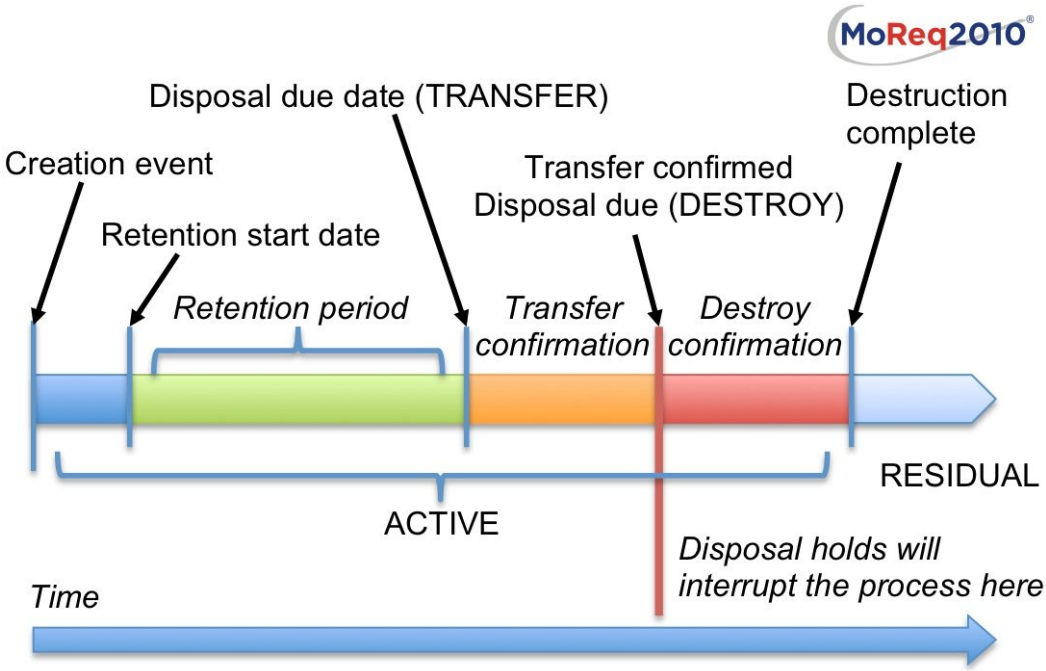


***Figure 8d** – If its disposal schedule specifies the transfer of a record then it is destroyed from the MCRS, but only after the transfer is confirmed as completed*

MoReq2010® allows for transfers to be cancelled. If a transfer is cancelled then the record must be given a new disposal schedule; the outcome will therefore be similar to the review shown in **Figure 8c**.

### 8.2.8 Destruction lifecycle

**Figure 8e** shows the record lifecycle as it applies to a disposal action of destroy. The destruction of records, either in response to a destroy disposal action, or as the second phase of a transfer, is subject to particular constraints. If a disposal hold is in place for the record then the MCRS must mark the record as being held and prevent it from being destroyed. In this case the destroy confirmation period must not begin until the disposal hold has been lifted. More information about disposal holds can be found in **10. Disposal Holding Service**.

It is important to note that once a disposal process has entered the confirmation period it is no longer possible for the imposition of a disposal hold to prevent the destruction of a record. This is because the instruction to destroy the content is assumed to have already been given and it is only awaiting confirmation that this has occurred. Once the instruction

is issued, MoReq2010® does not make any provision for a recall, and preventing the destruction of the record is therefore outside the control of the MCRS.

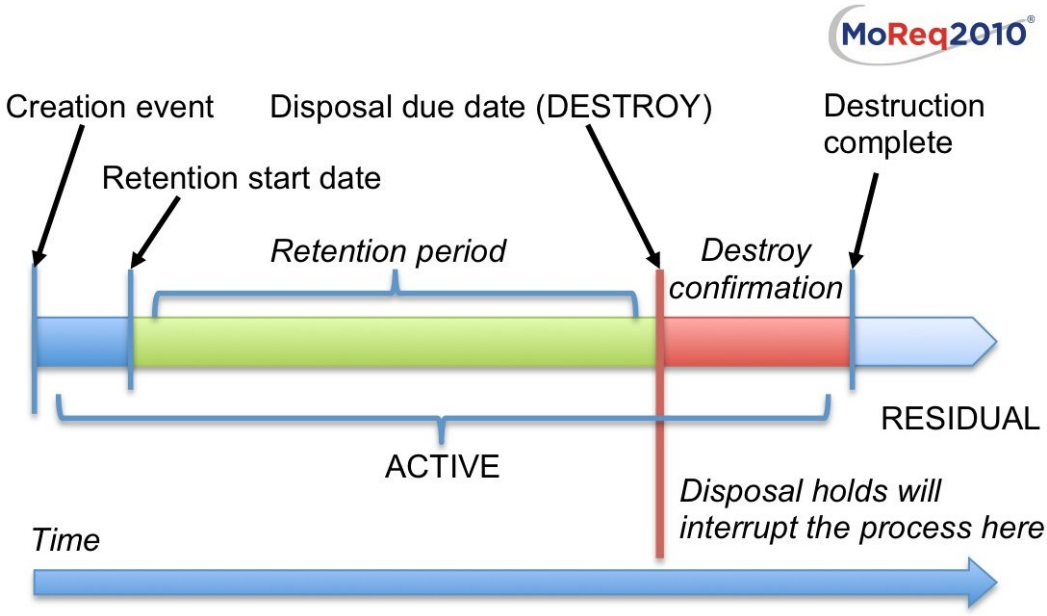


***Figure 8e** – If its disposal schedule specifies the destruction of a record then there is usually a confirmation period following the disposal due date*

How records are destroyed in an MCRS will depend upon the nature of the content of their components. This in turn may depend upon the design and the purpose of the MCRS. When records are created in the MCRS, their components will be created with content that the MCRS is either able to automatically delete or with content that requires confirmation of deletion. This is determined by the Automatic Deletion Flag included in the metadata of a component, under **R6.5.19**.

An MCRS that directly manages its own content repository is more likely to be capable of automatically deleting the content of components when records are destroyed. An MCRS that manages records in place, in other business systems, or records with physical rather than digital content is more likely to require that separate confirmation be received of the destruction of the content of records scheduled for destruction.

MoReq2010® does not constrain the architecture of MCRS solutions such that they must be able to automatically destroy record content. Nor does MoReq2010® specify that all MCRS solutions must be able to support record components that require confirmation. Many MCRS solutions will support both types of content or allow different content types to be configurable. Suppliers may build and support any of these options in their products.

For this reason, the MoReq2010® disposal process specifies that when a record is due to be destroyed, the MCRS must check its components to determine whether they can be automatically destroyed or must be destroyed after confirmation. The MCRS will then wait for confirmation of the destruction of the content of any components that cannot be automatically destroyed, before destroying the remaining components and the record itself.

Note that this approach, where automatic destruction is not specified as an attribute of the disposal schedule, but is instead a function of the nature of the content of the record, as well as the design and implementation of the MCRS, differs from previous versions of MoReq®.

### 8.2.9 Bottom up destruction

Under MoReq2010®, disposal schedules are only ever applied to records; they are not applied to aggregations. This is different to previous versions of the specification. Aggregations have classes but do not have disposal schedules, instead the destruction of aggregations is managed automatically by the MCRS using the principle of "bottom up" destruction.

Under this principle, individual records in an aggregation may be destroyed at different times. When this occurs the destroyed record becomes a residual record but there is no impact on the other records, whether active or residual, in the same aggregation. There is also no impact on the aggregation itself, until the last active record in the aggregation is destroyed.

Bottom up destruction means that when the last active record in an aggregation is destroyed then the aggregation itself will be destroyed automatically by the MCRS. However, automatic destruction of an aggregation only takes place when the aggregation is closed. An open aggregation cannot be destroyed. If the aggregation is already closed at the time of destruction of the last active record, or later if the aggregation is subsequently closed, then the aggregation will be destroyed automatically, provided no more active records have been added to it.

It is also worth noting that an aggregation, whether open or closed, cannot be automatically destroyed if it has never been used, although an aggregation that has never been used can be deleted instead.

**Figure 8f** shows how bottom up destruction applies to both open and closed aggregations when the last record in the aggregation is destroyed.

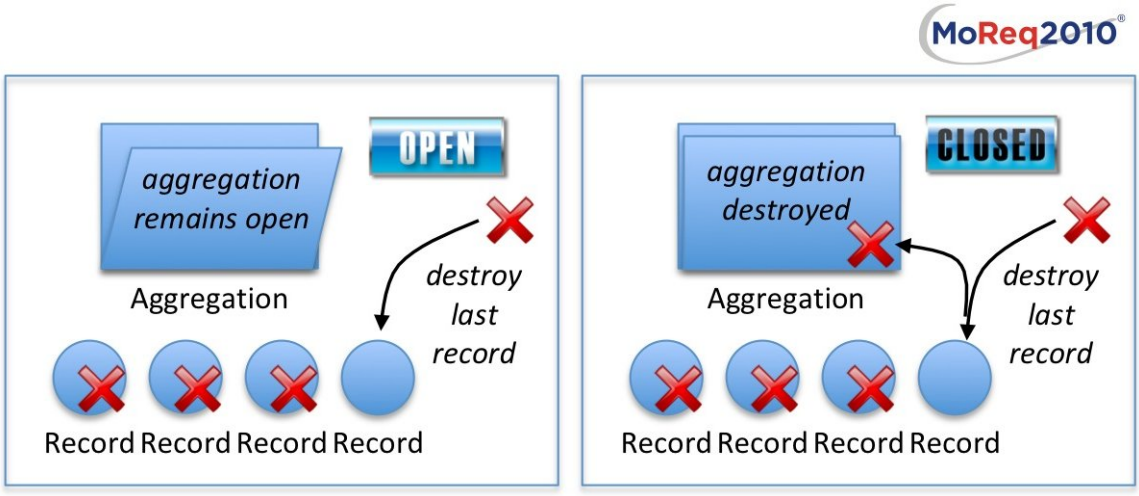


***Figure 8f** – According to the principle of bottom up destruction, when the last record in an aggregation is destroyed then the aggregation will be automatically destroyed, but only when it has been closed*

Bottom up destruction not only affects aggregations that contain records, but also aggregations that contain other aggregations. A parent aggregation will be automatically destroyed by an MCRS when all of its child aggregations have been destroyed. The same rule that the parent aggregation must be closed also applies.

This cumulative effect of the destruction of aggregations cascading upwards is shown in **Figure 8g**.

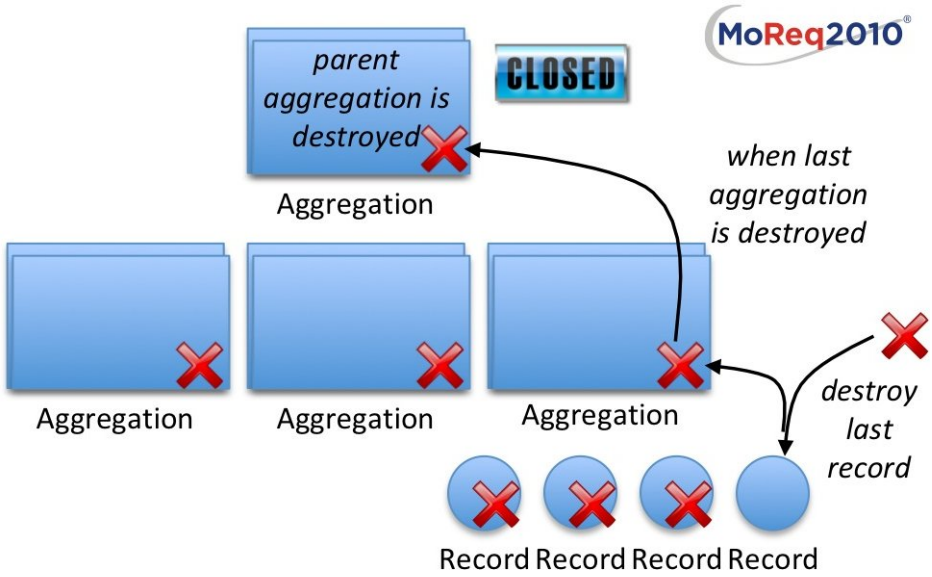


***Figure 8g** – A closed aggregation will be automatically destroyed when all of its child entities, either records or other aggregations, have been destroyed; this may trigger the destruction of its parent aggregation, and so on*

### 8.2.10 Disposal conflicts

Some specifications for records systems, including previous versions of MoReq®, allow disposal conflicts to occur where two or more disposal schedules are applied simultaneously to the same record. The resulting disposal conflict must then be managed, either by manual user intervention to resolve the ambiguity, or by the implementation of an algorithm that determines which of the conflicting disposal schedules is the more important and should therefore take precedence.

MoReq2010® has been deliberately designed to avoid disposal conflicts of this nature by ensuring that each record has one, and only one, disposal schedule applied to it at any time. The disposal schedule initially applied to each record is the default disposal schedule associated with the record's class. This default disposal schedule may then be overridden by an authorised user applying a different disposal schedule directly to the record itself. The new disposal schedule can itself be overridden, and so on, provided the record only ever has a single disposal schedule.

The most common reason why a record's default disposal schedule will be overridden is to apply a new disposal schedule as a result of a review decision.

The following is an example of the lifecycle of a record:

- A record is created with a default disposal schedule inherited from its class of, "Review 2 years after aggregation closed";
- The record's aggregation is subsequently closed;
- After two years the record is reviewed and, as a result, a new disposal schedule is applied, "Review 1 year after last review";
- One year later the record is reviewed again and a third disposal schedule is applied, "Destroy in 6 months";
- Six months later, or three and a half years in total after the record's aggregation was closed, the record is destroyed by the MCRS.

In this example, the record is created with a default disposal schedule which is then overridden twice during the active life of the record. At any point in time, however, the record will be subject to only one disposal schedule and will be following only one corresponding disposal process. In this way, MoReq2010® ensures that an MCRS never has to manage disposal conflicts or ambiguities.

### 8.2.11 The disposal process

An MCRS must ensure that the disposal process is carried out regularly for each of its active records. This may be done in real-time or it may be done as a scheduled activity but MoReq2010® specifies that it should take place on a daily basis so that each day authorised users are able to carry out records management activities on records due for disposal.

The flowchart in **Figure 8h** gives a logical view of the disposal process for each record, showing the various decision points and processing required.

This diagram is meant to represent visually the functional requirements contained in this section and is considered to be a logical view. It is not optimised in any way, and it is not intended to be reproduced in code, so long as MCRS solutions provide the same logical outcomes as described by the process for the same input data.

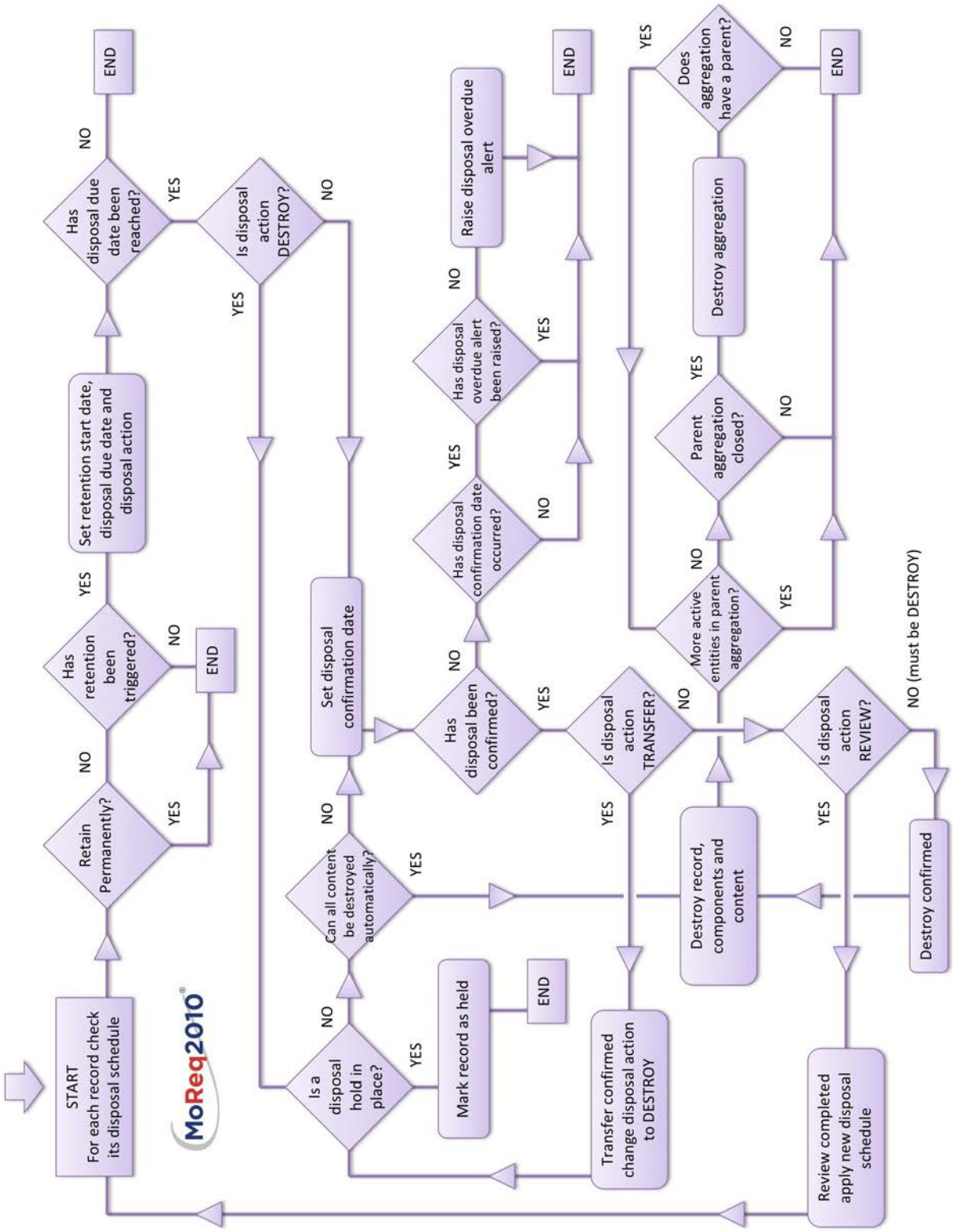


***Figure 8h** – The integrated disposal process illustrating all the disposal choices provided within MoReq2010®*

### 8.2.12 Disposal scheduling as a service

MoReq2010® describes disposal scheduling as a service that is used by an MCRS. Most MCRS software will have a disposal scheduling service built in, however it is also possible

that in the future standalone disposal scheduling services will be created that allow centralised management of an organisation's disposal schedules across multiple MCRS solutions. One possible use of such a service would be for an authority to issue and maintain an industry-wide set of disposal schedules in the form of a disposal scheduling service, available on the Internet, to be used across a particular sector.

## 8.4 Functional Requirements

### R8.4.1

The MCRS must allow an authorised user to create new disposal schedules (**E14.2.6**) with the following system metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- First Used Timestamp (**M14.4.32**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),
- Mandate (**M14.4.51**),
- Scope Notes (**M14.4.97**),
- Disposal Action Code (**M14.4.18**),
- Retention Trigger Code (**M14.4.94**),
- Retention Trigger Element Identifier (**M14.4.95**, ***or equivalent***),
- Retention Period Interval Code (**M14.4.90**),
- Retention Period Duration Number (**M14.4.89**),
- Retention Period Offset Code (**M14.4.91**),
- Retention Period Offset Month Code (**M14.4.92**),
- Confirmation Period Interval Code (**M14.4.7**),
- Confirmation Period Duration Number (**M14.4.6**), and
- Destroyed Timestamp (**M14.4.17**).

Each disposal schedule also has:

- Event history (see **2. System Services**),
- Access control list (***or equivalent***, see **4. Model Role Service**),

And may have:

- Contextual metadata (***or equivalent***, see **7. Model Metadata Service**)

*Note that many of the system metadata elements listed above refer to codes, intervals, offsets or durations. These may be unfamiliar terms. The purpose of these disposal controls is explained in detail in **14.4 System Metadata Element Definitions**, and in requirements **R8.4.2** through **R8.4.7**, below.*

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, a MoReq2010® access control list may not be present during system operation and may only be added to a disposal schedule at export.*

*Depending on the approach taken by the MCRS in implementing **7. Model Metadata Service**, the mechanism by which contextual metadata are added to disposal schedules may vary.*

*Function reference:* ***F14.5.71***

#### R8.4.2

The MCRS must allow the Disposal Action Code, under **R8.4.1**, to be set to one of the following values:

- RETAIN PERMANENTLY,
- REVIEW,
- TRANSFER, or
- DESTROY.

Whenever the Disposal Action Code is set to RETAIN PERMANENTLY, the MCRS must ensure that none of the following metadata elements are included in the disposal schedule:

- Retention Trigger Code,
- Retention Trigger Element Identifier,
- Retention Period Interval Code,
- Retention Period Duration Number,
- Retention Period Offset Code,
- Retention Period Offset Month Code,
- Confirmation Period Interval Code, or
- Confirmation Period Duration Number.

Whenever the Disposal Action Code is not set to RETAIN PERMANENTLY, the MCRS must ensure that at least the following system metadata elements are also included in the disposal schedule:

- Retention Trigger Code,
- Retention Period Interval Code,
- Confirmation Period Interval Code, and
- Confirmation Period Duration Number.

*A Disposal Action Code of RETAIN PERMANENTLY is used to ensure that records are never destroyed. The metadata elements listed are not applicable to records that are to be retained permanently.*

*Another additional Disposal Action Code, RETAIN ON HOLD, is automatically applied to records by the MCRS, under* ***R8.4.21****, but may not be used in disposal schedules.*

#### R8.4.3

Where it is included in a disposal schedule, under **R8.4.2**, the MCRS must allow the Retention Trigger Code, under **R8.4.1**, to be set to one of the following values:

- FROM NOW
- FROM DATE OF LAST REVIEW
- FROM RECORD ORIGINATED DATE
- FROM AGGREGATION ORIGINATED DATE
- FROM DATE ADDED TO AGGREGATION
- FROM DATE OF LAST ADDITION TO AGGREGATION
- FROM AGGREGATION CLOSED DATE
- FROM RECORD METADATA DATE

- FROM AGGREGATION METADATA DATE

Whenever the Retention Trigger Code is set to FROM RECORD METADATA DATE or FROM AGGREGATION METADATA DATE, the MCRS must ensure that a Retention Trigger Element Identifier is always included in the metadata of the disposal schedule. The Retention Trigger Element Identifier must not be included in the metadata of the disposal schedule for any other value of the Retention Trigger Code.

*The Retention Trigger Element Identifier indicates which metadata element associated with the record or with the record's parent aggregation is to be used to obtain its Retention Start Date.*

*A Retention Trigger Code value of FROM NOW is used as a means of initiating the retention period for a record immediately without waiting for a specific trigger.*

*Other retention triggers relate to specific metadata elements, specified under either* ***R6.5.10*** *for records or* ***R6.5.1*** *for aggregations, as follows:*

- *FROM DATE OF LAST REVIEW uses a record's Last Reviewed Timestamp,*
- *FROM RECORD ORIGINATED DATE uses a record's Originated Date/Time,*
- *FROM AGGREGATION ORIGINATED DATE uses the Originated Date/Time from a record's parent aggregation,*
- *FROM DATE ADDED TO AGGREGATION uses a record's Aggregated Timestamp,*
- *FROM DATE OF LAST ADDITION TO AGGREGATION uses the Last Addition Timestamp from a record's parent aggregation, and*
- *FROM AGGREGATION CLOSED DATE uses the Closed Timestamp from a record's parent aggregation.*

*The retention triggers, FROM RECORD METADATA DATE and FROM AGGREGATION METADATA DATE use a contextual metadata element associated with either the record or its aggregation. Depending on the approach taken by the MCRS in implementing* ***7. Model Metadata Service****, an equivalent method for identifying the appropriate system or contextual metadata element to use as the retention trigger may be substituted.*

*However, regardless of whether the model metadata service or an equivalent is used, the MCRS must ensure that the metadata element indicated always:*

- *Has a Datatype (see* ***R7.5.3****) of date, date/time or timestamp;*
- *Has a Max Occurs set to one (see* ***R7.5.4****), to ensure the metadata element has only a single unambiguous value;*
- *Is associated with the record or its aggregation (for example, by inclusion in a template, or equivalent under* ***R7.5.17*** *and* ***R7.5.18****); and*
- *May be meaningfully converted and exported as part of a MoReq2010® compliant disposal schedule regardless of how it is implemented in the MCRS (see* ***7.2 Complying with the Model Metadata Service****).*

#### R8.4.4

The MCRS must not allow a disposal schedule with a Retention Trigger Code of FROM DATE OF LAST REVIEW, under **R8.4.3**, to be associated with a class as the default disposal schedule of the class. Likewise, the MCRS must not allow such a disposal schedule to be applied directly to any record that has not previously been reviewed, except during the confirmation of a review decision, under **R8.4.16**.

*A default disposal schedule is associated with each class, under* ***R5.4.2*** *and* ***R5.4.4****. This default disposal schedule is inherited and adopted by all records under* ***R6.5.14****, unless it is overridden under* ***R6.5.15****.*

*Retention schedules with a Retention Trigger Code of FROM DATE OF LAST REVIEW are only applicable when used in conjunction with a current or previous review of the record. The MCRS must prevent them being associated with records that have never been reviewed. For this reason, they may not be associated with a class because they cannot become the default disposal schedule for newly created records.*

#### R8.4.5

Where it is included in a disposal schedule, under **R8.4.2**, the MCRS must allow the Retention Period Interval Code, under **R8.4.1**, to be set to one of the following values:

- NO RETENTION PERIOD,
- DAYS,
- WEEKS,
- MONTHS, or
- YEARS.

Whenever the Retention Period Interval Code is set to NO RETENTION PERIOD, the MCRS must ensure that none of the following metadata elements are included in the disposal schedule:

- Retention Period Duration Number,
- Retention Period Offset Code, and
- Retention Period Offset Month Code.

Whenever the Retention Period Interval Code is not set to NO RETENTION PERIOD, the MCRS must ensure that at least the following system metadata elements are also included in the disposal schedule:

- Retention Period Duration Number, and
- Retention Period Offset Code.

*If the Retention Period Interval Code is set to NO RETENTION PERIOD then a record's Disposal Action Due Date will be set to the same date as the Retention Start Date. In other words, the retention period is zero days in length and the disposal action immediately falls due as soon as the retention period is triggered.*

*Where the Retention Period Interval Duration is included in the disposal schedule it must be a whole number greater than zero. It is then interpreted in combination with the Retention Period Interval Code as either a number of days, weeks, months or years. Consequently the shortest retention period after NO RETENTION PERIOD is one day.*

#### R8.4.6

Where it is included in a disposal schedule, under **R8.4.2**, the MCRS must allow the Retention Period Offset Code, under **R8.4.1**, to be set to one of the following values:

- NO OFFSET
- START OF NEXT MONTH
- START OF NEXT QUARTER

- START OF SPECIFIED MONTH

Where the Retention Period Offset Code is set to NO OFFSET or START OF NEXT MONTH the MCRS must ensure that a Retention Period Offset Month Code is not included in the disposal schedule. Otherwise if the Retention Period Offset Code is set to START OF NEXT QUARTER or START OF SPECIFIED MONTH, the Retention Period Offset Month Code must be included and must be given a value corresponding to a month of the year.

*An offset period of NO OFFSET means that the retention period will end exactly on the calculated Disposal Action Due Date. An offset period of START OF NEXT MONTH means that the MCRS should extend the retention period to the end of the month in which it falls.*

*An offset period of START OF NEXT QUARTER will extend the retention period to the end of the quarter of the year in which it falls. For this option the Retention Period Offset Month Code is interpreted as referring to the first month of the first quarter of the year. For example, if the month were set to APRIL then the quarters would run April to June, July to September, October to December and January to March.*

*An offset period of START OF SPECIFIED MONTH will extend the retention period to the end of the month before the start of the retention period offset month. For example, to extend the retention period to the end of the calendar year in which it falls, the authorised user would set the Retention Period Offset Month Code to JANUARY.*

#### R8.4.7

Where it is included in a disposal schedule, under **R8.4.2**, the MCRS must allow the Confirmation Period Interval Code, under **R8.4.1**, to be set to one of the following values:

- DAYS, or
- WEEKS.

The Confirmation Period Duration Number must also be set to a whole number greater than zero.

*The confirmation period refers to the maximum time allocated to complete the disposal action nominated under **R8.4.2**. The shortest possible confirmation period for a disposal schedule is one day.*

*If the confirmation period is exceeded then the MCRS will raise an alert under **R8.4.14**.*

*Note that records with a Disposal Action Code of TRANSFER, under **R8.4.2**, have two confirmation periods. The first confirmation period covers the transfer of the records to another records system or location. The second confirmation period covers the destruction of the records. For simplicity, these two confirmation periods are calculated using the same values for the Confirmation Period Interval Code and the Confirmation Period Duration Number. In other words, both confirmation periods will always be the same length.*

#### R8.4.8

The MCRS must allow an authorised user to modify the following metadata for any active disposal schedule:

- Title,
- Description,
- Mandate

- Scope Notes, and
- Any contextual metadata elements.

*The mandate describes the authority for the disposal schedule. Different retention periods and disposal rules for records belonging to particular business classifications may be mandated by government legislation, industry regulations, national or international standards, etc.*

*Function reference:* ***F14.5.81***

#### R8.4.9

In addition to R8.4.8, the MCRS must allow an authorised user to modify the following metadata for any disposal schedule that has never been applied to a record:

- Disposal Action Code,
- Retention Trigger Code,
- Retention Trigger Element Identifier (***or equivalent***),
- Retention Period Interval Code,
- Retention Period Duration Number,
- Retention Period Offset Code,
- Retention Period Offset Month Code,
- Confirmation Period Interval Code, and
- Confirmation Period Duration Number.

*Once a disposal schedule has been used, by applying it to a record, the metadata values used to calculate the retention period and disposal process become fixed and can no longer be changed. Note that these are referred to as "disposal controls" in the rationale to* ***R8.4.13****.*

*Function reference:* ***F14.5.81***

#### R8.4.10

The MCRS must allow an authorised user to delete a disposal schedule that has never been applied to a record, provided it is not associated with any active class as its default disposal schedule.

*A disposal schedule can no longer be deleted once it has been used, however, it can be destroyed under* ***R8.4.11****. Before deleting a disposal schedule it must be replaced as the default disposal schedule for any active classes it is associated with, see* ***R5.4.4****.*

*Function reference:* ***F14.5.72***

#### R8.4.11

The MCRS must allow an authorised user to destroy any active disposal schedule, provided it is not applied to any active records and it is not the default disposal schedule of any active class.

*When a disposal schedule is destroyed, the MCRS retains a residual disposal schedule. A disposal schedule cannot be destroyed while it is applied as the disposal schedule for any active records, but it can be destroyed if the only records it is associated with are residual records. Before destroying a disposal schedule it must be replaced as the default disposal schedule for any active classes it is associated with, see* ***R5.4.4****, this is because only active disposal schedules may be associated with active classes.*

*Function reference:* ***F14.5.75***

**R8.4.12**

Subject to **R2.4.22**, the MCRS must allow an authorised user to browse across the disposal schedules in the disposal scheduling service and inspect their metadata.

*The terms "browse" and "inspect" are defined in* ***13. Glossary of Terms****.*

*Function reference:* ***F14.5.77***

**R8.4.13**

The MCRS must allow an authorised user to replace an active disposal schedule with another active disposal schedule, for all active records to which it is applied, and for all active classes where it is associated as the default disposal schedule.

*This function allows the authorised user to replace one active disposal schedule with another, throughout the MCRS.*

*Note that under* ***R8.4.9****, the controls in a disposal schedule, such as the determinants for the length of the retention period, cannot be modified once they have been applied to records. To change these disposal controls an authorised user must create a new disposal schedule defining different disposal controls and replace the previous disposal schedule where it occurs.*

*Function references:* ***F14.5.34****,* ***F14.5.138***

**R8.4.14**

The MCRS must update the disposal status of any record when requested by an authorised user and, either immediately or periodically, and at least daily, the MCRS must update the disposal status of all active records.

*The disposal status of all records must be updated in real time or as a scheduled activity at least once per day. The MCRS must also be able to update the disposal status of individual records on request.*

*Updating the disposal status of an active record includes:*

- *Checking whether the disposal schedule for the record has changed, see* ***R6.5.14*** *and* ***R6.5.15****, and, if so, clearing the previous disposal metadata;*
- *Checking whether the value of the retention trigger metadata element defined by the disposal schedule has changed, and if the retention trigger conditions have been met, see* ***R8.4.3****;*
- *Calculating and setting the Retention Start Date, Disposal Action Code, Disposal Action Due Date, and Disposal Confirmation Due Date for a record based on the disposal controls in its disposal schedule;*
- *Checking if there is an disposal hold in place for the record and, where necessary, generating events and applying the Disposal Action Code RETAIN ON HOLD to the record, see* ***R8.4.21****;*
- *Checking whether the disposal action has been cancelled, see* ***R8.4.18****, or confirmed, see* ***R8.4.17****,* ***R8.4.19*** *and* ***R8.4.20****;*
- *Raising a disposal overdue alert where the disposal action has not been confirmed by the Disposal Confirmation Due Date, see* ***R8.4.15****;*
- *Implementing the results of a review, including applying a new disposal schedule to the record, see* ***R8.4.17****;*

- *Changing the Disposal Action Code to DESTROY once a transfer has been confirmed, see **R8.4.19**;*
- *Destroying the content of records for components where the MCRS is able to do so, see **R8.4.20**;*
- *Destroying records and their components by pruning their metadata and event histories to leave a residual entity, see **R8.4.20**; and*
- *Destroying closed parent aggregations once their last child has been destroyed, see **R8.4.22** and **R8.4.23**.*

*See the disposal process flowchart, **Figure 8h** in section **8.2.11 The disposal process**, for further explanation.*

*Note that while MoReq2010® does specify what an MCRS must do to be compliant with the specification, it does not specify how the MCRS should implement this functionality or whether it should be done in real time or as a scheduled process. The level of specificity given here is necessary to support interoperability.*

*Function reference: **F14.5.140***

#### R8.4.15

The MCRS must alert all users authorised to receive alerts for an aggregation or record whenever the Disposal Confirmation Due Date has elapsed without the disposal action being carried out and confirmed.

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, the mechanism by which users are authorised to receive alerts will vary. When the model role service is implemented, receiving alerts can be allocated to a role, like any other function. Users are therefore authorised to receive alerts when they are allocated to that role via an access control list.*

*For example, a particular organisation defines a new role called "local records officer" (LRO). The function, receive alerts is included in the LRO role. A user is then granted an LRO role for a particular aggregation. This user will now be sent alerts by the MCRS for any records that belong to that aggregation whose confirmation is overdue.*

*MoReq2010® does not specify what alert mechanism must be implemented by an MCRS. Many different technologies are available and may be supported. As a rule of thumb an alert mechanism must be proactive and be traceable to show that alerts have been sent and have been received.*

*Acceptable mechanisms for raising alerts may include:*

- *By email;*
- *By push notification, for example SMS messaging;*
- *Published as an RSS/Atom feed;*
- *Via a subscription service, for example "twitter.com";*

*Mechanisms that are not acceptable without additional supporting infrastructure include:*

- *Writing alerts to an error log (this is not proactive in distributing the alert to authorised users); and*
- *By instant messaging (instant messaging does not leave an easily traceable conversation to show that the message was sent and received).*

*When an alert is sent, the MCRS must add a Disposal Overdue Alert Timestamp to the record. This ensures, among other things that alerts do not continue to be sent for the same records.*

*Depending on how the MCRS implements the disposal process, under **R8.4.14**, the MCRS may consolidate the alerts for multiple overdue records into a single alert to each authorised user if they are due to be raised at the same time. MoReq2010® does not specify how different alerts should be consolidated.*

*Function reference: **F14.5.125***

### R8.4.16

The MCRS must allow an authorised user to browse and inspect all active records that are due for disposal and to order and group them variously by their:

- Class;
- Aggregation, including both parent and higher level aggregations;
- Disposal schedule;
- Disposal Action Code;
- Disposal Action Due Date; and
- Disposal Confirmation Due Date.

*The MCRS must allow an authorised user to perform this function without having to construct individual search queries under **12. Searching and Reporting Service**, and in such a way that it facilitates the cancellation or confirmation of disposal actions simultaneously for groups of records, under **R8.4.17**, **R8.4.18**, **R8.4.19** and **R8.4.19**.*

*Note that the MCRS must not allow any user to browse or inspect entities through this requirement that the user is not normally able to inspect, or to find records due for destruction but subject to a disposal hold, see **R8.4.21**.*

*Function references: **F14.5.131**, **F14.5.178***

### R8.4.17

The MCRS must allow an authorised user to complete a review for any record or records with a Disposal Action Code of REVIEW that have reached their disposal due date, by applying a new disposal schedule and providing a review comment describing the outcome of the review, to either:

- A single record due for review, individually;
- Any nominated set of records that are all due for review;
- All records due for review under a nominated disposal schedule;
- All records due for review within a nominated aggregation, including both parent and higher level aggregations; or
- All records due for review within a nominated class.

*The review comment must be stored as the Event Comment in the corresponding event generated for the review. The MCRS must also apply a Last Review Comment and a Last Reviewed Timestamp to each record, see **R6.5.10**.*

*The ability to review records within nominated aggregations and classes is to ensure that records may be reviewed in context and the authorised user is able to unambiguously apply the same outcome simultaneously to the whole of the aggregation or class.*

*Function reference:* ***F14.5.118***

**R8.4.18**

The MCRS must allow an authorised user to cancel the transfer of any record or records with a Disposal Action Code of TRANSFER that have reached their Disposal Action Due Date, or to cancel the destruction of any record or records with a Disposal Action Code of DESTROY that require confirmation, for either:

- A single record due for transfer or destruction, individually;
- Any nominated group of records that are all due for transfer;
- Any nominated group of records that are all due for destruction;
- All records due for transfer under a nominated disposal schedule;
- All records due for destruction under a nominated disposal schedule;
- All records due for transfer within a nominated aggregation, including both parent and higher level aggregations;
- All records due for destruction within a nominated aggregation, including both parent and higher level aggregations;
- All records due for transfer within a nominated class; or
- All records due for destruction within a nominated class.

The MCRS must allow transfers or destruction requiring confirmation to be cancelled at any time after the Disposal Action Due Date and prior to the confirmation of these actions under **R8.4.19** and **R8.4.20**.

*To cancel the transfer or destruction the authorised user must apply a new disposal schedule and provide a mandatory cancellation comment, similar to a review in* ***R8.4.17****. The MCRS must not proceed with the previous disposal action and the new disposal schedule must take effect immediately.*

*Function references:* ***F14.5.116***, ***F14.5.117***

**R8.4.19**

The MCRS must allow an authorised user to confirm that a transfer has been completed for any record or records with a disposal action of TRANSFER that have reached their disposal due date, for either:

- A single record due for transfer, individually;
- Any nominated group of records that are all due for transfer;
- All records due for transfer under a nominated disposal schedule;
- All records due for transfer within a nominated aggregation, including both parent and higher level aggregations; or
- All records due for transfer within a nominated class.

*In response to confirmation that a transfer has been completed, the MCRS must set the Transferred Timestamp for a record, change its Disposal Action Code to DESTROY, set the record's Disposal Action Due Date to the date of confirmation, clear the record's Disposal Overdue Alert Timestamp (if set), and calculate a new Disposal Confirmation Due Date. Note that the disposal schedule for the record is not changed.*

*The ability to confirm the transfer of records by their nominated aggregations and classes is to ensure that the confirmation can be made more easily and in context where whole aggregations and classes have been exported to effect the transfer.*

*Function reference:* ***F14.5.120***

**R8.4.20**

Subject to **R8.4.21**, whenever an active record has a disposal action of DESTROY and reaches its due date, under **R8.4.14**, the MCRS must check whether all of its components are to be automatically deleted. If one or more components are not marked to be automatically deleted then the MCRS must set a confirmation period and allow an authorised user to confirm that destruction has been completed. The MCRS must allow the authorised user to confirm the deletion of components for either:

- A single record due for destruction, individually;
- Any nominated group of records that are all due for destruction;
- All records due for destruction under a nominated disposal schedule;
- All records due for destruction within a nominated aggregation, including both parent and higher level aggregations; or
- All records due for destruction within a nominated class.

If the MCRS is able to automatically delete all of the content of the components of a record, or upon confirmation of the deletion of all of the components of the record, the MCRS must ensure that it destroys the active record and its components, in its entirety, leaving a residual record with residual components.

*Destroying the record includes pruning metadata elements and their values from the metadata of the record and its components, pruning events from the event history of the record and its components, and deleting the content of the record's components where the MCRS is responsible for doing this.*

*The ability to confirm the destruction of records within their nominated aggregations and classes is to ensure that destruction can be carried out in context.*

*Function references:* ***F14.5.41***, ***F14.5.119***, ***F14.5.124***

**R8.4.21**

As part of the disposal process, under **R8.4.14**, the MCRS must determine if a disposal hold applies to any active record with a Disposal Action Code of DESTROY, that has not yet reached its Disposal Action Due Date. Where this occurs, the MCRS must change the record's Disposal Action Code to RETAIN ON HOLD and delete its Disposal Action Due Date and Disposal Confirmation Due Date, until the record is no longer subject to any disposal holds, or its disposal schedule is replaced.

*In accordance with* ***R9.4.4****, the MCRS must never allow the destruction of any record, under* ***R8.4.20****, that is subject to an active disposal hold. Furthermore the MCRS must never indicate that any record is due for destruction if it is subject to an active disposal hold, for example, by including it in the results of* ***R8.4.16****.*

*Where the disposal action has already become due before the disposal hold was applied and is awaiting confirmation, the MCRS must allow it to be confirmed. Note that disposal holds prevent the destruction of a record, including its destruction following a successful transfer, but do not prevent authorised users from reviewing the record or completing the transfer of the record to another records system prior to destroying it.*

*Whenever the MCRS changes a record's Disposal Action Code to RETAIN ON HOLD, or when all disposal holds are lifted from the record, as specified by this requirement, the MCRS must generate appropriate events, see* ***F14.5.128 Record – Held*** *and* ***F14.5.139 Record - Released****.*

*For more information on disposal holds see the key concepts in* ***9. Disposal Holding Service****.*

*Function references:* ***F14.5.128****,* ***F14.5.139***

#### R8.4.22

Following the destruction of records, under **R8.4.20**, the MCRS must automatically destroy any aggregations that no longer contain any active records, provided they are closed.

*Open aggregations are not destroyed until they are closed, under* ***R6.5.6****. Aggregations may be closed as part of confirming the destruction of the records they contain, under* ***R8.4.23****.*

*Note that the destruction of aggregations under this requirement cascades upwards so that any parent aggregation containing child aggregations will be destroyed when its last remaining active child aggregation is destroyed, provided the parent aggregation is not open.*

*Function reference:* ***F14.5.9***

#### R8.4.23

Whenever an authorised user confirms the destruction of records due for destruction within a nominated aggregation or aggregations, under **R8.4.20**, then the MCRS must allow the user to close the aggregation, including all of its descendant aggregations, as part of the same operation.

*Closing an aggregation ensures that the aggregation itself will be destroyed, under* ***R8.4.22****, if it no longer contains any active records. Under this requirement, a user authorised to close an aggregation, see* ***R6.5.6****, may choose to close it while confirming the destruction of records in that aggregation so ensuring the simultaneous destruction of the aggregation under* ***R8.4.22****.*

*Function reference:* ***F14.5.119***

#### R8.4.24

The MCRS must not allow any change to the disposal schedule applied to a residual record.

*Disposal schedules may be applied individually to active records or are inherited from their classification. However, once a record has been destroyed the MCRS must then preserve its relationship to the specific disposal schedule under which it was destroyed, and no longer allow that disposal schedule to be replaced by another.*

*It is essential for maintaining trust in the integrity of the MCRS that an auditor at a later date can with confidence confirm when a record was destroyed, the disposal rules and controls that were applied at the time, and all other relevant contextual metadata surrounding its disposal.*

*Function reference:* ***F14.5.124***

# 9. Disposal Holding Service

## 9.1 Service Information

| | |
|---|---|
| **Service Name** | **Disposal Holding Service** |
| **Service Version** | **1.0** |
| **Implements Service Identifier *(see M14.4.42)*** | 2e4a8618-c4b3-470f-8ccb-03e2d5e07026 |

## 9.2 Key Concepts

### 9.2.1 Imposing disposal holds

Disposal holds are an important and necessary part of modern records management. A disposal hold is a legal or other administrative order that interrupts the normal disposal process and prevents the destruction of some of an organisation's records while the disposal hold is in place.

Under MoReq2010® a disposal hold is created within an MCRS, as part of a disposal holding service, and the active disposal hold is then associated with entities such as records, aggregations and classes.

Where the disposal hold is associated with a record individually it prevents the destruction of that record while the disposal hold remains active. Once the disposal hold is destroyed the record's disposal process continues.

Where the disposal hold is associated with an aggregation as a whole then it prevents any record in the aggregation, or that is a descendant of the aggregation, from being destroyed. This applies equally to new records that are added to the aggregation after the hold has already been associated with the aggregation. However, records that are moved out of the held aggregation will not remain subject to the disposal hold, unless it is also applied to them individually.

Where the disposal hold is associated with a class then it prevents any record classified by that class from being destroyed. The disposal hold does not prevent a held record from being reclassified to a different class that is not subject to the disposal hold.

### 9.2.2 Impact of disposal holds

The impact of a disposal hold on the disposal scheduling service is described in detail in **8. Disposal Scheduling Service**. A disposal hold will prevent the destruction of a record by halting the disposal process at the point immediately before it is destroyed. Disposal holds do not prevent the taking of review decisions, changing a record's disposal schedule, or transferring a record up to the point when the record is destroyed from the MCRS.

A record or aggregation may be subject to more than one disposal hold simultaneously. All associated disposal holds must be lifted before the record or aggregation can be destroyed.

### 9.2.3 Lifting disposal holds

A disposal hold will remain in place, blocking the destruction of any records and aggregations it is associated with, until it is destroyed. This is sometimes described as "lifting" the disposal hold.

An authorised user can also disassociate held records, aggregations and classes from a disposal hold. However, this operation is intended only to correct associations between these entities and the disposal hold that have been applied by mistake, or where the parties to a dispute agree to narrow the scope of the disposal hold. The usual process for lifting the disposal hold from the entities that it affects is by destroying the disposal hold. Where this occurs, it is not necessary to individually disassociate entities from the disposal hold when it is lifted, indeed the continued association will provide useful historical information.

Once lifted, the disposal hold is destroyed and becomes a residual disposal hold. It is not possible to reverse this process. If the disposal hold is lifted in error then a new disposal hold must be created and the new disposal hold must then be associated with the same entities that were previously associated with the residual disposal hold. Note that until this occurs any records previously held by the old disposal schedule will not be prevented from being destroyed through the disposal process.

### 9.2.4 Disposal holding service

Disposal holds are usually institution-wide in their scope and often carry with them legal and financial penalties and liability. They may be relevant to information stored in several different business systems within an organisation.

The service based architecture of MoReq2010® makes possible the implementation of a centralised disposal holding service that is shared by several records systems in an organisation, allowing disposal holds to be created, managed and lifted centrally, while being simultaneously applied across multiple MoReq2010® compliant systems.

## 9.4 Functional Requirements

#### R9.4.1

The MCRS must allow an authorised user to create active disposal holds (**E14.2.5**) with the following system metadata:

- System Identifier (**M14.4.100**),
- Created Timestamp (**M14.4.9**),
- Originated Date/Time (**M14.4.61**),
- First Used Timestamp (**M14.4.32**),
- Held Record Identifier (**M14.4.39**),
- Held Aggregation Identifier (**M14.4.37**),
- Held Class Identifier (**M14.4.38**),
- Title (**M14.4.104**),
- Description (**M14.4.16**),
- Mandate (**M14.4.51**),
- Scope Notes (**M14.4.97**), and
- Destroyed Timestamp (**M14.4.17**).

Each disposal hold also has:

- Event history (see **2. System Services**),
- Access control list (***or equivalent***, see **4. Model Role Service**),

And may have:

- Contextual metadata (***or equivalent***, see **7. Model Metadata Service**)

*The Originated Date/Time may be used to record the date on which the legal or administrative order was issued, rather than the timestamp that marks when the disposal hold was created in the MCRS, see **R2.4.26**.*

*Depending on the approach taken by the MCRS in implementing **4. Model Role Service**, a MoReq2010® access control list may not be present during system operation and may only be added to a disposal hold at export.*

*Depending on the approach taken by the MCRS in implementing **7. Model Metadata Service**, the mechanism by which contextual metadata is added to disposal holds may vary.*

*Function reference: **F14.5.57***

#### R9.4.2

The MCRS must allow an authorised user to change the metadata of an active disposal hold, including its Title, Description, Mandate, Scope Notes and any contextual metadata.

*Disposal holds usually represent a legal or an administrative order. The Mandate describes the authority and jurisdiction under which the disposal hold operates. The Scope Notes provide additional information to users on how the disposal hold should be interpreted and applied.*

*Function reference: **F14.5.67***

#### R9.4.3

The MCRS must allow an authorised user to associate and disassociate an active disposal hold with active records, aggregations and classes.

*For example, it should be possible to search for records, aggregations and classes that fall under the scope of the legal or administrative order represented by the disposal hold, and associate the disposal hold with the entities in the search results.*

*Disassociating a disposal hold from a record, aggregation or class is intended only for those occasions when records, aggregations and classes have been added to the disposal hold by mistake. Records, aggregations and classes that have been properly included in the disposal hold will be released for destruction when the disposal hold is lifted under **R9.4.6**, and should not be individually disassociated from the disposal hold.*

*Function references: **F14.5.56**, **F14.5.69***

#### R9.4.4

The MCRS must prevent the destruction of any record, under **R8.4.21**, that:

- Is directly associated with the disposal hold;
- Is a child or descendent of any aggregation that is associated with the disposal hold; or

- Has been classified with a class that is associated with the disposal hold.

*See **8. Disposal Scheduling Service** for further details of how disposal holds interrupt the disposal process and prevent the destruction of aggregations and records.*

#### R9.4.5

The MCRS must allow an authorised user to delete a disposal hold provided it has never been associated with any records, aggregations or classes.

*Once a disposal hold has been used it becomes important to the history of those records, aggregations and classes that is has been associated with and can no longer be deleted from the MCRS. However, the disposal hold can be destroyed under **R9.4.6**.*

*Function reference: **F14.5.58***

#### R9.4.6

The MCRS must allow an authorised user to lift a disposal hold by destroying it, thereby allowing the destruction of any records directly or indirectly subject to the disposal hold.

*When a disposal hold is lifted then any records subject to the disposal hold will resume the disposal process described in **8. Disposal Scheduling Service** and may be destroyed, provided they are not subject to any other disposal holds.*

*Note that, under **R9.4.4**, records can be held under a disposal hold in a number of ways. The disposal hold may be directly associated with them, or they may be part of an aggregation that the disposal hold is associated with, or the disposal hold may be associated with the record's class. As a consequence, it is possible that records that are disassociated from a disposal hold may still be subject to the disposal hold by another means, for example, because their parent aggregation or their class remains associated with the disposal hold.*

*When records are no longer associated with a disposal hold, then they resume the disposal process described in **8. Disposal Scheduling Service** and may be destroyed, provided they are not subject to any other disposal holds.*

*Function reference: **F14.5.61***

#### R9.4.7

Subject to **R2.4.22**, the MCRS must allow an authorised user to browse the disposal holds in the disposal holding service, and associated entities in other services, and inspect their metadata in the following ways:

- Browse across all the disposal holds in the disposal holding service and inspect their metadata; and
- Browse from a disposal hold to any associated records, aggregations or classes and inspect their metadata.

*The terms "browse" and "inspect" are defined in **13. Glossary of Terms**.*

*Function references: **F14.5.12**, **F14.5.30**, **F14.5.63**, **F14.5.131***

# 10. Searching and Reporting Service

## 10.1 Service Information

| | |
|---|---|
| **Service Name** | **Searching and Reporting Service** |
| **Service Version** | **1.0** |
| **Implements Service Identifier *(see M14.4.42)*** | f09984a5-dd31-44d8-9607-22521667c78a |

## 10.2 Key Concepts

### 10.2.1 Discovery

There are two methods by which users can discover entities in an MCRS: a user may browse from one entity to its related entities (for example, from parent entities to their children, from aggregations to their classes, from users to their groups, from records to their components, and so on), or alternatively a user may search for entities that match a particular search query.

Experience shows that searching is a far more scalable option for discovery in records systems with large numbers of entities. In some records systems, it is also possible to find entities by searching that a user may not be able to access by browsing because of access control settings. This occurs, for example, when a user is able to inspect a child entity but not its parent. In this case the inaccessible parent may block the ability of the user to discover the child entity by browsing.

Often users will combine both methods by first searching for those entities that meet their general search criteria and then refining this further by browsing from the search results, once the number of entities in the results is reduced to a manageable number.

MoReq2010® requires that all records systems must have a search engine for finding entities based on the values in their metadata elements. The core requirements do not specify that an MCRS must provide users with the ability to search within the content of records, however many suppliers will provide this capability for certain types of record content.

An important non-functional requirement for searching an MCRS is consistency and completeness of results. This is particularly important in a records management environment. If the same user performs the same search multiple times then, assuming no changes to its underlying data, the MCRS should reliably provide the same set of search results back to the user in the same order.

### 10.2.2 Methods of searching

MoReq2010® does not specify how suppliers should implement searching in their MCRS solutions, however, the specification does require a minimum level of support for various methods of searching. These include:

- The MCRS must be able to search for any entity type by any of its system or contextual metadata, including events;

- The MCRS must be able to define search criteria to match the datatype of any metadata element definition, including both system and contextual metadata;
- The MCRS must be able to support full text searching using the same search term, entered once, across all textual metadata elements simultaneously;
- The MCRS must be able to search on a combination of separate search criteria for nominated metadata elements; and
- The MCRS must be able to combine the results of searches together to perform complex searches.

### 10.2.3 Searching text

MoReq2010® makes a distinction between two types of text based metadata element. "Textual" metadata elements are those which are intended to hold informative or explanatory text expressed as natural language, such as Title, Description and Comment. Under **R2.4.28**, textual metadata elements must always be accompanied by a language identifier.

Other metadata elements may be text based but are not intended to hold words or sentences in the context of a particular language. They may instead hold identifiers, or codes, etc.

MoReq2010® requires that textual metadata elements be able to be searched by full text searching. Full text searching means searching by whole word, rather than by a sequence of characters within the value of the metadata element. Using a modern Internet search engine such as Google, provides an example of full text searching.

Full text searching techniques can be very complex, for example, finding the word based on alternative spellings, different tenses, or a variety of suffixes and prefixes based on the known language of the text. MoReq2010® does not specify the level of sophistication required for full text searching, but it does require that the language of the metadata element be stored with its contents, to enable these types of techniques to be applied by a suitably equipped search engine.

The MCRS must also provide the means to search on metadata elements that are not textual. These may be text based or they may be numbers, timestamps, references to other entities, and so on.

### 10.2.4 Search results

Users search an MCRS in order to find entities that match their search query. Search results are therefore always expressed as a list of entities. MoReq2010® specifies that search results be user configurable so that the user may specify how the entities in the list of search results are ordered and which of the metadata elements belonging to the entities in the list are returned.

Although MoReq2010® does not specify any particular layout for search results they can be logically conceptualised as adhering to a tabular format where each entity occupies a row of the table and the columns represent the values of the different metadata elements belonging to each entity. This conceptual layout is shown in **Figure 10a**, but it must be reiterated that MoReq2010® does not specify that this layout be used by an MCRS for presenting search results back to the user.

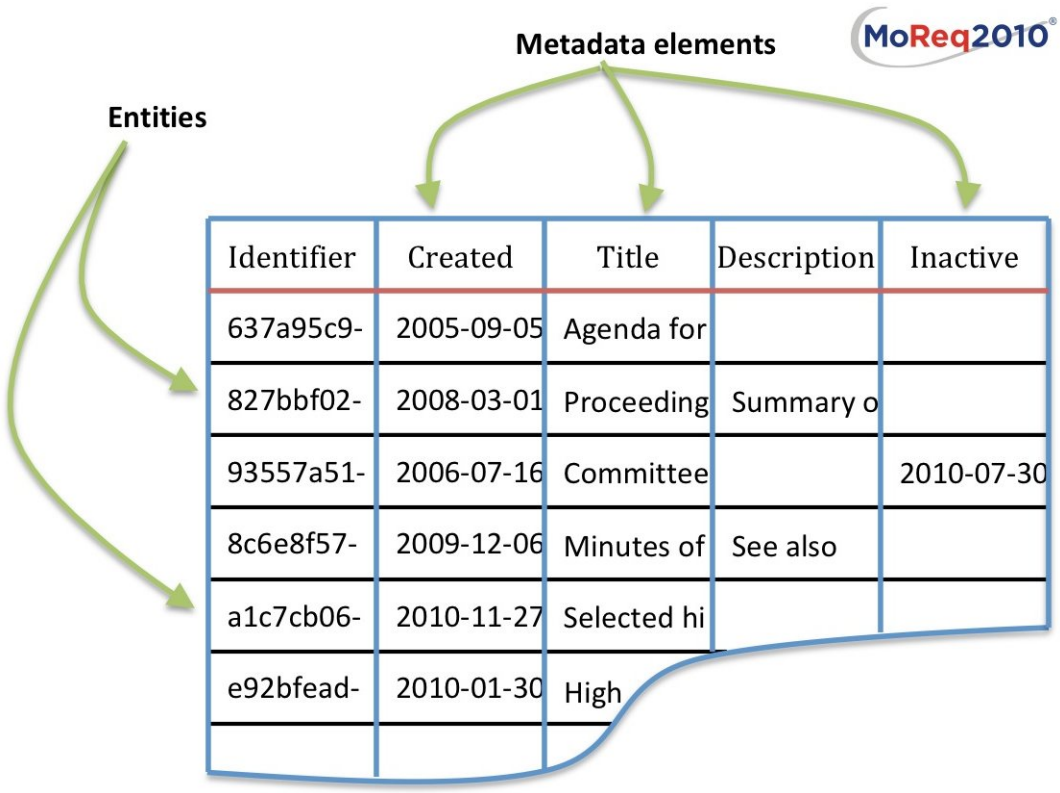


***Figure 10a** – However they are presented, a set of search results may be conceptually pictured as a list of entities and their selected metadata in a user defined order*

When searching an MCRS the total count of search results that match the original query may be a high number. Because of this, MoReq2010® specifies that an MCRS must provide pagination or other division of large sets of search results, so that the user is presented only a subset of the total results at a time, and can then request each subsequent page of results from the MCRS in turn.

### 10.2.5 Security

The results returned by searches will be user specific and related to the user's access control settings. Users who search an MCRS will only ever find entities that they have been granted access to inspect. MoReq2010® does not permit an MCRS to provide search results that include entities that a user has not been granted access to inspect.

From time to time when searching and browsing an entity that the user has full access to, on inspection, has an identifier to another entity which the user does not have the rights to access. This could be a parent, child, or any other related entity, such as a disposal hold associated with a record. Where this occurs the MCRS must prevent the user from accessing the second entity, or any of its metadata, by browsing to it, searching for it or inspecting it.

Where possible the entity should simply not appear in search results or browsing lists. Where the inaccessible entity would normally appear, say, in a column of a table of metadata such as that shown in **Figure 10a**, or in the metadata of an entity under inspection, the MCRS must anonymise its presence by leaving a blank space or replacing its title with a suitable token such as "Unknown entity".

### 10.2.6 Saved searches

Users are able to save their search queries so that they may reuse them later. This enables a user to run the same search again, or to use the previous search criteria as the starting point for building a new set of search criteria.

A saved search is not regarded by MoReq2010® as a managed entity. It is specific to a particular MCRS. There is no entity type definition or list of metadata elements associated with a saved search, and no requirement that a saved search be exported or transferred to another records system.

### 10.2.7 Reports

MoReq2010® requires that an MCRS support two different types of report: detailed reports and summary reports. Both types of report are related to searching, as described above.

Detailed reports imitate normal searches and are based around a single search query returning a subset of the metadata for each entity in the results. Detailed reports will usually adopt a tabular layout, similar to **Figure 10a**. Detailed reports differ from searches by delivering all results together as a single document in a common report format, although MoReq2010® does not specify the format the report should be in.

By comparison, summary reports are based on multiple search queries, but do not return the set of results for each search query but rather the total number of entities found that meet each search query. As with detailed reports, MoReq2010® does not specify any particular report format for summary reports.

### 10.2.8 Saved reports

Like saved searches, report definitions once constructed, can also be saved so that they may be rerun at a later date or used as the basis for constructing a subsequent report. Also, like saved searches, saved reports are not required to conform to an entity type definition or be transferrable to other records systems.

## 10.4 Functional Requirements

**R10.4.1**

The MCRS must allow users to find, using a search query, any entities that they have been granted authorisation to browse or inspect.

*To find entities by searching a user constructs a search query. The MCRS then finds the entities that match that query, provided they are also entities that the user is authorised to inspect (see **4. Model Role Service**). The list of all entities that match the search query comprises the set of search results.*

*MoReq2010® does not specify any particular format for search queries, they may be scripted, such as SQL statements, or visual, such as query by form, etc.*

*Function reference: **F14.5.195***

**R10.4.2**

The MCRS must allow users to restrict the results of searching, under **R10.4.1** to entities of a particular entity type or types.

*For example, a user might search only among events, or only for users, groups and roles that match the search query.*

**R10.4.3**

The MCRS must allow a user to specify a search query, under **R10.4.1**, that comprises a single full text search carried out across all textual metadata elements.

*Full text searching involves searching for one or more whole words or phrases. All textual metadata elements must be included in a full text search.*

**R10.4.4**

When performing full text searching, under **R10.4.3**, the MCRS must calculate a relevancy score for each entity found.

*The relevancy score should rank the search results along a continuum of best fit to worst fit against the search query. MoReq2010® does not specify the algorithm to be used by the MCRS' search engine for calculating the relevancy score.*

**R10.4.5**

The MCRS must allow a user to specify a search query, under **R10.4.1**, that consists of one or a combination of search criteria, where each search criterion compares a particular system or contextual metadata element against a value provided by the user.

*For example, a user may search for group entities, based on the value of their Title.*

**R10.4.6**

The MCRS must allow a user to specify a search criterion, under **R10.4.5**, that returns a match for any value of the specified metadata element.

*For example, a user must be able to find only closed aggregations based on a search criteria that specifies that the Closed Timestamp exists and has a value; or conversely find only open aggregations by searching for aggregations where the Closed Timestamp does not have a value.*

**R10.4.7**

The MCRS must allow a user to specify a search criterion, under **R10.4.5**, that returns a match for textual metadata based on full text searching.

*This is similar to **R10.4.3** except that this requirement narrows the full text search to a single specified textual metadata element rather than across all textual metadata elements simultaneously.*

**R10.4.8**

The MCRS must allow a user to specify a search criterion, under **R10.4.5**, that returns a match for date, date/time and timestamp metadata based on any of the following:

- Values occurring before a particular date, date/time or timestamp;
- Values occurring after a particular date, date/time or timestamp;
- Values occurring on a particular date;
- Values occurring today;
- Values occurring yesterday;
- Values occurring tomorrow;

- Values occurring this week;
- Values occurring last week;
- Values occurring next week;
- Values occurring this calendar month;
- Values occurring last calendar month;
- Values occurring next calendar month;
- Values occurring this organisational quarter;
- Values occurring last organisational quarter;
- Values occurring next organisational quarter;
- Values occurring this organisational year;
- Values occurring last organisational year;
- Values occurring next organisational year;
- Values occurring this calendar year;
- Values occurring last calendar year; or
- Values occurring next calendar year.

*See the example in the rationale to **R10.4.5**. Being able to define a date/time range in a relative sense, such as "this calendar month" is particularly important for saved searches and reporting.*

#### R10.4.9

The MCRS must allow an authorised user to set for the searching and reporting service the first day of the week and the first month of the first quarter of the organisational year.

*These values are required when searching under **R10.4.8** to determine criteria such as "last week", "this organisational quarter", and "next organisational year".*

*For many organisations the organisational year does not match the calendar year, which runs from January to December. An organisational year may, for example, run from April to March.*

#### R10.4.10

When searching using timestamp based criteria under **R10.4.8** the MCRS must be able to factor in the user's local time zone to accurately find metadata values that fall into the specified period or on the specified day.

*For example, the following may all represent exactly the same time during the Northern Hemisphere summer months:*

- *0030, Wednesday, Central European Summer Time (UTC + 0300)*
- *0130, Wednesday, Eastern European Summer Time (UTC + 0200)*
- *2330, Tuesday, Western European Summer Time (UTC + 0100)*
- *2230, Tuesday, Universal Coordinated Time (UTC)*

*Note that this requirement applies only to timestamps (which include time zone information) and does not apply to date and date/time based metadata elements.*

*For example, it would not be appropriate to factor in a relative time zone difference when searching for a personnel aggregation using the employee's date of birth as a criterion.*

#### R10.4.11

The MCRS must allow a user to specify a search criterion, under **R10.4.5**, for numeric metadata that matches the user provided value based on any of the following:

- Equality;
- Greater than; or
- Less than.

*Note that by combining search criteria together, under **R10.4.15**, it is possible achieve further numeric comparisons than just those listed above. For example, "greater than OR equal to", "less than OR equal to", "greater than [min] AND less than [max]", and so on.*

**R10.4.12**

The MCRS must allow a user to specify a search criterion, under **R10.4.5**, for Boolean metadata flags that checks whether the value of the element is true (set) or false (cleared).

*Boolean metadata elements are referred to as flags in MoReq2010®.*

**R10.4.13**

The MCRS must allow a user to specify a search criterion, under **R10.4.5**, for metadata elements that contain system identifiers, based on a match with the user provided entity.

*For example, find all the entities that have been classified under a specified class.*

**R10.4.14**

The MCRS must allow a user to specify a search criterion, under **R10.4.5**, for records, aggregations and components that are descendants of any aggregation.

*Descendants do not have to be the immediate children of the aggregation.*

**R10.4.15**

The MCRS must allow users to combine different search criteria, under **R10.4.5**, using the Boolean operators AND, OR and NOT in any combination, and to change the order of precedence by which search criteria are evaluated by using parentheses or an equivalent method.

*These operators come from Boolean algebra. AND refers to the intersection of two sets, OR to the union of two sets and NOT to the complement of a set. Under the conventions of Boolean logic, NOT has the highest precedence, followed by AND, with OR having the lowest precedence. Parentheses can be used to change the precedence of operators (operations inside parentheses are always performed first).*

*Note that MoReq2010® does not specify that the MCRS must explicitly use the terms "AND", "OR" and "NOT", nor must it explicitly use parentheses to change the order of operations. Instead the MCRS may use any logically equivalent means of representing the same operators and concepts.*

**R10.4.16**

The MCRS must allow a user to combine, chain, or join, the results of several search queries so as to answer complex search enquiries.

*MoReq2010® does not specify how different sets of search results should be combined by an MCRS to provide a complex search capability. This may, for example, be by the equivalent of an SQL join between two database tables or it may be by using a different technique.*

*The following are typical examples of the types of complex search results that users will expect an MCRS to provide:*

- *Find all the open aggregations that do not contain any active records;*
- *Construct an "activity history" for an aggregation by finding and combining together all events for the aggregation and for each of its descendant entities;*
- *Find all the aggregations, or all the records, that have a particular class, regardless of whether they inherit their classification from their parent aggregation or it is directly applied to them;*
- *Find all the records that have a particular disposal schedule, regardless of whether they inherit it from their class or it is directly applied to them;*
- *Find all the records that are subject to a disposal hold, regardless of whether they have been added to the disposal hold individually or because they belong to a particular class or aggregation that has been added to the disposal hold;*
- *Monitor the activities of a particular group by finding all the events generated by users of that group over a specified time period;*
- *Find all the entities which have a particular contextual metadata element definition associated with them where the metadata element contains a specific value or range of values;*
- *Find all the entities which were created by a particular user in the last week, by tracing the creation function through the creation event; and*
- *Find all the records that were transferred by members of a particular group in the last month.*

#### R10.4.17

The MCRS must, by default, return only active entities in search results, under **R10.4.1**, unless the user performing the search specifies the inclusion of both active and residual entities in the search results.

*By default, residual entities will not be included in search results. Where both active and residual entities are specified by the user, then their status should be clearly indicated in the search results.*

#### R10.4.18

The MCRS must, allow a user who initiates a search, under **R10.4.1**, to specify:

- Which metadata elements to include in the search results;
- Whether to include the entity type of each entity in the search results;
- Whether to include the relevancy score, calculated under **R10.4.4**;
- Whether to include both active and residual entities, under **R10.4.17**;
- Whether to order the search results by the relevancy score (if included); and if not
- Which metadata element(s) to use to order the search results.

*The user may elect to include, for example, the title and description of each entity, its entity type, and its Created Timestamp in the search results, and to order the results by the Created Timestamp.*

*Note that when ordering search results using a timestamp, the MCRS should factor in the time zone when determining the order of entities, see also **R10.4.10**.*

#### R10.4.19

The MCRS must, for large sets of search results, implement a method of pagination, or other alternative, where only a subset of the total search results is provided back to the user and additional subsets are provided when required.

*MoReq2010® does not define a "large" set of search results or how the MCRS should implement pagination or any other method of providing subsets of the full search results sequentially to the user, however, pagination (or an alternative) should be demonstrable in the MCRS during testing. Page sizes in different implementations typically vary between 10 and 100 items per page.*

#### R10.4.20

The MCRS must provide the total number of entities that match the search query as part of the search results, this total must not include entities that are excluded from the search results under **R10.4.21**.

*Where large sets of search results are paginated, under **R10.4.19**, the total number of matching search results must be returned with the first and all subsequent subsets.*

*The total number of search results is a valuable indication both immediately to the user and later statistically when stored in a matching event, of search activity and usage patterns across the MCRS. Depending on the search engine used by the MCRS, this number may be an approximation.*

#### R10.4.21

The MCRS must never allow a user by searching, browsing or any other method, to access entities, or their metadata, that the user does not have authorisation to inspect. All such entities should be excluded from search results.

*See also **R10.4.1. S**earch results must not include any inaccessible entities that the user is not authorised to access. Also, inaccessible entities must be excluded from totals of search results, under **R10.4.20**.*

#### R10.4.22

Whenever a user performs a search under **R10.4.1**, the MCRS must generate an event and include it in the event history of the user's user entity. The event must include a description of the search query performed and the total number of entities found, see **R10.4.20**.

*MoReq2010® does not specify how search queries should be described in the Search Query. It may be in a structured expression language or in natural language.*

*Note that MoReq2010® does not require that the generated search event should link to each of the search results as participating entities, but the event should contain the total number of search results returned.*

*Function reference: **F14.5.195***

#### R10.4.23

The MCRS must allow authorised users to save, modify, delete and share search queries.

*Note that these are commonly referred to as "saved searches", although it is only the search query that is saved, rather than the search results.*

*MoReq2010® does not specify how search queries are to be saved or in what format. This detail is specific to the individual MCRS and its search engine. For this reason, saved searches cannot be exported from the MCRS and imported into another records system, under **11. Export Service**, and this functionality is not included in the function definitions in **14.5 Function Definitions**.*

### R10.4.24

The MCRS must allow authorised users to generate detailed reports based on any search query, in a common reporting format, with the following configurable items:

- A report header provided by the user;
- The date and time the report was generated;
- Page numbering;
- Details of the MCRS and the searching and reporting service generating the report, see **R2.4.2**;
- Details of the user generating the report, see **R3.4.1**;
- A description of the search query used for the report, see **R10.4.22**;
- The total number of search results in the report, see **R10.4.20**; and
- Columns and column headings based on the metadata elements selected for the report, see **R10.4.18**.

*Suppliers must provide a list of the reporting formats they support when their product is certified. Common reporting formats include:*

- *Comma or tab separated values;*
- *Spreadsheet formats, such as OOXML and ODF;*
- *XML and HTML based formats; and*
- *PDF or other document formats.*

*Function reference:* ***F14.5.184***

### R10.4.25

The MCRS must allow authorised users to generate summary reports based on multiple search queries, in a common reporting format, with the following configurable items:

- A report header provided by the user;
- The date and time the report was generated;
- Page numbering;
- Details of the MCRS and the searching and reporting service generating the report, see **R2.4.2**;
- Details of the user generating the report, see **R3.4.1**;
- A description of each of the search queries used by the report, see **R10.4.22**; and
- The total number of search results found for each search query, see **R10.4.20**.

*A summary report gives only the totals for the number of entities that meet each of the specified set of search queries. Only the total number of matching entities is given. Unlike a detailed report, there is no body to a summary report that lists the entities or their metadata.*

*Common reporting formats are listed in the rationale to* ***R10.4.24****.*

*Function reference:* ***F14.5.196***

### R10.4.26

Whenever a user generates a detailed report under **R10.4.24**, or a summary report under **R10.4.25**, the MCRS must generate an event and include it in the event history of the user's user entity. The event comment must include description of the search(es) performed in generating the report, and the total number of entities found.

*See also* ***R10.4.22***.

*Function references:* ***F14.5.184***, ***F14.5.196***

### R10.4.27

The MCRS must allow authorised users to save, modify, delete and share report definitions for both detailed and summary reports.

*See also* ***R10.4.23***.

# 11. Export Service

## 11.1 Service Information

| | |
|---|---|
| **Service Name** | **Export Service** |
| **Service Version** | **1.0** |
| **Implements Service Identifier *(see M14.4.42)*** | 2777ab81-057e-4aa4-9595-69459ec2dc1e |

## 11.2 Key Concepts

### 11.2.1 Purpose of export

Export in MoReq2010® is the operation by which entities in an MCRS can be described in sufficient detail in a common XML data format, belonging to the specification, so that their metadata values, event histories, access controls and content can be preserved and transferred to another MCRS.

The complementary operation to export is import. The purpose of import is to take MoReq2010® formatted XML data that has been exported from one MCRS, and use it to create new entities in a different MCRS, so that they can be accessed and managed in the same way as they were previously.

Ideally both export and import should be "lossless" operations; they should not divest the entity of any of its significance, content or context. The ability to export entities from one MCRS and usefully import them into another MCRS, without loss of business context, is referred to by MoReq2010® as achieving "interoperability".

Some of the common reasons for exporting entities from an MCRS include:

- **Transfer** – where entities are relocated to the management of a different system, organisation or archive. Transfer is mostly performed as a consequence of following a disposal schedule as part of the disposal process described in **8. Disposal Scheduling Service**.
- **Migration** – where entities are moved from the management of one MCRS to another MCRS within an organisation. This may be done as part of replacing, upgrading or decommissioning the original MCRS.
- **Secondary hosting** – where entities are regularly copied to one or more secondary and possibly read only systems. If the secondary host is regularly updated in this way, then only the differences between the entities it holds and those in the source system will need to be imported; and
- **Replication** – so as to provide a copy for reference or safekeeping of the contents of an MCRS in a non-proprietary and easily understood format that is transportable to other compliant records systems. Note that the MoReq2010® export service is neither intended, nor optimised, to provide routine operational backups as part of the general provision of disaster recovery services. However, unlike a system backup of data which is made in the supplier's own data format and can only be used for

restoring the system it was made from, the export service does allow a copy of all or part of an MCRS to be made into the widely understood MoReq2010® XML format.

### 11.2.2 Partial export

The MoReq2010® export service is intended for the export of complete entities with their metadata, event histories, access controls and content intact. When entities are exported from the MoReq2010® export service, other related entities that provide context for the entity, must also be exported with them, some as placeholders, as described below. This ensures that the full context of entities is transferred from system to system. However, it also requires the export of sufficient data to ensure this.

Some records systems may additionally allow for other variations of export where only some of the requirements of the MoReq2010® export service are supported. These variations may include:

- Only some of the metadata,
- Only some of the events in the event history,
- Only some of the access control information, and/or
- Only some of the related entities.

MoReq2010® describes these approaches as implementing "partial export" because of the incomplete integrity of the data that is exported. A partial export is, by definition, "lossy" rather than lossless.

While partial export may have some applicability in certain business situations, for example as a means of providing a temporary copy, or a summary set of records, to an external authority, it is generally unsuitable for the practice of good records management. Lossy transfers of entities may result in the unintended stripping away of important attributes and business context that are found to be required later. Internal consistency within a records system cannot be guaranteed, especially in the longer term.

For this reason, partial export is not required and not tested for compliance with the MoReq2010® specification, although a supplier may offer some form or forms of partial export as an additional product feature. Note also that because the export data format provided by MoReq2010® is XML based, it should be equally possible, for an organisation to generate any partial export from a complete export by subsequently applying XML transformation.

Implementation of the full MoReq2010® export service, as defined by these requirements, is an essential quality mandated for compliance with the core services.

### 11.2.3 Use of XML

MoReq2010® is accompanied by an XML schema, published and maintained by the DLM Forum®, that defines how data should be described as a post-condition of export and a precondition of import. Every MCRS must provide a full implementation of this schema, which is extensible to allow for the definition and capture of contextual metadata elements and variations introduced by MoReq2010® extension modules and plug-ins.

MoReq2010® currently defines the export service as writing the exported entities to an XML datafile. However, it recognises that writing exported data to XML datafiles, while practical

for small amounts of data, cannot be considered a scalable solution for medium to large MCRS environments. This is because:

- Operating systems place limits on datafile sizes and there are also practical limits imposed by storage media. For this reason, large amounts of data cannot be written to a single datafile but must be split across many datafiles. There is no industry standard for this.
- Often XML datafiles are compressed for more efficient storage, and while XML is itself standardised, different compression algorithms are not.
- XML datafiles must be stored somewhere, even temporarily, leaving them vulnerable to security threats, such as unauthorised access, accidental deletion and even tampering.

In the future, larger MCRS solutions will need to transfer thousands, tens of thousands, hundreds of thousands and even millions of records and related entities, as part of a single secure export/import operation.

### 11.2.4 XML data streaming

For the reasons given above, exporting to XML datafiles is seen as a temporary solution to the problem of exporting and importing data from an MCRS. The DLM Forum Foundation's, MoReq Governance Board intends to investigate and explore options for streaming support for exported data in a future version of MoReq2010®.

XML data streams have the following advantages over XML datafiles:

- Streams are not limited by fixed datafile size limits or the storage capacity of physical media, although streams may be captured into one or more datafiles if required;
- A stream may be interrupted at any point during transfer and resumed later, they are therefore a robust means of transfer;
- Streams may be sent across encrypted channels, allowing secure system to system communication;
- During transfer, streams do not require any intermediate physical storage, such as DVDs, which might be lost, damaged or unlawfully copied in transit; and
- Streaming allows the direct transfer of data in real time.

As a consequence, it can be harder to intercept or interfere with streamed data and this combined with the immediacy of real time communication will, in the future, contribute to enhancing the authenticity of the transferred records. One possible standard that may be introduced to the MoReq2010® export service in the future is the use of the W3C EXI (Efficient XML Interchange) standard. At the time of publication of MoReq2010®, EXI is not a "recommendation" of the World Wide Web Consortium (W3C), but it has been a candidate recommendation since December 2009.

### 11.2.5 Importing entities

MoReq2010® is intended to be suitable for use by both large scale multi-purpose records systems implementations and small scale single purpose implementations. While export is a necessary part of the core services provided by an MCRS, the complementary import is not a core requirement of the specification. This reflects the complexity of the import process. Implementing an import service that can successfully import data from every other MCRS

necessarily requires a more sophisticated solution than one which is only required to export data. The result of building more sophisticated solutions is a longer time to develop with potentially a higher cost to the end consumer, at least until such systems are commonplace, and a corresponding fall in the number of compliant records systems.

Therefore, while MoReq2010® specifies that every MCRS must support export so that no records system can become a trap for data which, once having been put into the system cannot be retrieved from it, it does not specify that every MCRS must necessarily make provision for an import service. Some dedicated front line business systems may never support import, while other MCRS suppliers will only include import in their second generation MoReq2010® compliant products. Generally, it will be the larger and more generic records systems, implemented as enterprise middleware with the capacity to integrate with and support multiple front line business systems, that will require import functionality sooner than other types of MCRS.

The import service will be provided as an extension module to MoReq2010® and consumer organisations can specify this functionality should they require it.

### 11.2.6 Exporting from non-compliant systems

As MoReq2010® compliant systems become more commonplace, it is anticipated that suppliers will want to build adaptors that enable the records and other entities in legacy information systems to be exported to the MoReq2010® XML format. These older, non-compliant systems may never be updated to new versions that can achieve MoReq2010® compliance. However, organisations will want either the original supplier, or a third party supplier, to provide a migration solution so that the contents of these records systems may be transferred to an MCRS.

Once records and entities have been successfully migrated to, or recreated in, an MCRS they may be held indefinitely, as all MCRS solutions support an export service and can therefore transfer them on to the next records system.

Other than providing an XML data format, MoReq2010® does not define any requirements for the transfer of records and entities from non-compliant records systems. Each consumer organisation must be independently satisfied of the completeness and correctness of any migration tool it licenses or develops for extracting records and entities from legacy systems.

### 11.2.7 Export context and placeholders

The entities in an MCRS are richly interrelated and MoReq2010® requires each entity to be exported in context. This means that when one entity is exported, information about the entity, and other related entities are exported with it.

For the purpose of export the full context of each entity is made up of the following:

1. System metadata elements and their values;
2. Contextual metadata elements and their values;
3. Related entities referred to by system identifiers held in these metadata elements;
4. Significant entities, such as a record's disposal schedule, whether or not they are referred to directly by a system identifier held in a metadata element;

5. Included entities, such as the components of a record, the users in a group or the records in an aggregation;
6. The entity's access control list and its access control entries;
7. The access control lists of related and significant entities, and their access control entries;
8. The users, groups and roles referred by these access control entries;
9. The events in the entity's event history; and
10. The entities referred to by system identifiers held in the metadata elements belonging to each event.

To export an entity complete with its context from the MCRS, all of the items in the list above must be exported with it. Note however that some items in the list, notably 3, 4, 6, 7 and 8 call for additional entities to be exported. If these related entities are likewise exported under the same rules as above, then this might potentially result in an ever increasing pool of entities to export.

Instead of this, any related entities that are not directly included in the set of entities to export are exported with a reduced context. Specifically these entities have only the following items from the list above:

1. System metadata elements and their values;
4. Significant entities, such as a record's disposal schedule, whether or not they are referred to directly by a system identifier held in a metadata element;
6. The entity's access control list and its access control entries;
7. The access control lists of related and significant entities, and their access control entries;
8. The users, groups and roles referred by these access control entries;

Entities exported with their full context are described as being "exported in full". Entities exported with a reduced context are described as being "exported as placeholders". Placeholders do not include contextual metadata, related entities other than significant entities or any event history.

While export placeholders are useful for providing context for other entities they are not considered to be complete entities, as some of their own context is missing. They represent a two dimensional snapshot taken at a particular moment in time rather than a "living" entity.

Entities that have been exported in full can be imported by another MCRS and managed as active entities. By comparison, when an MCRS imports a placeholder it creates from it an "inactive" entity, which is neither active nor residual. Inactive entities are only relevant in the context of the MoReq2010® **501. Import Service**, they are not part of the MoReq2010® core services, nor is it necessary to implement them to achieve compliance with the core requirements.

### 11.2.8 Exporting metadata

Each entity has associated with it a set of metadata elements, some are system metadata elements and some may be contextual metadata elements. Each metadata element has two important parts to it:

- Its related metadata element definition, and

- Its value (or values).

When exporting metadata elements, system metadata element definitions are not included in the exported data. It can be safely assumed that the importing MCRS already knows these definitions, as they are specified by MoReq2010®. System metadata elements are therefore exported as a value only.

When contextual metadata elements are exported it is necessary to export the corresponding contextual metadata element definition along with the value. This allows the importing MCRS to "recognise" the metadata element. Relevant contextual metadata element definitions must be exported as placeholders, where necessary.

When an entity is exported in full, then both its system metadata values and its contextual metadata values, and any corresponding contextual metadata element definitions, are exported with it. When an entity is exported as a placeholder then only its system metadata values are exported.

Each metadata element, including for both system and contextual metadata, is either:

- A value of a particular datatype, such as a text value, a number or a flag; or
- A system identifier that refers to another entity in the MCRS.

If a metadata element contains a datatype then its value is simply exported. However, if a metadata element contains a system identifier then the entity it refers to is described as a "related entity". It is the relationships between related entities that give them context.

When an entity is exported in full, any related entities referred to by either its system metadata or its contextual metadata, are exported as placeholders. When an entity is exported as a placeholder then only its system metadata values are exported, not any related entities that these values may refer to.

### 11.2.9 Exporting significant entities

Some entities are more significant in providing the context of an entity than others. In MoReq2010®, not all related entities are significant entities and not all significant entities are related entities. For example, the disposal schedule of a record is very significant. However, the record may inherit its class from its parent aggregation and its disposal schedule from its class. The disposal schedule is therefore not an immediately related entity to the record even though it is a highly significant one.

For classes, the following entities are significant:

- The class's disposal schedule; and
- Any disposal holds associated with the class.

For aggregations, the following entities are significant:

- The aggregation's class, whether inherited or directly applied to the aggregation;
- The aggregation's parent aggregation, and any ancestor aggregations, up to and including the root aggregation; and
- Any disposal holds associated with the aggregation.

For records, the following entities are significant:

- The record's class, whether inherited or directly applied to the record;

- The record's disposal schedule, whether inherited from its class or directly applied to the record;
- The record's parent aggregation, and any ancestor aggregations, up to and including the root aggregation; and
- Any disposal holds associated with the record.

For users, the following entities are significant:

- The groups to which the users belong.

**Figure 11a** shows some of the significant entities for a record.

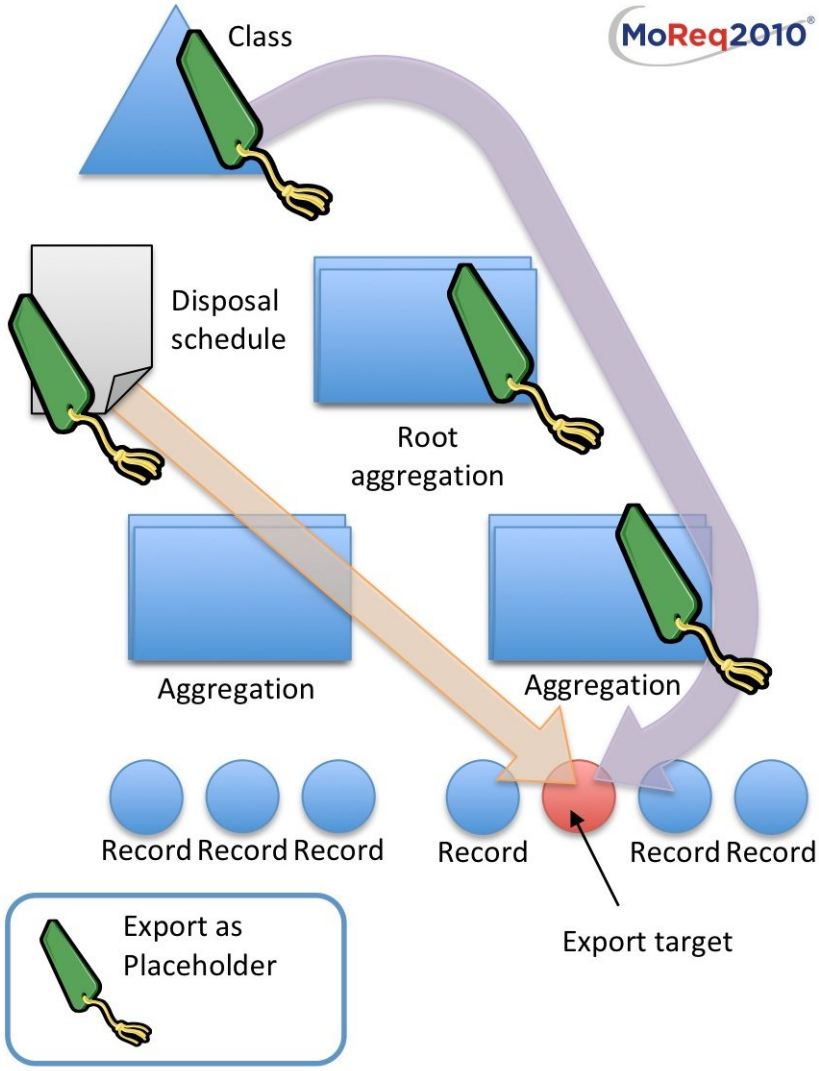


***Figure 11a** – Significant entities, such as the class, disposal schedule and ancestor aggregations for a record must be exported as placeholders.*

Significant entities are so significant that even when an entity is exported as a placeholder, its significant entities must also be exported with it as placeholders.

### 11.2.10 Exporting included entities

Some entities may be regarded as containing other entities; these are their "included" entities. The following is a list of included entities:

- The included entities of records are components;
- The included entities of aggregations are child aggregations and records;
- The included entities of groups are users;
- The included entities of templates are contextual metadata element definitions.

When entities with included entities are exported in full then their included entities are also exported in full. This is shown in **Figures 11b** and **11c**. Included entities are never exported as placeholders.

When entities are exported as placeholders, their included entities are not exported.

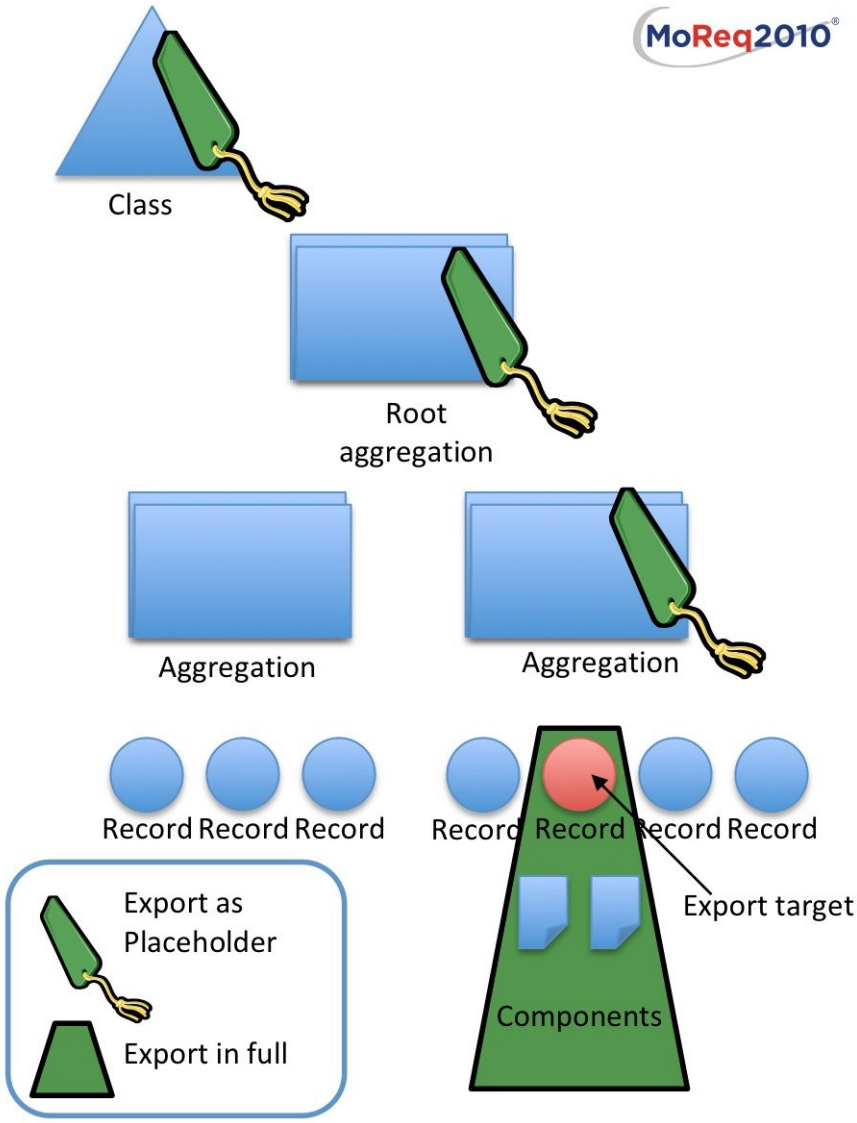


***Figure 11b** – An example of included entities are the components of a record; when the record is exported in full, the components are also exported in full*

All included entities must be exported in full, cascading down to any depth. For example, if a root aggregation is exported in full then any child aggregations are exported in full, any records in those child aggregations are exported in full and the components of those records are exported in full, and so on.

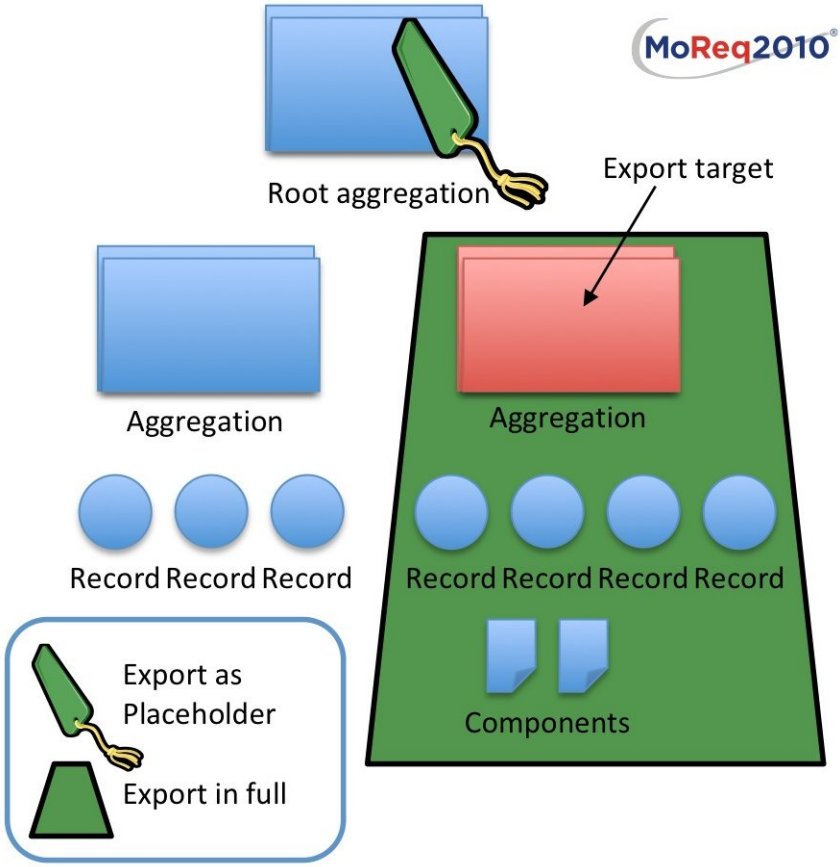


***Figure 11c** – Another example of included entities are the children of aggregations; all included entities are exported in full, so the included entities of included entities will be exported in full*

When an entity is exported in full because it is an included entity, then it has the same effect as if the included entity was one of the entities originally nominated for export. All entities are exported in full and any significant entities related to any of the included entities are also exported as placeholders, as shown in **Figure 11d**.

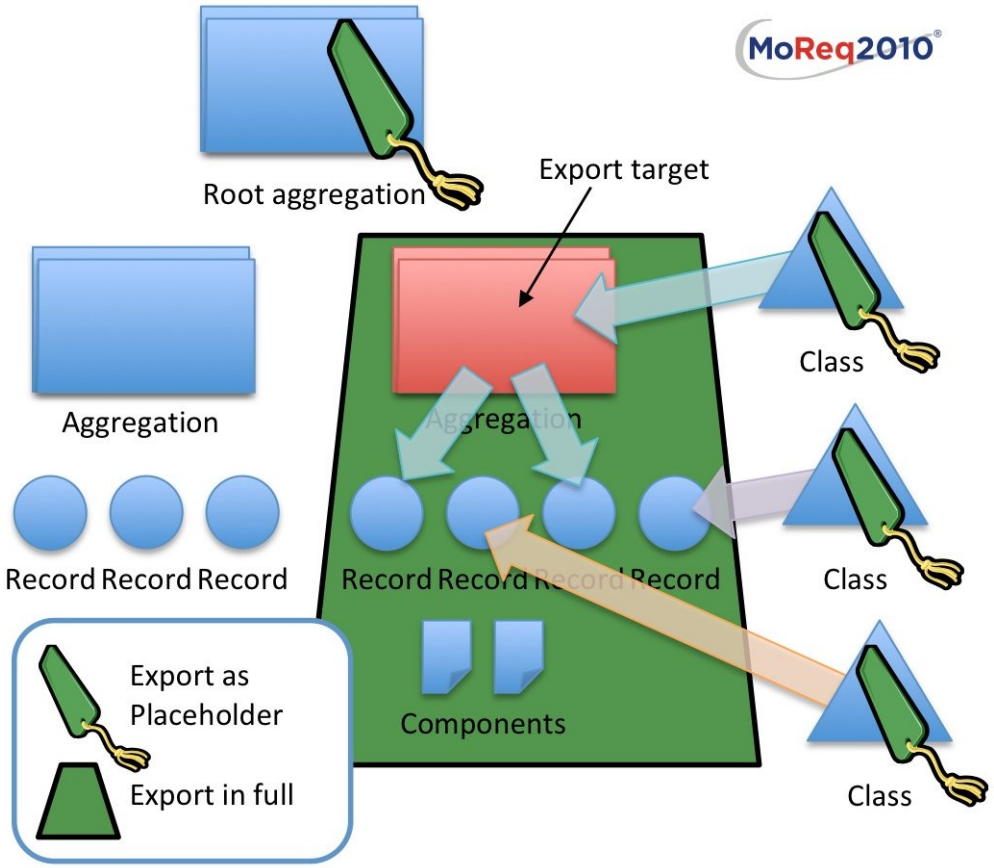


***Figure 11d** – Showing both included entities which are exported in full, and significant entities which are exported as placeholders*

### 11.2.11 Exporting access control lists

Access control lists contain access control entries that each associate a particular user or group with one or more roles. **Figure 11e** shows a typical access control list for an entity.

When entities are exported as placeholders it is still important that access to the information they contain is controlled. For this reason, the MCRS must export the access control lists of both entities that are exported in full and entities that are exported as placeholders.

Because the model role service supports the inheritance of access control lists from services, parents and classes, under **R4.5.11**, the MCRS must also export the access control lists of services relevant to the entities and placeholders being exported. In this way, the full access control list for an entity can be assembled on import from a combination of the service, significant entities such as parents and classes exported as placeholders, and the exported entity itself.

When an access control list is exported, whether for an entity that is exported in full or for a placeholder entity, each of the entities referred to by the access control list must also be exported as placeholders, including users, groups and roles. This is shown in **Figure 11f**.

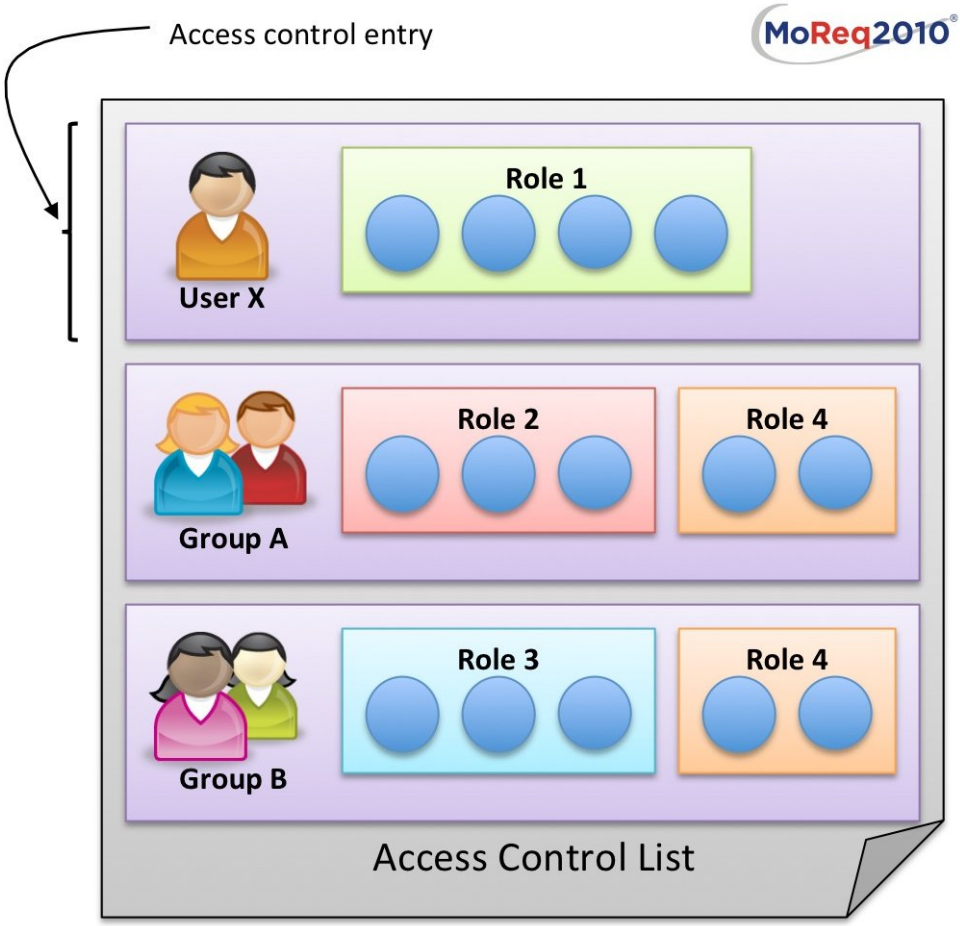


***Figure 11e*** *– A typical access control list; each access control entry associates a user or group with one or more roles*

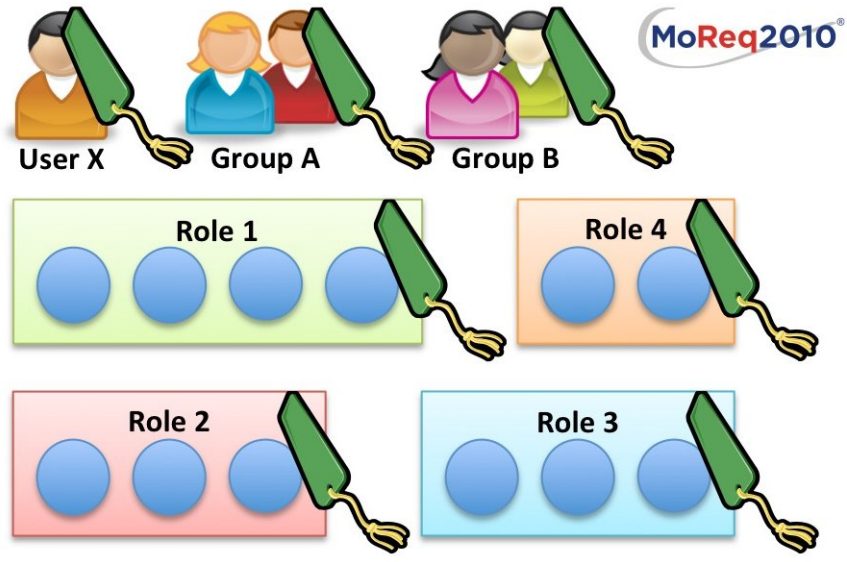


***Figure 11f*** *– All of the entities referred to by the access control list must be exported as placeholders*

### 11.2.12 Exporting events

Each entity's event history consists of a set of events in which the entity participated. For each of these events there may have been other participating entities, and the metadata associated with the event will refer to various related entities.

When an entity is exported in full then its event history is exported with it. This means all of the events in which the entity was a participating entity. In addition to exporting the metadata of the events, all of the related entities to the events should also be exported as placeholder entities.

When an entity is exported as a placeholder then its event history is not exported. None of the events in which the entity participated are exported with a placeholder entity.

### 11.2.13 Export summary table

The description given by **11.2.7** through **11.2.12** (above) indicates what an MCRS must do to compile a set of data for export. Starting with the entities nominated by the user, the MCRS must trace all of their related entities and decide which entities to export in full and which to export as placeholders. The MCRS must then compile this data by service, de-duplicate it, and export is as a coherent set of data.

The following table summarises these export decisions of the MCRS:

| What to export | For entities exported in full | For placeholders |
|---|---|---|
| **System metadata elements** | Export values | Export values |
| **Contextual metadata elements** | Export values | Do not export |
| **Contextual metadata element definitions** | Export as placeholders | Do not export |
| **Related entities (referred to by metadata elements)** | Export as placeholders | Do not export |
| **Significant entities (for example, a record's class)** | Export as placeholders | Export as placeholders |
| **Included entities (for example, the child entities of an aggregation)** | Export in full | Do not export |
| **Access control lists** | Export values | Export values |
| **Entities referred to by access control entries** | Export as placeholders | Export as placeholders |
| **Events and their metadata** | Export values | Do not export |
| **Entities related to events (referred to by metadata)** | Export as placeholders | Do not export |

### 11.2.14 Entities that are not exported

By default only active entities are exported from an MCRS. Residual entities are usually excluded from export, but may be optionally included by the user performing the export.

Note that saved searches and saved reports, referred to in **10. Searching and Reporting Service** are considered to be application specific objects, rather than MoReq2010® entities, and the specification makes no provision for their export.

### 11.2.15 Export security

A user cannot export in full or as a placeholder, any entity which the user is not authorised to inspect, and the MCRS must not allow entities to be exported by that user if they are normally inaccessible.

An export must not be carried out by the MCRS if the user does not have full access to export the requested entities and their placeholders.

### 11.2.16 Export completeness

MCRS solutions may be built around different architectures. The specification allows different products to implement their own access controls (see **4. Model Role Service**) and their own contextual metadata definitions (see **7. Model Metadata Service**). MoReq2010® also presents a service based approach where entities of different types are managed by discrete services, but MoReq2010® does not enforce the use of this approach, allowing each supplier to choose their own method of implementing an MCRS, including as a single non-modular application.

These freedoms must be balanced by the necessity for exporting all data to a common format that can be understood and used by any MCRS that implements the import service requirements. This means the following post-conditions must be true when any data is exported:

- At least in the data that is exported, MCRS solutions must export access controls as MoReq2010® access control lists, access control entities, and roles;
- At least in the data that is exported, MCRS solutions must export system and contextual metadata as MoReq2010® metadata element definitions and templates;
- At least in the data that is exported, MCRS solutions must not export any custom or proprietary entities or metadata that is not described by a contextual metadata element definition that is also exported; and
- At least in the data that is exported, MCRS solutions must export entities grouped together by type into their discrete services.

The last of these requirements, that entities of the same type are grouped together and exported as services, is necessary because MCRS solutions that support the import service must also be able to support multiple services of the same type. For example, an MCRS with its own classification service may then import a classification service from another MCRS. The imported classification service may even represent a different classification scheme structure (see the **200. Classification Series** modules).

When an MCRS imports a service, such as classification, from another records system then the classes in that classification service may either be mapped across to classes in its own

classification service, or they may be imported and managed by the MCRS as inactive classes that mirror the structure under which they were held in the originating records system. The ability to choose either of these import methods implies that the data being imported identifies entities as belonging to discrete services, even if they were not arranged in this way in the original records system.

### 11.2.17 The export process

Each time an MCRS performs an export it must create an Export Identifier for that export. This Export Identifier is included in the XML data as well as the export event generated for each entity (see **R2.4.16**). The same Export Identifier is used for all entities and placeholders exported together as part of a single export operation. The Export Identifier allows a user to find by searching which entities were exported together for any given export event. It also allows the import service to track which entities came from a single MCRS source as a snapshot of a particular moment in time.

When exporting entities, all MCRS solutions must follow the same export process. The following gives an overview of the export process:

1. Create an Export Identifier, as described above;
2. For each entity to be exported in full, build lists of related entities to be exported in full and related placeholders be exported as placeholders;
3. Initiate the export and write out header information including information about the MCRS and which services it complies with, the Export Identifier, and so on;
4. Export each relevant service in turn, and within each service perform the export of each entity in turn, exporting a block of placeholders followed by a block of entities in full for each entity type;
5. As each entity is exported generate an export event and add it to that entity's event history;
6. Complete the export; or
7. In the event of any failure or interruption to the export write any errors to the external error log (see **R2.4.7**).

The data format to export to and the order in which entities and their metadata elements are exported, are determined by the schema for the MoReq2010® XML format.

### 11.2.18 De-duplication

When exporting entities, the MCRS must ensure that it includes each relevant entity once in any export. If an entity is exported in full then it should not also be exported as a placeholder in the same export. If many entities refer to the same entity then it should be exported once regardless of how many times it is referenced.

### 11.2.19 Limiting access to data once it has been exported

When entities are exported from an MCRS then the exported entities are translated into a stream of XML data with none of the usual controls present that are implemented by an MCRS. This raises a number of security considerations that must be taken into account by any organisation using the export service:

- Care must be taken that the user undertaking the export has access to all of the entities that are to be included in the export – users cannot export entities that they are not themselves able to access and inspect in the MCRS;
- Operational procedures must be put into place to protect the export and its informational content - once it has been exported from the MCRS the data described by the export is no longer protected by the system and is in a simple XML format, organisations should consider encrypting and otherwise protecting it;
- If the XML data is in readable form it must be stored securely and access should be limited to those users who would be authorised to access it inside the MCRS;
- Where the data is sensitive it must be imported into the destination records system in such a way that the MCRS importing it simultaneously makes it secure so that other users of the destination records system cannot automatically access it. This is particularly important as access control lists including users, groups and roles from the source records system may not be respected in the destination records system.

Further measures may be required if data is stored outside an MCRS for an extended period of time. By preference, and especially for sensitive data, an XML export should only be retained temporarily and destroyed as soon as possibly after its successful transfer is confirmed.

## 11.4 Functional Requirements

### R11.4.1

The MCRS must allow an authorised user to export entities to an XML datafile that can be validated using the MoReq2010® XML schema.

The MCRS must allow the user to export any of the following:

- All users and groups collectively from a user and group service;
- Any nominated groups individually with the user entities that belong to those groups;
- Any nominated users individually;
- All roles collectively from a role service;
- Any nominated roles individually;
- All classes collectively from a classification service;
- Any nominated classes individually;
- All aggregations and records, with their components, from a record service;
- Any nominated aggregations individually and the records, with their components, they contain;
- Any nominated records individually, with their components;
- All element definitions and templates collectively from a metadata service;
- Any nominated templates individually with the contextual metadata element definitions that form part of the template;
- Any nominated contextual metadata element definitions individually;
- All disposal schedules collectively from a disposal scheduling service;
- Any nominated disposal schedules individually;
- All disposal holds collectively from a disposal holding service; or
- Any nominated disposal holds individually.

*See **11.2.4 XML data streaming**, the MoReq Governance Board is investigating replacing the export of entities to a datafile with the export of entities to a standardised data stream instead.*

*This future initiative is intended to resolve current technical and proprietary limitations on export such as how to provide a standardised mechanism for breaking large exports across multiple datafiles and what compression technologies an MCRS must support.*

*Function reference: **F14.5.185***

#### R11.4.2

Whenever an MCRS exports entities, under **R11.4.1** or **R11.4.3**, the MCRS must not, by default, export residual entities unless the authorised user specifically includes them. However, where the user so chooses, the MCRS must export all entities, including both active and residual.

*For example, if the user exports an aggregation, only the active records in the aggregation would normally be included in the export. However, the MCRS must also provide the user with an option to export the aggregation and include both its active and residual entities.*

#### R11.4.3

When preparing to export under **R11.4.1**, the MCRS must first determine which entities to export in full, these will be the following:

- The entities nominated by the authorised user under **R11.4.1**;
- The included entities of any entities to be exported in full (see **11.2.10** for a definition of inclusion); and
- The events of any entities to be exported in full.

The MCRS must then determine which entities to export as placeholders, these will be the following:

- All entities referred to by the metadata elements of entities to be exported in full, including their events;
- The contextual metadata element definitions for all the contextual metadata elements of entities to be exported in full;
- All entities that are significant to entities to be exported in full or as placeholders (see **11.2.9** for a definition of significance); and
- All entities that are referred to by the access control lists of entities to be exported in full or as placeholders.

*These rules must be applied iteratively until all entities to be exported in full and all entities to be exported as placeholders are identified; subject to **R11.4.2**, meaning that by default only active entities are to be included unless the user also includes residual entities. No entity should be included twice and no entity should be exported as a placeholder if it is simultaneously being exported in full.*

#### R11.4.4

When an authorised user exports one or more entities under **R11.4.1**, then the MCRS must generate a universally unique identifier for the export.

*See **R2.4.24**. Each separate export from an MCRS represents a snapshot in time of a part of the MCRS. Clearly and uniquely identifying each export as it is made allows another MCRS to import a series of exports over time and more easily and precisely piece the information back together.*

*The Export Identifier is stored as part of the export event for each entity included in the export, see **R11.4.7**.*

#### R11.4.5

When an authorised user exports entities under **R11.4.1** then the MCRS must allow the user to provide a text comment to be included in the export data under **R11.4.6**, and the export event under **R11.4.9**.

*The export comment is an explanation of why the export was performed and it contains by the user initiating the export.*

#### R11.4.6

Once an export has been initiated under **R11.4.1** the MCRS must export the following:

- An Export Commencing Timestamp;
- The Export Identifier generated under **R11.4.4**;
- The export comments included under **R11.4.5**;
- An export header containing full identification and information on the services supported by the MCRS, see **R2.4.5**;
- Metadata and access control lists for each service, see **11.2.11 Exporting access control lists**;
- For each service the entities to be exported, grouped into placeholders or entities exported in full, see **R11.4.3**; and
- An Export Completed Timestamp.

*The MoReq2010® XML schema specifies the complete and detailed export format that must be met.*

#### R11.4.7

For each entity to be exported in full, under **R11.4.3**, the MCRS must export the values of each of its metadata elements, including both system and contextual metadata elements, and its access control list.

*The MCRS must export both system metadata elements and the contextual metadata elements for entities to be exported in full, including events. All the metadata in the entity's access control list and its access control entries must also be exported.*

#### R11.4.8

For each component to be exported in full, under **R11.4.7**, the MCRS must export the content of the component, subject to the separate provisions specified by the applicable MoReq2010®, **300. Component Series** module for the component's entity sub-type.

*The content of different types of components will be exported in different ways, as specified by the plug-in module for the appropriate component type. Generally, the content of components will either be included in the export XML, or exported separately but referenced by the export XML, or exported separately and not referenced by the export XML.*

#### R11.4.9

For each entity exported as a placeholder, under **R11.4.3**, the MCRS must export the values of each of its system metadata elements, and its access control list.

*The values of contextual metadata elements are not exported for placeholder entities. All the metadata in the placeholder's access control list and its access control entries must be exported.*

#### R11.4.10

For each entity exported in full, under **R11.4.7**, and for each entity exported as a placeholder, under **R11.4.9**, the MCRS must add to the event history of the entity an event that includes:

- The Export Identifier, see **R11.4.4**;
- An Exported In Full Flag; and
- The export comment, as the Event Comment, see **R11.4.5**.

*If an export is only partially completed, either because it is cancelled, or fails, or for any reason then events should only be generated for those entities that were exported successfully prior to the termination of the export process.*

*The Exported In Full Flag indicates whether the entity was exported in full or otherwise as a placeholder.*

*Note that as service information is exported with every export, under **R11.4.6**, and services are not exported as entities, an exported event is not generated when a service is exported.*

*Function reference: **F14.5.10**, **F14.5.29**, **F14.5.43**, **F14.5.52**, **F14.5.62**, **F14.5.76**, **F14.5.100**, **F14.5.127**, **F14.5.148**, **F14.5.170**, **F14.5.185**, **F14.5.186***

# 12. Non-functional Requirements

## 12.1 Key Concepts

### 12.1.1 Non-functional requirements in MoReq®

Non-functional requirements have always been an important part of the MoReq® specification since the publication of the original MoReq® in 2001. They specify those qualitative aspects of the records system which are not necessarily made explicit by the functional requirements alone. Functional requirements tend to focus on the specific behaviour required from the system, without focussing on these closely related environmental, operational, infrastructural and comfort factors.

By their very nature, non-functional requirements are less definitive and more subjective than functional requirements. They are more difficult to specify in a universally applicable way, are more open to interpretation, and are more difficult to quantify, measure and test.

Nevertheless, experience with previous versions of MoReq® shows that both suppliers and consumer organisations often draw important and practical information from the specification's non-functional requirements. This has helped suppliers to enhance the quality of their products, and consumer organisations to select records systems that are well suited to their business needs and environments. It is interesting to note that during the consultative phases leading up to the development of MoReq2010® contributors showed a clear preference for retaining and augmenting the part of the specification covering non-functional requirements.

### 12.1.2 Functional and non-functional requirements in MoReq2010®

Within any statement of requirements, the boundary between the purely functional and the non-functional aspects of the specification is often subjective and difficult to define with clarity. MoReq2010® has been purposely designed so that its non-functional requirements can fulfil a different and separate capacity to that of its functional requirements.

The characteristics of the two different types of requirements within MoReq2010® may be expressed as follows:

**Functional requirements**

- Functional requirements in MoReq2010® are expressed as closed statements, for example, "The MCRS must…" or "The MCRS must not…";
- Each functional requirement relates directly to one (or sometimes more than one) explicit function that must be performed by an MCRS;
- Most functional requirements in MoReq2010® are accompanied by a rationale providing additional clarification of the requirement;
- The explicit functions described by the functional requirements are each individually identifiable, are listed under **14.5 Function Definitions**, and are utilised and referred to by the architecture of MoReq2010®, both within the access control model (see 4. **Model Role Service**) and the event model (see **2. System Services**); and
- The functional requirements of MoReq2010® are verifiable by test cases and scripts within the test framework that accompanies the specification. Testing of a product or

installation by an accredited test centre against these functional requirements can lead to the award of a certificate of compliance by the DLM Forum®.

**Non-functional requirements**

- A non-functional requirement in MoReq2010® relates to a desirable characteristic or quality of the MCRS;
- For each non-functional requirement in MoReq2010® the rationale, reason, or premise is listed first and is then followed by the non-functional requirement;
- Each non-functional requirement is expressed as an open or closed question that is addressed to the supplier of an MCRS solution, for example, "How does the records system ensure…" or "What provision does the records system make for…";
- MoReq2010® does not require that products or installations be tested for compliance against non-functional requirements and makes no provision for this in the MoReq2010® test framework, although non-functional requirements may be assessed by individual organisations outside the test framework;
- However, to be certified against MoReq2010® each supplier must document and submit, as part of the pre-qualification phase of testing and certification, its responses to the non-functional requirements, as they relate specifically to the product or service being put forward for testing (this process is described under **12.2.4 Addressing the non-functional requirements**).

### 12.1.3 What the non-functional requirements cover

The non-functional requirements take into consideration the following aspects of an MCRS, and are described in greater detail in **12.3 The Non-functional Aspects of a Records System**. Each records system must have the qualities of:

- Performance,
- Scalability,
- Manageability,
- Portability,
- Security,
- Privacy,
- Usability,
- Accessibility,
- Availability,
- Reliability,
- Recoverability,
- Maintainability,
- Supported,
- Warranted, and
- Compliance.

MoReq2010® does not necessarily provide a comprehensive, nor exhaustive, list of non-functional requirements for records systems, covering all these aspects. In many cases these will be specific to a particular organisation, industry, type of system, environment or legislative or regulatory regime. The importance placed on particular non-functional requirements will also be different for different stakeholders.

### 12.1.4 Addressing the non-functional requirements

The DLM Forum's procedures for certifying products and installations against the MoReq2010® specification require a number of steps. Before formal testing of a product or installation can begin, the supplier must complete a pre-qualification process. As part of this process the supplier must describe the product or installation that is being put forward for certification. This includes providing detailed written responses to each of the requirements, both functional and non-functional, for the core services and modules of MoReq2010® being tested.

For this reason, the non-functional requirements of MoReq2010® are expressed as questions and suppliers must document their answers to each of these questions in respect to their particular product or installation. The accredited test centre will then test the product or installation against the functional (only) requirements of MoReq2010® using the test framework.

Upon the successful completion of the functional testing phase, and following any necessary corrections to the supplier's original responses, the supplier's responses to both the functional and non-functional requirements will then be incorporated into the full test verification report along with the test results and the test centre's recommendation. Under various legal terms and conditions, separately specified and managed by the MoReq Governance Board, test verification reports for certified products will be made available to be accessed and viewed by members of the DLM Forum®.

Therefore, while the supplier's product or installation is never formally tested against the non-functional requirements, by their inclusion in the test verification report, the supplier's responses to the questions raised by the non-functional requirements does become an important reference resource. The supplier's responses can become instrumental to helping consumer organisations find the best fit between their own local, organisational needs and the range of MoReq2010® certified solutions that are available on the market.

This inclusion of non-functional requirements in the certification process, even though they are not formally tested, gives additional importance to their consideration under MoReq2010® than in previous versions of the specification.

### 12.1.5 Testing the non-functional requirements

While the MoReq2010® test framework does not make provision for formal testing against the non-functional aspects of records systems, this should not be interpreted as meaning that non-functional requirements can never be empirically tested or measured. Many organisations may wish to do this as part of their own evaluation processes against their own local assessment criteria.

Unlike the testing of functional requirements, where a records system is likely to be passed or failed on each individual criterion, evaluating compliance or non-compliance against non-functional requirements is a more subjective exercise. Often this will result in the records system receiving a score against a scale or continuum.

For example, a non-functional requirement may state that a records system must be provided with adequate user documentation. How is this to be evaluated, and what is the meaning of the word "adequate"? Each organisation must judge this for itself.

One way to assess the quality of user documentation may be to give it to a sample group of users drawn from the organisation who are asked to evaluate it and use it under various simulated test conditions, and then rate it against various criteria, such as:

- Was it indexed and organised logically?
- How long did it take to find the relevant section?
- How much assistance did it provide in a carrying out the task?
- Were there any areas where insufficient documentation was provided?
- Did it use suitable language and could you understand any jargon?
- And so on.

The users may subsequently be asked to rate their experiences with the user documentation on a five point Likert scale, for example:

1. Unusable, unintelligible or missing;
2. Poor or patchy documentation;
3. Acceptable and understandable once the section was found;
4. Good quality, well laid out and indexed; or
5. Excellent, relevant, easy to find and extremely helpful in use.

This process of trialling a system with a pilot group of users from the organisation and collating their responses is usually described as "user acceptance testing" of a records system, and is one of a number of evaluation approaches that include, but are not limited to:

- User acceptance testing;
- Security/penetration testing;
- Load testing;
- Stress testing;
- Installation and configuration testing;
- Disaster recovery testing;
- Interoperability testing; and
- Environmental testing.

The non-functional aspects of records systems may also be evaluated against external standards and specifications where these are applicable, for example, assessment against the ISO/IEC 27000 series, information security standards. There may also be localised requirements applicable to a particular jurisdiction. For example within the United Kingdom, the British standard BSI DISC PD0008 relating to the "Legal Admissibility and Evidential Weight of Information Stored Electronically" (2009) is accompanied by a workbook that allows an organisation to carry out an independent assessment in relation to a particular installation.

Where there are relevant standards, such as those described above, they often provide a benchmark against which an individual products and installations can be judged. Otherwise the assessment of non-functional requirements may need to be relative rather than absolute, requiring the evaluation of two or more records systems against each other, to discover by direct side-by-side comparison which is the more suitable when judged against a particular non-functional requirement.

Apart from direct observation and evaluation, organisations may also usefully verify and adjudicate suppliers' responses to the non-functional requirements for MCRS solutions, as

contained in the test verification report, by undertaking reference checks and site visits to existing installations that have already deployed and are already using the supplier's solution. Other comparable organisations can often supply useful empirical data against the non-functional requirements, such as:

- The number of support issues which have been raised;
- The suppliers responsiveness to critical issues;
- The amount of user training required;
- The percentage of system uptime over a particular period;
- The frequency of product upgrades;
- The general level of satisfaction amongst the user community;
- And so on.

It is important when assessing and evaluating non-functional requirements that the needs of all the various stakeholders within the organisation are taken into account. Organisations may wish to formalise the acceptable level of performance against the non-functional requirements of MoReq2010® by entering a service level agreement (SLA) with the supplier.

## 12.2 The Non-functional Aspects of a Records System

### 12.2.1 Performance

Performance relates to the responsiveness, efficiency and throughput of the records system under load. It is difficult to properly evaluate on a demonstration system or in a pilot study or model office and may change considerably when the whole organisation is using the records system. Performance is often also highly dependent on hardware, such as network bandwidth, CPU cores and cycles, memory capacity, available storage space on hard drives, and so on. One bottleneck may cause the whole installation to slow down.

The organisation and the supplier may put into place a service level agreement over performance that stipulates, for example, how long it should take to create a record, how long it should take to find and retrieve a record, etc.

### 12.2.2 Scalability

Scalability relates to the performance and capacity of the system over time and under increasing load. As the number of records grows along with the number of users and the consequent load on the system, how easy is it for the records system to maintain the same level of performance stipulated under **12.3.1**?

Suppliers often seek to make provision for scalability by either:

- Scaling up – increasing the size and capacity of the records system; or
- Scaling out – balancing the increased load between several records systems, or across multiple services.

Organisations may wish to stress test new records systems before they deploy them to assess both their performance and their scalability.

### 12.2.3 Manageability

A records system must make provision for its own management and administration. There are two aspects to this.

Technical administration of the records system includes:

- Installation and configuration;
- Monitoring the utilisation of system resources;
- Adding additional storage and other capacities as required;
- Accessing and managing the error log;
- Upgrading the records system; and
- Trouble-shooting and resolving technical issues.

Management of the records system from the perspective of the records manager, includes:

- Administrative reporting;
- Usage statistics on usage of the system by users, for example, number and types of searches performed;
- Collation and monitoring of statistics on the number of records and related entities of various types under management;
- Provision of facilities to allow auditing of the records system;
- And so on.

### 12.2.4 Portability

Portability refers to the ability of the records system to operate successfully within different environments. A records system may be designed to execute only within a single technology stack provided by a single operating system vendor, such as within Microsoft's Windows server and client environments; or a records system may be cross-platform.

Some records systems have a server component, which is tied to a particular operating environment, and support a client component that is not. In the interests of portability, a supplier may choose to provide a range of different clients for the web and for particular platforms, operating systems, and mobile devices.

For records systems that are intended to interface directly to other business systems, portability is dependent on the solution's support for a range of different interfacing standards and technologies, for example, a Java interface, or web services, or REST based APIs.

Where a records system is only a specialised component of a wider business system, there may be little choice about which technology or technologies it supports. For example, a records system may be developed to support and be a part of a particular customer relationship management (CRM) solution.

Portability also extends to the degree with which a solution can be customised for different environments, although customisation and installation options, while useful, can also be problematic for records systems as explained under **12.2.15 Compliance**.

### 12.2.5 Security

Security relates to the external integrity of the records system and its ability to withstand unauthorised access, hacking or tampering, computer viruses, and other forms of accidental or malicious damage. Penetration testing and security assessments under the ISO 27000 information security standards are recommended.

Ideally a records system should be:

- Physically secure – with restricted access to hardware, equipment and installed software integral to its operation;
- Secure in its data – ensuring that the information stored on server and client devices is not accessible except through the application itself;
- Secure from unauthorised access – requiring one or more factors of authentication;
- Secure in its communications – utilising digital certificates and encryption where possible to ensure that information is exchanged only with the intended recipient; and
- Internally secure – enforcing access controls that do not allow different users to perform functions and access entities where they are not explicitly granted permission to do so.

### 12.2.6 Privacy

Closely related to security, it is important that any records system respect the privacy of personal information and data. This is particularly important for records systems intended to hold sensitive information, such as medical records.

Many countries, both in and outside Europe, observe provisions about the privacy of personal information. Each records system must comply with these regulations and protect the rights of citizens in the local jurisdiction where it is deployed.

For example the European data privacy directive, Directive 95/46/EC of the European Parliament and of the Council, regulates the processing of personal data; makes provision for notifying subjects when their personal data is held, including the guarantee of a right of access and a right to object; and places restrictions on the transfer of personal data to third countries.

This has direct relevance to records systems, for example, where the entities in a records system may be stored, particularly in relation to an international or cloud based storage service provided by an MCRS.

### 12.2.7 Usability

Ease of use is an important consideration in records systems, especially for user acceptance. Experience shows that users will bypass a system that is overly complex or time consuming to use. This has serious consequences for records systems in organisations if important records are not being captured or employees are not taking advantage of the accumulated corporate knowledge they contain.

Some of the features of a records system that contributes most to usability include:

- Clean, uncluttered and familiar interfaces;
- Consistency throughout the application and with the operating environment;
- System responsiveness;
- Informative error messages and dialogues;
- Automated processing, providing useful defaults, and other ways of minimising the number of decisions users must make;
- Minimising the number of user actions required to perform an operation;
- Providing alternative ways of executing common functions, such as through keyboard key combinations, tool bar buttons, and so on;

- Enabling bulk processing;
- Support for internationalisation;
- Support for personalisation and localisation;
- Context sensitive help facilities;
- High quality user documentation;
- Frequently asked questions; and
- Online videos and tutorials.

Training and education programmes are often essential to the uptake and acceptance of new systems and records systems are no different in this respect.

### 12.2.8 Accessibility

Closely related to usability, records system should ideally be accessible to all types of user with different capabilities, including those with specific disabilities. Such users have an important part to play in all areas of human activity where they will regularly interact with records systems. Many exemplar business organisations protect the rights of their employees to access any of their technical systems, and as a consequence will only purchase solutions that incorporate a wide provision for accessibility.

A leading proponent of the active evaluation of non-functional requirements for accessibility is the World Wide Web Consortium (W3C) Web Accessibility Initiative (WAI). The W3C WAI issues the Web Content Accessibility Guidelines (WCAG) which cover recommendations for making Web content more accessible.

"Following these guidelines will make content accessible to a wider range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited movement, speech disabilities, photosensitivity and combinations of these. Following these guidelines will also often make your Web content more usable to users in general." (WCAG 2.0:2008, Abstract)

Against each guideline, the Web Content Accessibility Guidelines provide success criteria which may be tested and evaluated. These test criteria allow for three levels of conformance:

- A – the solution meets the mandatory checkpoints;
- AA – the solution meets the mandatory and the highly desirable checkpoints; and
- AAA – the solution meets the mandatory, highly desirable and desirable checkpoints within the WCAG specification.

Although the WCAG specification is intended for web applications, accessibility is an important consideration for all records systems, and these same principles should be adopted across all platforms.

### 12.2.9 Availability

Availability requirements are important considerations for assessing which records systems implementations will be suitable for an organisation's business practices and is often expressed as a percentage or a ratio of system uptime compared to downtime.

Depending on the nature of the organisation, some organisations require access to the records system during business hours on weekdays, whereas other organisations require 24x7 access and support. Not all records systems are capable of running indefinitely without

requiring regularly planned maintenance, upgrade and backup windows when they must be taken off line. However, every business also has certain critical periods where system availability is essential.

The level of availability that a particular solution can provide should be clearly stated by the supplier in response to the non-functional requirements. This should also be included in any service level agreements between the supplier and the organisation. Where the records system is hosted by a third party provider, a separate service level agreement covering the provision of the hosting service may also be necessary. Likewise the organisation should be realistic about its need for utilising the records system at all times of the day or night, as 24 hour service and support from the supplier or service integrator will necessarily come at a higher cost to the organisation.

If an organisation has agreed to a particular level of records system availability in a service level agreement, then system availability should be closely monitored to see whether the service target's that have been set are being met or exceeded. Aside from any penalty clauses included in the service level agreement, monitoring system uptime, especially against system utilisation, will provide useful feedback to both the supplier and to setting the expectations of the organisation around the availability of its records systems.

### 12.2.10 Reliability

Reliability describes the internal integrity of a system, the precision and accuracy of its software, and its resilience to defects, malfunctions and unexpected operating conditions. It is possible to apply proof of correctness algorithms to records systems, as well as assessing their tolerance of invalid data and other unexpected occurrences, such as a sudden loss of communication. Truly robust records systems will be able to gracefully handle error conditions, without crashing or sudden failure.

Reliability is also closely related to system availability, and reliability is often measured as the "mean time between failures". More reliable systems will therefore have more system uptime and far less unplanned downtime, making them more available. Equally, a more reliable records system will generally be one that is easier to maintain and actively supported by the supplier.

The MoReq2010® test framework can be used by accredited test centres to check the functional requirements of a compliant records system, and this provides one measure of the reliability of a records system in performing the essential tasks of records management. However, the test framework does not by itself test all the possible inputs and outputs of a records system, and an organisation may not wish to rely solely on this single measure when comparing the quality and reliability of different records systems.

### 12.2.11 Recoverability

Should a records system fail for any reason, it is important that the organisation is able to recover it with its data largely intact. Operations that were occurring at the moment of failure may not be able to be recovered, however, the organisation must be equally sure that it is not dependent on backed up information that is more than a day old, especially in high volume environments.

Not only must the records system be recovered or rebuilt, but this must also be done in a timely manner, to avoid any unnecessary impact on the critical business of the organisation. A disaster recovery process that means that the records system is offline for several days or weeks may not be suitable for many businesses. For other businesses, unscheduled downtime of even a few hours may be too long.

As with other non-functional requirements, requirements around disaster recovery will vary substantially between different organisations. It is essential that the needs of the organisation are assessed prior to any disaster occurring and that a full and comprehensive business continuity plan be put in place across the organisation.

Experience shows that business continuity requirements must be planned across all business critical systems, and not for individual systems, such as the records system, alone. This is because the records system forms only part of the corporate infrastructure. If the records system, for example, depends on particular hardware then the organisation must have access to replacement hardware in a timeframe that meets the targets set for recovery of the records system.

It is also good practice to regularly perform disaster recovery testing so that staff know the process and procedures for restoring the records system from any system failure and realistic timings for full system recovery are known. As the amount of data managed by the records system grows over time, the organisation's expectations over the period of time required to recover the records system may not keep pace with the actual period required to physically transfer the organisation's data into a newly rebuilt system.

Some mission critical organisations cannot allow for hours or even minutes of downtime for disaster recovery. These organisations must consider running multiple parallel systems with mirrored data, that enable them to switch between their live and reserve records systems on hot standby, warm standby or cold standby. Or alternatively, an organisation may run a single system but with many layers of built in redundancy that allows, for example, the hot swapping of individual hardware and software components such as computers, routers and hard drives while the system, as a whole, remains operational.

Recoverability, like availability, will cost the organisation more for each higher degree of redundancy and immediacy specified. Organisations must be careful, therefore, to understand their own disaster recovery needs and to specify realistic targets within their business continuity plans. Where meeting these targets is mission critical, the organisation should seek to test recoverability and to put in place a service level agreement with the supplier to ensure that the records system will comply with the organisation's operational requirements.

### 12.2.12 Maintainability

A records system must be maintainable. Meaning that it must be relatively easy to repair and upgrade. Most suppliers will have a system of major, minor and maintenance versions, which might variously be called something like: new versions, service packs, and patches.

Each of these will have a cost associated with its deployment in terms of the resources and time taken to deploy across the organisation, and where they incorporate new features and functions, there may also be a retraining and educational cost for users. As a general rule,

the more major the release the more likely that there will be an impact on the organisation, including possibly migrating data from the earlier version.

Records systems are not always built from components from a single supplier. Often suppliers licence (or "OEM") components such as search engines and databases from other suppliers, or deploy open source equivalents within their applications. Reuse, or bringing different technologies together can be far more efficient that a single supplier reinventing every records system component. However, organisations should be aware of the underlying patchwork of different technologies within their records systems as these may individually need upgrading at different times and for different reasons.

For example, a records system may use a well known database engine made by an independent database supplier. Whenever that supplier upgrades or patches the database software, these changes must be retested for compatibility with the records system solution. Upgrades to the records system must therefore keep pace with upgrades to its database engine. This same situation may equally apply to the search engine, the storage solution, and other parts of the records system.

### 12.2.13 Supported

Regardless of how maintainable the records system is, the supplier must actively support it. Experience shows that many organisations have legacy systems where the supplier is no longer in business, or has decided to maintain but not to upgrade a particular records system solution.

Therefore, before purchasing a records system, an organisation should request details of the level of support given to the product, how often it is upgraded, when the last version was released, and what the supplier's roadmap is for the records system.

Organisations should understand that the modern information technology marketplace is highly volatile and even these precautions may not prevent a particular records system from failing to be supported in the future, even by the biggest of suppliers. Fortunately, MoReq2010® provides some consolation through its support for interoperability. At worst the organisation may have to migrate its records from its old MCRS to a new MCRS solution.

Support also refers to the level of day-to-day support provided by the supplier organisation or a third-party on behalf of the supplier organisation. Organisations should seek to learn how to request assistance from the supplier, how to report errors and software bugs, and what level of on site support and assistance to expect. Many suppliers have active user groups where different organisations come together to share experiences, hints and tips, and other information about how best to utilise a particular records system.

### 12.2.14 Warranted

Organisations should be fully aware of the licensing and other terms and conditions related to installing and using a particular records system from a particular supplier. Even open source solutions have intellectual property rights and usage conditions associated with them.

In agreeing to the supplier's terms and conditions around implementing a records system, an organisation must also ensure that it receives a warrant from the supplier covering the use of the records system and agreeing to fix any issues encountered in its deployment and use by the organisation.

Many organisations require that the source code of a records system be placed into escrow so that should anything happen to the supplier, the records system remains supportable. Again, as with the previous section, MoReq2010® actively seeks to remove the possibility that data will be lost from an unsupported records system by ensuring that every MCRS includes the basic functionality of full system export, to enable the lossless migration of records and entities to another MCRS.

### 12.2.15 Compliance

The final non-functional aspect of a records system is its level of compliance. This has already been mentioned within other criteria. Records systems may need to be compliant with industry standards and local regulations in the following ways:

- They must be compliant with the MoReq2010® specification;
- They must comply with all legislative and regulatory standards that apply to the organisation, in the industry and under the jurisdiction where they are deployed, such as health and safety regulations or freedom of information;
- They should comply with widely used and accepted industry standards in technology, and on the platforms where they are deployed, such as HTML and HTML5 for web browser based applications; and
- As required by the organisation, they should seek to be compliant with popular document formats, such as PDF, enabling the records system to examine the structure of these documents, extract their metadata, and index their content for searching; especially within organisations where the purpose of the records system is to manage and maintain records in these formats.

Because the compliance requirements of records systems can be so broad and can vary depending on local conditions, many suppliers build a degree of configurability into their solutions. While configurability is often a very positive feature of an application, it can also be a threat to compliance when applied to a records system. This is especially true if the configuration and installation options allow essential records management functionality to be manipulated or turned off.

The non-functional requirements of MoReq2010® require that suppliers indicate if their systems allow customisation options which may invalidate the functional requirements of MoReq2010®. Where this occurs it is essential that the organisation checks that the records system remains MoReq2010® compliant after it has been installed and become operational. This can be done using the compliance reporting features of MoReq2010®, see functional requirement **R2.4.5**.

## 12.3 Non-functional Requirements for Performance

#### N12.3.1

*In assessing the performance of a records system it is important to understand the purpose of the records system and the nature of the records that can be created and stored.*

What is the records system designed for and what types of records can it manage?

#### N12.3.2

*It is also necessary to understand the size and complexity of the average deployment. Often this is described in terms of the:*

- *Number of simultaneous users;*
- *System utilisation percentage per user;*
- *Number of records under management;*
- *Space occupied by an average record;*
- *Amount and type of storage space required, including search indices and other system requirements; and*
- *Number of servers and types of server required.*

What is a typical small deployment, medium deployment and large deployment of the records system?

(Give examples where possible.)

#### N12.3.3

*Records systems have different cycles of usage. It is important to understand whether a particular records system will suit the sustained workload of the organisation.*

For each of the typical deployments described in **N12.3.2**, describe their typical usage of the records system during normal operations and indicate what might be considered periods of peak load?

(Give examples where possible.)

#### N12.3.4

*Throughput can be measured by the number of records that can be captured into the records system.*

For each of the typical deployments described in **N12.3.2**, how many records can be captured and simultaneously retrieved per hour on average, during normal operation and in periods of peak load, as described in **N12.3.3**?

#### N12.3.5

*An important measure of system performance is how long users spend searching on average.*

For each of the typical deployments described in **N12.3.2**, how long on average is a search across three metadata elements, such as Title, Class and Created Timestamp, that returns 100 records, during normal operation and in periods of peak load, as described in **N12.3.3**?

#### N12.3.6

*Some records systems implement a timeout interval if searches take too long.*

What is the longest possible search time for any search and can it be configured?

#### N12.3.7

*Another measure of system performance is how regularly each record is assessed for disposal. Some records systems may do this in real time, or periodically at scheduled intervals.*

MoReq2010® allows the disposal process described in **R8.4.14** to occur periodically, and it must be performed at least daily, how regularly should the records system perform this process for each of the typical deployments described in **N12.3.2**?

(Where the records system adopts an alternative design approach, provide an explanation of its effects, both beneficial and detrimental, on performance, scalability and other factors.)

## 12.4 Non-functional Requirements for Scalability

### N12.4.1

*Some systems are restricted by technical or other limitations such as database size, file system segmentation, single server utilisation, and so on. These limitations may apply to:*

- *The maximum number of simultaneous users;*
- *The maximum number of concurrent users;*
- *The maximum number of entities, including records;*
- *The storage space used by the records system; and*
- *The hardware that can be deployed to support the records system.*

*In particular, organisations want to achieve scalability while still protecting their investment in hardware and other resources. Records systems should therefore provide means by which additional capacity can be added to an existing system without migrating it to a new environment.*

What are the upper limits of scalability for each of the typical deployments described in **N12.3.2**, without replacing existing hardware?

### N12.4.2

*The MoReq2010® specification imposes no theoretical upper limit on how many entities, how much metadata, how many users and how much content an MCRS can hold. There may, however be practical considerations with any records system.*

For each response to **N12.4.1**, what strategies should an organisation put in place to expand its deployment of the records system beyond these technical limits, assuming the number of users doubles and the number of records increases fivefold over a period of three years?

### N12.4.3

*As a records system is scaled up and out, the performance of each of the functions performed by the system may be affected.*

For each response to **N12.4.2** what will be the impact on:

- The throughput of the system described in **N12.3.3**?
- The average search time described in **N12.3.4**?
- The search timeout described in **N12.3.5**?
- The regularity of the periodic disposal process described in **N12.3.6**?

### N12.4.4

*Records systems may also impose internal limitations on the numbers, types and relationships between entities. For example, the classification service may have an upper limit on the number of classes it may contain.*

What are the technical limits in the records system to each of the following:

- The number of entities that can be managed by any service, or bundle of services under **R2.4.1**?
- The number of root aggregations that can be added to the record service?
- The number of entities, either child aggregations or records, that can be added to an aggregation?
- The depth or number of levels of aggregation under a root aggregation?
- The number of components in a record?

#### N12.4.6

*As the number of entities in a records system grows, not just search times but the number of results returned by a search will also grow. Some search engines approximate the total number of entities rather than calculate it for searches that return a large number of entities. Search engines also have different ways of determining the relevance of searches.*

What is the maximum number of search results that the records system will find and return and what mitigating strategies does the search engine use to make the first few search results more relevant?

#### N12.4.7

*Requirement **R10.4.16** requires that an MCRS allow users to chain or join several search queries so as to answer complex search enquiries.*

What limits the number of chains or joins the records system can include in a search under **R10.4.16**, and what is the impact of chaining or joining searches on:

- The search response time?
- The relevance of search results?

## 12.5 Non-functional Requirements for Manageability

#### N12.5.1

*It is important to understand how the organisation should go about installing and configuring the records system. Often this is done as part of a wider project.*

How is a new instance of the records system installed and configured, and who should undertake this work?

#### N12.5.2

*While the records system is in operation its resource usage should be monitored to ensure that the system has adequate reserves. Measuring of resource usage can extend to:*

- *Number of users accessing the system and at what times and on what days;*
- *Amount of storage being used and rate of increase;*
- *Average search time and rate of increase or decrease;*
- *Traffic, in terms of functions performed, and rate of increase or decrease;*
- *Average response time to all functions; and*
- *CPU and memory utilisation.*

What means are employed by the records system for measuring resource usage?

#### N12.5.3

*Resource usage may reach a critical point where more resources must be given to the system. It is important that technical administrative staff anticipate this and add more resources as required, for example, increasing the storage space available before the existing storage space is exhausted.*

How does the records system, while monitoring resource usage under **N12.5.2**, warn technical administrators of anticipated resource shortages, and can resource thresholds be pre-set?

#### N12.5.4

*Resources available to the records system may be able to be increased, but this may be a difficult task, and for some records systems requiring a maintenance window of system down time.*

What capacity exists for increasing the resources available to the records system, how is this done?

#### N12.5.5

*In addition to monitoring and warnings for resource usage it is useful to collate reports and statistics over time, so that trends emerge.*

What long term statistical reporting facilities exist for the records system for the analysis of resource utilisation under **N12.5.2**?

#### N12.5.6

*Requirement **R2.4.7** stipulates the use of an external error log. Different types of records system use different error logs.*

Describe the error log used by the records system under **R2.4.7**. How is it accessed and used, and what mechanisms exist for warning technical administrators when the records system fails to perform a function?

#### N12.5.7

*Requirement **R8.4.15** requires that users authorised to receive alerts for an aggregation or record whenever a disposal action has not been carried out and confirmed by the due date. Different types of records system use different alerting mechanisms.*

Describe the alert mechanism used by the records system under **R8.4.15**. How can authorised users receive the alert, and what mechanisms exist for consolidating alerts, as described in the rationale to **R8.4.15**?

#### N12.5.8

*From time to time the records system may be audited. Auditors may wish to check (this list is not exhaustive):*

- *That only appropriate users and groups have access to the records system;*
- *That all appropriate users and groups have access to the records system;*
- *That appropriate security controls and access controls have been set;*
- *That users are not accessing records and other entities they are not permitted to access;*
- *That the classification service configuration is appropriate for the business;*
- *That the disposal scheduling service configuration is appropriate for the business;*

- *That all relevant records are being captured by the records system;*
- *That records are being placed into the appropriate aggregations;*
- *That records are being classified correctly;*
- *That users are not inappropriately overriding the default disposal schedules of records;*
- *That no records or other entities are being deleted from the records system, outside the disposal process;*
- *That disposal periods are being monitored and disposal due dates are being met;*
- *That confirmations occur within the disposal due date and records are not overdue for disposal;*
- *That record content is being disposed of correctly; and*
- *That copies of record content are being deleted from secondary sources within the organisation immediately following or in concert with the formal disposal of the record.*

What facilities exist for auditing the records system and how should this be carried out?

## 12.6 Non-functional Requirements for Portability

### N12.6.1

*The records system may operate on many platforms or just one. It may have a limited number of server configurations but support several different clients.*

What operating systems and platforms does the records system use, and which parts of the records system including server and client based system modules are deployed onto which technologies?

### N12.6.2

*As described in **3. User and Group Service**, an MCRS may utilise a popular directory service, such as an LDAP directory, and provide a wrapper for capturing historical data about users and groups, or it may provide its own user and group management service.*

Which directory services does the records system interface to, if any, and how is historical information about users and groups preserved as entities within the user and group service?

### N12.6.3

*Many records systems utilise third-party software components such as database technologies and search engines. Where these are integrated into the product they have the advantage of reuse but the issues associated with managing independent development cycles.*

What OEM, third-party or open source system services does the records system incorporate?

### N12.6.4

*Many records systems provide interfaces and API sets to other applications.*

What other business systems can the records system integrate to, if any, what API sets are available, and what technologies do they support?

### N12.6.5

*Platform support may place limitations on metadata and templates. For example, MoReq2010® does not restrict the maximum length of a text field, however this may be restricted by database table sizes.*

*The records system may adopt the model metadata service (see **7. Model Metadata Service**) or it may implement its own approach to metadata.*

What approach is used by the records system to managing metadata and what is the impact of this approach over:

- The number of contextual metadata elements that can be applied to an entity of any entity type?
- The number of contextual metadata element definitions that can be included in a template?
- The use of templates?
- The maximum length of a metadata field?
- The datatypes supported by the records system?

## 12.7 Non-functional Requirements for Security

### N12.7.1

***R3.4.1** specifies that the MCRS must only by accessed by authenticated users. The MCRS may support one or a number of commercial or proprietary authentication services, or the authentication service may be built in.*

On each of the platforms listed under **N12.6.1**, and for each of the directory implementations listed under **N12.6.2**, which authentication technologies does the records system support?

### N12.7.2

*Depending on the records system's implementation of the model role service, there may be particular restrictions that apply to roles and access control.*

*The records system may adopt the model role service (see **4. Model Role Service**) or it may implement its own approach to access control. If the records system uses its own approach to access control then it might not have the same level of granularity as the MoReq2010® model role service.*

How does the records system implement internal access control and what constraints does it impose on:

- The roles that are predefined and fixed by the records system?
- The number of additional roles that may be defined?
- The function definitions that may be included in roles?
- The entities that have access control lists?
- Inheritance and other features of access control entries?

### N12.7.3

*Authentication and access control are of little value if the information stored by the records system can be accessed directly.*

What mechanisms does the records system rely on for restricting access to its stored data?

### N12.7.4

*Similarly, when the different components of the records system communicate with one another either internally, for example between client and server, or externally, for example with another business system, their communications should be secured to avoid snooping and man-in-the-middle attacks.*

What technologies are used to ensure that communication between the components listed in **N12.6.1**, **N12.6.2**, **N12.6.3** and external systems listed in **N12.6.4** is secured?

#### N12.7.5

*Suitable controls should be embedded into the records system, and/or its operating environment, to prevent it being exploited by viruses, Trojan horses, and other malicious code. For example, a records system may be vulnerable to SQL injection attack through the metadata elements of entities.*

What anti-virus and other security strategies are integrated into the records system or are recommended as part of the records system's normal operational environment?

#### N12.7.6

*The records system may have been designed and implemented to meet various well known security standards and penetration tests. Standards such as ISO 27000 do not assess individual products alone but are made in relation to an organisation's whole security strategy and practice.*

What types of secure environments is the records system designed for, which national or international regulations and under what jurisdictions?

(The product or an existing site installation may have received an independent security assessment or rating.)

## 12.8 Non-functional Requirements for Privacy

#### N12.8.1

*The records system may incorporate mechanisms for addressing important issues for privacy, such as the protection of personal data, but also, and especially in government contexts the individual's right to access public information.*

How does the records system address the following issues, where applicable:

- Data protection?
- Freedom of information?

(Indicate the jurisdictions under which individual responses apply.)

#### N12.8.2

*The records system may have been assessed for its relevance to particular privacy laws, such as the European data privacy directive (95/46/EC). Where this has occurred, the records system may incorporate special features, such as restrictions on which data is moved between different internal data stores. This is particularly relevant if the deployment of the records system crossed international boundaries.*

Has the records system been specifically designed for conformance with any national, international, or good practice privacy guidelines or regulations?

(The product or an existing site installation may have received an independent assessment or rating against specific regulations in a particular jurisdiction.)

## 12.9 Non-functional Requirements for Usability

**N12.9.1**

*The records system must be accompanied by user and technical documentation to facilitate its assessment by a MoReq2010® accredited test centre.*

What user and technical documentation is available to users and technical administrators of the records system?

**N12.9.2**

*Users generally need training and education to utilise a records system effectively. This is especially true of specialised users such as technical administrators, security managers, auditors and, most importantly, records managers.*

What training do users with different degrees of specialisation require to use the records system effectively, and what training courses, tutorials and other educational and learning resources are available for general users and specialised users?

***Other non-functional requirements for usability are addressed by the non-functional requirements for plug-in modules in the MoReq2010® 100. Interface Series.***

## 12.10 Non-functional Requirements for Accessibility

**N12.10.1**

*The records system may have been assessed under the W3C WAI Web Content Accessibility Guidelines (WCAG). These guidelines provide a rating of A (lowest), AA or AAA (highest).*

Has the records system been independently assessed and rated against WCAG?

(Include details of the assessment, who undertook it, and the ratings achieved.)

***Other non-functional requirements for accessibility are addressed by the non-functional requirements for plug-in modules in the MoReq2010® 100. Interface Series.***

## 12.11 Non-functional Requirements for Availability

**N12.11.1**

*Records system availability is dependent on whether the records system and other necessary support systems are running. Sometimes these systems must be taken offline for planned maintenance and upgrade. On other occasions they fail.*

For each of the typical deployments described in **N12.3.2**, what is the anticipated records system availability as a percentage or ratio of records system uptime against downtime over the course of a calendar year?

**N12.11.2**

*Where a service level agreement exists covering records system availability some mechanism must also be provided or suggested to allow this to be measured.*

Where recommended for inclusion in a service level agreement, how should the percentages or ratios given under **N12.11.1** be measured and calculated for the records system, what tools should be used, and are they available in the records system or from a third-party?

(Provide a description where a different form of service level agreement is recommended to ensure a minimum level of records system availability.)

#### N12.11.3

*Some administrative operations, such as backup require that the records system be taken offline.*

What administrative operations require that the records system be taken offline?

#### N12.11.4

*As systems grow they take longer to perform certain functions, such as backup.*

How long would a typical backup take for each of the available technologies (see **N12.13.1**), given the typical deployments described in **N12.3.2**, and how will this vary given the growth scenarios in **N12.4.2**?

(Take into consideration full backups and incremental backups.)

#### N12.11.5

*Organisations must work to a schedule when administering their records systems. See also **N12.14.4**.*

For each of the responses to **N12.11.1** what are the recommended planned backup, maintenance or upgrade windows that should be reserved, per day, week and month throughout the year?

#### N12.11.6

*Hosted records systems require additional consideration as they are limited by their host's availability.*

If the records system is a hosted system, what additional availability constraints and guarantees should be provided by the host system?

## 12.12 Non-functional Requirements for Reliability

#### N12.12.1

***R2.4.7** requires that functions be performed atomically and that no function should be only partially successful. Atomic functions are required to ensure the integrity of the data if the MCRS fails during an operation. If the function is not successful then it should be rolled back.*

How does the records system support atomic operations, and how is the integrity of the records system ensured if a transaction fails or the records system fails prior to its completion?

#### N12.12.2

*The design of the records system plays a significant role in its reliability and its ability to withstand error conditions.*

How does the architecture of the records system affect its reliability?

#### N12.12.3

*In addition to its architecture under **N12.12.2**, the methodology used to develop the records system, including designing, checking and unit testing, all play a part in the resulting quality of the final product.*

What quality assurance controls are used in the manufacture of the records system to ensure its correctness?

#### N12.12.4

*The records system may have been previously assessed and given a rating for the mean time between failures or measured against some other reliability or availability instrument.*

Has the records system been independently verified, benchmarked, or rated in a previous installation for its reliability or availability?

(Include details of the assessment, who undertook it, and the ratings achieved.)

#### N12.12.5

*Many records systems are built using a service orientated architecture (or other resilient architecture) and some parts of the records system may be redundant allowing individual services to fail and be restarted while the records system remains operational.*

What parts of the system, listed in the responses to **N12.6.1** and **N12.6.3** can fail or be stopped and restarted while the rest of the records system remains operational?

#### N12.12.6

*Gracefully handling system failure means that all processes are shut down in the right order to preserve the integrity of the system and allow it to be restarted and continue operation once normal operating conditions have been achieved.*

What hardware, software and systems support is required within the operating environment of the records system to allow it to shut itself down gracefully in the event of a power outage or in response to another external threat?

### 12.13 Non-functional Requirements for Recoverability

#### N12.13.1

*Different records systems adopt different backup strategies for redundancy of data to ensure business continuity in the event of data loss or hardware, software or system failure. Different strategies may have different advantages, such as speed, or disadvantages, such as cost. Backup strategies also may be partially dependent on the backup media used. There are many different types of backup media available, for example, magnetic discs, magnetic tapes, optical media, cloud storage, etc.*

What backup strategies does the records system support, what are their relative advantages and disadvantages, and which are recommended for the usage scenarios in **N12.11.4**?

#### N12.13.2

*All organisations should develop a business continuity plan. This should include the time required to restore systems including the records system, in the event of a system failure.*

For each of the strategies listed under **N12.13.1** and for each of the usage scenarios in **N12.11.4**, what is the recommended time required to restore the records system, if:

- Only the data needs to be recovered; or
- The records system needs to be rebuilt and the data recovered?

#### N12.13.3

*The business continuity plan should contain step by step instructions on how to recover the records system, and at least one copy should be stored outside the records system.*

For recovering the data of a records system under **N12.13.2**, what approaches are required for each of the backup strategies listed under **N12.13.1**, and what critical considerations do they raise or address?

#### N12.13.4

***N12.12.1*** *indicates that the records system should support atomic transactions. Where these are supported and stored in a transaction log it may be possible to restore a system up to the point where it fails.*

When restoring from backup under **N12.13.2**, does the records system support incremental backup and during data recovery is it able to roll forward to the transaction immediately before the failure?

(Describe any alternative design approach adopted.)

#### N12.13.5

*Not all records systems rely on backup and recovery to provide redundancy. Some can be set up to mirror data and to fail over to a standby system or site. Others rely on redundancy at the level of individual drives which can be "hot swapped" while the records system stays online.*

Other than backup and recovery, what other mechanisms does the records system employ to ensure business continuity?

(Describe the mechanisms available.)

#### N12.13.6

*Where records systems are built from many different integrated system components, some provided by different suppliers, this may have an affect on how the records system is managed and how it is backed up and recovered. For example, the database may be backed up using one technology while a content store is backed up using a different technique. They may even be backed up at different times, which can create a synchronisation issue in putting them back together if they both need to be restored on system failure.*

What is the impact of the OEM, third-party and open source system services, listed under **N12.6.3**, on backing up and restoring the records system or on otherwise providing for data and system redundancy and failover?

(Explain in particular how any synchronisation issues are managed if different parts of the records system are backed up and restored separately.)

## 12.14 Non-functional Requirements for Maintainability

### N12.14.1

*Most records systems support major versions, minor versions and maintenance versions (otherwise called new versions, service packs and patches or builds). New features are usually introduced with major releases, enhancements with minor releases and bug fixes with maintenance releases.*

How are the different versions of the records system denoted, what types of upgrade do the different types of version cover, and what is the supplier's release cycle for each type of version?

### N12.14.2

*Even when the upgrade is free there will be costs associated with moving from one version of a product to the next. Using the information from **N12.14.1** and **N12.11.5** it should be possible for an organisation to schedule planned upgrades to its records system.*

For each of the releases listed under **N12.14.1**, what recommended upgrade planning should be undertaken by the organisation using the records system?

### N12.14.3

*From time to time it is necessary to fix bugs and other issues outside the scheduled maintenance and upgrade windows, for example, if an immediate security issue is discovered and publicised which causes a serious threat to the organisation's records system, or if the system becomes unstable in operation for any reason.*

What is the supplier's policy for emergency product support and issuing hot fixes or patches for critical errors?

### N12.14.4

*Where records systems incorporate different parts sourced from different suppliers this may have an impact on how product releases are made. This is especially true if the third-party system components are independently sourced by the organisation and do not come directly from the supplier of the records system. Even records systems that are substantially build by a single supplier may be dependent on different operating environments from different sources.*

How are different OEM, third-party and open source system services, listed under **N12.6.3**, managed in terms of the different releases of the records system listed under **N12.14.1**, and their own separate product development and release cycles?

### N12.14.5

*Not all records systems are maintained by the organisation, some are hosted by the supplier or a third party. Where this occurs, regular updates may be made to the records system at scheduled times over which the organisation has little control, especially where the organisation shares a multi-tenanted hosting system.*

If the records system is hosted, what are the typical arrangements made for upgrading the hosted records system, how and when customers are notified, and the potential impact on customer organisations, for each of the different types of upgrade listed under **N12.14.1**?

**N12.14.6**

*Upgrading records systems and adding additional features also requires additional user education, to learn the new features included in the release and to change old habits.*

What release notes are issued with each release, and what are the retraining and educational requirements usually associated with the different releases of the records system listed under **N12.14.1**?

## 12.15 Non-functional Requirements for Supported

**N12.15.1**

*A records system must be maintainable, but it must also be currently supported by the supplier.*

When were the last releases of each of the versions of the records system noted under **N12.14.1**, and what were the versions and release dates?

(Include the date the list was compiled as the latest releases will change over time.)

**N12.15.2**

*It is also important to know as much as possible about the supplier's future plans for the records system, although some of these may be trade secrets and they may also be subject to change in response to new technologies and priorities.*

Is there a product roadmap for the records system, what periods does it cover (for example, the next 18 months, 3 years and 5 years), how often is it updated, and how is the product roadmap shared with customers and prospective customers?

**N12.15.3**

*The organisation may also wish to know how it can request new product features and influence their priority and development in the records system.*

How are customers involved in requesting, and prioritising new features?

(Provide an example of a feature developed as a result of a customer request.)

**N12.15.4**

*Issues of all types should be able to be reported to the supplier and fixed within an agreed timeframe. The supplier's policy on issuing patches for emergency support has already been listed under **N12.14.3**.*

What levels of support are available for the records system, can it include a 24 x 7 live support line for reporting critical issues, and how do customers and prospective customers learn about the support processes available and how to use them?

**N12.15.5**

*The supplier should make provision for issue ticketing and tracking, especially when issues are raised by organisations through the support process.*

Do organisations have a view of the issues they raise through the supplier's service desk so that they may monitor the progress of their issues?

**N12.15.6**

*Organisations should seek to understand how quickly they may expect an issue to be resolved.*

Are the categories of issues used, under **N12.15.5**, and the average resolution times across each category over the last year available to customers and prospective customers?

**N12.15.7**

*Many suppliers now support active user groups, on line forums, including support forums, and conferences.*

Apart from the information already listed in this section, what other opportunities exist for organisations to engage with each other and with the supplier of the records system?

## 12.16 Non-functional Requirements for Warranted

**N12.16.1**

*It is important to understand what assurance the supplier provides in terms of quality and performance in relation to the records system.*

Does the supplier warrant the records system and, if so, what parts of the records system are covered for what types of issues?

**N12.16.2**

*The organisation may also wish to agree on a service level agreement covering the records system and aspects such as performance, availability, and so on.*

Does the supplier enter into service level agreements with its customers and, if so, what aspects of the non-functional requirements of records systems do they cover, and are they the same or individual for each agreement?

**N12.16.3**

*The supplier may have standard terms and conditions for customers of its records systems. It should be noted that even open source software has licensing conditions associated with it.*

Are there standard terms and conditions available to customers or prospective customers, or are they negotiated individually for each contract?

**N12.16.4**

*It must be possible for the organisation to obtain access to the source code for the records system should the supplier ever go out of business.*

Is the records system an open source software application and if not what other means may be used to protect the customer's access, such as lodging a copy of the source code in escrow with a neutral third-party?

**N12.16.5**

*It is also possible to place data into escrow, especially for hosted services.*

Does the records system support data escrow?

#### N12.16.6

*Where a records system is hosted or different parts of the system are provided by different suppliers, this can have unexpected consequences on the main supplier's terms and conditions, licence agreement, service level agreements, and so on.*

What is the impact of different parts of the system being provided by different suppliers, under **N12.6.3**, or hosted under **N12.11.6** and **N12.14.5**, to the responses to **N12.16.1**, **N12.16.2**, **N12.16.3**, **N12.16.4** and **N12.16.5** above?

## 12.17 Non-functional Requirements for Compliance

#### N12.17.1

*The records system may already comply with national or international standards on records management, or related disciplines such as content management, document management, etc.*

Apart from MoReq2010®, what other records management, or related, standards and specifications does the records system comply with, and has this been independently verified?

(For each such independent verification include details of the assessment, who undertook it, and the grade achieved.)

#### N12.17.2

*The records system may comply with other regulatory and legislative frameworks outside the immediate field of records management, at either the national or international level. For example, the records system may be compliant with the European Markets in Financial Instruments Directive (2004/39/EC) also known as "MiFID".*

Apart from those standards listed under **N12.17.1**, what other national or international standards or regulations does the records system comply with, and has this been independently verified?

(For each such independent verification include details of the assessment, who undertook it, and the grade achieved.)

#### N12.17.3

*The records system may comply with particular technical standards, such as HTTPS, HTML5, MD5, XML, and so on.*

What technical standards does the records system use and comply with?

#### N12.17.4

*It is useful for an MCRS to provide a high level of precision for timestamps generated under **R2.4.27**. Millisecond or better precision is useful for maintaining the exact order in which events occur in high throughput systems.*

What precision does the records system use for timestamps?

#### N12.17.5

*It is important that, under **R2.4.24**, an MCRS does not generate the same system identifiers for entities as other MCRS solutions. The algorithm used to generate universally unique identifiers must*

*be suitable for creating large numbers of UUIDs without repetition, pattern or overlap with other systems.*

What algorithm does the records system use for generating UUIDs?

**N12.17.6**

*Some records systems have configuration options that allow some or all of their MoReq2010® compliant functionality to be switched off or replaced by non-compliant functionality.*

*Where provision for this is included in the configuration options of the records system, especially for a MoReq2010® certified product, it is important that safeguards are put in place to ensure that it does not happen inadvertently. Even thought the consequences of particular installation options may be clearly explained in technical documentation and when the records system is installed, a different technical administrator may unknowingly change the configuration of the records system at a later time with unintended consequences.*

*For this reason, MoReq2010® includes an option for compliance reporting under* ***R2.4.5*** *that allows users to check the current MoReq2010® compliance status of the records system at any time during routine operation.*

How does the records system support MoReq2010® compliance reporting under **R2.4.5**, and how does it check the current status of the records system when reporting, to ensure that it has not been reconfigured in a non-compliant way?

# 13. Glossary of Terms

| Term | Explanation and relationship to general concepts |
|---|---|
| **Access** | (*verb*) To interact with an **information system** so as to perform **functions** and to **browse** and **inspect** its **entities.**<br>(*noun*) The degree of functionality a **user** is permitted to have with an **information system** described as a "level of access". |
| **Access control** | (*concept*) The concept of managing **users' access** to **entities** and to **functions** in an **information system.** |
| **Access control entry** | (*data structure*) An individual entry within an **access control list** that **grants** one or more **roles** to a single **user** or a **group.**<br>See also **access control list**. |
| **Access control list** | (*data structure*) A list of **access control entries** associated with an **entity** that defines which **users** may **perform functions** on the **entity** by assigning **roles** to **users** and **groups**.<br>See also **access control entry**. |
| **Accessibility, Principle of** | (*concept*) A **non-functional** aspect of an **information system** that describes the extent to which it supports users with different capabilities and learning rates, including those with specific disabilities.<br>(disambiguation) The term "accessibility" is derived from a broader meaning of the word "**access**" than that used by **MoReq2010®**. |
| **ACE** | (*acronym*) An **access control entry**. |
| **ACL** | (*acronym*) An **access control list**. |
| **Active entity** | (*noun*) An **entity** that has been **created** in an **MCRS** and has never been **deleted** or **destroyed**.<br>See also **residual entity**. |
| **Activity** | (*noun*) An organised **process** designed to achieve an outcome. Both human and computer activities are common within **information systems**.<br>See **business activity** and **scheduled activity**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Add** | (*operation*) The **function** of including an **entity** within a set of entities, usually by moving it from elsewhere. For example, adding a **record** to an **aggregation**.<br><br>See also **move** and **remove**. |
| **Administrative role** | (*noun*) A **role** that applies uniformly, through **inheritance**, to all the **descendants** of an **entity**, or if applied to a **service** to all the entities in the service. By comparison, the inheritance of **non-administrative roles** is selective.<br><br>See also **non-administrative role**. |
| **Administrator** | (*noun*) A technical administrator of an **MCRS** is a person charged with managing the infrastructure and technical environment that support its operation. The technical administrator will have access and influence over external aspects of the MCRS such as its physical data storage areas and the **error log**, but is not necessarily a **user** of the MCRS.<br><br>(*disambiguation*) The term "administrator" may also be used in a non-specific way (though not in **MoReq2010®**) to describe a **user** of the MCRS with:<br>• A generally high level of **access** to **entities** and **functions**;<br>• The task of configuring such entities as **classes**, **metadata element definitions**, **templates** or **roles**;<br>• Specific **records management** responsibilities such as the **transfer** or **destruction** of **records**; or<br>• One or more **administrative roles** that the user has been **granted**.<br><br>See **administrative role**. |
| **Aggregate** | (*verb*) The activity of bringing together entities with shared characteristics. Specifically within **MoReq2010®** to **create** or **move records** into **aggregations**.<br><br>See also **add**. |
| **Aggregation** | (*entity*) Aggregations of records are accumulations of related record entities that, when combined, may exist at a level above that of a single record.<br><br>Aggregations of records may reflect relationships such as shared characteristics or attributes, or the existence of sequential relationships between related records.<br><br>(Adapted from ISO 16175-3:2010, 2.3.1) |

| Term | Explanation and relationship to general concepts |
|---|---|
| | See also **child aggregation**, **parent aggregation** and **root aggregation**. |
| **Alert** | (*noun*) A **disposal** alert is raised automatically by an **MCRS**, and sent to **users** authorised to receive it, whenever a record has become due for **disposal**, and has exceeded its subsequent **confirmation period** without receiving the necessary **confirmation** of disposal. **MoReq2010®** allows each MCRS to implement its own alert technologies and solutions.<br>(*verb*) The act of raising an alert and sending it to users authorised to receive it. |
| **Ancestor** | (*noun*) In a **hierarchical** structure, starting with a given **entity**, every other entity that can be reached by tracing only a unidirectional path from **child** entities to their **parents**.<br>See also **child**, **descendant** and **parent**. |
| **Anonymise** | (*verb*) A process of obscuring or concealing sensitive information so that the source cannot be determined.<br>See also **secure**. |
| **Application** | (*noun*) Applications are any computer software. They may include but are not restricted to **products** or **information systems**. |
| **Architecture** | (*noun*) The layout and structure of an information system, particularly highlighting the design intent of each part, their different construction, and the relationships between them.<br>See **service based architecture**. |
| **Assign** | (*operation*) Giving a value to a **metadata element**, particularly by using it to **associate** one **entity** with another.<br>See also **associate** and **modify**. |
| **Associate** | (*operation*) Form a **relationship** between two **entities**, for example, associating a **record** with an **aggregation**, usually by modifying the **metadata** of one or both to hold a **reference** to the other.<br>See also **assign**, **modify** and **remove**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Atomic** | (*concept*) The concept that a particular unit of information should be managed holistically. For example, a **record** is considered to be atomic even though it is made up of different **entities**, such as **components**. The record is either wholly **active** or **destroyed** in its entirety, it cannot be partially destroyed.<br>Atomicity in **MoReq2010®** also extends to the way an **MCRS performs functions**. They must be wholly performed, never partially performed. This is important for maintaining internal consistency and stability, and when the MCRS is **restored** from **backup** |
| **Audit trail** | (*noun*) In a traditional **business system**, a centralised log of all, or significant, system activity. An **MCRS** keeps a trail of its activities as a sequence of **events**, which may be viewed across the system as a whole, but are more commonly accessed as an **event history** for an individual entity.<br>See **event history**. |
| **Authenticate** | (*operation*) Operation to check the identity of a user usually by asking the user for a previously agreed password. |
| **Authenticity, Principle of** | (*concept*) Along with integrity, reliability and usability, one of the central characteristics of a record according to ISO 15489.<br>An authentic record is one that can be proven to be what it purports to be.<br>(Adapted from ISO 15489-1:2001, 7.2.2)<br>See also **integrity**, **reliability** and **usability**. |
| **Authorised user** | (*noun*) An **authenticated user** with the **authority** to **perform** a **function**. |
| **Authority** | (*concept*) Having the ability to **perform** a particular **function**, especially with respect to a particular **entity**. Authority is given to **users** by **granting** them **roles** in different parts of the system or in respect to different **entities** using an **access control list**. |
| **Automatic** | (*adjective*) An **operation** or **function** performed by the **system** in accordance with its own internal processing rules. There is no **user** or **manual** intervention.<br>See also **manual**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Automatic destruction** | (*operation*) The **content** of a **component** of a **record** may be **destroyed automatically** when the record reaches its **disposal action due date**. Whether or not the content is destroyed automatically or destroyed following **disposal confirmation** is determined by the nature of the content and the design of the **MCRS**. All components have a **flag** indicating whether or not their content is to be destroyed automatically. |
| **Availability, Principle of** | (concept) A **non-functional** aspect of an **information system** that describes the period during which the system is fully operational and stable, especially by contrast with how long the system spends partially operational or offline. This is sometimes expressed as a ratio or percentage. |
| **BCS** | (*acronym*) A **business classification scheme**. |
| **Backup** | (*verb*) To take a redundant copy of the data of an **information system** so that it may be **restored** following a system failure or disaster.<br>(*noun*) A redundant copy of the data of an information system held in secure storage so that it may be later used to restore the system, if required.<br>See also **restore** and **recover**. |
| **Boolean** | (*concept*) A **value** with only two possible states: true or false.<br>See also **flag**. |
| **Boolean operator** | (*concept*) An operator used to logically combine **Boolean** values so as to calculate a single result. There are only three basic Boolean operators AND, OR and NOT from which all other Boolean operations can be derived.<br>See also **Boolean**. |
| **Bottom-up destruction** | (*concept*) An approach to managing **disposal** where the **retention period** of each **record** is assessed individually and they may be separately disposed of. The disposal of **aggregations** of records is then tied by processing rules to their contents, so that when all the records have been destroyed, the aggregation is **automatically** destroyed. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Browse** | (*operation*) Discover **entities** by exploring their **relationships** with other entities. For example, starting from a **record** discovering its **parent aggregation** using the **parent/child relationship**, or by using other relationships discovering its **components**, **class**, **disposal schedule**, associated **disposal holds**, and so on.<br>(*disambiguation*) Browsing in an **MCRS** should not be confused with using a **web browser**. |
| **Business classification** | (*verb*) See **classification**. |
| **Business classification scheme** | (*noun*) See **classification scheme**. |
| **Business activity** | (*noun*) Activity carried out by a business so as to constitute or fulfil an overarching business function, this can include any of the areas of activity that an organisation might engage in, or be required to undertake by external regulatory or other controls.<br>"An analysis of business activity and processes will provide an understanding of the relationship between the organisation's business and its records." (ISO 15489-2:2001, 3.2.3) |
| **Business function** | (*noun*) An area of business activity pursued by an organisation, usually related to the purpose or mission of the organisation and the execution of its business strategy and policies. **ISO 15489** describes business functions as supporting the pursuit of an organisation's goals and strategies. (ISO 15489-2:2001, 4.2.2.2)<br>(*disambiguation*) A business function used in **classification** should not be confused with **performing** a **function**. |
| **Business transaction** | (*noun*) A discrete stage or "constituent step" (ISO 15489-2:2001, 4.2.2.2) in a business activity for which an evidential record is kept. The record might include information related to:<br>• The business transaction that was undertaken;<br>• When it occurred; and<br>• Who participated. |
| **Business system** | (*noun*) Any **information system** used as part of conducting business. For example, a financial management system. |
| **Cancellation** | (*operation*) See **disposal cancellation**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Capture** | (*concept*) An activity leading up to the **creation** of a **record** in an **MCRS**. Other terms may also be used for this, such as **declaring** a record. Often this is dependent on the **user's** perception as to whether the content of the record must be moved into a new storage facility (capture), or it can be made a record **in place** (declare).<br>See also **declare**. |
| **Cascade** | (*concept*) The concept that a change to an entity will have an impact on its **descendants**. For example, if the descendants of an **root aggregation**, both **child aggregations** and **records**, inherit their **classification** from their **parent aggregation**, then changing the root aggregation's class will have a cascading effect from parent to child, resulting in all the descendants of the root aggregation being **reclassified**.<br>See also **hierarchical**. |
| **Certification** | (*adjective*) The act of formally acknowledging that a **records system** fully meets the requirements of **MoReq2010®** in respect to the core services and nominated modules. Only the **DLM Forum®** may certify a records system, subject to testing by an accredited **MoReq2010®** test centre.<br>See also **testing**. |
| **Child** | (*noun*) An **entity** that is part of a set of entities that belong and are subordinate to another **parent** entity. A parent entity may have multiple children, but each child entity has only one parent. The link between parent and child entities is known as a **parent/child relationship**.<br>See also **ancestor**, **descendant**, **parent** and **parent/child relationship**. |
| **Child aggregation** | (*noun*) Any **aggregation** that is not a **root aggregation**.<br>See also **parent aggregation** and **root aggregation**. |
| **Class** | (*entity*) A unit of **classification** that may be **associated** with an **aggregation** or a **record**.<br>In **MoReq2010®** classes always have a **default disposal schedule** which is inherited by any record they classify, in accordance with the principle from **ISO 15489**, that "Classification of business activities acts as a powerful tool to assist the conduct of business and in many of the processes involved in the |

| Term | Explanation and relationship to general concepts |
|---|---|
| | management of records including… determining appropriate retention periods and disposal actions for records." (ISO 15489-1:2001, 9.5.1)<br>See also **classification**. |
| **Classification** | (*operation*) The act of **associating** a **class** from a **classification scheme** to an **aggregation** or **record**.<br>(*noun*) A **class** applied to an aggregation or record.<br>See also **class**, **default classification** and **reclassification**. |
| **Classification scheme** | (*noun*) Representation of the business functions, business activities and business transactions of the organisation as a set of discrete **classes** that can be associated with **records** and **aggregations** of records.<br>"Classification systems may be derived from analysis of business processes to ensure that the records and their metadata descriptions accurately represent the business processes that created them." (ISO 15489-2:2001, 4.2.2.2) |
| **Classification service** | (*service*) A **logically** separate **service** within an **MCRS** operationally responsible for maintaining **class** entities within a **classification scheme**. |
| **Clear** | (*operation*) When used in conjunction with a **flag** or **Boolean value**, meaning to assign it the value "false".<br>See also **flag**. |
| **Close** | (*operation*) The **function** of closing an **aggregation** so that it can no longer accept additional **children** that may be **moved** or **created** in it. A **user** cannot close an **aggregation** unless all of its **child aggregations** are also previously or simultaneously closed.<br>See also **open**. |
| **Closed** | (*noun*) For aggregations, the state of having been closed. A **child aggregation** or **record** may be moved out of a closed aggregation but no additional child aggregations or records may be moved or created in it. |
| **Code** | (*datatype*) A **metadata element** that can only hold one of a limited set of predefined **values**.<br>See also **datatype** and **value**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Column** | (*noun*) Where the **metadata elements** or properties of **entities** are arranged into **tables**, each **row** of the table usually contains the metadata of a single entity, while each column contains the values for each entity of a single metadata element.<br>See also **row** and **table**. |
| **Comment** | (*metadata*) An additional notation, usually provided by a **user** but sometimes constructed **automatically**, that provides explanatory and later historical detail to the purpose, motivation or outcome of taking a significant action.<br>See also **event comment**, **export comment** and **last review comment**. |
| **Common off-the-shelf** | (*adjective*) A **product** that can be installed and run in a variety of different environments with minimum of reconfiguration. COTS products may be differentiated from other custom **applications** that are purpose built for a specific environment or situation. **MoReq2010®** applies equally to both COTS products and individual site installations.<br>Also referred to as **COTS**. |
| **Completeness, Principle of** | (*concept*) When applied to the **component** of a **record** ensures that collectively the **content** of the record comprises a whole record ensuring the **integrity** of the record.<br>See also **destructibility**, **discreteness** and **immutability**. |
| **Complex search** | (*noun*) A search that is made by chaining together several search queries into a compound search query. Complex searches are necessary because entities are related to one another with sometimes complex relationships.<br>See also **search**. |
| **Compliance, Principle of** | (concept) Compliance is a **non-functional** aspect of a **records system** that assesses its suitability within a particular industry, or legislative jurisdiction, by assessing its support for and adoption of various standards and regulations. Standards and regulations may apply to technologies, obligations, policies, rights, communication instruments, information formats and the processing rules implemented by a records system. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Compliant** | (*adjective*) A statement of the compliance status of a **system** or **process**, used especially to indicate that a **records system** complies with the **MoReq2010®** specification.<br>See also **certified** and **MoReq2010® compliant records system**. |
| **Component** | (*entity*) A part of a **record** that represents a discrete item of **content**. For **completeness** a **record** including all its components and their content must be managed **atomically**. |
| **Component content** | (*noun*) The actual item of record, whether it be a physical object or a digital sequence. All the other **entities** in an **MCRS** are purely representational, holding **metadata** related to, or extracted from, the content. |
| **Confirmation** | (*operation*) See **disposal confirmation**. |
| **Confirmation period** | (*noun*) The period between the **disposal action due date** and the **disposal confirmation due date**, in which a **user** must ensure that the **disposal** of a **record** has been carried out and confirm it with the **MCRS**. The disposal of a record is frozen pending either its **confirmation** or **cancellation**.<br>See also **disposal cancellation**, **disposal confirmation** and **disposal confirmation due date**. |
| **Content** | (*noun*) See **component content**. |
| **Contextual metadata** | (*noun*) Metadata that is not mandated by **MoReq2010®** but is created within an **MCRS** in a local context to support the local business needs and operations of an organisation. |
| **Contextual metadata element definition** | (*entity*) The definition of a **contextual metadata** element. Contextual metadata element definitions must be **exported** whenever contextual metadata is exported to ensure that an **MCRS** that imports the **export data** can interpret the metadata element and represent it correctly. |
| **Core service** | (*noun*) One of the nine services that is fundamental and necessary to an **MCRS** solution. |
| **Co-requisite** | (*noun*) A module for which additional requirements and consideration is necessary if it is implemented alongside another module.<br>See also **module**, **prerequisite** and **service**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **COTS** | (*acronym*) **Common off-the-shelf**, especially in relation to software products. |
| **Create** | (*operation*) The **function** of adding a new entity to an **MCRS**. |
| **CRUD** | (*acronym*) Create, read, update and delete. Often considered to be the four essential operations of a **system** in relation to its **data**.<br>(*disambiguation*) **MoReq2010®** has a different and more specialised definition of the terms "**create**", "update" and "**delete**". |
| **Data** | (*noun*) Any information stored in an electronic format or communicated electronically.<br>See also **export data**. |
| **Data streaming** | (*verb*) See **XML data streaming**. |
| **Data structure** | (*data structure*) Compound metadata consisting of more than one interrelated **metadata element** bound together into a structure to preserve their **relationship** to one another. Data structures are part of entities in the same way as simple metadata elements.<br>All data structures are provided as part of the **MoReq2010® information model** including the unique **system identifier** for each data structure. For example, the system identifier for the data structure "Access Control List" is "60124baa-2625-4795-bf14-7e67f2224ccf". |
| **Database** | (*noun*) A dataset usually divided in to **tables** of entities of the same **entity types**. The tables are often related together by storing **identifiers** in one column that refer to entities in another table.<br>See also **column**, **row** and **table**. |
| **Datafile** | (*noun*) A machine readable computer file containing **data** in any digital format. The term "datafile" has been coined specifically for use in **MoReq2010®** to avoid any ambiguities with other terms used in **records management**.<br>(*disambiguation*) The word "file" used in this context does not refer to an **aggregation**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Datatype** | (*noun*) An **XML** datatype used to define the characteristics of metadata elements. XML datatypes are used throughout **MoReq2010®** as this standardisation allows different **MCRS** solutions to share **metadata element definitions**. |
| **Date** | (*datatype*) A metadata element based on a day, month and year. A date does not include time or **time zone** information and is therefore only accurate to within 24 hours. |
| **Date/time** | (*datatype*) A metadata value based on a **date** and a time of day that can be modified by users and is therefore not as precise as an automatically generated **timestamp**. A date/time value does not include a **time zone**.<br>See also **originated date/time** and **timestamp**. |
| **Declare** | (*concept*) A related term to **capture** that describes the **user** action that may precede the creation of a **record** in an **MCRS**.<br>See also **capture**. |
| **De-duplication** | (*concept*) The practice of not storing or transmitting the same data more than once. De-duplication is of most importance to the **MoReq2010®** export process. When a number of entities are exported they may share data in common, for example the **class** of a **record**. If several records with the same class are exported together, then the class information will only be exported once. |
| **Definition** | (*entity*) See **function definition** and **metadata element definition**. |
| **Default** | (*concept*) An **entity** or **value** that is used whenever a replacement is not explicitly specified. For example, a record has a default **class** (**inherited** from its **parent aggregation**) unless it is **overridden** by applying a different classification directly to the record itself.<br>See **default classification**, **default disposal schedule**, **default language identifier** and **default value**. |
| **Default classification** | (*noun*) The **class** of a **child aggregation** or **record** that it automatically **inherits** from its **parent**, unless it is **overridden**.<br>See also **class**, **default** and **classification**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Default disposal schedule** | (*noun*) The **disposal schedule** of a **record** that it **automatically** inherits from its **class**, unless it is **overridden**.<br><br>See also **default** and **disposal schedule**. |
| **Default language identifier** | (*metadata*) The **language identifier** that is used for a **textual metadata element** unless a different language is specified when the metadata element is given a value.<br><br>See also **default** and **language identifier**. |
| **Default value** | (*metadata*) The **value** of a **metadata element** that is it is given **automatically** whenever the metadata element is instantiated. The default value is stored in the **metadata element definition**.<br><br>See also **default** and **value**. |
| **Delete** | (*operation*) Erase data, especially an entity, from an **MCRS** so that no trace remains. **MoReq2010®** only permits entities to be deleted if they have not been used. Once an entity has been used then it cannot be deleted and must be **destroyed**, leaving a **residual entity**.<br><br>There is an important distinction made between deleting an entity and destroying an entity. |
| **Descendant** | (*noun*) In a **hierarchical** structure, starting with a given **entity**, every other entity that can be reached by tracing all possible unidirectional paths from **parent** entities to their **children**.<br><br>See also **ancestor**, **child** and **parent**. |
| **Description** | (*metadata*) Optional extended information describing an **entity**.<br><br>See also **title**. |
| **Destroy** | (*operation*) A managed **process** where **active entities** are made into a **residual entities** through the controlled deletion of:<br>• Some of their **metadata**<br>• Some of the **events** from their **event histories**, and<br>• For records, their **content**.<br><br>There is an important distinction made between destroying an entity and **deleting** an entity.<br><br>See also **bottom-up destruction**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Destructibility, Principle of** | (*concept*) When applied to the **component** of a **record**, ensures that the **content** of the record can be permanently **destroyed** as a result of undertaking the disposal process in response to the record's **disposal schedule**.<br>See also **completeness**, **discreteness** and **immutability**. |
| **Detailed report** | (*noun*) A **report** based on a **search query** that lists **entities** and their **metadata**, usually as a **table**.<br>See also **report** and **summary report**. |
| **Directory** | (*noun*) An external **business system** common to modern organisational environments that maintains lists of **users** and **groups** and related **metadata**. Directories are intended deployed as a central resource allowing other business systems, including **records systems**, to interface to them and reuse this information. Common protocols include X.500 and LDAP. Directories do not typically keep good historical information about users and their group memberships and an **MCRS** has additional responsibilities to ensure that it is preserved where an external directory is used. |
| **Directory service** | (*noun*) See **directory**. |
| **Discovery** | (*concept*) In relation to a **user**, finding **entities**, their **metadata** and their **relationships** to other entities by **searching** and **browsing**. |
| **Discrete** | (*concept*) Individual, clearly discernable as being **logically** or **physically** separated from other entities or units of information. |
| **Discreteness, Principle of** | (*concept*) When applied to the **component** of a **record**, ensures that the **content** of the component is a single item that is separately identifiable from the content of other components and records.<br>See also **completeness**, **destructibility** and **immutability**. |
| **Dispose** | (*operation*) To execute a scheduled **disposal action** on a **record** which will result in either **destroying** it or passing it on to a new phase of its lifecycle. Records can only be disposed of in strict accordance with their **disposal schedule**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Disposal action** | (*noun*) An action that is taken to **dispose** of a **record** in response to the record's **disposal schedule**. Where a record is not **retained permanently** there are only three possible disposal actions:<br>• **Review**,<br>• **Transfer**, and<br>• **Destroy**. |
| **Disposal action due date** | (*metadata*) The date, marking the end of the **retention period**, when a **disposal action** should be carried out on a **record** in accordance with its **disposal schedule**.<br>See also **disposal confirmation due date**. |
| **Disposal cancellation** | (*operation*) Where a **disposal action** of **transfer** or **destroy** requires **confirmation**, it may be cancelled rather than being confirmed. Cancelling a disposal action involves assigning a new **disposal schedule** to the **record**. |
| **Disposal completion** | (*operation*) A **disposal action** of **review** must be completed. Reviewing a **record** involves assigning it a new **disposal schedule**, as part of a **review decision**. |
| **Disposal confirmation** | (*operation*) The **disposal actions** of **transferring** and **destroying records** must be confirmed by a **user**, except where records have **components** that are subject to **automatic destruction**. The user is confirming that the transfer was successful or that the **content** of the records has actually been destroyed.<br>See also **automatic destruction**. |
| **Disposal confirmation due date** | (*metadata*) The date, marking the end of the **confirmation period**, by which a **disposal action** requiring **confirmation** should have been carried out and confirmed by a **user**. If the disposal action is not confirmed by this date then the **MCRS** will raise an **alert**.<br>See also **alert** and **confirmation period**. |
| **Disposal hold** | (*entity*) A legal or other administrative order preventing the destruction of **records**. Despite their name, disposal holds do not prevent the **review** or **transfer** of records, however they do prevent them being **destroyed** by changing their **disposal action** to **retain on hold**. Disposal holds may be applied to whole **classes** and whole **aggregations**, as well as to individual records. |
| **Disposal holding service** | (*service*) A **logically** separate **service** within an **MCRS** operationally responsible for managing **disposal holds**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Disposal process** | (*noun*) The **process** by which a **record** is updated and managed, by checking for changes to its **disposal schedule** and **retention trigger**, calculating its **retention period**, and applying **disposal actions** when they are due. |
| **Disposal schedule** | (*entity*) A schedule detailing the lifecycle of a record and detailing:<br>• The **retention trigger** (used to determine the **retention start date**);<br>• The **retention period**;<br>• The **disposal action**; and<br>• The **confirmation period**.<br>See also **default disposal schedule**. |
| **Disposal scheduling service** | (*service*) A **logically** separate **service** within an **MCRS** operationally responsible for managing **disposal schedules**. |
| **DLM** | (*acronym*) Document Lifecycle Management (formerly "Données Lisibles par Machine").<br>See **DLM Forum®**. |
| **DLM Forum®** | (*noun*) A community of public archives and interested parties in archive, records, document and information lifecycle management throughout the European Union, and beyond. In February 2010 the DLM Forum® became a not for profit foundation and is now officially the DLM Forum Foundation.<br>See also **MoReq Governance Board**. |
| **Due date** | (*noun*) See **disposal action due date** and **disposal confirmation due date**. |
| **Duplicate** | (*noun*) An **entity** that is an exact copy of another entity. **MoReq2010®** allows **records**, and their **events** and **components** to be duplicated, so as to allow for example a duplicate of the same record to be placed into two different **aggregations**. Each duplicate will then follow its own separate lifecycle.<br>(*operation*) The **function** of duplicating a **record**. |
| **Electronic** | (*adjective*) Having a purely digital representation that is stored and transmitted electronically.<br>See also **physical**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Element** | (*noun*) See **metadata element**. |
| **Entity** | (*noun*) Entities represent individual and discrete units of information within an **information system**. In an **MCRS** each entity must be of a particular **entity type** and has some or all of the following:<br>• **System metadata**;<br>• **Contextual metadata**;<br>• **Access control list**;<br>• **Event history**;<br>The system metadata, and sometimes the contextual metadata, link the entity to other entities, forming **relationships**. |
| **Entity relationship diagram** | (*noun*) A technique for modelling relationships between units of data when describing an **information system**.<br>(*disambiguation*) The "entities" and "relationships" in an entity relationship diagram are not necessarily the same as **entities** and **relationships** as defined by **MoReq2010®**. |
| **Entity sub-type** | (*entity*) A specialised derivation of an **entity type** that may exhibit different behaviours, and have additional **metadata elements** and **functions**. For example, a **contextual metadata element definition** is a sub-type of a **metadata element definition**. **MoReq2010®** is fully extensible, allowing for additional entity types and entity sub-types to be introduced, as required.<br>See also **entity type**. |
| **Entity type** | (*entity*) A definition of an **entity**. The following entity types appear in the **MoReq2010® core services**:<br>• **Aggregations**,<br>• **Classes**,<br>• **Components**,<br>• **Disposal Holds**,<br>• **Disposal Schedules**,<br>• Entity Types,<br>• **Events**,<br>• **Function Definitions**,<br>• **Groups**,<br>• **Metadata Element Definitions**,<br>• **Records**,<br>• **Roles**, |

| Term | Explanation and relationship to general concepts |
|---|---|
| | • **Templates**, and<br>• **Users**.<br><br>All entity types and sub-types are provided as part of the **MoReq2010® information model** including the unique **system identifier** for each. For example, the system identifier for the entity type "Record" is "3ac228ef-c008-4524-9e41-5c4564eaa7f0".<br><br>See also **entity sub-type**. |
| **Entry** | (*data structure*) See **access control entry** and **metadata change entry**. |
| **Error** | (*noun*) A fault in a computer program or application which prevents the successful completion of a **function**.<br><br>See also **error log** and **extended error information**. |
| **Error log** | (*noun*) A log of **errors** that that have occurred within the **MCRS**. The error log is maintained externally to the MCRS and contains details and **extended error information** for every error. The error log is held externally for diagnostic reasons, to allow it to be accessed even when the MCRS has crashed or fails to start.<br><br>Also called an **external log**. |
| **Event** | (*entity*) An **entity** that is generated by **performing** a **function**. The event preserves metadata about:<br><br>• The function that was performed,<br>• When it was performed,<br>• Who performed it,<br>• The participating entities,<br>• What metadata was changed, and may include<br>• An **event comment**.<br><br>See also **function** and **metadata change entry**. |
| **Event comment** | (*metadata*) A comment that is included in an **event** to provide more detail about the event.<br><br>See also **comment**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Event history** | (*noun*) All of the **events** in which a particular **entity** has participated. An **MCRS** maintains an event history for all entities in accordance with the principle expressed in **ISO 15489** that:<br>"Records systems should contain complete and accurate representations of all transactions that occur in relation to a particular record. These include the processes associated with individual records." (ISO 15489-1:2001, 8.3.2) |
| **Evidential** | (*concept*) Having weight as evidence, especially in a legal sense. Part of the reason for using an **MCRS** is to enhance the evidential value of an organisation's records because of the manner in which they can be shown to have been managed. |
| **Export** | (*operation*) A **function performed** by a **user** where one or more entities, and their related entities, are output as **XML data**, either **exported in full** or as **placeholders**.<br>(*noun*) The **export data** produced as a result of exporting entities in the **MoReq2010® export data format**.<br>See also **import**, **lossless**, **lossy** and **transfer**. |
| **Export comment** | (*metadata*) A **comment** provided by the **user** who **performs** an **export**, which is subsequently included in the **export data** and in the **event comment** generated by the export.<br>See also **comment** and **event comment**. |
| **Export data** | (*noun*) A **datafile** output from an **MCRS** containing an **export** and in the **MoReq2010® export data format**.<br>See also **export data format** and **XML data**. |
| **Export data format** | (*noun*) A specific **XML data** format designed to accompany **MoReq2010®** which provides a valid schema for the export of all entities.<br>See also **export data** and **XML data**. |
| **Export identifier** | (*metadata*) An **MCRS** must generate a unique identifier, called the export identifier, for each **export**. This is included in the **export data** and in the **events** generated by the export, to show which entities were **exported in full** or as **placeholders**.<br>See also **UUID**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Export placeholder** | (*noun*) When an **entity** is exported as an export placeholder, then its **system metadata**, **access control list** and **significant entities** are all exported with it, but not its **contextual metadata**, **included entities** or **event history**. |
| **Export process** | (*noun*) The **process** by which **entities** are exported from an **MCRS**. The export process involves:<br>• Assembling the entities to **export**,<br>• Determining whether they are to be **exported in full** or as **placeholders**,<br>• **De-duplication**, and<br>• **Performing** the export operation. |
| **Export service** | (*service*) A **logically** separate **service** within an **MCRS** operationally responsible for undertaking the **export process** and managing **exports**. |
| **Exported in full** | (*verb*) When an **entity** is exported in full, then its **system metadata**, **contextual metadata**, **access control list**, **included entities**, **significant entities** and **event history** are all exported with it. |
| **Extended error information** | (*noun*) Detailed diagnostic information about why a particular error occurred. |
| **Extension module** | (*noun*) An extension to the requirements of **MoReq2010®** developed by the **MoReq Governance Board** and issued by the **DLM Forum®**. An extension module may contain:<br>• Key concepts,<br>• **Functional requirements**,<br>• **Non-functional requirements**,<br>• New glossary terms, and<br>• Extensions to the **information model**.<br>Each extension module also has its own test cases and can be **tested** and **certified** independently of the **core services**, provided the **MCRS** has already been certified as **MoReq2010® compliant**.<br>Extension modules are used to flexibly add additional capabilities to the core services in relation to particular technologies, industries and compliance standards. |
| **External** | (*concept*) Operating autonomously outside the control of the **MCRS** with separate data storage that is not managed by the **MCRS**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **External log** | (*noun*) See **error log**. |
| **Fail** | (*verb*) See **error**. |
| **First day of the week** | (*noun*) The day on which a week is considered to begin, used when searching and reporting. For example, the search query, "Find **records** with a **disposal action due date** next week" could have different results depending on when "next week" nominally begins and ends. Traditionally the first day of the week is Sunday, but many organisations use Monday since it usually represents the first day of a working week. |
| **First used** | (*concept*) The **lifecycle** concept used throughout the specification that an **entity** can be **created**, **modified** and **deleted** up until the moment when it is first associated with another entity. After first use, some metadata will become permanent and the entity must be **destroyed**, rather than being deleted, leaving a **residual entity**. The concept of first use is not applicable to certain specified entities, such as **records**, **components**, and **events**, as well as the **entity types**, **system metadata element definitions** and **function definitions** defined by **MoReq2010®**.<br>See also **first used timestamp**. |
| **First used timestamp** | (*metadata*) A **timestamp** applied to an **entity** when it is **first used**.<br>See also **timestamp**. |
| **Fixed role** | (*noun*) A built-in **role** that is part of the design of an **MCRS** and cannot be **modified**, **deleted** or **destroyed**. Some products are supplied with certain roles pre-defined by the **supplier**.<br>See also **preconfigured role**. |
| **Flag** | (*datatype*) A metadata element based on a **Boolean value** that can only be "true" or "false". Changing the value to "true" is described as **setting** the flag while changing the value to "false" is described as **clearing** the flag.<br>See also **Boolean**, **clear** and **set**. |
| **Flowchart** | (*noun*) A technique for modelling **processes** in an **information system**. |
| **Format** | (*noun*) See **export data format** and **reporting format**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Full text searching** | (*concept*) A technique of searching on **text** using whole words, rather than pattern matching. Full text searching is particularly relevant to **textual** information, and is enhanced by knowledge of the particular **language** in which the text is composed. This, for example, enables the **search engine** to ignore prefixes and suffixes (known as "word stemming"), to search for synonyms, and to place a relative importance on different words in a phrase. Full text searching is "fuzzy" and most search engines place a weighting on different results that orders them by best fit to the search criteria. This weighting is called a **relevancy score**.<br>See also **relevancy score** and **textual**. |
| **Function** | (*operation*) A pre-defined **operation** involving one or more **participating entities** that a **user performs**. Each function has an expected outcome that is specified by **MoReq2010®**. Functions are closely related to the **functional requirements** which specify the functionality required of an **MCRS**.<br>(*disambiguation*) A function should not be confused with a **business function** used in **classification**. |
| **Function definition** | (*entity*) A definition of a **function** that is represented as an **entity**. **Function definitions** are used for both **access control** and in **events** that are generated by **performing** functions. For access control, function definitions are included in **roles** which are then **granted** to **users** and **groups**. To perform a function a user must have been granted a role that includes its function definition. When events are generated the function definition of the function that was performed is included in the event.<br>All function definitions are provided as part of the **MoReq2010® information model** including the unique **system identifier** for each function. For example, the system identifier for the function "Group – Remove User" is "c3713f12-feb6-459e-a21a-7e63aaeeea6c". |
| **Functional requirement** | (*noun*) A **requirement** stating what an **MCRS** must do. Functional requirements may be **tested** to determine whether a particular MCRS is compliant with **MoReq2010®**.<br>See also **non-functional requirement**. |
| **Generate** | (*verb*) **Automatic** creation of **entities**, such as **events**, and other information by an **MCRS**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Good practice** | (*concept*) Accepted ways of working that experience shows produce above average outcomes, especially in relation to a discipline such as **records management**. |
| **Grant** | (*operation*) Award a **role** to a **user** or **group**. Granting a role will update the **access control list** belonging to an **entity** or **service** and **add** or **modify** an **access control entry**.<br>See also **rescind** and **role**. |
| **Group** | (*entity*) An **entity** that usually represents a team or business unit within the organisation, and has various **user** entities as members. |
| **Hierarchical** | (*concept*) Structured using **parent/child relationships** so that each **entity** may be either a **parent** entity, or a **child** entity or both. Hierarchies are connected, so that every entity must be included by having at least one parent/child relationship with another entity, and non-cyclical so that no entity may be the **ancestor** (or **descendant**) of itself.<br>See also **inherit**. |
| **Heterogeneous** | (*concept*) Comprising entities that are different according to some fundamental characteristic. **MoReq2010®** uses this term to refer to **aggregations** that contain **records** where the records are **classified** using different **classes**.<br>See also **homogenous**. |
| **Homogenous** | (*concept*) Comprising entities that are the same or similar according to some fundamental characteristic. **MoReq2010®** uses this term to refer to **aggregations** that contain **records** where the records have all have the same **class** which is inherited from their **parent aggregation**.<br>See also **heterogeneous**. |
| **Identifier** | (*metadata*) See **language identifier**, **MCRS certification identifier**, **module compatibility identifier**, **service type identifier**, and **system identifier**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Immutability, Principle of** | (*concept*) When applied to the **component** of a **record**, ensures that the **content** of the component remains unaltered. Once the record has been created the content of its components must not change over time.<br>See also **completeness**, **discreteness** and **destructibility**. |
| **Implement** | (*verb*) Interpreting and encoding the **requirements** of **MoReq2010®** into an **application**. To implement the **specification**, **suppliers** must build the requirements into an **implementation**. |
| **Implementation** | (*noun*) An **application** that implements the **requirements** of **MoReq2010®**.<br>Also referred to as a **solution**. |
| **Implements module identifier** | (*metadata*) See **module identifier**. |
| **Implements service identifier** | (*metadata*) See **service identifier**. |
| **Import** | (*operation*) Reconstitution of facsimiles of the **entities** that have been **exported** from one **MCRS** in another different MCRS by input of **XML data** in the **MoReq2010® export data format**.<br>See also **export**, **lossless**, **lossy** and **transfer**. |
| **In full** | (*qualifier*) See **exported in full**. |
| **In place** | (*qualifier*) See **manage in place**. |
| **Inaccessible** | (*adjective*) Unable to be **inspected** because the **user** does not have **access** to the **entity**.<br>See also **access** and **access control**. |
| **Included entity** | (*concept*) Some **entities** include other entities. When they are **exported in full** their included entities must also be exported in full. For example, an **aggregation** includes its **child aggregations** and **records**, and a record includes its **components**. If an aggregation is exported in full then all of its **descendants** will be exported in full as well, and the components of any records that are descendants of the aggregation will also be exported. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Information** | (*noun*) Facts known to a person or organisation. Where information provides evidence of an organisation's **business activities** or **transactions** it should be **captured** as a **record**. An organisation's records may therefore be described as a sub-set of the information available to that organisation.<br><br>Information may be managed by an **information system** and **records** may be managed by a specialised type of information system known as a **records system**. A records system **implements** particular functionality that makes it suitable for **records management**. |
| **Information model** | (*noun*) A structured functional and metadata model that underlies all of **MoReq2010®** and maps every **entity**, **metadata element** and **function** to one another. Strict observance of the information model is critical in ensuring that the **export data** from different **MCRS** solutions is fully compatible. |
| **Information system** | (*noun*) An integrated set of components for collecting, storing, processing, and communicating **information**.<br><br>The main components of information systems are computer hardware and software, **databases**, telecommunications systems, human resources, and procedures.<br><br>From information system (2011). In *Encyclopædia Britannica*. Retrieved from http://www.britannica.com/EBchecked/topic/287895/information-system |
| **Inherit** | (*verb*) See **inheritance**. |
| **Inheritance** | (*concept*) The adoption by one **entity** of the characteristics or properties of another entity by association with it. Inheritance usually takes place through the **parent/child relationship**, but not in all cases. For example, a **record** inherits its **default disposal schedule** from its **class**.<br><br>The principle of inheritance is used in MoReq2010® in the following main areas:<br>• **Access control**,<br>• **Classification**,<br>• **Disposal schedules**,<br>• **Disposal holds**, and<br>• **Export**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Inspect** | (*operation*) Examine an **entity** and its **metadata**. **Access** to the inspect **function** is fundamental for **discovery**. If a **user** is not able to inspect an entity then it is considered **inaccessible** and there must not be any indication of the entity provided to the user by the **MCRS** while **browsing**, **searching** or in any other way. |
| **Install** | (*verb*) Initialise an instance of an **MCRS** by setting up and configuring the necessary hardware and software. |
| **Integration** | (*concept*) Interfacing to and interoperating with another **business system**. Tight integration implies close cooperation. |
| **Integrity, Principle of** | (*concept*) Along with **authenticity**, **reliability** and **usability**, one of the central characteristics of a **record** according to **ISO 15489**.<br>"The integrity of a record refers to its being completed and unaltered." (ISO 15489-1:2001, 7.2.4)<br>See also **authenticity**, **reliability** and **usability**. |
| **Interface** | (*noun*) The part of an **information system** that offers up its functionality to **users** and/or other systems. By interacting with the interface of an **MCRS**, a **user** is able to **perform functions**. |
| **Interoperability** | (*concept*) The ability for one system to be able to operate using the data and information provided by another system. Specifically in **MoReq2010®** interoperability between records systems is defined as the ability of a source system to export its entities and of a destination system to import and represent them as complete and fully functional entities alongside its own. |
| **Language** | (*noun*) A human language. Languages are specified in **MoReq2010®** using language identifiers.<br>See also **default language** and **language identifier**. |
| **Language identifier** | (*datatype*) **MoReq2010®** language identifiers must be compliant with RFC5646 and the IANA Language Subtag Registry.<br>See also **textual** and **default language identifier**. |
| **Last review comment** | (*metadata*) The **comment** made by the **user** that last **reviewed** the record. The last review comment is applied as **metadata** to the record to allow users to understand what considerations went into the previous review.<br>See also **comment**, **last reviewed timestamp** and **review**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Last reviewed timestamp** | (*metadata*) System set date and time of the last **review**. The last reviewed timestamp is useful when considering the **last review comment** to see how long ago the assessment was made.<br>See also **last review comment**, **review** and **timestamp**. |
| **Level** | (*noun*) The depth of a **hierarchical** structure as measured by counting the number of entities from the common **ancestor** of all entities in the hierarchy, tracing from **parent** to **child** to its furthest **descendant**, where the furthest descendant has the most intervening **parent/child relationships**.<br>See also **hierarchical** and **max levels of aggregation**. |
| **Lifecycle** | (*concept*) Every **entity** in an **MCRS** has a pre-determined lifecycle which begins with the **creation** of the entity and has various phases depending on its **entity type**. An entity is initially **active** until it is **destroyed**, when it becomes a **residual entity**. |
| **Lift** | (*operation*) Term used to describe the removal of a **disposal hold** from a **class**, **aggregation** or **record**. When a disposal hold is **destroyed** then it is immediately lifted. |
| **Litigation hold** | (*noun*) In **MoReq2010®** the term **disposal hold** is used instead of "litigation hold".<br>See **disposal hold**. |
| **Localisation** | (*concept*) Customising an individual instance of an **MCRS** to meet local **requirements**. This can be done in many different ways. Using different **languages**, creating specific **roles**, constructing an organisational **classification scheme**, defining **contextual metadata elements**, using different **titles** and **descriptions** for **entity types** and **function definitions** are all aspects of localisation. |
| **Log** | (*noun*) See **error log**. |
| **Logical** | (*concept*) Conceptually different. For an example, a **service** may be logically differentiated within an **MCRS** from other services, while in reality they all remain part of the same codebase, or two different **components** may have the same **content** but use pointers to make it appear as if they have different **content**, thereby satisfying the principle of **discreteness**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Lossless** | (*concept*) When used in the context of **interoperability**, or **export** and **import**, refers to the idea that important contextual information is not lost during **transfer**.<br>See also **export**, **import**, **lossy** and **transfer**. |
| **Lossy** | (*concept*) When used in the context of **interoperability**, or **export** and **import**, indicates that some contextual information, such as **metadata**, **events**, **access control lists** or **relationships** with other **entities** is lost during **transfer**.<br>See also **export**, **import**, **lossless** and **transfer**. |
| **Maintainability, Principle of** | (concept) A **non-functional** aspect of a **records system** that describes how relatively easy a given **application** or site **installation** is to maintain an upgrade. A system is considered less maintainable if it is difficult to service, must be taken offline for a significant period to apply upgrades and patches, or requires costly expenditure on third-party support or maintenance contracts. |
| **Major version** | (*noun*) See **version**. |
| **Manage in place** | (*concept*) The management of **records** with **components** whose **content** is located in another **external information system**. |
| **Manageability, Principle of** | (concept) A **non-functional** aspect of a **records system** that refers to how it is managed and **administered**. The technical **administrator's** task is to ensure the records system remains operational, while the records manager is charged with ensuring the records system is used effectively. |
| **Mandate** | (*metadata*) The legal or other instrument that provides the procedural authority for a disposal schedule or a disposal hold |
| **Manual** | (*adjective*) An **operation** or **function** performed by a **user**, especially when contrasted with one performed automatically by the **MCRS** in accordance with its own processing rules. It is important to note that a user may be another **business system**. As a result a "manual" function or operation may not necessarily be performed by a human.<br>See also **automatic**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Max levels of aggregation** | (*metadata*) The number of **levels** of **aggregation** allowed below a given **root aggregation**. If max levels of aggregation is assigned a value of zero then no **child aggregations** can be added below the root.<br>See also **level**. |
| **MCRS** | (*acronym*) A **MoReq2010® compliant records system**. |
| **MCRS certification identifier** | (*metadata*) A **universally unique identifier** issued by the **DLM Forum®** upon the award of a **MoReq2010®** certificate of compliance. The MCRS certification identifier is used to report the compliance status of an **MCRS.**<br>See also **certify** and **system identifier**. |
| **Metadata** | (*noun*) Information about an **entity** made up of several **metadata elements**.<br>See **metadata element**. |
| **Metadata change entry** | (*data structure*) A **data structure** included in an **event** whenever one or more **metadata elements** were modified by the event. Each metadata change entry contains the value of a metadata element both before and after it was modified. |
| **Metadata element** | (*noun*) An item of **metadata** described by a **metadata element definition** that has zero or more **values** given to it by **users** and by the **MCRS**. |
| **Metadata element definition** | (*entity*) A definition of a **metadata element** that indicates, among other properties, its:<br>• Title – the name of the metadata element;<br>• Datatype – what type of metadata it may contain;<br>• Cardinality – how many values it may have; as well as<br>• Whether these values may be changed by a user.<br>See **contextual metadata element definition** and **system metadata element definition**. |
| **Metadata service** | (*service*) A **logically** separate **service** within an **MCRS** operationally responsible for managing **metadata element definitions** and metadata **templates**.<br>See also **model metadata service**. |
| **Metadata template** | (*entity*) A **template**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Metadata value** | (*noun*) See **value**. |
| **MGB** | (*acronym*) The **MoReq Governance Board**. |
| **Migration** | (*operation*) A variant of **transfer** where the intention is to move all or a substantive number of often active **entities** from one **records system** to a new **MCRS** allowing them to be managed by a new owner, in a new environment, or by a new technology or **solution**.<br>Usually the purpose of migration is to decommission the previous **records system**. Sometimes the source is not a **MoReq2010® compliant records system** and the migration is a singular exercise intended to rearrange the older **records** into the **export data format** used by MoReq2010®. Once they have been migrated into an **MCRS**, the entities can be **transferred** on again in the future, **losslessly**. |
| **Minor version** | (*noun*) See **version**. |
| **Model metadata service** | (*service*) The **MoReq2010®** preferred implementation of the **role service**.<br>See also **metadata service**. |
| **Model role service** | (*service*) The **MoReq2010®** preferred implementation of the **role service**.<br>See also **role service**. |
| **Model service** | (*concept*) A **service** defined by the **MoReq2010®** specification, that may be replaced in operation by a functionally similar service, with a different set of processing rules, that has been defined for a particular **MCRS solution**. To use a replacement service for a model service, an MCRS must have been **tested** to ensure a close match in overall functionality, and it must be able to export its own **proprietary** data structures to the **MoReq2010® export data format**.<br>See **model metadata service** and **model role service**. |
| **Modify** | (*operation*) **Assign** a new **value** to a **metadata element**, or change or **delete** a **previous value**.<br>See also **assign** and **associate**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Modifiable** | (*adjective*) Indication of whether or not a **user** may **modify** a particular **metadata element**. A metadata element that cannot be modified is said to be "**read only**" |
| **Module** | (*noun*) See **extension module** and **plug-in module**. |
| **Module identifier** | (*metadata*) A **universally unique identifier** provided by the **DLM Forum®** that is used in compliance reporting by an **MCRS** to indicate that it implements a particular **major version** of a **module** of **MoReq2010®**.<br><br>See also **module**, **system identifier** and **version**. |
| **MoReq®** | (*acronym*) Modular Requirements for Records Systems (formerly "Model Requirements for the Management of Electronic Records").<br><br>(*noun*) As a generic term used to refer to any of the various specifications published under the acronym MoReq®, including MoReq® (2001), **MoReq2®** (2008) and **MoReq2010®**.<br><br>(*disambiguation*) As a specific term used to refer to the original MoReq® specification published in 2001.<br><br>Can also be used informally to imply the most recent version of the MoReq® specification, which is **MoReq2010®**. |
| **MoReq Governance Board** | (*noun*) A sub-committee of the **DLM Forum®** appointed by the executive committee to manage the **MoReq® specification** and related activities, including all:<br><br>• Future development, extension, localisation and adaption through sponsorship and joint working groups;<br>• Translation of the specification into different languages;<br>• Issuing guidance notes and supplementary resources;<br>• Maintenance, clarification and update;<br>• Education, workshops and training programmes;<br>• Marketing and publicity;<br>• Brand protection; and managing the accompanying<br>• **Testing** and **certification** programme through a network of accredited test centres.<br><br>The **MoReq® Governance Board** is commonly referred to as the **MGB**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **MoReq2®** | (*noun*) The immediate successor to the original **MoReq® specification** developed by the **DLM Forum®** following a scoping study in 2005, and launched in March, 2008.<br><br>**MoReq2®** (2008) superseded MoReq® (2001) and was itself superseded by **MoReq2010®** in May, 2011. |
| **MoReq2010®** | (*noun*) The most recent version of the **MoReq® specification**, developed under the MoReq2010® work programme, following the **MGB's** 2009 roadmap.<br><br>The MoReq2010® work programme was initiated by the **DLM Forum®** at its meeting in Madrid in May 2010, and MoReq2010® was launched, after a year's development, at the DLM Forum® meeting in Budapest in May 2011.<br><br>MoReq2010® supersedes **MoReq2®** (2008). |
| **MoReq2010® compliant records system** | (*noun*) A **records system** which is fully **compliant** with the **core services** of **MoReq2010®** and may also be fully compliant with one or more **modules**. A MoReq2010® compliant records system is usually referred to as an **MCRS**. To provide surety that an MCRS is fully compliant with MoReq2010®, it should be **tested** and **certified** by the **DLM Forum®**. |
| **Move** | (*operation*) Used in the context of an **aggregation** to indicate the **removal** of a **child entity** or **record** from one aggregation and its **addition** to another aggregation. An aggregation can be moved to become a **root aggregation** or moved from a root aggregation to become a **child aggregation**. Records must always belong to an aggregation.<br><br>See also **add** and **remove**. |
| **Multi-tenanted** | (*concept*) In respect of a **records system**, one that is shared by a number of different organisations such that each organisation has its own part of the records system and does not have access to the **records** and other **entities** belonging to other organisations. |
| **Native metadata model** | (*noun*) The **metadata model** of an **MCRS** that does not use the **model metadata service**. A native metadata model will usually not be compatible with the metadata model of an MCRS from a different **supplier**.<br><br>See **proprietary**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Native permissions model** | (*noun*) The **role model** of an **MCRS** that does not use the **model role service**. A native permissions model will usually not be compatible with the role model, or native permissions model, of an MCRS from a different **supplier**.<br><br>See **proprietary**. |
| **Nominated** | (*adjective*) In relation to **entities**, the entity or entities that have been chosen or selected by the **user**. |
| **Non-administrative role** | (*noun*) A **role** that is only **inherited** when it is included by the **access control list** of a **child entity**. Non-administrative roles may be specifically excluded from being inherited. **Administrative roles** may never be blocked.<br><br>See also **administrative role**. |
| **Non-compliant system** | (*noun*) A records system that is not **MoReq2010®** compliant, specifically one that has not been certified by the **DLM Forum®**.<br><br>See also **information system** and **MoReq2010® compliant records system**. |
| **Non-functional** | (*adjective*) A qualitative aspect or assessment, as distinct from taking a purely functional perspective.<br><br>See also **non-functional requirement**. |
| **Non-functional requirement** | (*noun*) An important requirement of a **system** that is not expressed as a particular **function** that the system must perform. Non-functional requirements assess not what a system does but how well it does it. This includes a qualitative assessment of the following principles:<br><br>• **Performance**,<br>• **Scalability**,<br>• **Manageability**,<br>• **Portability**,<br>• **Security**,<br>• **Privacy**,<br>• **Usability**,<br>• **Accessibility**,<br>• **Availability**,<br>• **Reliability**,<br>• **Recoverability**,<br>• **Maintainability**,<br>• **Supported**, |

| Term | Explanation and relationship to general concepts |
|---|---|
| | • **Warranted**, and<br>• **Compliance**.<br><br>See also **functional requirement**. |
| **Open** | (*noun*) For **aggregations**, a state allowing new **child aggregations** or **records** to be **created** in the aggregation, or existing aggregations and records to be **moved** to it.<br><br>See also **closed**.<br><br>(*operation*) The **function** of opening a **closed aggregation** so that it can accept additional **children** that may be moved or created in it.<br><br>See also **close**. |
| **Operation** | (*noun*) Any action generally taken by an **MCRS** in response to an internal processing rule or an external stimulus. A **function** is a specialised form of operation which is usually (though not exclusively) initiated by a **user**, involves one or more **participating entities**, and may result in an **event** being **generated**.<br><br>See also **function**. |
| **Organisational quarter** | (*noun*) A quarter, consisting of a three month period, in an organisational year.<br><br>See also **organisational year**. |
| **Organisational year** | (*noun*) Many organisations have a financial or administrative year consisting of four quarters, each of three months, that is not aligned with the calendar year.<br><br>See also **organisational quarter**. |
| **Originated Date/Time** | (*metadata*) The **date** and time from which an **entity** originates. By default, **MoReq2010®** uses the date and time on which the entity was **created** in the **MCRS**, however the entity's actual originated date/time may be earlier than this.<br><br>See also **date/time**. |
| **Override** | (*verb*) Provide an explicit replacement for a **default entity** or **value**, especially with respect to overriding the **inherited classification** of an **aggregation** or **record**, or overriding the **default disposal schedule** for a record, derived from its **class**.<br><br>See **default** and **inherit**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Pagination** | (*concept*) Specifically with respect to **search results**, to divide a large set of search results up into smaller subsets that are returned individually to the **user** performing the **search**.<br>See also **search results**. |
| **Parent** | (*noun*) An **entity**, in a **hierarchical** structure, that has one or more **child** entities. The link between parent and child entities is known as a parent/child relationship.<br>See also **ancestor**, **child** and **descendant**. |
| **Parent aggregation** | (*noun*) An **aggregation** that contains either **child aggregations** or **records**.<br>See also **child aggregation** and **root aggregation**. |
| **Parent/child relationship** | (*noun*) The **relationship** between a **parent** and a **child** in a **hierarchical** structure. In **MoReq2010®** this relationship is established by storing a **reference** to the parent as part of the **metadata** of the child **entity**. |
| **Participating entity** | (*noun*) An **entity** that is considered to have participated in an **event**, or for which the event is important. The event forms a part of the **event history** of every participating entity.<br>See also **event**. |
| **Perform** | (*verb*) To carry out or execute a **function** in relation to one or more **entities**.<br>See also **error** and **function**. |
| **Performance, Principle of** | (concept) A **non-functional** aspect of a **records system** concerned with the speed and efficiency of processing. |
| **Physical** | (*adjective*) Having a physical or real world presence. For example, a **record** that has **content** consisting of paper with writing on it, as compared to a record with **electronic** content.<br>(*disambiguation*) The term "physical" is also used as the opposite of **logical**.<br>See also **electronic** and **logical**. |
| **Placeholder** | (*noun*) See **export placeholder**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Plug-in module** | (*noun*) A **module** of **MoReq2010®** that plugs into the **core services** and provides specialised functionality in a particular area. Plug-in modules are organised into series and to be **compliant** each **MCRS** must implement at least one plug-in module from each series. An MCRS may implement more than one plug-in module from the same series. |
| **Pointer** | (*concept*) A technique for implementing the principle of **discreteness** in a **logical**, rather than a **physical**, way. For example, assuming two different **records** have **components** that both contain the same **content**. Rather than keep two copies of the content, the MCRS keeps two pointers to the content and a counter of the number of components that point to the same content. When the first component is **destroyed** its pointer is **deleted** and the pointer counter is decremented from two to one. When the second component is destroyed it pointer is also deleted and as the pointer counter decrements to zero, the content is also deleted. |
| **Portability, Principle of** | (concept) A **non-functional** aspect of a **records system** that assesses the extent to which the system can be deployed onto different platforms, devices, and operating environments. |
| **Preconfigured role** | (*noun*) A role, like a **fixed role**, that has been pre-defined by the **supplier** of an **MCRS solution** to allow its configuration. Unlike a fixed role, a preconfigured role may be later **modified** or **destroyed**.<br>See also **fixed role**. |
| **Prerequisite** | (*noun*) A **module** which must be **implemented** before the nominated module is able to be implemented. Used to indicate that one module of **MoReq2010®** is reliant on the functionality of another also being implemented by the **MCRS**.<br>See also **co-requisite**, **module** and **service**. |
| **Presentation order** | (*metadata*) A **user** specified order in which items should appear, specifically the order in which **metadata elements** should be presented when an **entity** is **inspected**. |
| **Privacy, Principle of** | (concept) A **non-functional** aspect of a **records system**, privacy is concerned with the degree to which the system supports the principles of data protection as well as the broader issue of handling any form of sensitive information. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Process** | (*noun*) A step-by-step workflow described by **MoReq2010®** that should be implemented by all **MCRS solutions** in the same way, so that it will reliably produce the same outcomes each time it is **performed**. The most important processes described by MoReq2010® are the **disposal process** and the **export process**. |
| **Product** | (*noun*) **COTS** software, usually licensed from a **supplier**, that can be **installed** as a standalone **application** to provide an **implementation** of an **MCRS**. |
| **Proprietary** | (*concept*) An **implementation** that is specific to a particular **product** from a particular **supplier** and is incompatible with other **records system** implementations from other suppliers. One of the primary design goals of **MoReq2010®** is to avoid incompatible proprietary records systems that are not fully interoperable.<br>See also **native metadata model** and **native permissions model**. |
| **Prune** | (*verb*) When **entities** are **destroyed**, to **automatically** and selectively **delete** pre-determined **metadata elements**, as well as to delete **events** from their **event histories**, based on which **functions** they represent. |
| **Read only** | (*concept*) See **modifiable**. |
| **Reclassification** | (*operation*) To change the **classification** of an **aggregation** or **record**. Reclassification may occur as a result of **overriding** the **default** classification or, where the aggregation or record **inherits** its **class**, it may occur as a result of reclassifying a **parent** or **ancestor entity**.<br>See also **classification**. |
| **Record** | (*entity*) Any "information created, received, and maintained as evidence and information by an organization or person, in pursuance of legal obligations or in the transaction of business" (ISO 15489-1:2001, 3.15)<br>In **MoReq2010®** a record may be further characterised as follows:<br>• It has an extensible set of **metadata** that describe it;<br>• It has one or more **components** that represent its **content**;<br>• It is **classified** with a business **classification**;<br>• It has a **disposal schedule** that describes explicitly if, how and when it will be **disposed** of or **destroyed**;<br>• It belongs to an **aggregation** of records; |

| Term | Explanation and relationship to general concepts |
|---|---|
| | • **Access** to it is controlled and limited to **authorised users**;<br>• Its destruction may be prevented by a **disposal hold**; and<br>• It may be **exported** to another **MCRS** while retaining all of the characteristics listed above. |
| **Record content** | (*noun*) See **component content**. |
| **Record service** | (*service*) A **logically** separate **service** within an **MCRS** operationally responsible for managing **aggregations**, **records** and their **components**. |
| **Records management** | (*concept*) The field of management responsible for the efficient and systematic control of the creation, receipt, maintenance, use and **disposal** of **records**, including **processes** for capturing and maintaining **evidence** of, and information about, **business activities** and **transactions** in the form of records (from ISO 15489-1:2001, 3.16). |
| **Records system** | (*noun*) An **information system** which captures, manages and provides access to **records** through time (ISO 15489-1:2001, 3.17).<br>A **MoReq2010® compliant records system** delivers the functionality of a records system in a standardised way through its **implementation** of a set of **core services**, defined by the **MoReq2010® specification**, which may be further extended, adapted and **localised** through additional **modules**. |
| **Recover** | (*verb*) To cope with a system failure or disaster by replacing damaged hardware and software, and **restoring** the system's data to bring it back to a previously known and stable state.<br>See also **backup** and **recover**. |
| **Recoverability, Principle of** | (concept) Non-functional aspect of a records system that assesses its ability to recover from a system failure or disaster. Also known as "disaster recovery" or "business continuity". |
| **Reference** | (*noun*) Specifically in the context of **relationships** between entities and the **metadata** of an **entity**, a **metadata element** is said to contain a reference when its whole purpose is to store the unique **system identifier** of another entity. **Relationships** between entities are realised by one entity referencing the other. If a **user** is **authorised** to **inspect** both entities then the **MCRS** must allow the user to **browse** the relationship, from one to the other, in either direction.<br>(*disambiguation*) **MoReq2010®** also uses the term "reference" in its |

| Term | Explanation and relationship to general concepts |
|---|---|
| | more general sense. For example, most **functional requirements** have a "function reference" which is simply an index to the information model that allows the reader to look up the **function definition** matching the requirement. |
| **Relationship** | (*noun*) See **entity relationship diagram**, **parent/child relationship** and especially **reference**. |
| **Relevancy score** | (*metadata*) The, usually **proprietary**, weighting given to a **full text search** that allows more relevant results to be returned first.<br>See **full text searching**. |
| **Reliability, Principle of** | (*concept*) Along with **authenticity**, **integrity** and **usability**, one of the central characteristics of a **record** according to **ISO 15489**.<br>"A reliable record is one whose contents can be trusted as a full and accurate representation of the transactions, activities or facts to which they attest..." (ISO 15489-1:2001, 7.2.3)<br>(*alternative usage*) Reliability is also a **non-functional** aspect of a **records system**, and when used in this sense it refers to the resilience of the system as a whole, and is often measured as the "mean time between failures".<br>See also **authenticity**, **integrity** and **usability**. |
| **Remove** | (*operation*) The **function** of disassociating an **entity** from a set of entities, usually for the purpose of moving it elsewhere. For example, removing a **record** from its **parent aggregation** so as to **move** it to a new parent aggregation.<br>See also **add**, **associate** and **move**. |
| **Report** | (*noun*) The structured assembly of information, using a preconfigured outline, in response to a specific enquiry or a more general enquiry, such as a **search query**. Searching and reporting are closely related activities and often overlap. A **search** that is subsequently made into a report is sometimes called an ad hoc report. Traditionally a report has a **report format** that enables it to be stored as **electronic content** and potentially to be **declared** as a **record** itself. **MoReq2010®** requires an MCRS to be able to provide both specific reports, in response to some **functional requirements**, and two types of general report: a **detailed report** and a **summary report**.<br>See also **detailed report** and **summary report**.<br>(*operation*) The **function** of generating a report. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Reporting format** | (*noun*) A common and widely recognised format for a report, such as:<br>• Comma or tab separated values;<br>• Spreadsheet formats, such as OOXML and ODF;<br>• XML and HTML based formats; and<br>• PDF or other document formats.<br>See also **format** and **report**. |
| **Requirement** | (*noun*) See **functional requirement** and **non-functional requirement**. |
| **Rescind** | (*operation*) To prevent the further use of a **role** that was previously **granted** to a **user** or **group**. Rescinding a role will update the **access control list** belonging to an **entity** or **service** and **delete** or **modify** an **access control entry**.<br>See also **grant** and **role**. |
| **Residual entity** | (*noun*) An **entity** that has been **destroyed**. A residual entity is no longer **active** and upon its destruction it may have been **pruned** of some of its **metadata** and some of the **events** from its **event history**.<br>See also **active entity**. |
| **Restore** | (*verb*) To **recover** an **information system** to a previously known and stable state using a **backup** following a system failure or disaster.<br>See also **backup** and **recover**. |
| **Retain** | (*verb*) Keep a **record** in the **MCRS** and ensure that it is not **deleted** or **destroyed**.<br>See also **dispose**. |
| **Retain on hold** | (*verb*) **Retain** a **record** and prevent its **destruction** for the duration of a **disposal hold**.<br>See also **disposal hold**. |
| **Retain permanently** | (*verb*) **Retain** a **record** indefinitely and prevent its **destruction** unless or until it is given a different **disposal schedule**.<br>See also **disposal action**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Retention period** | (*noun*) The period of time that a **record** is **retained** from its **retention start date**, initiated by its **retention trigger**, until its **disposal due date**, marking the end of the retention period. |
| **Retention start date** | (*metadata*) The **date** on which the conditions of the **retention trigger** in the **record's disposal schedule** are met and the **retention period** begins. |
| **Retention trigger** | (*noun*) One of a number of possible conditions that may cause a **record's retention period** to commence. Each **disposal schedule** lists a **retention trigger**. For example, a record's retention trigger may be the **date** it was added to its **parent aggregation**. |
| **Review** | (*operation*) The **function** of assessing a **record** that is due for review and determining what **disposal schedule** to apply to it. An authorised user may complete a review by applying a new disposal schedule to the record and entering a review comment describing the **review decision**.<br>See also **last review comment** and **last reviewed timestamp**. |
| **Review decision** | (*noun*) The decision made by the reviewer of a **record** when **completing** a **review**.<br>See also **disposal completion** and **review**. |
| **Role** | (*entity*) An **entity** representing a set of **function definitions**. **Granting** a role to a **user** or **group** in relation to an **entity**, enables that user or any member of that group to **perform** that role on the entity and its **descendants**. **Roles** are generally constructed to mirror the tasks of a staff member filling a particular position within the organisation. For example, different roles may be constructed around each of the following usage types:<br>• Office clerk,<br>• Local records officer,<br>• Senior records manager,<br>• Personnel manager,<br>• Sales representative,<br>• Auditor,<br>• External contractor,<br>• Guest or office temp,<br>• Executive personal assistant,<br>• Senior executive officer,<br>• And so on. |

| Term | Explanation and relationship to general concepts |
|---|---|
| | See also **fixed role**, **grant**, **preconfigured role** and **rescind**. |
| **Role service** | (*service*) A **logically** separate **service** within an **MCRS** operationally responsible for managing **roles**.<br>See also **model role service**. |
| **Root aggregation** | (*noun*) An **aggregation** that is not a **child** of another aggregation. Each root aggregation is created directly under the **record service**.<br>See **child aggregation** and **parent aggregation**. |
| **Row** | (*noun*) Where the **metadata elements** or properties of **entities** are arranged into **tables**, each row of the table usually contains the metadata of a single entity, while each **column** contains the values for each entity of a single metadata element.<br>See also **column** and **table**. |
| **Saved report** | (*noun*) A **report** where the **report format** and **search query** used are saved by the **MCRS** and stored so the saved report may be shared with other **users** and generated again, as required. A saved report is not an entity.<br>(*disambiguation*) A "saved report" refers to the saving of the report definition, not the report results. |
| **Saved search** | (*noun*) A **search query** which is saved so that it can be shared with other **users**, run again when required, and used as the basis of further **searches**. A saved search is not an entity.<br>(*disambiguation*) A saved "search refers" to the saving of the search query, not the search results. |
| **Scalability, Principle of** | (concept) A **non-functional** aspect of a **records system** that assesses the extent to which it can grow to support increased capacity without requiring replacement or extensive reconfiguration. Systems typically need to be scaled as a result of organisational expansion, increased workload, peak usage and/or the regular accumulation of records and related entities over time. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Scheduled activity** | (*noun*) A planned **process** or routine carried out by an **information system** at regular intervals; particularly a resource intensive **operation** that is scheduled for a time of the day or week when usage of the system is low.<br>See also **activity**. |
| **Scope notes** | (*metadata*) Guidance notes indicating how best to apply a particular **entity** and stating any organisational policies or constraints on its use. For example, a **class** may have scope notes indicating which **aggregations** and **records** the class should be applied to and which aggregations and records are unsuited for classification under that class. Scope notes can be given to **aggregations**, **classes**, **disposal holds**, **disposal schedules** and **contextual metadata elements**. |
| **Search** | (*operation*) **Discover entities** by specifying **search criteria** that partially or fully match the **values** of their **metadata elements**.<br>See also **complex search** and **full text searching**. |
| **Search engine** | (*noun*) The heart of the **MCRS solution's searching and reporting service**. A search engine will typically index the metadata in the MCRS, apply the **search criteria**, execute and evaluate the search, and assemble and sort the **search results**. Different search engines may be more or less sophisticated in the functionality they offer. As search engines are often highly specialised, like **databases**, some suppliers will make use of third-party search engines in their **MCRS solutions**. |
| **Search criteria** | (*noun*) Individual conditions included in a **search query**. Each search query will be made up of one or more search criteria. Each search criterion is made up of **metadata elements** to search on, search operators and **search terms**. |
| **Search description** | (*metadata*) A description of a **search query** that is included in an **event** so as to retain evidence of the **search** that was **performed**. The search description may be in natural language or in a structured expression language.<br>See also **event comment**. |
| **Search query** | (*noun*) An enquiry made up of one or more **search criteria** that a **user** constructs so as to **perform** a **search**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Search results** | (*noun*) The results of a **search** expressed as an ordered list of **entities** and their **metadata**.<br>See also **pagination**. |
| **Search term** | (*noun*) A part of a **search criterion** that holds a value that can be compared or searched on. |
| **Searching and reporting service** | (*service*) A **logically** separate **service** within an **MCRS** that is, or is integrated with, a **search engine** and **performs searches** and generates **reports** for **authorised users**.<br>See also **model role service**. |
| **Secure** | (*concept*) A unit of information is secure when only **authorised users** may **access** it and manipulate it. In an **MCRS**, internal security is provided by **access controls** while external security is assessed under the **non-functional** principle of **security**.<br>See also **anonymise**. |
| **Security, Principle of** | (concept) A **non-functional** aspect of a **records system** referring to, and measuring, its integrity and its ability to withstand unauthorised access. |
| **Sentencing on creation** | (*concept*) The general principle that the context of a **record** is best **captured** when the record is first **created**, and that this context should be used to govern the record's **disposal**. In **MoReq2010®**, the principle of "sentencing on creation" is embodied by the functionality required of the **MCRS**, where every record created must be **classified** and every **class** has a **default disposal schedule** associated with it. |
| **Service** | (*noun*) A logical subset of the total functionality of an **MCRS** that focuses on managing only one or a small group of **entity types**. For example, the **disposal scheduling service** only manages **disposal schedules**.<br>See also **classification service**, **disposal holding service**, **disposal scheduling service**, **metadata service**, **model metadata service**, **model role service**, **searching and reporting service**, **service based architecture**, **role service** and **user and group service**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Service based architecture** | (*concept*) An **architecture** adopted within the specification where **functional requirements** within the **core** of **MoReq2010®** are separated into discrete **services**. Each service represents a bundle of functionality that could theoretically be undertaken by a separate **application**. Dividing the core in this way helps to introduce the key concepts in a logical order, but it also points towards the future where one **MCRS** may have two services of the same type, such as two different **classification services** for different types of **record** or **aggregation**, or may share a service, such as its **metadata service**, with another MCRS. |
| **Service identifier** | (*metadata*) A **universally unique identifier** provided by the **DLM Forum®** that is used in compliance reporting by an **MCRS** to indicate that it implements a particular **major version** of a **core service** of **MoReq2010®**.<br>See also **module**, **system identifier** and **version**. |
| **Set** | (*operation*) When used in conjunction with a **flag** or **Boolean value**, meaning to assign it the value "true".<br>(*disambiguation*) **MoReq2010®** also uses the term "set" as a collective noun to refer to groupings of entities.<br>See also **flag**. |
| **Significant entity** | (*concept*) An **entity** that is particularly important to another entity and must therefore always be **exported** with it as a **placeholder**, whenever it is exported. For example, a **record's class** is a significant entity to the record. The record cannot be **exported in full** or as a placeholder without its class also being exported as a placeholder. |
| **Snapshot** | (*noun*) A representation of any data or unit of information, but especially an active **entity** and its **metadata**, frozen at a particular moment in time. |
| **Solution** | (*noun*) See **implementation**. |
| **Stream** | (*verb*) See **XML data streaming**. |
| **Sub-type** | (*noun*) See **entity sub-type**. |
| **Summary report** | (*noun*) A report based on statistics rather than individual line items<br>See also **report** and **detailed report**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Supplier** | (*noun*) The manufacturer or developer of a **records system**, or a body that holds some or all of the intellectual property rights, or a supplier's representative, such as a reseller or service integrator. |
| **Supported, Principle of a system being** | (concept) A **non-functional** aspect of a **records system**, a supported system is one that is being actively maintained and upgraded by the **supplier** and for which there exists an operational support facility for reporting issues and receiving information about new releases and software fixes. |
| **System** | (*noun*) See **information system**. |
| **System identifier** | (*metadata*) A **universally unique identifier** generated by the system for an entity or other purpose, such as an **export identifier**. Some system identifiers are provided by **MoReq2010®**, including **service identifiers**, **module identifiers**, **entity type** identifiers, **data structure** identifiers, **system metadata element definition** identifiers and **function definition** identifiers. |
| **System metadata** | (*noun*) **Metadata** that is mandated by **MoReq2010®** and is predefined by the specification which includes the **system metadata element definitions** and their **system identifiers**. Each **entity type** has its own set of system metadata associated with it. |
| **System metadata element definition** | (*entity*) A **metadata element definition** for a **system metadata element**.<br>All system metadata element definitions are provided as part of the **MoReq2010® information model** including the unique **system identifier** for each. For example, the system identifier for the system metadata element "**Title**" is "077fc367-48ba-44a8-8afb-012d05ed1a16". |
| **Table** | (*noun*) An actual or conceptual laying out of the **metadata** of entities into **rows** and **columns**. This layout is most successful when all of the **entities** in a table have the same **metadata elements** or properties, which usually means that they are of the same **entity type**.<br>See also **column**, **database** and **row**. |
| **Technical administration** | (*concept*) See **administrator**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Template** | (*entity*) A representation of a set of **contextual metadata element definitions** that can be applied to an **entity**. Whenever a template is applied, each of the **contextual metadata elements** is instantiated and associated with the entity.<br>See also **default template** and **contextual metadata element definition**. |
| **Testing** | (verb) The activity of proving the correctness of software by performing functions and comparing the results to expected outcomes. Specifically for **MoReq2010®**, formally testing a records system against the functional requirements using the MoReq2010® test framework.<br>See also **certification**. |
| **Text** | (*datatype*) Information transmitted using characters and typically expressed using words delineated by spaces and punctuation. All text in an **MCRS** must use the **Unicode** character set.<br>See also **Unicode**. |
| **Textual** | (*adjective*) Descriptor for a text based metadata element that is expected to contain a value expressed in a particular **language**. The information may not be **accessible** to a user who does not understand that language. Additionally, the language of a textual value may alter how it is automatically processed. For example, it may be indexed differently by a search engine, sorted into a different order in a list, or rendered using different styles or fonts.<br>See also **language identifier**. |
| **Title** | (*metadata*) Mandatory name or identifying text for an entity. Usually shorter than a full **description**. In **MoReq2010®** the titles of entities do not need to be unique, even for records within the same aggregation, although this is considered good practice.<br>See also **description**. |
| **Time zone** | (*metadata*) The offset between a time measured locally and UTC (Universal Coordinated Time). Time zone information must be incorporated into all **timestamps** generated by an **MCRS** to avoid **events** appearing to occur out of sequence. This is particularly necessary to support interoperability.<br>See also **timestamp**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Timestamp** | (*noun*) A highly precise system generated value for the moment in time at which an **event** or other significant **operation** occurred. **Timestamps** must include the full date, time and time zone and should be precise at least to the second, and should seek to be even more precise, such as for example, millisecond precision.<br>See also **first used timestamp**, **last reviewed timestamp** and **time zone**. |
| **Transaction** | (*noun*) An operation performed by a **business system** that is stored in a transaction log so that it can be repeated should the system need to be **recovered** from a disaster. User of transactions ensures that less data is lost if a system fails as the transaction log can be used to roll forward from the last backup to the moment immediately prior to the disaster.<br>(*disambiguation*) The type of transaction described above should not be confused with a **business transaction**.<br>See **business transaction**. |
| **Transfer** | (*operation*) A **disposal action** where **records** are moved to a secondary storage or archival facility. Transfer will therefore usually involve **export** and **import**, followed by the destruction of the records in the originating sytem. Successful transfer must be confirmed before the records in the originating **MCRS** are **destroyed**. Transfer can also be referred to as **migration**, especially where all, or a substantive number, of the entities in an **MCRS**, or a **service**, are being transferred.<br>See also **disposal action**, **disposal confirmation**, **export**, **import** and **migration**. |
| **Tree** | (*noun*) A hierarchical structure.<br>See also **hierarchical**. |
| **Unicode** | (*noun*) The Unicode standard provides standardised character encodings and processing rules for text in all European languages and nearly all human languages. |
| **Uniform resource identifier** | (*noun*) A uniform resource identifier, or **URI**, is used to identify and locate a resource, such as a **datafile**, on the internet. Uniform resource identifiers can be extended to identify resources in other locations, such as in the local operating system. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Universally unique identifier** | (*noun*) An identification number made up of 128 bits and commonly described as a **UUID**. If generated using a suitable algorithm there is only a tiny probability that any two UUID **values** will ever be the same, even among billions of **entities**. UUIDs are invaluable in their support for **interoperability** as they allow **system identifiers** to be exchanged between different generating **information systems**.<br><br>Suitable algorithms for generating UUIDs can be found in RFC4122, and these can even support "high allocation rates of up to 10 million per second per machine" (RFC4122:2005, 2.). |
| **Usability, Principle of** | (*concept*) Along with **authenticity**, **integrity** and **reliability**, one of the central characteristics of a **record** according to **ISO 15489**.<br><br>"A usable record is one that can be located, retrieved, presented and interpreted." (ISO 15489-1:2001, 7.2.5)<br><br>(*alternative usage*) Usability is also a **non-functional** aspect of a **records system**, and when used in this sense it refers to the system as a whole and the quality of the **user** experience, including how difficult it is to learn, and its ease of operation in day to day use.<br><br>See also **authenticity**, **integrity** and **reliability**. |
| **User** | (*entity*) A person or **system** with an account that enables **access** to and use of an **MCRS**. A user does not have to be a human and could be another **business system**. Users must be **authenticated** before they can use an MCRS. |
| **User and group service** | (*entity*) A **logically** separate **service** within an **MCRS** operationally responsible for managing **users** and **groups**. The user and group service may integrate with or provide a wrapper for a **directory**. |
| **UUID** | (*acronym*) A **universally unique identifier**. |
| **URI** | (*acronym*) A **uniform resource identifier**. |
| **Value** | (*noun*) The data that is placed into a **metadata element**. Values must adhere to a strict **datatype** described by the equivalent **metadata element definition**.<br><br>See also **code**, **datatype** and **default value**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Version** | (*noun*) Each **service** and each **module** of **MoReq2010®** has a version number made up of a **major version** and a **minor version**. A minor version represents a change to the service or module that introduces a clarification, correction or additional **non-functional requirement** without changing the underlying **information model**. All implementations with the same major version number should therefore be compatible with one another even if their minor versions are different. A major version represents a change to the information model, such as the addition of new **functions** or **system metadata elements**, that makes an **MCRS** that only implements one major version, incompatible with an MCRS that only implements another. MoReq2010® provided **Service identifiers** and **module identifiers** are only replaced when major version numbers change.<br>See also **major version** and **minor version**. |
| **Warranted, Principle of a system being** | (*concept*) A **non-functional** aspect of a **records system**, a warranted system is one which is issued with a **warrant** from the **supplier** covering its use. |
| **Web browser** | (*noun*) An **application** used to visit a website and load and view web pages.<br>(*disambiguation*) The terms **browse** and **inspect** have a special meaning in **MoReq2010®** that should not be confused with using a web browser. |
| **XML** | (*acronym*) Extensible Markup Language, a document format for expressing data in machine readable text, where elements and entities are identified by markup inserted into the text. |
| **XML data** | (*noun*) Data representing **entities** that are formatted in the **export data format** used by **MoReq2010®** for **transfers**, including **migration**. |
| **XML data streaming** | (*verb*) **XML data** transmitted electronically as a stream, or a series of data packets. |
| **XML transformation** | (*verb*) Manipulation of **XML data** so to represent it in an alternative **XML** format or as other types of data. |

# 14. Information Model

## 14.1 Index to the information model

## 14.2 Entity Types

### E14.2.1 Aggregation

| | |
|---|---|
| **System Identifier** | c4bd4f18-e3f5-4dba-819d-8d58cbd0aed4 |
| **Title** | **Aggregation** |
| **Description** | Aggregation of individual records or higher level aggregation of aggregations of records |
| **Service** | Record Service |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• First Used Timestamp (**M14.4.32**)<br>• Last Addition Timestamp (**M14.4.48**)<br>• Class Identifier (**M14.4.4**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• Closed Timestamp (**M14.4.5**)<br>• Destroyed Timestamp (**M14.4.17**)<br>• Max Levels Of Aggregation (**M14.4.52**)<br>• Parent Aggregation Identifier (**M14.4.63**)<br>• Aggregated Timestamp (**M14.4.1**) |
| **Functions** | • Aggregation – Add Aggregation (**F14.5.1**)<br>• Aggregation – Add Contextual Metadata (**F14.5.2**)<br>• Aggregation – Add Record (**F14.5.3**)<br>• Aggregation – Close (**F14.5.4**)<br>• Aggregation – Create (**F14.5.5**)<br>• Aggregation – Delete (**F14.5.6**)<br>• Aggregation – Delete Residual Event (**F14.5.7**)<br>• Aggregation – Delete Residual Metadata (**F14.5.8**)<br>• Aggregation – Destroy (**F14.5.9**)<br>• Aggregation – Exported (**F14.5.10**)<br>• Aggregation – Inherit Default Class (**F14.5.11**)<br>• Aggregation – Inspect (**F14.5.12**)<br>• Aggregation – Inspect ACL (**F14.5.13**)<br>• Aggregation – Inspect Event (**F14.5.14**)<br>• Aggregation – Modify ACL (**F14.5.15**)<br>• Aggregation – Modify Max Levels Of Aggregation (**F14.5.16**)<br>• Aggregation – Modify Metadata (**F14.5.17**)<br>• Aggregation – Modify Originated Date/Time (**F14.5.18**)<br>• Aggregation – Open (**F14.5.19**) |

| | |
|---|---|
| | • Aggregation – Override Class (**F14.5.20**)<br>• Aggregation – Remove Aggregation (**F14.5.21**)<br>• Aggregation – Remove Record (**F14.5.22**) |

### E14.2.2 Class

| | |
|---|---|
| **System Identifier** | 5a5240e2-939b-43dd-a50b-2d3284d81735 |
| **Title** | **Class** |
| **Description** | Business classification applied to records and aggregations of records |
| **Service** | Classification Service |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• First Used Timestamp (**M14.4.32**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• Default Disposal Schedule Identifier (**M14.4.11**)<br>• Destroyed Timestamp (**M14.4.17**) |
| **Functions** | • Class – Add Contextual Metadata (**F14.5.23**)<br>• Class – Create (**F14.5.24**)<br>• Class – Delete (**F14.5.25**)<br>• Class – Delete Residual Event (**F14.5.26**)<br>• Class – Delete Residual Metadata (**F14.5.27**)<br>• Class – Destroy (**F14.5.28**)<br>• Class – Exported (**F14.5.29**)<br>• Class – Inspect (**F14.5.30**)<br>• Class – Inspect ACL (**F14.5.31**)<br>• Class – Inspect Event (**F14.5.32**)<br>• Class – Modify ACL (**F14.5.33**)<br>• Class – Modify Default Disposal Schedule (**F14.5.34**)<br>• Class – Modify Metadata (**F14.5.35**)<br>• Class – Modify Originated Date/Time (**F14.5.36**) |

### E14.2.3 Component

| | |
|---|---|
| **System Identifier** | 7af81a86-c6d4-43f7-b62d-6c7b905231dd |
| **Title** | **Component** |
| **Description** | Individual component of a record representing its content |

| | |
|---|---|
| **Service** | Record Service |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Record Identifier (**M14.4.86**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Duplicate Identifier (**M14.4.23**)<br>• Automatic Deletion Flag (**M14.4.3**)<br>• Destroyed Timestamp (**M14.4.17**) |
| **Functions** | • Component – Add Contextual Metadata (**F14.5.37**)<br>• Component – Create (**F14.5.38**)<br>• Component – Delete Residual Event (**F14.5.39**)<br>• Component – Delete Residual (**F14.5.40**)<br>• Component – Destroy (**F14.5.41**)<br>• Component – Duplicate (**F14.5.42**)<br>• Component – Exported (**F14.5.43**)<br>• Component – Inspect (**F14.5.44**)<br>• Component – Inspect Event (**F14.5.45**)<br>• Component – Modify Metadata (**F14.5.46**)<br>• Component – Modify Originated Date/Time (**F14.5.47**) |

### E14.2.4 Contextual Metadata Element Definition

| | |
|---|---|
| **System Identifier** | 5effc721-2184-4014-a4f0-5e399b41be57 |
| **Title** | **Contextual Metadata Element Definition** |
| **Description** | Definition of the properties of a contextual metadata element |
| **Sub-type of** | Metadata Element Definition (**E14.2.11**) |
| **Service** | Metadata Service |
| **Additional system metadata** | *As for Metadata Element Definition (**E14.2.11**) plus the following additional system metadata:*<br>• Created Timestamp (**M14.4.9**),<br>• Originated Date/Time (**M14.4.61**),<br>• First Used Timestamp (**M14.4.32**), and<br>• Destroyed Timestamp (**M14.4.17**). |
| **Additional functions** | *As for Metadata Element Definition (**E14.2.11**) plus the following additional functions:*<br>• Contextual Metadata Element Definition – Create (**F14.5.48**) |

| | |
|---|---|
| | • Contextual Metadata Element Definition – Delete (**F14.5.49**)<br>• Contextual Metadata Element Definition – Delete Residual Event (**F14.5.50**)<br>• Contextual Metadata Element Definition – Destroy (**F14.5.51**)<br>• Contextual Metadata Element Definition – Exported (**F14.5.52**)<br>• Contextual Metadata Element Definition – Modify Before Use (**F14.5.53**)<br>• Contextual Metadata Element Definition – Modify Originated Date/Time (**F14.5.54**) |

### E14.2.5 Disposal Hold

| | |
|---|---|
| **System Identifier** | 93645ca5-29c2-428b-9adc-b61b56d2c8bb |
| **Title** | **Disposal Hold** |
| **Description** | Legal or other hold preventing the scheduled destruction of records |
| **Service** | Disposal Holding Service |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• First Used Timestamp (**M14.4.32**)<br>• Held Record Identifier (**M14.4.39**)<br>• Held Aggregation Identifier (**M14.4.37**)<br>• Held Class Identifier (**M14.4.38**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Mandate (**M14.4.51**)<br>• Scope Notes (**M14.4.97**)<br>• Destroyed Timestamp (**M14.4.17**) |
| **Functions** | • Disposal Hold – Add Contextual Metadata (**F14.5.55**)<br>• Disposal Hold – Add Entity (**F14.5.56**)<br>• Disposal Hold – Create (**F14.5.57**)<br>• Disposal Hold – Delete (**F14.5.58**)<br>• Disposal Hold – Delete Residual Event (**F14.5.59**)<br>• Disposal Hold – Delete Residual Metadata (**F14.5.60**)<br>• Disposal Hold – Destroy (**F14.5.61**)<br>• Disposal Hold – Exported (**F14.5.62**)<br>• Disposal Hold – Inspect (**F14.5.63**)<br>• Disposal Hold – Inspect ACL (**F14.5.64**)<br>• Disposal Hold – Inspect Event (**F14.5.65**)<br>• Disposal Hold – Modify ACL (**F14.5.66**)<br>• Disposal Hold – Modify Metadata (**F14.5.67**)<br>• Disposal Hold – Modify Originated Date/Time (**F14.5.68**) |

| | |
|---|---|
| | • Disposal Hold – Remove Entity (**F14.5.69**) |

### E14.2.6 Disposal Schedule

| | |
|---|---|
| **System Identifier** | 00b35d5d-301e-4000-ad18-211de45edb32 |
| **Title** | **Disposal Schedule** |
| **Description** | Schedule by which records are retained for a specified period of time followed by their planned disposal |
| **Service** | Disposal Scheduling Service |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• First Used Timestamp (**M14.4.32**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Mandate (**M14.4.51**)<br>• Scope Notes (**M14.4.97**)<br>• Disposal Action Code (**M14.4.18**)<br>• Retention Trigger Code (**M14.4.94**)<br>• Retention Trigger Element Identifier (**M14.4.95**)<br>• Retention Period Interval Code (**M14.4.90**)<br>• Retention Period Duration Number (**M14.4.89**)<br>• Retention Period Offset Code (**M14.4.91**)<br>• Retention Period Offset Month Code (**M14.4.92**)<br>• Confirmation Period Interval Code (**M14.4.7**)<br>• Confirmation Period Duration Number (**M14.4.6**)<br>• Destroyed Timestamp (**M14.4.17**) |
| **Functions** | • Disposal Schedule – Add Contextual Metadata (**F14.5.70**)<br>• Disposal Schedule – Create (**F14.5.71**)<br>• Disposal Schedule – Delete (**F14.5.72**)<br>• Disposal Schedule – Delete Residual Record (**F14.5.73**)<br>• Disposal Schedule – Delete Residual Metadata (**F14.5.74**)<br>• Disposal Schedule – Destroy (**F14.5.75**)<br>• Disposal Schedule – Exported (**F14.5.76**)<br>• Disposal Schedule – Inspect (**F14.5.77**)<br>• Disposal Schedule – Inspect ACL (**F14.5.78**)<br>• Disposal Schedule – Inspect Event (**F14.5.79**)<br>• Disposal Schedule – Modify ACL (**F14.5.80**)<br>• Disposal Schedule – Modify Metadata (**F14.5.81**)<br>• Disposal Schedule – Modify Originated Date/Time (**F14.5.82**) |

### E14.2.7 Entity Type

| | |
|---|---|
| **System Identifier** | 5f423557-6130-43dc-930f-f95c6700e630 |
| **Title** | **Entity Type** |
| **Description** | Definition of an entity, including a list of its system metadata and the functions that can be performed on it |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**) |
| **Functions** | • Entity Type – Inspect (**F14.5.83**)<br>• Entity Type – Inspect ACL (**F14.5.84**)<br>• Entity Type – Inspect Event (**F14.5.85**)<br>• Entity Type – Modify ACL (**F14.5.86**) |

### E14.2.8 Event

| | |
|---|---|
| **System Identifier** | 97ff1eaa-30cd-4ea1-8ec2-2cbc50732d56 |
| **Title** | **Event** |
| **Description** | Description of the outcome of a function that was performed previously that is retained to show the history of an entity |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Event Timestamp (**M14.4.27**)<br>• Event Function Identifier (**M14.4.26**)<br>• Performed By User Identifier (**M14.4.83**)<br>• Event Comment (**M14.4.25**)<br>• Duplicate Identifier (**M14.4.23)**<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Deleted Metadata Function Definition Identifier (**M14.4.15**)<br>• Export Commencing Timestamp (**M14.4.28**)<br>• Export Completed Timestamp (**M14.4.29**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Historical Date/Time (**M14.4.40**)<br>• Overdue Disposal Action Code (**M14.4.58**)<br>• Overdue Disposal Action Due Date (**M14.4.59**) |

| | |
|---|---|
| | • Overdue Disposal Confirmation Due Date (**M14.4.60**)<br>• Participating Aggregation Identifier (**M14.4.64**)<br>• Participating Class Identifier (**M14.4.65**)<br>• Participating Component Identifier (**M14.4.66**)<br>• Participating Disposal Hold Identifier (**M14.4.67**)<br>• Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Participating Duplicate Identifier (**M14.4.69**)<br>• Participating Entity Type Identifier (**M14.4.70**)<br>• Participating Event Identifier (**M14.4.71**)<br>• Participating Function Definition Identifier (**M14.4.72**)<br>• Participating Group Identifier (**M14.4.73**)<br>• Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Participating New Parent Identifier (**M14.4.75**)<br>• Participating Previous Parent Identifier (**M14.4.76**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Participating Role Identifier (**M14.4.78**)<br>• Participating Service Identifier (**M14.4.79**)<br>• Participating Template Identifier (**M14.4.80**)<br>• Participating User Identifier (**M14.4.81**)<br>• Participating User or Group Identifier (**M14.4.82**)<br>• Rescinded Role Identifier (**M14.4.87**)<br>• Search Query (**M14.4.98**)<br>• Total Entities (**M14.4.105**) |
| **Functions** | *See the other entity types – once generated an event is part of the event history of an entity and accessed through that entity* |

### E14.2.9 Function Definition

| | |
|---|---|
| **System Identifier** | 5c433e4e-f926-4206-ba3f-998b14b8dabb |
| **Title** | **Function Definition** |
| **Description** | Definition of function that can be performed with an entity by a user |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Generate Event Flag (**M14.4.34**)<br>• Retain On Destruction Flag (**M14.4.88**) |
| **Functions** | • Function Definition – Inspect (**F14.5.87**)<br>• Function Definition – Inspect ACL (**F14.5.88**)<br>• Function Definition – Inspect Event (**F14.5.89**)<br>• Function Definition – Modify ACL (**F14.5.90**)<br>• Function Definition – Modify Event Generation (**F14.5.91**) |

| | |
|---|---|
| | • Function Definition – Modify Retain Event On Destruction (**F14.5.92**) |

### E14.2.10 Group

| | |
|---|---|
| **System Identifier** | 9cac7661-62c9-4a9d-8c64-d000210674ee |
| **Title** | **Group** |
| **Description** | Group of users |
| **Service** | User and Group Service |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• First Used Timestamp (**M14.4.32**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Destroyed Timestamp (**M14.4.17**) |
| **Functions** | • Group – Add Contextual Metadata (**F14.5.93**)<br>• Group – Add User (**F14.5.94**)<br>• Group – Create (**F14.5.95**)<br>• Group – Delete (**F14.5.96**)<br>• Group – Delete Residual Event (**F14.5.97**)<br>• Group – Delete Residual Metadata (**F14.5.98**)<br>• Group – Destroy (**F14.5.99**)<br>• Group – Exported (**F14.5.100**)<br>• Group – Inspect (**F14.5.101**)<br>• Group – Inspect ACL (**F14.5.102**)<br>• Group – Inspect Event (**F14.5.103**)<br>• Group – Modify ACL (**F14.5.104**)<br>• Group – Modify Metadata (**F14.5.105**)<br>• Group – Modify Originated Date/Time (**F14.5.106**)<br>• Group – Remove User (**F14.5.107**)<br>• Group – Report User Membership (**F14.5.108**) |

### E14.2.11 Metadata Element Definition

| | |
|---|---|
| **System Identifier** | e6682264-1902-434c-beb3-18a29795aaf4 |
| **Title** | **Metadata Element Definition** |
| **Description** | Definition of the properties of a system metadata element |

| | |
|---|---|
| **Service** | Metadata Service |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• Presentation Order (**M14.4.84**)<br>• Min Occurs (**M14.4.56**)<br>• Max Occurs (**M14.4.53**)<br>• Is Modifiable Flag (**M14.4.46**)<br>• Is Entity Reference Flag (**M14.4.45**)<br>• Entity Reference Type Identifier (**M14.4.24**)<br>• Datatype (**M14.4.10**)<br>• Is Textual Flag (**M14.4.47**)<br>• Default Value (**M14.4.13**)<br>• Default Language Identifier (**M14.4.12**)<br>• Retain On Destruction Flag (**M14.4.88**) |
| **Functions** | • Metadata Element Definition – Inspect (**F14.5.109**)<br>• Metadata Element Definition – Inspect ACL (**F14.5.110**)<br>• Metadata Element Definition – Inspect Event (**F14.5.111**)<br>• Metadata Element Definition – Modify ACL (**F14.5.112**)<br>• Metadata Element Definition – Modify Metadata (**F14.5.113**)<br>• Metadata Element Definition – Modify Retain On Destruction (**F14.5.114**) |

### E14.2.12 Record

| | |
|---|---|
| **System Identifier** | 3ac228ef-c008-4524-9e41-5c4564eaa7f0 |
| **Title** | **Record** |
| **Description** | Record of a business transaction made up of one or more components that are managed atomically |
| **Service** | Record Service |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Duplicate Identifier (**M14.4.23**)<br>• Parent Aggregation Identifier (**M14.4.63**)<br>• Aggregated Timestamp (**M14.4.1**) |

| | |
|---|---|
| | • Class Identifier (**M14.4.4**)<br>• Disposal Schedule Identifier (**M14.4.22**)<br>• Retention Start Date (**M14.4.93**)<br>• Disposal Action Code (**M14.4.18**)<br>• Disposal Action Due Date (**M14.4.19**)<br>• Disposal Confirmation Due Date (**M14.4.20**)<br>• Disposal Overdue Alert Timestamp (**M14.4.21**)<br>• Last Review Comment (**M14.4.49**)<br>• Last Reviewed Timestamp (**M14.4.50**)<br>• Transferred Timestamp (**M14.4.106**)<br>• Destroyed Timestamp (**M14.4.17**) |
| **Functions** | • Record – Add Contextual Metadata (**F14.5.115**)<br>• Record – Cancel Destruction (**F14.5.116**)<br>• Record – Cancel Transfer (**F14.5.117**)<br>• Record – Complete Review (**F14.5.118**)<br>• Record – Confirm Destruction (**F14.5.119**)<br>• Record – Confirm Transfer (**F14.5.120**)<br>• Record – Create (**F14.5.121**)<br>• Record – Delete Residual Event (**F14.5.122**)<br>• Record – Delete Residual Metadata (**F14.5.123**)<br>• Record – Destroy (**F14.5.124**)<br>• Record – Disposal Alert (**F14.5.125**)<br>• Record – Duplicate (**F14.5.126**)<br>• Record – Exported (**F14.5.127**)<br>• Record – Held (**F14.5.128**)<br>• Record – Inherit Default Class (**F14.5.129**)<br>• Record – Inherit Default Disposal Schedule (**F14.5.130**)<br>• Record – Inspect (**F14.5.131**)<br>• Record – Inspect ACL(**F14.5.132**)<br>• Record – Inspect Event (**F14.5.133**)<br>• Record – Modify ACL (**F14.5.134**)<br>• Record – Modify Metadata (**F14.5.135**)<br>• Record – Modify Originated Date/Time (**F14.5.136**)<br>• Record – Override Class (**F14.5.137**)<br>• Record – Override Disposal Schedule (**F14.5.138**)<br>• Record – Released (**F14.5.139**)<br>• Record – Update Disposal (**F14.5.140**) |

### E14.2.13 Role

| | |
|---|---|
| **System Identifier** | fc9f9333-097f-4a84-96f2-11ad6b444ebf |
| **Title** | **Role** |

| | |
|---|---|
| **Description** | Function definitions that have been arranged into a role that may be granted to a user or a group to allow them access to entities |
| **Service** | Role Service |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• First Used Timestamp (**M14.4.32**)<br>• Is Administrative Role Flag (**M14.4.44**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• Function Definition Identifier (**M14.4.33**)<br>• Destroyed Timestamp (**M14.4.17**) |
| **Functions** | • Role – Add Contextual Metadata (**F14.5.141**)<br>• Role – Add Function Definition (**F14.5.142**)<br>• Role – Create (**F14.5.143**)<br>• Role – Delete (**F14.5.144**)<br>• Role – Delete Residual Event (**F14.5.145**)<br>• Role – Delete Residual Metadata (**F14.5.146**)<br>• Role – Destroy (**F14.5.147**)<br>• Role – Exported (**F14.5.148**)<br>• Role – Inspect (**F14.5.149**)<br>• Role – Inspect ACL (**F14.5.150**)<br>• Role – Inspect Event (**F14.5.151**)<br>• Role – Modify ACL (**F14.5.152**)<br>• Role – Modify Metadata (**F14.5.153**)<br>• Role – Modify Originated Date/Time (**F14.5.154**)<br>• Role – Remove Function Definition (**F14.5.155**)<br>• Role – Report Function Definitions (**F14.5.156**) |

### E14.2.14 Service

| | |
|---|---|
| **System Identifier** | 363d5464-db6c-464b-980a-1851464cab45 |
| **Title** | **Service** |
| **Description** | Specific service within an MCRS that manages particular entity types |
| **System metadata** | • System Identifier (**M14.4.100**),<br>• Implements Service Identifier (**M14.4.42**)<br>• Implements Module Identifier (**M14.4.41**)<br>• MCRS Certification Identifier (**M14.4.54**) |

| | |
|---|---|
| | • Supplier Information (**M14.4.99**)<br>• Default Language Identifier (**M14.4.12**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Owner Information (**M14.4.62**) |
| **Functions** | • Service – Add Contextual Metadata (**F14.5.157**)<br>• Service – Inspect (**F14.5.158**)<br>• Service – Inspect ACL (**F14.5.159**)<br>• Service – Inspect Event (**F14.5.160**)<br>• Service – Modify ACL (**F14.5.161**)<br>• Service – Modify Metadata (**F14.5.162**) |

### E14.2.15 Template

| | |
|---|---|
| **System Identifier** | 92596e33-da23-4c9f-8de6-e585f027157a |
| **Title** | **Template** |
| **Description** | Template set of contextual metadata element definitions that can be used to add contextual metadata elements to entities at creation or later |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• First Used Timestamp (**M14.4.32**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Template Entity Type Identifier (**M14.4.102**)<br>• Template Service Identifier (**M14.4.103**)<br>• Template Class Identifier (**M14.4.101**)<br>• Contextual Metadata Element Definition Identifier (**M14.4.8**)<br>• Destroyed Timestamp (**M14.4.17**) |
| **Functions** | • Template – Add Contextual Metadata (**F14.5.164**)<br>• Template – Create (**F14.5.165**)<br>• Template – Delete (**F14.5.166**)<br>• Template – Delete Residual Event (**F14.5.167**)<br>• Template – Delete Residual Metadata (**F14.5.168**)<br>• Template – Destroy (**F14.5.169**)<br>• Template – Exported (**F14.5.170**)<br>• Template – Inspect (**F14.5.171**)<br>• Template – Inspect ACL (**F14.5.172**)<br>• Template – Inspect Event (**F14.5.173**)<br>• Template – Modify ACL (**F14.5.174**)<br>• Template – Modify Metadata (**F14.5.175**) |

| | |
|---|---|
| | • Template – Modify Originated Date/Time (**F14.5.176**) |

### E14.2.16 User

| | |
|---|---|
| **System Identifier** | 19c91384-1dfb-470a-bf1e-c1574ea4ba51 |
| **Title** | **User** |
| **Description** | Entity representing a user of the MCRS |
| **System metadata** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• First Used Timestamp (**M14.4.32**)<br>• Group Identifier (**M14.4.36**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Destroyed Timestamp (**M14.4.17**) |
| **Functions** | • User – Add Contextual Metadata (**F14.5.177**)<br>• User – Browse Records Due For Disposal (**F14.5.178**)<br>• User – Create (**F14.5.179**)<br>• User – Delete (**F14.5.180**)<br>• User – Delete Residual Event (**F14.5.181**)<br>• User – Delete Residual Metadata (**F14.5.182**)<br>• User – Destroy (**F14.5.183**)<br>• User – Detailed Report (**F14.5.184**)<br>• User – Export (**F14.5.185**)<br>• User – Exported (**F14.5.186**)<br>• User – Inspect (**F14.5.187**)<br>• User – Inspect ACL (**F14.5.188**)<br>• User – Inspect Event (**F14.5.189**)<br>• User – Modify ACL (**F14.5.190**)<br>• User – Modify Metadata (**F14.5.191**)<br>• User – Modify Originated Date/Time (**F14.5.192**)<br>• User – Report Authorisation (**F14.5.193**)<br>• User – Report Group Membership (**F14.5.194**)<br>• User – Search (**F14.5.195**)<br>• User – Summary Report (**F14.5.196**) |

## 14.3 Data Structures

### D14.3.1 Access Control Entry

| | |
|---|---|
| **System Identifier** | 60124baa-2625-4795-bf14-7e67f2224ccf |
| **Title** | **Access Control Entry** |
| **Description** | Compound metadata structure within the access control list of an entity which associates a user or a group with one or more roles |
| **Entity Type** | Access Control List (**D14.3.2**) |
| **System metadata** | • User Or Group Identifier (**M14.4.107**)<br>• Role Identifier (**M14.4.96**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |

### D14.3.2 Access Control List

| | |
|---|---|
| **System Identifier** | 082da683-1a04-4cf3-9096-98837a711cbe |
| **Title** | **Access Control List** |
| **Description** | Compound metadata structure within an entity used for determining access control, which maintains a list of access control entries |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Class (**E14.2.2**)<br>• Contextual Metadata Element Definition (**E14.2.4**)<br>• Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**)<br>• Entity Type (**E14.2.7**)<br>• Function Definition (**E14.2.9**)<br>• Group (**E14.2.10**)<br>• Metadata Element Definition (**E14.2.11**)<br>• Record (**E14.2.12**)<br>• Role (**E14.2.13**)<br>• Service (**E14.2.14**)<br>• Template (**E14.2.15**)<br>• User (**E14.2.16**) |
| **System metadata** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |

| | |
|---|---|
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |

### D14.3.3 Metadata Change Entry

| | |
|---|---|
| **System Identifier** | 7e32c9c9-e00a-4dbe-8c5f-c05421e632c2 |
| **Title** | **Metadata Change Entry** |
| **Description** | Compound metadata structure used by events to track the changes made to metadata when functions are performed |
| **Entity Type** | Event (**E14.2.8**) |
| **System metadata** | • Metadata Element Definition Identifier (**M14.4.55**)<br>• Previous Value (**M14.4.85**)<br>• New Value (**M14.4.57**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |

## 14.4 System Metadata Element Definitions

### M14.4.1 Aggregated Timestamp

| | |
|---|---|
| **System Identifier** | 05fd550b-ff6c-4fd0-b1f6-b27df905076f |
| **Title** | **Aggregated Timestamp** |
| **Description** | System set date and time when the child aggregation or record was created in, or added to, its current aggregation |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Record (**E14.2.12**) |
| **Min Occurs** | 0 (for root aggregations)<br>1 (for child aggregations and records) |
| **Max Occurs** | 0 (for root aggregations)<br>1 (for child aggregations and records) |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Timestamp |

### M14.4.2 Applied Template Identifier

| | |
|---|---|
| **System Identifier** | 65a2d42c-dfe3-4fbf-a894-8db373ac2d45 |
| **Title** | **Applied Template Identifier** |
| **Description** | Template that was applied when the function was performed to add contextual metadata elements to entities |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Template (**E14.2.15**) |

| | |
|---|---|
| **Datatype** | UUID |

### M14.4.3 Automatic Deletion Flag

| | |
|---|---|
| **System Identifier** | 4a9c44df-a9ac-4b06-8c87-d5940cd76046 |
| **Title** | **Automatic Deletion Flag** |
| **Description** | Flag that indicates whether the content of the component can be automatically deleted by the system when its record is destroyed |
| **Entity Type** | Component (**E14.2.3**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Boolean |

### M14.4.4 Class Identifier

| | |
|---|---|
| **System Identifier** | 7d935531-e7e6-4a28-a474-431c68522cfa |
| **Title** | **Class Identifier** |
| **Description** | The classification of the entity, used by child aggregations and records to override the class they inherit from their parent aggregation |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Record (**E14.2.12**) |
| **Min Occurs** | 0 (for child aggregations and records)<br>1 (for root aggregations) |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (by reclassifying the aggregation or record) |
| **Entity Reference?** | Yes |
| **Reference Type** | Class (**E14.2.2**) |

| | |
|---|---|
| **Datatype** | UUID |
| ***Usage notes*** | *For child aggregations and records only used when the default disposal schedule is overridden* |

### M14.4.5 Closed Timestamp

| | |
|---|---|
| **System Identifier** | b0194c04-bd20-4582-af83-5da3d55c258a |
| **Title** | **Closed Timestamp** |
| **Description** | System set date and time when the aggregation was closed |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Timestamp |

### M14.4.6 Confirmation Period Duration Number

| | |
|---|---|
| **System Identifier** | 64d99feb-2654-4c09-accd-a00417f78596 |
| **Title** | **Confirmation Period Duration Number** |
| **Description** | Number of days or weeks allowed for confirming the disposal of the record |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (until disposal schedule is first used)<br>No (once disposal schedule has been used) |
| **Entity Reference?** | No |

| | |
|---|---|
| **Datatype** | Positive integer |
| ***Usage notes*** | *Used with Confirmation Period Interval Code (**M14.4.7**)* |

### M14.4.7 Confirmation Period Interval Code

| | |
|---|---|
| **System Identifier** | 30cd2d6f-9957-4904-b0a7-ade8f181cb68 |
| **Title** | **Confirmation Period Interval Code** |
| **Description** | Unit of measurement of time as either days or weeks |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (until disposal schedule is first used)<br>No (once disposal schedule has been used) |
| **Entity Reference?** | No |
| **Refers To Type** | Code |
| **Valid values** | 1 = DAYS<br>2 = WEEKS |
| ***Usage notes*** | *Used with Confirmation Period Duration Number (**M14.4.6**)* |

### M14.4.8 Contextual Metadata Element Definition Identifier

| | |
|---|---|
| **System Identifier** | 1b4ee523-d46d-4614-a3a6-edad609f46ab |
| **Title** | **Contextual Metadata Element Definition Identifier** |
| **Description** | Contextual metadata element included in a template that can be applied to entities as additional contextual metadata |
| **Entity Type** | Template (**E14.2.15**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |

| | |
|---|---|
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Contextual Metadata Element Definition (**E14.2.4**) |
| **Datatype** | UUID |

### M14.4.9 Created Timestamp

| | |
|---|---|
| **System Identifier** | 466a4378-db7d-4b2c-b35e-696722f58c6b |
| **Title** | **Created Timestamp** |
| **Description** | System set date and time when the entity was created |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Class (**E14.2.2**)<br>• Component (**E14.2.3**)<br>• Contextual Metadata Element Definition (**E14.2.4**)<br>• Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**)<br>• Event (**E14.2.8**)<br>• Group (**E14.2.10**)<br>• Record (**E14.2.12**)<br>• Role (**E14.2.13**)<br>• Template (**E14.2.15**)<br>• User (**E14.2.16**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Timestamp |

### M14.4.10 Datatype

| | |
|---|---|
| **System Identifier** | e7f28bcc-0857-4361-b6e4-1bebffdd7578 |
| **Title** | **Datatype** |

| | |
|---|---|
| **Description** | XML datatype definition giving the precise format required for the value of a system or contextual metadata element |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (for contextual metadata prior to their first use)<br>No (for system metadata and after the definition has been applied) |
| **Entity Reference?** | No |
| **Datatype** | XML Datatype Definition |

### M14.4.11 Default Disposal Schedule Identifier

| | |
|---|---|
| **System Identifier** | b5a0997f-3285-4606-a20b-1d938b1415c7 |
| **Title** | **Default Disposal Schedule Identifier** |
| **Description** | The disposal schedule for the class which is inherited by default by records with this classification |
| **Entity Type** | Class (**E14.2.2**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Disposal Schedule (**E14.2.6**) |
| **Datatype** | UUID |

### M14.4.12 Default Language Identifier

| | |
|---|---|
| **System Identifier** | ad13c98d-6ff4-4e0d-9ccd-a7d7238591f8 |
| **Title** | **Default Language Identifier** |

| | |
|---|---|
| **Description** | The default language for a service or a metadata element definition, used as a default by textual metadata elements (only) |
| **Entity Type** | • Metadata Element Definition (**E14.2.11**)<br>• Service (**E14.2.14**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Valid language identifier |

### M14.4.13 Default Value

| | |
|---|---|
| **System Identifier** | ae0b3b8e-f6fa-49e7-b411-d03bda7013f4 |
| **Title** | **Default Value** |
| **Description** | The default value for a metadata element when it is first applied to an entity |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Limited to the maximum number of occurrences allowed for the metadata element in the metadata element definition |
| **Modifiable?** | Yes |
| **Entity Reference?** | Dependent on the settings in the metadata element definition |
| **Refers To Type** | Dependent on the settings in the metadata element definition |
| **Datatype** | Dependent on the settings in the metadata element definition |
| **Textual?** | Dependent on the settings in the metadata element definition |

### M14.4.14 Deleted Event Function Definition Identifier

| | |
|---|---|
| **System Identifier** | eec57404-3539-4c24-b3f5-bf55f1a8f99e |

| | |
|---|---|
| **Title** | **Deleted Event Function Definition Identifier** |
| **Description** | Function or functions indicating the type or types of event that have been deleted from the event history of the entity |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depends on the number and types of events that were deleted) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Function Definition (**E14.2.9**) |
| **Datatype** | UUID |

### M14.4.15 Deleted Metadata Element Definition Identifier

| | |
|---|---|
| **System Identifier** | bb2a7c80-a367-4734-9676-a2bb2fab2e26 |
| **Title** | **Deleted Metadata Element Definition Identifier** |
| **Description** | Metadata element definition or definitions indicating the type or types of metadata elements that have been deleted from the entity |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depends on the number and types of metadata elements that were deleted) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Metadata Element Definition (**E14.2.11**) |
| **Datatype** | UUID |

### M14.4.16 Description

| | |
|---|---|
| **System Identifier** | c7e6aad5-27dd-47a4-97a6-ab541b47b37f |
| **Title** | **Description** |
| **Description** | Description of the entity |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Class (**E14.2.2**)<br>• Component (**E14.2.3**)<br>• Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**)<br>• Entity Type (**E14.2.7**)<br>• Function Definition (**E14.2.9**)<br>• Group (**E14.2.10**)<br>• Metadata Element Definition (**E14.2.11**)<br>• Record (**E14.2.12**)<br>• Role (**E14.2.13**)<br>• Service (**E14.2.14**)<br>• Template (**E14.2.15**)<br>• User (**E14.2.16**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | No |
| **Datatype** | Text |
| **Textual?** | Yes (must be accompanied by a language identifier) |

### M14.4.17 Destroyed Timestamp

| | |
|---|---|
| **System Identifier** | dbcf3076-c193-41fb-8043-635c8bc299b2 |
| **Title** | **Destroyed Timestamp** |
| **Description** | System set date and time when the entity was destroyed |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Class (**E14.2.2**)<br>• Component (**E14.2.3**)<br>• Contextual Metadata Element Definition (**E14.2.4**) |

| | |
|---|---|
| | • Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**)<br>• Group (**E14.2.10**)<br>• Record (**E14.2.12**)<br>• Role (**E14.2.13**)<br>• Template (**E14.2.15**)<br>• User (**E14.2.16**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Timestamp |

### M14.4.18 Disposal Action Code

| | |
|---|---|
| **System Identifier** | 83de3514-3888-4eff-88bd-d58cfb043cc7 |
| **Title** | **Disposal Action Code** |
| **Description** | Code describing the action to be taken on disposal of the record |
| **Entity Type** | • Disposal Schedule (**E14.2.6**)<br>• Record (**E14.2.12**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (for disposal schedules prior to their first use)<br>No (for records and disposal schedules after their use) |
| **Entity Reference?** | No |
| **Datatype** | Code |
| **Valid values** | 0 = RETAIN ON HOLD (not valid for disposal schedules)<br>1 = RETAIN PERMANENTLY<br>2 = REVIEW<br>3 = TRANSFER<br>4 = DESTROY |

| | |
|---|---|
| ***Usage notes*** | *For records this value is updated automatically by the system* |

### M14.4.19 Disposal Action Due Date

| | |
|---|---|
| **System Identifier** | 92e083e8-fe2b-4594-a287-685fad30c2b7 |
| **Title** | **Disposal Action Due Date** |
| **Description** | The calculated date that the record is due for disposal |
| **Entity Type** | Record (**E14.2.12**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No (updated automatically) |
| **Entity Reference?** | No |
| **Datatype** | Date |

### M14.4.20 Disposal Confirmation Due Date

| | |
|---|---|
| **System Identifier** | 39971796-dfd3-47b4-b36c-b2d5d0d770c7 |
| **Title** | **Disposal Confirmation Due Date** |
| **Description** | The calculated date by which confirmation of carrying out the disposal action is due |
| **Entity Type** | Record (**E14.2.12**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No (updated automatically) |
| **Entity Reference?** | No |
| **Datatype** | Date |

### M14.4.21 Disposal Overdue Alert Timestamp

| | |
|---|---|
| **System Identifier** | df4d965b-6e65-4131-b58b-0840c0bfd69d |
| **Title** | **Disposal Overdue Alert Timestamp** |
| **Description** | System set date and time when an alert was sent because the record was overdue for disposal |
| **Entity Type** | Record (**E14.2.12**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Date |

### M14.4.22 Disposal Schedule Identifier

| | |
|---|---|
| **System Identifier** | aee900e0-bb8b-4b9b-b714-738ed3bc14a3 |
| **Title** | **Disposal Schedule Identifier** |
| **Description** | The disposal schedule for the record when the inherited disposal schedule from the record's classification has been overridden |
| **Entity Type** | Record (**E14.2.12**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Disposal Schedule (**E14.2.6**) |
| **Datatype** | UUID |
| ***Usage notes*** | *Only used when the default disposal schedule is overridden* |

### M14.4.23 Duplicate Identifier

| | |
|---|---|
| **System Identifier** | b887bda3-f34d-4ddd-8069-630567d949d2 |
| **Title** | **Duplicate Identifier** |
| **Description** | Reference to another entity that has been created by duplicating the record, component or event, and is an exact copy up to the event of duplication, with an identical provenance |
| **Entity Type** | • Component (**E14.2.3**)<br>• Event (**E14.2.8**)<br>• Record (**E14.2.12**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | • Component (**E14.2.3**)<br>• Event (**E14.2.8**)<br>• Record (**E14.2.12**) |
| **Datatype** | UUID |

### M14.4.24 Entity Reference Type Identifier

| | |
|---|---|
| **System Identifier** | 04ad1126-6673-4a73-a0c3-34871bb49905 |
| **Title** | **Entity Reference Type Identifier** |
| **Description** | Restricts the values of a metadata element that refers to another entity to entities of a particular type, or types |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 0 (not present if the metadata element definition is not an entity reference or there are no restrictions on the types of entities that the element can refer to) |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes (for contextual metadata prior to their first use)<br>No (for system metadata and after the definition has been applied) |

| | |
|---|---|
| **Entity Reference?** | Yes |
| **Refers To Type** | Entity Type (**E14.2.7**) |
| **Datatype** | UUID |
| ***Usage notes*** | *Used when the Is Entity Reference Flag (**M14.4.45**) is set* |

### M14.4.25 Event Comment

| | |
|---|---|
| **System Identifier** | 1f82121e-dcb9-475b-814e-6719031ca30b |
| **Title** | **Event Comment** |
| **Description** | Comment giving additional detailed information about an event, or the reason for an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Text |
| **Textual?** | Yes (must be accompanied by a language identifier) |

### M14.4.26 Event Function Identifier

| | |
|---|---|
| **System Identifier** | ea21de8f-1a0c-4bc5-939a-dc06dcff44c1 |
| **Title** | **Event Function Identifier** |
| **Description** | The function that was performed to generate the event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |

| | |
|---|---|
| **Entity Reference?** | Yes |
| **Refers To Type** | Function Definition (**E14.2.9**) |
| **Datatype** | UUID |

### M14.4.27 Event Timestamp

| | |
|---|---|
| **System Identifier** | b51c08ed-d332-4bb5-974c-7cee3cb5c753 |
| **Title** | **Event Timestamp** |
| **Description** | System set date and time for when a function was performed |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Timestamp |

### M14.4.28 Export Commencing Timestamp

| | |
|---|---|
| **System Identifier** | a252f442-6ab5-4325-94e5-00adaf908614 |
| **Title** | **Export Commencing Timestamp** |
| **Description** | System set date and time for when an export commenced |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Timestamp |

| | |
|---|---|
| ***Usage notes*** | *See R11.4.6* |

### M14.4.29 Export Completed Timestamp

| | |
|---|---|
| **System Identifier** | 4220eaa9-fc78-4e33-ae94-0394824d4616 |
| **Title** | **Export Completed Timestamp** |
| **Description** | System set date and time for when an export completed |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Timestamp |
| ***Usage notes*** | *See R11.4.6* |

### M14.4.30 Export Identifier

| | |
|---|---|
| **System Identifier** | 885f5945-e724-466b-a84f-6ae3c603768b |
| **Title** | **Export Identifier** |
| **Description** | The system generated identifier for an individual export operation, used in the event history of all entities that are successfully exported |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | UUID |
| ***Usage notes*** | *See R11.4.4* |

### M14.4.31 Exported In Full Flag

| | |
|---|---|
| **System Identifier** | 633bd60f-50df-4709-ab22-32cf63790598 |
| **Title** | **Exported In Full Flag** |
| **Description** | Flag indicating whether an entity was exported in full or as a placeholder |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Boolean |

### M14.4.32 First Used Timestamp

| | |
|---|---|
| **System Identifier** | a937d097-856d-472f-836b-a2fae1811550 |
| **Title** | **First Used Timestamp** |
| **Description** | System generated date and time indicating when an entity was first used; generally taken as the last time it can be modified or deleted without formally destroying it |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Class (**E14.2.2**)<br>• Contextual Metadata Element Definition (**E14.2.4**)<br>• Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**)<br>• Group (**E14.2.10**)<br>• Role (**E14.2.13**)<br>• Template (**E14.2.15**)<br>• User (**E14.2.16**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |

| | |
|---|---|
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Timestamp |

### M14.4.33 Function Definition Identifier

| | |
|---|---|
| **System Identifier** | 55fcb00b-1d7a-4b6b-899c-3a9a9762dbf8 |
| **Title** | **Function Definition Identifier** |
| **Description** | A function that is included in a role to allow users and groups assigned that role to perform the function |
| **Entity Type** | Role (**E14.2.13**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Function Definition (**E14.2.9**) |
| **Datatype** | UUID |

### M14.4.34 Generate Event Flag

| | |
|---|---|
| **System Identifier** | 50a7cd32-ed4f-47b6-bcbb-bc1112e06eb9 |
| **Title** | **Generate Event Flag** |
| **Description** | Flag indicating whether an event should be generated by the system, when a function is performed |
| **Entity Type** | Function Definition (**E14.2.9**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |

| | |
|---|---|
| **Entity Reference?** | No |
| **Datatype** | Boolean |

### M14.4.35 Granted Role Identifier

| | |
|---|---|
| **System Identifier** | 1f50d91a-dac6-42ec-bb6e-d5dd5c0f4770 |
| **Title** | **Granted Role Identifier** |
| **Description** | Indicator in an event of a role that was awarded to the participating user or group |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Role (**E14.2.13**) |
| **Datatype** | UUID |

### M14.4.36 Group Identifier

| | |
|---|---|
| **System Identifier** | 1d32086d-3855-4815-a294-f7d1977ce03a |
| **Title** | **Group Identifier** |
| **Description** | Group which the user is a member of |
| **Entity Type** | User (**E14.2.16**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Group (**E14.2.10**) |

| | |
|---|---|
| **Datatype** | UUID |

### M14.4.37 Held Aggregation Identifier

| | |
|---|---|
| **System Identifier** | 93856ba9-a0dd-4cfa-b8a9-edf031774076 |
| **Title** | **Held Aggregation Identifier** |
| **Description** | Aggregation which is associated with a disposal hold and as a result no records in the aggregation, or its descendant aggregations, may be destroyed |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Aggregation (**E14.2.1**) |
| **Datatype** | UUID |

### M14.4.38 Held Class Identifier

| | |
|---|---|
| **System Identifier** | 29abe491-0c58-4816-97c2-8f60be62d9fa |
| **Title** | **Held Class Identifier** |
| **Description** | Class which is associated with a disposal hold and as a result no records which have been classified with the class, may be destroyed |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Class (**E14.2.2**) |

| | |
|---|---|
| **Datatype** | UUID |

### M14.4.39 Held Record Identifier

| | |
|---|---|
| **System Identifier** | 374957cb-23ce-4987-81e9-5ccc00cfe9e3 |
| **Title** | **Held Record Identifier** |
| **Description** | Record which has been associated with a disposal hold and, as a result, may not be destroyed |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Record (**E14.2.12**) |
| **Datatype** | UUID |

### M14.4.40 Historical Date/Time

| | |
|---|---|
| **System Identifier** | 8274fbf7-3145-4c23-a748-6c4682d6b7ed |
| **Title** | **Historical Date/Time** |
| **Description** | Date and time for which a report was run to obtain historical information about a user, group or role |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Date/Time |

### M14.4.41 Implements Module Identifier

| | |
|---|---|
| **System Identifier** | 92bddc26-aae5-493d-9278-94fbb00a2e44 |
| **Title** | **Implements Module Identifier** |
| **Description** | Compliance indicator providing assurance that the system implements the identified module of MoReq2010® |
| **Entity Type** | Service (**E14.2.14**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | UUID |
| ***Usage notes*** | *Each major version of a MoReq2010® module has a different unique identifier* |

### M14.4.42 Implements Service Identifier

| | |
|---|---|
| **System Identifier** | 3960b7a5-624e-41e1-9976-248895b9352e |
| **Title** | **Implements Service Identifier** |
| **Description** | Compliance indicator providing assurance that the system implements the core service of MoReq2010® that is identified |
| **Entity Type** | Service (**E14.2.14**) |
| **Min Occurs** | 1 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | UUID |
| ***Usage notes*** | *Each major version of a MoReq2010® core service has a different unique identifier* |

### M14.4.43 Include Inherited Roles Flag

| | |
|---|---|
| **System Identifier** | 4ad73ab1-4df8-40b7-b863-f8d63a584037 |
| **Title** | **Include Inherited Roles Flag** |
| **Description** | Flag indicating whether administrative and/or non-administrative roles are inherited by an access control list |
| **Entity Type** | Access Control List (**D14.3.2**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | No |
| **Datatype** | Boolean |
| ***Usage notes*** | • *When the flag is set both administrative and non-administrative roles will be inherited from the parent's access control list*<br>• *When the flag is cleared only administrative roles will be inherited (administrative roles may not be overridden)*<br>• *The specific rules of inheritance are given in the rationale to* ***R4.5.11*** |

### M14.4.44 Is Administrative Role Flag

| | |
|---|---|
| **System Identifier** | d2e3b3da-a109-4ec0-ab26-5e6f53ad3673 |
| **Title** | **Is Administrative Role Flag** |
| **Description** | Flag indicating that a role is an administrative role and will, as a result, always be inherited |
| **Entity Type** | Role (**E14.2.13**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (for roles prior to their first use)<br>No (once a role has been granted) |
| **Entity Reference?** | No |

| | |
|---|---|
| **Datatype** | Boolean |
| ***Usage notes*** | *See also Include Inherited Roles Flag (**M14.4.43**)* |

### M14.4.45 Is Entity Reference Flag

| | |
|---|---|
| **System Identifier** | b52c2bbf-f27b-41c9-8885-3d906b4bd275 |
| **Title** | **Is Entity Reference Flag** |
| **Description** | Flag indicating whether a metadata element contains a reference to an entity or a different datatype value |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (for contextual metadata prior to their first use)<br>No (for system metadata and after the definition has been applied) |
| **Entity Reference?** | No |
| **Datatype** | Boolean |
| ***Usage notes*** | *See also Entity Reference Type Identifier (**M14.4.24**) and Datatype (**M14.4.10**)* |

### M14.4.46 Is Modifiable Flag

| | |
|---|---|
| **System Identifier** | 7ff6a66e-b137-417c-aab0-aa327a0b119b |
| **Title** | **Is Modifiable Flag** |
| **Description** | Flag indicating whether a metadata element is modifiable by users |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (for contextual metadata prior to their first use)<br>No (for system metadata and after the definition has been applied) |

| | |
|---|---|
| **Entity Reference?** | No |
| **Datatype** | Boolean |

### M14.4.47 Is Textual Flag

| | |
|---|---|
| **System Identifier** | 7e5ec95d-423c-4ac9-ac94-a835d3713e91 |
| **Title** | **Is Textual Flag** |
| **Description** | Flag indicating whether a text based metadata element should be accompanied by a language identifier |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (for contextual metadata prior to their first use)<br>No (for system metadata and after the definition has been applied) |
| **Entity Reference?** | No |
| **Datatype** | Boolean |
| ***Usage notes*** | *Should be set for metadata elements where descriptive text is stored in the user's own language* |

### M14.4.48 Last Addition Timestamp

| | |
|---|---|
| **System Identifier** | 71a6cf2d-c039-4b27-bd31-dcba471081ac |
| **Title** | **Last Addition Timestamp** |
| **Description** | System set date and time indicating when the most recent record or child aggregation was added to the parent aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |

| | |
|---|---|
| **Entity Reference?** | No |
| **Datatype** | Timestamp |

### M14.4.49 Last Review Comment

| | |
|---|---|
| **System Identifier** | 7036ec2e-644a-4309-9690-178690a6bc4f |
| **Title** | **Last Review Comment** |
| **Description** | Comment made by the user who last reviewed the record explaining the disposal decision made by that review |
| **Entity Type** | Record (**E14.2.12**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Text |
| **Textual?** | Yes (must be accompanied by a language identifier) |
| ***Usage notes*** | *Used with Last Reviewed Timestamp (**M14.4.50**)* |

### M14.4.50 Last Reviewed Timestamp

| | |
|---|---|
| **System Identifier** | dbbeb8a5-b019-46e1-983e-5f728f1dcc3a |
| **Title** | **Last Reviewed Timestamp** |
| **Description** | System set date and time of when the last review was completed |
| **Entity Type** | Record (**E14.2.12**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |

| | |
|---|---|
| **Datatype** | Timestamp |
| ***Usage notes*** | *Used with Last Review Comment (**M14.4.49**)* |

### M14.4.51 Mandate

| | |
|---|---|
| **System Identifier** | a6986d60-1257-4046-bafe-fef680d5dc6c |
| **Title** | **Mandate** |
| **Description** | Textual reference to a legal or other instrument that provides the authority for a disposal schedule or a disposal hold |
| **Entity Type** | • Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | No |
| **Datatype** | Text |
| **Textual?** | Yes (must be accompanied by a language identifier) |

### M14.4.52 Max Levels Of Aggregation

| | |
|---|---|
| **System Identifier** | d0da657f-9345-4965-b2a8-17a0c300a896 |
| **Title** | **Max Levels Of Aggregation** |
| **Description** | The maximum number of levels of aggregation allowed below a root aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | No |

| | |
|---|---|
| **Datatype** | Positive Integer |

### M14.4.53 Max Occurs

| | |
|---|---|
| **System Identifier** | 4a69e008-50df-4506-b9a7-d6279b0c21d5 |
| **Title** | **Max Occurs** |
| **Description** | The maximum number of values that can be assigned to a metadata element for a single entity |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (for contextual metadata prior to their first use)<br>No (for system metadata and after the definition has been applied) |
| **Entity Reference?** | No |
| **Datatype** | Positive Integer |
| ***Usage notes*** | • *Used with Min Occurs (**M14.4.56**)*<br>• *Where this value is not set it indicates that the metadata element is a list with an unlimited number of values* |

### M14.4.54 MCRS Certification Identifier

| | |
|---|---|
| **System Identifier** | bf8efec0-e182-47ec-bdfa-a83a24c602c8 |
| **Title** | **MCRS Certification Identifier** |
| **Description** | Compliance indicator giving the DLM Forum issued identifier for a certificate of compliance against MoReq2010® |
| **Entity Type** | Service (**E14.2.14**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | No |

| | |
|---|---|
| **Entity Reference?** | No |
| **Datatype** | UUID |
| ***Usage notes*** | *Only the DLM Forum may issue a certificate of compliance against MoReq2010®* |

### M14.4.55 Metadata Element Definition Identifier

| | |
|---|---|
| **System Identifier** | 8afb23cd-a741-4be0-97fd-f76aef411503 |
| **Title** | **Metadata Element Definition Identifier** |
| **Description** | Reference in an event to the metadata element of an entity that was changed when a function was performed |
| **Entity Type** | Metadata Change Entry (**D14.3.3**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Metadata Element Definition (**E14.2.11**) |
| **Datatype** | UUID |
| ***Usage notes*** | *Used in a metadata change entry in combination with New Value (**M14.4.57**) and Previous Value (**M14.4.85**)* |

### M14.4.56 Min Occurs

| | |
|---|---|
| **System Identifier** | 6e182700-4679-4ebf-938a-ef110828cba6 |
| **Title** | **Min Occurs** |
| **Description** | The minimum number of values that can be assigned to a metadata element for a single entity |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 1 |

| | |
|---|---|
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (for contextual metadata prior to their first use)<br>No (for system metadata and after the definition has been applied) |
| **Entity Reference?** | No |
| **Datatype** | Positive Integer |
| ***Usage notes*** | • *Used with Max Occurs (**M14.4.53**)*<br>• *Where this value is set to zero it indicates a optional metadata element, where it is set to one a value is mandatory* |

### M14.4.57 New Value

| | |
|---|---|
| **System Identifier** | d3d4f2ae-43f0-4567-99ae-ddd58e6cf2dd |
| **Title** | **New Value** |
| **Description** | The value of a metadata element after a function was performed |
| **Entity Type** | Metadata Change Entry (**D14.3.3**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | Dependent on the settings in the metadata element definition |
| **Refers To Type** | Dependent on the settings in the metadata element definition |
| **Datatype** | Dependent on the settings in the metadata element definition |
| **Textual?** | Dependent on the settings in the metadata element definition |
| ***Usage notes*** | • *Used in a metadata change entry with Previous Value (**M14.4.85**) in conjunction with Metadata Element Definition Identifier (**M14.4.55**)*<br>• *Where New Value does not occur it indicates that the value of the metadata element was deleted* |

### M14.4.58 Overdue Disposal Action Code

| | |
|---|---|
| **System Identifier** | 60027dfb-2e3d-4456-b341-54737e83243c |

| | |
|---|---|
| **Title** | **Overdue Disposal Action Code** |
| **Description** | The disposal action that was overdue on the record at the time an alert was raised |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Code |
| **Valid values** | Same as for Disposal Action Code (**M14.4.18**) |
| ***Usage notes*** | *Used with Overdue Disposal Action Due Date (**M14.4.59**) and Overdue Disposal Confirmation Due Date (**M14.4.60**)* |

### M14.4.59 Overdue Disposal Action Due Date

| | |
|---|---|
| **System Identifier** | 75491c12-0c90-4f99-85ad-b583df082b45 |
| **Title** | **Overdue Disposal Action Due Date** |
| **Description** | The due date that was set for the disposal action that was overdue on a record at the time an alert was raised |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Date/Time |
| ***Usage notes*** | *Used with Overdue Disposal Action Code (**M14.4.58**) and Overdue Disposal Confirmation Due Date (**M14.4.60**)* |

### M14.4.60 Overdue Disposal Confirmation Due Date

| | |
|---|---|
| **System Identifier** | 1ac9514a-7e0e-4fd8-a543-4956c22f4089 |
| **Title** | **Overdue Disposal Confirmation Due Date** |
| **Description** | The due date that was set for confirmation of a disposal action that was overdue, causing an alert to be raised |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Date/Time |
| ***Usage notes*** | *Used with Overdue Disposal Action Code (**M14.4.58**) and Overdue Disposal Action Due Date (**M14.4.59**)* |

### M14.4.61 Originated Date/Time

| | |
|---|---|
| **System Identifier** | 50864023-dc92-4b1e-8e8a-66c284d40942 |
| **Title** | **Originated Date/Time** |
| **Description** | The date and time of origin of a record or other entity which may vary from the creation date of the entity in the system |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Class (**E14.2.2**)<br>• Component (**E14.2.3**)<br>• Contextual Metadata Element Definition (**E14.2.4**)<br>• Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**)<br>• Group (**E14.2.10**)<br>• Record (**E14.2.12**)<br>• Role (**E14.2.13**)<br>• Template (**E14.2.15**)<br>• User (**E14.2.16**) |
| **Min Occurs** | 1 |

| | |
|---|---|
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | No |
| **Datatype** | Date/Time |

### M14.4.62 Owner Information

| | |
|---|---|
| **System Identifier** | ae99f4f8-3edf-4874-aea2-ea50fa7a73f9 |
| **Title** | **Owner Information** |
| **Description** | Information provided by the owner of a system that may be used to identify the originating system when entities are later exported to another system |
| **Entity Type** | Service (**E14.2.14**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | No |
| **Datatype** | Text |
| **Textual?** | Yes (must be accompanied by a language identifier) |

### M14.4.63 Parent Aggregation Identifier

| | |
|---|---|
| **System Identifier** | a74597a5-190a-4874-a350-083ac030aa55 |
| **Title** | **Parent Aggregation Identifier** |
| **Description** | The parent aggregation for a child aggregation or record |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Record (**E14.2.12**) |
| **Min Occurs** | 0 (for root aggregations)<br>1 (for child aggregations and records) |

| | |
|---|---|
| **Max Occurs** | 0 (for root aggregations)<br>1 (for child aggregations and records) |
| **Modifiable?** | Yes (by moving the child aggregation or record) |
| **Entity Reference?** | Yes |
| **Refers To Type** | Aggregation (**E14.2.1**) |
| **Datatype** | UUID |

### M14.4.64 Participating Aggregation Identifier

| | |
|---|---|
| **System Identifier** | daafbb4c-3183-40bd-9eff-759e88c725fc |
| **Title** | **Participating Aggregation Identifier** |
| **Description** | Aggregation that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Aggregation (**E14.2.1**) |
| **Datatype** | UUID |

### M14.4.65 Participating Class Identifier

| | |
|---|---|
| **System Identifier** | f046bdfa-5ef5-4217-9c4f-e24d0f15ea9a |
| **Title** | **Participating Class Identifier** |
| **Description** | Class that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |

| | |
|---|---|
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Class (**E14.2.2**) |
| **Datatype** | UUID |

### M14.4.66 Participating Component Identifier

| | |
|---|---|
| **System Identifier** | 0686271b-b41f-40db-be27-9e6626b9ee54 |
| **Title** | **Participating Component Identifier** |
| **Description** | Component that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Component (**E14.2.3**) |
| **Datatype** | UUID |

### M14.4.67 Participating Disposal Hold Identifier

| | |
|---|---|
| **System Identifier** | f9553093-24cb-4bd4-920e-c1341572b6f5 |
| **Title** | **Participating Disposal Hold Identifier** |
| **Description** | Disposal hold that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |

| | |
|---|---|
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Disposal Hold (**E14.2.5**) |
| **Datatype** | UUID |

### M14.4.68 Participating Disposal Schedule Identifier

| | |
|---|---|
| **System Identifier** | dfaf8f03-b7eb-49c0-b889-2c9b61ce1e3c |
| **Title** | **Participating Disposal Schedule Identifier** |
| **Description** | Disposal schedule that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Disposal Schedule (**E14.2.6**) |
| **Datatype** | UUID |

### M14.4.69 Participating Duplicate Identifier

| | |
|---|---|
| **System Identifier** | 51afb74e-8729-46be-aac2-761c56e1b7d1 |
| **Title** | **Participating Duplicate Identifier** |
| **Description** | Duplicate entity that was created by duplicating a record or a component |
| **Entity Type** | • Event (**E14.2.8**) |

| | |
|---|---|
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (normally only one duplicate will be made at a time but this allows for two or more duplicates to be created simultaneously) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | • Component (**E14.2.3**)<br>• Record (**E14.2.12**) |
| **Datatype** | UUID |

### M14.4.70 Participating Entity Type Identifier

| | |
|---|---|
| **System Identifier** | 7acf7c1d-d537-4548-a24d-ce92cf7cd68b |
| **Title** | **Participating Entity Type Identifier** |
| **Description** | Entity type that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Entity Type (**E14.2.7**) |
| **Datatype** | UUID |

### M14.4.71 Participating Event Identifier

| | |
|---|---|
| **System Identifier** | b41e6d9c-c19d-41c1-8969-3090306b987f |
| **Title** | **Participating Event Identifier** |
| **Description** | Event that is a participating entity in the function that was performed that generated an event |

| | |
|---|---|
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Event (**E14.2.8**) |
| **Datatype** | UUID |

### M14.4.72 Participating Function Definition Identifier

| | |
|---|---|
| **System Identifier** | 56ee4c6d-ad65-468f-9de2-b9d5cf8b0557 |
| **Title** | **Participating Function Definition Identifier** |
| **Description** | Function definition that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Function Definition (**E14.2.9**) |
| **Datatype** | UUID |

### M14.4.73 Participating Group Identifier

| | |
|---|---|
| **System Identifier** | a6bf28b7-334b-465f-837b-34d8d09c7f2d |
| **Title** | **Participating Group Identifier** |
| **Description** | Group that is a participating entity in the function that was performed that generated an event |

| | |
|---|---|
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Group (**E14.2.10**) |
| **Datatype** | UUID |

### M14.4.74 Participating Metadata Element Definition Identifier

| | |
|---|---|
| **System Identifier** | bbbef444-bf3c-49ba-8560-0b5dfaeb8a88 |
| **Title** | **Participating Metadata Element Definition Identifier** |
| **Description** | Metadata element definition that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Metadata Element Definition (**E14.2.11**) |
| **Datatype** | UUID |

### M14.4.75 Participating New Parent Identifier

| | |
|---|---|
| **System Identifier** | 236e52dd-b117-4b6a-b60b-286d5dc5acd0 |
| **Title** | **Participating New Parent Identifier** |
| **Description** | The new parent for a child entity that has been created or moved into it |
| **Entity Type** | Event (**E14.2.8**) |

| | |
|---|---|
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Aggregation (**E14.2.1**) *or other (as specified by the function definition)* |
| **Datatype** | UUID |
| ***Usage notes*** | *In the core services this metadata element is used exclusively for aggregations, however it may be reused by other modules for functions related to different hierarchical structures* |

### M14.4.76 Participating Previous Parent Identifier

| | |
|---|---|
| **System Identifier** | 736b0697-2078-4e50-b740-a52d4f16f18d |
| **Title** | **Participating Previous Parent Identifier** |
| **Description** | Parent entity from which a child entity was moved |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Aggregation (**E14.2.1**) *or other (as specified by the function definition)* |
| **Datatype** | UUID |
| ***Usage notes*** | *In the core services this metadata element is used exclusively for aggregations, however it may be reused by other modules for functions related to different hierarchical structures* |

### M14.4.77 Participating Record Identifier

| | |
|---|---|
| **System Identifier** | 74f1568c-bde8-45aa-895a-ac600c2de1c7 |
| **Title** | **Participating Record Identifier** |
| **Description** | Record that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Record (**E14.2.**) |
| **Datatype** | UUID |

### M14.4.78 Participating Role Identifier

| | |
|---|---|
| **System Identifier** | a55947ec-b23e-413a-a840-21566aae0b4b |
| **Title** | **Participating Role Identifier** |
| **Description** | Role that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Role (**E14.2.13**) |
| **Datatype** | UUID |

### M14.4.79 Participating Service Identifier

| | |
|---|---|
| **System Identifier** | a5d104c2-e343-4c2b-82a3-8a30046e0e07 |
| **Title** | **Participating Service Identifier** |
| **Description** | Service that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Service (**E14.2.14**) |
| **Datatype** | UUID |

### M14.4.80 Participating Template Identifier

| | |
|---|---|
| **System Identifier** | 01afa124-8a2b-422b-888a-db7829564b4f |
| **Title** | **Participating Template Identifier** |
| **Description** | Template that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Template (**E14.2.15**) |
| **Datatype** | UUID |

### M14.4.81 Participating User Identifier

| | |
|---|---|
| **System Identifier** | 35d1c4e2-4821-4e23-add8-fa14b9e38ca5 |
| **Title** | **Participating User Identifier** |
| **Description** | User that is a participating entity in the function that was performed that generated an event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited (depending on the function that was performed) |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | User (**E14.2.16**) |
| **Datatype** | UUID |

### M14.4.82 Participating User Or Group Identifier

| | |
|---|---|
| **System Identifier** | 2c159083-3b5b-4119-8bbd-ae39b2577ca2 |
| **Title** | **Participating User Or Group Identifier** |
| **Description** | User or group that is a participating entity in an event generated by granting or rescinding roles |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | • Group (**E14.2.10**)<br>• User (**E14.2.16**) |
| **Datatype** | UUID |

### M14.4.83 Performed By User Identifier

| | |
|---|---|
| **System Identifier** | ee568750-32b5-4441-b67b-d01ae9df5820 |
| **Title** | **Performed By User Identifier** |
| **Description** | The user that performed the function that generated the event |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 (if no user is specified then the function was performed by the system) |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | User (**E14.2.16**) |
| **Datatype** | UUID |

### M14.4.84 Presentation Order

| | |
|---|---|
| **System Identifier** | 92c9459d-aa29-4784-8646-6fc663d08c87 |
| **Title** | **Presentation Order** |
| **Description** | A value used for determining the order in which items, such as metadata elements, should be logically presented, particularly by systems that are not the originating system |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | No |
| **Datatype** | Positive Integer |
| ***Usage notes*** | • *The values given for Presentation Order do not have to be contiguous, but they must be unique, so that no two entities have the same value*<br>• *An entity with a lower Presentation Order value will be presented first before an entity with a higher Presentation Order* |

### M14.4.85 Previous Value

| | |
|---|---|
| **System Identifier** | ce3ec75a-7193-4fd2-9811-1c70419c7185 |
| **Title** | **Previous Value** |
| **Description** | The value of a metadata element prior to a function being performed |
| **Entity Type** | Metadata Change Entry (**D14.3.3**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | Dependent on the settings in the metadata element definition |
| **Refers To Type** | Dependent on the settings in the metadata element definition |
| **Datatype** | Dependent on the settings in the metadata element definition |
| **Textual?** | Dependent on the settings in the metadata element definition |
| ***Usage notes*** | • *Used in a metadata change entry with New Value (**M14.4.57**) in conjunction with Metadata Element Definition Identifier (**M14.4.55**)*<br>• *Where Previous Value does not occur it indicates that the metadata element value was added to the entity* |

### M14.4.86 Record Identifier

| | |
|---|---|
| **System Identifier** | 87daefe3-b429-4b7d-96ea-5dc9eabd8f56 |
| **Title** | **Record Identifier** |
| **Description** | The record that the component belongs to |
| **Entity Type** | Component (**E14.2.3**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |

| | |
|---|---|
| **Refers To Type** | Record (**E14.2.12**) |
| **Datatype** | UUID |

### M14.4.87 Rescinded Role Identifier

| | |
|---|---|
| **System Identifier** | 6c20a22d-0ca8-4dbb-a932-c950a1fce1e5 |
| **Title** | **Rescinded Role Identifier** |
| **Description** | Indicator in an event of a role that was rescinded from the participating user or group |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | No |
| **Entity Reference?** | Yes |
| **Refers To Type** | Role (**E14.2.13**) |
| **Datatype** | UUID |

### M14.4.88 Retain On Destruction Flag

| | |
|---|---|
| **System Identifier** | 222811d2-8c74-4f47-ae3f-c68698f55e2c |
| **Title** | **Retain On Destruction Flag** |
| **Description** | Flag that indicates whether an event generated by performing a particular function, or a metadata element belonging to a particular metadata element definition, should be deleted when the entity it belongs to is destroyed |
| **Entity Type** | • Function Definition (**E14.2.9**)<br>• Metadata Element Definition (**E14.2.11**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |

| | |
|---|---|
| **Entity Reference?** | No |
| **Datatype** | Boolean |

### M14.4.89 Retention Period Duration Number

| | |
|---|---|
| **System Identifier** | 1cf79293-5818-4141-876e-761448ae465b |
| **Title** | **Retention Period Duration Number** |
| **Description** | Number of days, weeks, months or years specified for retaining a record after the retention period is triggered |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (until disposal schedule is first used)<br>No (once disposal schedule has been used) |
| **Entity Reference?** | No |
| **Datatype** | Positive Integer |
| ***Usage notes*** | *Used with Retention Period Interval Code (**M14.4.90**)* |

### M14.4.90 Retention Period Interval Code

| | |
|---|---|
| **System Identifier** | 47aa0949-37a9-4f4f-ab0e-f33b0dd7d5fb |
| **Title** | **Retention Period Interval Code** |
| **Description** | Unit of measurement of time as either days, weeks, months or years |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (until disposal schedule is first used)<br>No (once disposal schedule has been used) |

| | |
|---|---|
| **Entity Reference?** | No |
| **Datatype** | Code |
| **Valid values** | 0 = NO RETENTION PERIOD<br>1 = DAYS<br>2 = WEEKS<br>3 = MONTHS<br>4 = YEARS |
| ***Usage notes*** | *Used with Retention Period Duration Number (**M14.4.89**)* |

### M14.4.91 Retention Period Offset Code

| | |
|---|---|
| **System Identifier** | 3882b1a6-48d7-45a2-a539-5871856a1b95 |
| **Title** | **Retention Period Offset Code** |
| **Description** | Offset for the disposal of records if they fall due for disposal in a particular period |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes (until disposal schedule is first used)<br>No (once disposal schedule has been used) |
| **Entity Reference?** | Yes |
| **Refers To Type** | x (**E14.2.**) |
| **Datatype** | Code |
| **Valid values** | 0 = NO OFFSET<br>1 = START OF NEXT MONTH<br>2 = START OF NEXT QUARTER<br>3 = START OF SPECIFIED MONTH |
| ***Usage notes*** | *START OF SPECIFIED MONTH is used with Retention Period Offset Month Code (**M14.4.92**)* |

### M14.4.92 Retention Period Offset Month Code

| | |
|---|---|
| **System Identifier** | 557d941f-4be7-4dad-8113-39757547530a |
| **Title** | **Retention Period Offset Month Code** |
| **Description** | Month to offset the disposal of records to if they fall due for disposal in a particular period |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (until disposal schedule is first used)<br>No (once disposal schedule has been used) |
| **Entity Reference?** | No |
| **Datatype** | Code |
| **Valid values** | 1 = JANUARY<br>2 = FEBRUARY<br>3 = MARCH<br>4 = APRIL<br>5 = MAY<br>6 = JUNE<br>7 = JULY<br>8 = AUGUST<br>9 = SEPTEMBER<br>10 = OCTOBER<br>11 = NOVEMBER<br>12 = DECEMBER |
| ***Usage notes*** | *Used with Retention Period Offset Code (**M14.4.91**) when START OF SPECIFIED MONTH is specified.* |

### M14.4.93 Retention Start Date

| | |
|---|---|
| **System Identifier** | 199494ea-0ca4-47ef-883a-4bcc5ccdcc02 |
| **Title** | **Retention Start Date** |
| **Description** | System generated date calculated from the record's disposal schedule indicating the start of its retention period |
| **Entity Type** | Record (**E14.2.12**) |
| **Min Occurs** | 0 (not used for records with a Disposal Action Code (**M14.4.18**) of RETAIN PERMANENTLY) |
| **Max Occurs** | 1 |
| **Modifiable?** | No (automatically updated) |
| **Entity Reference?** | No |
| **Datatype** | Date |

### M14.4.94 Retention Trigger Code

| | |
|---|---|
| **System Identifier** | 30891ecb-5b36-453d-baf0-9aaa19eba4a6 |
| **Title** | **Retention Trigger Code** |
| **Description** | The specific conditions required for triggering the start of the retention period for a record |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (until disposal schedule is first used)<br>No (once disposal schedule has been used) |
| **Entity Reference?** | No |
| **Datatype** | Code |

| | |
|---|---|
| **Valid values** | 0 = FROM NOW<br>1 = FROM DATE OF LAST REVIEW<br>2 = FROM RECORD ORIGINATED DATE<br>3 = FROM AGGREGATION ORIGINATED DATE<br>4 = FROM DATE ADDED TO AGGREGATION<br>5 = FROM DATE OF LAST ADDITION TO AGGREGATION<br>6 = FROM AGGREGATION CLOSED DATE<br>7 = FROM RECORD METADATA DATE<br>8 = FROM AGGREGATION METADATA DATE |
| ***Usage notes*** | *FROM RECORD METADATA DATE and FROM AGGREGATION METADATA DATE are used in combination with Retention Trigger Element Identifier (**M14.4.95**)* |

### M14.4.95 Retention Trigger Element Identifier

| | |
|---|---|
| **System Identifier** | 72fc68f0-5437-4d8e-a719-b4c4ecd1b649 |
| **Title** | **Retention Trigger Element Identifier** |
| **Description** | The metadata element associated with a record, or its parent aggregation, which contains the trigger date for a record's retention period |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes (until disposal schedule is first used)<br>No (once disposal schedule has been used) |
| **Entity Reference?** | Yes |
| **Refers To Type** | Metadata Element Definition (**E14.2.11**) |
| **Datatype** | UUID |

| | |
|---|---|
| ***Usage notes*** | • *Used with Retention Trigger Code (**M14.4.94**) when FROM RECORD METADATA DATE and FROM AGGREGATION METADATA DATE are specified*<br>• *Must be a reference to a metadata element definition of date, date/time or timestamp* |

### M14.4.96 Role Identifier

| | |
|---|---|
| **System Identifier** | 0394b604-5865-42e2-ab23-2e983ecdf454 |
| **Title** | **Role Identifier** |
| **Description** | Role included in an access control entry so that it is granted to a user or a group |
| **Entity Type** | Access Control Entry (**D14.3.1**) |
| **Min Occurs** | 1 |
| **Max Occurs** | Unlimited (more than one role may be granted to the user or group in a single access control entry) |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Role (**E14.2.13**) |
| **Datatype** | UUID |

### M14.4.97 Scope Notes

| | |
|---|---|
| **System Identifier** | c5c12c32-d263-44a3-91ae-26f80cb75a3c |
| **Title** | **Scope Notes** |
| **Description** | Guidance to authorised users indicating how best to apply a particular entity and stating any organisational policies or constraints on its use |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Class (**E14.2.2**)<br>• Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**)<br>• Metadata Element Definition (**E14.2.11**)<br>• Role (**E14.2.13**) |

| | |
|---|---|
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | No |
| **Datatype** | Text |
| **Textual?** | Yes (must be accompanied by a language identifier) |

### M14.4.98 Search Query

| | |
|---|---|
| **System Identifier** | e7f82f8c-80de-4cb5-a28e-6761abca27b0 |
| **Title** | **Search Query** |
| **Description** | Description in an event of the search query used to search for, or report on, entities in the system |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Text |
| **Textual?** | Yes (must be accompanied by a language identifier) |
| ***Usage notes*** | *There is no specified style for how the Search Query should be described; it may be using a structured expression language or using natural language, see* ***R10.4.22*** |

### M14.4.99 Supplier Information

| | |
|---|---|
| **System Identifier** | 8276d84b-a3fe-4b49-82a4-a8a58de14a7e |
| **Title** | **Supplier Information** |

| | |
|---|---|
| **Description** | Information provided by the supplier of a system that may be used to identify the type of the originating system, including its particular hardware and software versions and configuration, especially when entities are exported to another system |
| **Entity Type** | Service (**E14.2.14**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Text |
| **Textual?** | Yes (must be accompanied by a language identifier) |

### M14.4.100 System Identifier

| | |
|---|---|
| **System Identifier** | 01806231-2c8c-4482-9bb1-8e47e747784f |
| **Title** | **System Identifier** |
| **Description** | Universally unique identifier for an entity that is generated automatically by the system and stays with the entity forever |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Class (**E14.2.2**)<br>• Component (**E14.2.3**)<br>• Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**)<br>• Entity Type (**E14.2.7**)<br>• Event (**E14.2.8**)<br>• Function Definition (**E14.2.9**)<br>• Group (**E14.2.10**)<br>• Metadata Element Definition (**E14.2.11**)<br>• Record (**E14.2.12**)<br>• Role (**E14.2.13**)<br>• Service (**E14.2.14**)<br>• Template (**E14.2.15**)<br>• User (**E14.2.16**) |
| **Min Occurs** | 1 |

| | |
|---|---|
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | UUID |

### M14.4.101 Template Class Identifier

| | |
|---|---|
| **System Identifier** | ee03c826-39ee-4cdf-b2c7-6cc0c2782967 |
| **Title** | **Template Class Identifier** |
| **Description** | Class associated with the template such that when new entities, such as aggregations or records, are created with that classification, the template will be automatically applied to them by the system, giving them additional contextual metadata elements |
| **Entity Type** | Template (**E14.2.15**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Class (**E14.2.2**) |
| **Datatype** | UUID |
| ***Usage notes*** | *Subject to the entity being of the appropriate entity type, see Template Entity Type Identifier (**M14.4.102**)* |

### M14.4.102 Template Entity Type Identifier

| | |
|---|---|
| **System Identifier** | caaaf2e6-2ed4-4a53-adba-9363385127ac |
| **Title** | **Template Entity Type Identifier** |
| **Description** | Specifies the types of entities to which the template may be applied |
| **Entity Type** | • Template (**E14.2.15**) |

| | |
|---|---|
| **Min Occurs** | 1 |
| **Max Occurs** | Unlimited (the template may apply to more than one entity type) |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Entity Type (**E14.2.7**) |
| **Datatype** | UUID |

### M14.4.103 Template Service Identifier

| | |
|---|---|
| **System Identifier** | 2dd54e70-5b60-4d5a-89be-5f967735d515 |
| **Title** | **Template Service Identifier** |
| **Description** | Service associated with the template such that when new entities are created in that service, the template will be automatically applied to them by the system, giving them additional contextual metadata elements |
| **Entity Type** | Template (**E14.2.15**) |
| **Min Occurs** | 0 |
| **Max Occurs** | Unlimited |
| **Modifiable?** | Yes |
| **Entity Reference?** | Yes |
| **Refers To Type** | Service (**E14.2.14**) |
| **Datatype** | UUID |
| ***Usage notes*** | *Subject to the entity being of the appropriate entity type, see Template Entity Type Identifier (**M14.4.102**)* |

### M14.4.104 Title

| | |
|---|---|
| **System Identifier** | 077fc367-48ba-44a8-8afb-012d05ed1a16 |
| **Title** | **Title** |

| | |
|---|---|
| **Description** | The identifying name or title of the entity |
| **Entity Type** | • Aggregation (**E14.2.1**)<br>• Class (**E14.2.2**)<br>• Component (**E14.2.3**)<br>• Disposal Hold (**E14.2.5**)<br>• Disposal Schedule (**E14.2.6**)<br>• Entity Type (**E14.2.7**)<br>• Function Definition (**E14.2.9**)<br>• Group (**E14.2.10**)<br>• Metadata Element Definition (**E14.2.11**)<br>• Record (**E14.2.12**)<br>• Role (**E14.2.13**)<br>• Service (**E14.2.14**)<br>• Template (**E14.2.15**)<br>• User (**E14.2.16**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes |
| **Entity Reference?** | No |
| **Datatype** | Text |
| **Textual?** | Yes (must be accompanied by a language identifier) |
| ***Usage notes*** | *MoReq2010® does not require that titles be unique, for example, for two records within the same aggregation – however this is generally good practice* |

### M14.4.105 Total Entities

| | |
|---|---|
| **System Identifier** | 2131929b-7c06-471d-a96c-19b3560069d8 |
| **Title** | **Total Entities** |
| **Description** | The total number of entities returned by a search or included in a report |
| **Entity Type** | Event (**E14.2.8**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |

| | |
|---|---|
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Positive Integer |

### M14.4.106 Transferred Timestamp

| | |
|---|---|
| **System Identifier** | 09c440b3-b040-465f-a79f-9dae3243cce6 |
| **Title** | **Transferred Timestamp** |
| **Description** | System set date and time indicating when the transfer of the record was confirmed |
| **Entity Type** | Record (**E14.2.12**) |
| **Min Occurs** | 0 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |
| **Datatype** | Timestamp |

### M14.4.107 User Or Group Identifier

| | |
|---|---|
| **System Identifier** | 2b1820f7-97ef-4010-8de8-f0745e0855c0 |
| **Title** | **User Or Group Identifier** |
| **Description** | Identifier for a user, or for a group, that is granted one or more roles in an access control entry |
| **Entity Type** | Access Control Entry (**D14.3.1**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | Yes (by modifying the access control list) |
| **Entity Reference?** | Yes |

| | |
|---|---|
| **Refers To Type** | • Group (**E14.2.10**)<br>• User (**E14.2.16**) |
| **Datatype** | UUID |

## 14.5 Function Definitions

### F14.5.1 Aggregation – Add Aggregation

| | |
|---|---|
| **System Identifier** | f6c7d6a4-c69e-4d33-9d4a-4137274b68da |
| **Title** | **Aggregation – Add Aggregation** |
| **Description** | Add a child aggregation to the open aggregation by moving it from the root or its previous parent |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata** | *The following metadata element belonging to the aggregation will be modified:*<br>• Last Addition Timestamp (**M14.4.48**)<br>*The following metadata element belonging to the aggregation may also be modified (if it has not been set previously):*<br>• First Used Timestamp (**M14.4.32**)<br>*The following metadata elements belonging to the participating child aggregation will be modified:*<br>• Parent Aggregation Identifier (**M14.4.63**)<br>• Aggregated Timestamp (**M14.4.1**)<br>*The following metadata element belonging to the participating child aggregation will be removed (if it exists):*<br>• Max Levels Of Aggregation (**M14.4.52**) |
| ***From functional requirement(s)*** | **R6.5.8, R6.5.14** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Previous Parent Identifier (**M14.4.76**)<br>• Participating New Parent Identifier (**M14.4.75**)<br>• Participating Aggregation Identifier (**M14.4.64**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is always performed in conjunction with **F14.5.22 Aggregation – Remove Aggregation***<br>• *Before a user can move an aggregation the user must have the authority to perform this function on its new parent, as well as the authority to remove the aggregation from its previous parent or root*<br>• *To move an aggregation so that it becomes a root aggregation, the user must have the authority to perform this function for the record service as a whole* |

| | |
|---|---|
| | • *This function only applies to adding child aggregations by moving them from elsewhere, it does not apply to adding child aggregations by creating them in the aggregation (see **F14.5.5 Aggregation - Create**)* |

### F14.5.2 Aggregation - Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | 746b7ffc-d9a4-43d9-9dfa-01f6d1e8f671 |
| **Title** | **Aggregation – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.3 Aggregation - Add Record

| | |
|---|---|
| **System Identifier** | 0ef1d20b-a65f-4b0a-b2a0-e7b3a9a665f4 |
| **Title** | **Aggregation – Add Record** |
| **Description** | Add a record to the open aggregation by moving it from its previous aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |

| | |
|---|---|
| **Entity metadata** | *The following metadata element belonging to the aggregation will be modified:*<br>• Last Addition Timestamp (**M14.4.48**)<br>*The following metadata element belonging to the aggregation may also be modified (if it has not been set previously):*<br>• First Used Timestamp (**M14.4.32**)<br>*The following metadata elements belonging to the record will be modified:*<br>• Parent Aggregation Identifier (**M14.4.63**)<br>• Aggregated Timestamp (**M14.4.1**)<br>*The following metadata element belonging to the record may be modified depending on whether the authorised user chooses to retain or to replace the record's previous classification, under **R6.5.13**:*<br>• Class Identifier (**M14.4.4**) |
| ***From functional requirement(s)*** | **R6.5.13, R6.5.14** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Previous Parent Identifier (**M14.4.76**)<br>• Participating New Parent Identifier (**M14.4.75**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *This function is always performed in conjunction with **F14.5.22 Aggregation – Remove Record***<br>• *Before a user can move a record the user must have the authority to perform this function on its new parent aggregation, as well as the authority to remove the record from its previous parent aggregation*<br>• *This function only applies to adding records to a new parent aggregation by moving them from a previous parent aggregation, it does not apply to adding records to the aggregation by creating them in the aggregation (see **F14.5.121 Record - Create**)* |

### F14.5.4 Aggregation - Close

| | |
|---|---|
| **System Identifier** | 09fb9edc-d179-49dc-b069-a435f162e6fd |
| **Title** | **Aggregation – Close** |
| **Description** | Close the active aggregation |

| | |
|---|---|
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | • Closed Timestamp (**M14.4.5**) |
| ***From functional requirement(s)*** | **R6.5.6** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *Closing an aggregation may result in its automatic destruction if all of its contents have previously been destroyed (see **R8.4.21** and **F14.5.9 Aggregation – Destroy**)* |

### F14.5.5 Aggregation - Create

| | |
|---|---|
| **System Identifier** | 6054ae16-2036-424e-9bb7-aedb6e8229cc |
| **Title** | **Aggregation – Create** |
| **Description** | Create an aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Class Identifier (**M14.4.4**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• Closed Timestamp (**M14.4.5**)<br>• Max Levels Of Aggregation (**M14.4.52**)<br>• Parent Aggregation Identifier (**M14.4.63**)<br>• Aggregated Timestamp (**M14.4.1**)<br>• *Contextual metadata elements*<br>*If the aggregation is created in a parent aggregation then the following metadata element belonging to the parent aggregation will be modified:*<br>• Last Addition Timestamp (**M14.4.48**)<br>*The following metadata element belonging to the parent aggregation may also be modified (if it has not been set previously):* |

| | |
|---|---|
| | • First Used Timestamp (**M14.4.32**)<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R2.4.25, R6.5.1, R6.5.2, R7.5.18** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Participating New Parent Identifier (**M14.4.75**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *The aggregation may be created closed with a Closed Timestamp*<br>• *Root aggregations (with no parent) must have a Class Identifier*<br>• *Only root aggregations can have Max Levels Of Aggregation*<br>• *Only child aggregations have a Parent Aggregation Identifier and an Aggregated Timestamp; the event generated will also include a Participating Parent Aggregation Identifier and appear in the event histories of both the aggregation being created and its parent aggregation*<br>• *The aggregation may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event* |

### F14.5.6 Aggregation - Delete

| | |
|---|---|
| **System Identifier** | ae8dd3fe-3e02-4aa5-b9d0-840e0ea5b68b |
| **Title** | **Aggregation – Delete** |
| **Description** | Delete the unused aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | *The unused aggregation is deleted along with its metadata and event history* |

| | |
|---|---|
| ***From functional requirement(s)*** | **R6.5.7** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | *No event is generated* |

### F14.5.7 Aggregation – Delete Residual Event

| | |
|---|---|
| **System Identifier** | bff0f6be-8b87-454e-b4a9-a8ca584e574b |
| **Title** | **Aggregation – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.8 Aggregation – Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | 8e6f41c0-fe66-4147-865b-7c3fd0d3b3aa |
| **Title** | **Aggregation – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the residual aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R7.5.7** |

| | |
|---|---|
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R7.5.7**)* |

### F14.5.9 Aggregation – Destroy

| | |
|---|---|
| **System Identifier** | 60aa99a4-cf98-4e03-b64d-1fd137e90296 |
| **Title** | **Aggregation – Destroy** |
| **Description** | Destroy the aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R6.5.6**, **R8.4.22** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS when the aggregation is closed, under **R6.5.6**, and all of its children have previously been destroyed, under **R8.4.22***<br>• *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the aggregation on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.10 Aggregation – Exported

| | |
|---|---|
| **System Identifier** | 3f124e3f-64e8-4627-ac32-d1aa7b95ffa9 |
| **Title** | **Aggregation – Exported** |
| **Description** | The aggregation has been exported in full or as a placeholder |

| | |
|---|---|
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.11 Aggregation - Inherit Default Class

| | |
|---|---|
| **System Identifier** | d54f1e11-36c9-451e-abff-9cfa6b7b28e7 |
| **Title** | **Aggregation – Inherit Default Class** |
| **Description** | Inherit the default classification of the parent aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata** | • Class Identifier (**M14.4.4**) |
| ***From functional requirement(s)*** | **R6.5.4, R6.5.14** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |

| | |
|---|---|
| ***Usage notes*** | • *This function may only be performed for child aggregations where the default class has been overridden (see **F14.5.20 Aggregation – Override Class**)*<br>• *Performing this function removes the Class Identifier from the child aggregation ensuring it inherits its parent's classification* |

### F14.5.12 Aggregation - Inspect

| | |
|---|---|
| **System Identifier** | f607266f-e7fd-4bba-bbd3-462772c1d653 |
| **Title** | **Aggregation – Inspect** |
| **Description** | Browse to the aggregation, or discover it by searching, and inspect its metadata |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R6.5.9**, **R6.5.17**, **R9.4.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the aggregation and not when it is identified while browsing or included in search results* |

### F14.5.13 Aggregation - Inspect ACL

| | |
|---|---|
| **System Identifier** | c1de7c62-b4be-4622-99f5-f78ee88f41da |
| **Title** | **Aggregation – Inspect ACL** |
| **Description** | Inspect the access control list of the aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | *No metadata elements are modified* |

| | |
|---|---|
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**) |

### F14.5.14 Aggregation - Inspect Event

| | |
|---|---|
| **System Identifier** | db9c7774-0799-45e0-999e-0018cbd71e97 |
| **Title** | **Aggregation – Inspect Event** |
| **Description** | Browse the event history of the aggregation and inspect its events |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Participating Event Identifier (M14.4.71) |

### F14.5.15 Aggregation - Modify ACL

| | |
|---|---|
| **System Identifier** | 2b8f6950-8dce-41ba-aec3-a3b8181a4739 |
| **Title** | **Aggregation – Modify ACL** |
| **Description** | Modify the access control list for the aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |

| | |
|---|---|
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the aggregation is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.16 Aggregation – Modify Max Levels Of Aggregation

| | |
|---|---|
| **System Identifier** | 148427cb-6e55-498d-8352-c13729ad09c5 |
| **Title** | **Aggregation – Modify Max Levels Of Aggregation** |
| **Description** | Modify the maximum number of levels of aggregation that are allowed under a root aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata** | • Max Levels Of Aggregation (**M14.4.52**) |
| ***From functional requirement(s)*** | **R6.5.5** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *Only root aggregations may have Max Levels Of Aggregation* |

### F14.5.17 Aggregation - Modify Metadata

| | |
|---|---|
| **System Identifier** | d4104970-759a-45fc-aa83-43ca632a2307 |
| **Title** | **Aggregation – Modify Metadata** |
| **Description** | Modify the metadata of the active aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• *Contextual metadata elements* |
| ***From functional requirement(s)*** | **R6.5.3** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the aggregation*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.18 Aggregation - Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | 3ed4bcd1-ae2f-4c34-a0e1-5ddde22d1453 |
| **Title** | **Aggregation – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.19 Aggregation - Open

| | |
|---|---|
| **System Identifier** | 7c533508-1967-401c-9aa4-a6ad85fb63d5 |
| **Title** | **Aggregation – Open** |
| **Description** | Open the active aggregation that has previously been closed |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata modified** | • Closed Timestamp (**M14.4.5**) |
| ***From functional requirement(s)*** | **R6.5.6** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *Opening an aggregation removes the Closed Timestamp* |

### F14.5.20 Aggregation - Override Class

| | |
|---|---|
| **System Identifier** | a938c62a-6dff-4f7f-9490-9780cfc74f8b |
| **Title** | **Aggregation – Override Class** |
| **Description** | Override the previous classification of the aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata** | • Class Identifier (**M14.4.4**) |
| ***From functional requirement(s)*** | **R5.4.8**, **R6.5.4**, **R6.5.14** |

| | |
|---|---|
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Aggregation Identifier (**M14.4.64**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *Performing this function adds a Class Identifier directly to the aggregation or replaces it if the aggregation already has a Class Identifier* |

### F14.5.21 Aggregation - Remove Aggregation

| | |
|---|---|
| **System Identifier** | 41388b8a-b380-4997-8925-9f678c1655fc |
| **Title** | **Aggregation – Remove Aggregation** |
| **Description** | Remove a child aggregation from the aggregation by moving it to the root or another parent |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata** | *See related function **F14.5.1 Aggregation – Add Aggregation*** |
| ***From functional requirement(s)*** | **R6.5.8** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | • *Before a user can move a child aggregation out of the aggregation, the user must have the authority to perform this function, as well as the authority to add an aggregation to the new parent or root*<br>• *To remove a root aggregation and give it a parent, the user must have the authority to perform this function for the record service as a whole*<br>• *This function is always performed in conjunction with **F14.5.1 Aggregation – Add Aggregation**, which describes the metadata that is modified and the event that is generated*<br>• *This function does not separately modify metadata or generate an event* |

### F14.5.22 Aggregation - Remove Record

| | |
|---|---|
| **System Identifier** | d7d6dc0f-3d13-4f98-8fd5-b2ad9163d2cc |
| **Title** | **Aggregation – Remove Record** |

| | |
|---|---|
| **Description** | Remove a record from the aggregation by moving it to a different parent aggregation |
| **Entity Type** | Aggregation (**E14.2.1**) |
| **Entity metadata** | *See related function **F14.5.3 Aggregation – Add Record*** |
| ***From functional requirement(s)*** | **R6.5.8** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | • *Before a user can move a record out of the aggregation, the user must have the authority to perform this function, as well as the authority to add a record to the new parent aggregation*<br>• *This function is always performed in conjunction with **F14.5.3 Aggregation – Add Record**, which describes the metadata that is modified and the event that is generated*<br>• *This function does not separately modify metadata or generate an event* |

### F14.5.23 Class - Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | 5e4be488-5911-4fee-bb5b-bbc1e76bf8d4 |
| **Title** | **Class – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |

| | |
|---|---|
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.24 Class - Create

| | |
|---|---|
| **System Identifier** | c4285bfb-b62b-403c-9078-49522d881a85 |
| **Title** | **Class – Create** |
| **Description** | Create a class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• Default Disposal Schedule Identifier (**M14.4.11**)<br>• *Contextual metadata elements*<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R2.4.25, R5.4.2, R7.5.18** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *The class entity may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event*<br>• *Where the class's inherited access controls are modified on creation* |

| | |
|---|---|
| | *then separate **F14.5.33 Class – Modify ACL** events must be generated for each change made to the access control list* |

### F14.5.25 Class - Delete

| | |
|---|---|
| **System Identifier** | f6134d1a-649d-4b0a-9da4-ce9e4b713d6f |
| **Title** | **Class – Delete** |
| **Description** | Delete the unused class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | *The unused class is deleted along with its metadata and event history* |
| ***From functional requirement(s)*** | **R5.4.5** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | *No event is generated* |

### F14.5.26 Class - Delete Residual Event

| | |
|---|---|
| **System Identifier** | e8adbb3a-6e5d-49d2-989f-dabbcc384133 |
| **Title** | **Class – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.27 Class - Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | 145d6256-0974-46b1-b698-bc0d0ecadb57 |
| **Title** | **Class – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R7.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R7.5.7**)* |

### F14.5.28 Class - Destroy

| | |
|---|---|
| **System Identifier** | fed36daf-26ce-4d44-8452-b0cc3607ab75 |
| **Title** | **Class – Destroy** |
| **Description** | Destroy the active class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R5.4.6** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Event Comment (**M14.4.25**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the class on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.29 Class - Exported

| | |
|---|---|
| **System Identifier** | 476a72fd-0d53-470c-b962-afe9b1780759 |
| **Title** | **Class – Exported** |
| **Description** | The class has been exported in full or as a placeholder |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.30 Class - Inspect

| | |
|---|---|
| **System Identifier** | 1ce37d0a-be50-410b-9518-a4919921255f |

| | |
|---|---|
| **Title** | **Class – Inspect** |
| **Description** | Browse to the class, or discover it by searching, and inspect its metadata |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R5.4.7**, **R6.5.9**, **R6.5.17**, **R9.4.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the class and not when it is identified while browsing or included in search results* |

### F14.5.31 Class - Inspect ACL

| | |
|---|---|
| **System Identifier** | 1107cc8a-798c-4d6f-8d95-55c9f87b43bc |
| **Title** | **Class – Inspect ACL** |
| **Description** | Inspect the access control list of the class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**) |

### F14.5.32 Class - Inspect Event

| | |
|---|---|
| **System Identifier** | 8753d035-a402-4346-be10-5010e1f278b3 |
| **Title** | **Class – Inspect Event** |
| **Description** | Browse the event history of the class and inspect its events |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.33 Class - Modify ACL

| | |
|---|---|
| **System Identifier** | caced2b7-3496-4f0c-b679-6a8abc692c4a |
| **Title** | **Class – Modify ACL** |
| **Description** | Modify the access control list for the class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |

| | |
|---|---|
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the class is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.34 Class - Modify Default Disposal Schedule

| | |
|---|---|
| **System Identifier** | 7308ee79-510a-4738-bf79-07fd0e85f4af |
| **Title** | **Class – Modify Default Disposal Schedule** |
| **Description** | Modify the Default Disposal Schedule Identifier of the active class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | • Default Disposal Schedule Identifier (**M14.4.11**) |
| ***From functional requirement(s)*** | **R5.4.4**, **R8.4.13** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |

### F14.5.35 Class - Modify Metadata

| | |
|---|---|
| **System Identifier** | aa06e05a-2d32-4a08-9902-bf1628ce506e |
| **Title** | **Class – Modify Metadata** |
| **Description** | Modify the metadata of the active class |
| **Entity Type** | Class (**E14.2.2**) |

| | |
|---|---|
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• *Contextual metadata elements* |
| ***From functional requirement(s)*** | **R5.4.3** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the class*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.36 Class – Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | db27260a-e681-42b9-ad2f-64c5929b4f33 |
| **Title** | **Class – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active class |
| **Entity Type** | Class (**E14.2.2**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.37 Component - Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | 4c36756b-c66d-4469-9379-c3d979a777dc |
| **Title** | **Component – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the component |
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (M14.4.66)<br>• Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *This function may be performed by any user authorised to perform **F15.5.115 Record – Add Contextual Metadata***<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.38 Component - Create

| | |
|---|---|
| **System Identifier** | 0d6d5d46-4bab-4e92-8311-7d93ea476fd1 |
| **Title** | **Component – Create** |
| **Description** | Create the component of a record |
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Record Identifier (**M14.4.86**) |

| | |
|---|---|
| | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Automatic Deletion Flag (**M14.4.3**)<br>• *Contextual metadata elements*<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R2.4.25**, **R6.5.19**, **R6.5.21**, **R7.5.18** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *The component is created simultaneously with its record (see **F14.5.121 Record – Create**)*<br>• *The component may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event* |

### F14.5.39 Component - Delete Residual Event

| | |
|---|---|
| **System Identifier** | a0efddae-9c93-4bd4-bca1-11e4e12086d4 |
| **Title** | **Component – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual component |
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21**, **R6.5.21** |
| ***Purpose*** | Event generation only |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function may be performed by any user authorised to perform **F15.5.122 Record – Delete Residual Event***<br>• *This function always generates an event (see **R2.4.14**)* |

### F14.5.40 Component - Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | 18378b4a-db17-4309-9c04-a88e15888e1e |
| **Title** | **Component – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the residual component |
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R6.5.21, R7.5.7** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function may be performed by any user authorised to perform **F15.5.123 Record – Delete Residual Metadata***<br>• *This function always generates an event (see **R7.5.7**)* |

### F14.5.41 Component - Destroy

| | |
|---|---|
| **System Identifier** | 4bc532be-b33b-407b-9c59-28bb6c65e1ff |
| **Title** | **Component – Destroy** |
| **Description** | Destroy the component as part of the destruction of a record |
| **Entity Type** | Component (**E14.2.3**) |

| | |
|---|---|
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R8.4.20** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as part of destroying a record (see **F14.5.124 Record – Destroy**)*<br>• *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the component on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.42 Component - Duplicate

| | |
|---|---|
| **System Identifier** | 9cd765ac-a511-4da1-9bee-90b40f769d60 |
| **Title** | **Component – Duplicate** |
| **Description** | Duplicate a component |
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata modified** | • Duplicate Identifier (**M14.4.23**) |
| ***From functional requirement(s)*** | **R6.5.16** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Participating Duplicate Identifier (**M14.4.69**)<br>• Duplicate Identifier (**M14.4.23**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as part of duplicating a record (see **F14.5.126 Record – Duplicate**)*<br>• *Two duplicate events will be generated, one for the first component that identifies the second component as the duplicate using the Participating Duplicate Identifier, and one for the second component* |

| | |
|---|---|
| | *that identifies the first component as the duplicate using the Participating Duplicate Identifier*<br>• *The duplicate events will be linked by the Duplicate Identifier in the event entity* |

### F14.5.43 Component - Exported

| | |
|---|---|
| **System Identifier** | cad7c66b-50ef-479c-9403-f61af0cffea4 |
| **Title** | **Component – Exported** |
| **Description** | The component has been exported in full or as a placeholder |
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.44 Component - Inspect

| | |
|---|---|
| **System Identifier** | 31b1b287-7de3-4f67-a637-c0f7063d19ac |
| **Title** | **Component – Inspect** |
| **Description** | Browse to the component, or discover it by searching, and inspect its metadata |

| | |
|---|---|
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R6.5.17, R6.5.21** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**) |
| ***Usage notes*** | • *This function may be performed by any user authorised to perform **F15.5.131 Record – Inspect***<br>• *An event should only be generated for this function when the user examines the metadata of the component and not when it is identified while browsing or included in search results* |

### F14.5.45 Component - Inspect Event

| | |
|---|---|
| **System Identifier** | 351ea1cf-87e0-4499-a397-f608ad3033c3 |
| **Title** | **Component – Inspect Event** |
| **Description** | Browse the event history of the component and inspect its events |
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19, R6.5.21** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Participating Event Identifier (**M14.4.71**) |
| ***Usage notes*** | *This function may be performed by any user authorised to perform **F15.5.133 Record – Inspect Event*** |

### F14.5.46 Component - Modify Metadata

| | |
|---|---|
| **System Identifier** | fbc13347-7576-4aaa-9109-d5ff69a43021 |
| **Title** | **Component – Modify Metadata** |
| **Description** | Modify the metadata of the active component |
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• *Contextual metadata elements* |
| ***From functional requirement(s)*** | **R6.5.20, R6.5.21** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *This function may be performed by any user authorised to perform **F15.5.135 Record – Modify Metadata***<br>• *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the component*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.47 Component - Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | 94480ab5-c230-4c54-b0a4-4980441bde4c |
| **Title** | **Component – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active component |
| **Entity Type** | Component (**E14.2.3**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26, R6.5.21** |
| ***Purpose*** | Event generation only |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *This function may be performed by any user authorised to perform **F15.5.136 Record – Modify Originated Date/Time***<br>• *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.48 Contextual Metadata Element Definition - Create

| | |
|---|---|
| **System Identifier** | e12e910b-8a2e-4939-a34b-1f0eb06a9697 |
| **Title** | **Contextual Metadata Element Definition – Create** |
| **Description** | Create a contextual metadata element definition |
| **Entity Type** | Contextual Metadata Element Definition (**E14.2.4**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• Presentation Order (**M14.4.84**)<br>• Min Occurs (**M14.4.56**)<br>• Max Occurs (**M14.4.53**)<br>• Is Modifiable Flag (**M14.4.46**)<br>• Is Entity Reference Flag (**M14.4.45**)<br>• Entity Reference Type Identifier (**M14.4.24**)<br>• Datatype (**M14.4.10**)<br>• Is Textual Flag (**M14.4.47**)<br>• Default Value (**M14.4.13**)<br>• Default Language Identifier (**M14.4.12**)<br>• Retain On Destruction Flag (**M14.4.88**) |
| ***From functional requirement(s)*** | **R2.4.25, R7.5.2, R7.5.3, R7.5.4** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event*<br>• *Where the contextual metadata element definition's inherited access controls are modified on creation then separate **F14.5.112 Metadata Element Definition – Modify ACL** events must be generated for each change made to the access control list* |

### F14.5.49 Contextual Metadata Element Definition - Delete

| | |
|---|---|
| **System Identifier** | be1eadc6-d5a2-4d39-9a79-43c501b70cdb |
| **Title** | **Contextual Metadata Element Definition – Delete** |
| **Description** | Delete the unused contextual metadata element definition |
| **Entity Type** | Contextual Metadata Element Definition (**E14.2.4**) |
| **Entity metadata modified** | *The unused contextual metadata element definition is deleted along with its metadata and event history* |
| ***From functional requirement(s)*** | **R7.5.10** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | *No event is generated* |

### F14.5.50 Contextual Metadata Element Definition - Delete Residual Event

| | |
|---|---|
| **System Identifier** | f791a9a7-2470-491c-b5e5-31cc04a90788 |
| **Title** | **Contextual Metadata Element Definition – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual contextual metadata element definition |
| **Entity Type** | Contextual Metadata Element Definition (**E14.2.4**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |

| | |
|---|---|
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.51 Contextual Metadata Element Definition - Destroy

| | |
|---|---|
| **System Identifier** | 59e9e6a6-b87a-4ca2-a615-36d04518e064 |
| **Title** | **Contextual Metadata Element Definition – Destroy** |
| **Description** | Destroy the active contextual metadata element definition |
| **Entity Type** | Contextual Metadata Element Definition (**E14.2.4**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R7.5.11** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Event Comment (**M14.4.25**) |

### F14.5.52 Contextual Metadata Element Definition - Exported

| | |
|---|---|
| **System Identifier** | c1a541d1-d68a-4dc4-93d0-ae5596263d73 |
| **Title** | **Contextual Metadata Element Definition – Exported** |
| **Description** | The contextual metadata element definition has been exported in full or as a placeholder |
| **Entity Type** | Contextual Metadata Element Definition (**E14.2.4**) |
| **Entity metadata modified** | *No metadata elements are modified* |

| | |
|---|---|
| ***From functional requirement(s)*** | **R11.4.10** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.53 Contextual Metadata Element Definition – Modify Before Use

| | |
|---|---|
| **System Identifier** | 31e98553-840c-48ab-8514-df2a77e9ae87 |
| **Title** | **Contextual Metadata Element Definition – Modify Before Use** |
| **Description** | Modify the metadata of the contextual metadata element definition that has never been used previously |
| **Entity Type** | Contextual Metadata Element Definition (**E14.2.4**) |
| **Entity metadata** | • Min Occurs (**M14.4.56**)<br>• Max Occurs (**M14.4.53**)<br>• Is Modifiable Flag (**M14.4.46**)<br>• Is Entity Reference Flag (**M14.4.45**)<br>• Entity Reference Type Identifier (**M14.4.24**)<br>• Datatype (**M14.4.10**)<br>• Is Textual Flag (**M14.4.47**) |
| ***From functional requirement(s)*** | **R7.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *This function may only be performed for contextual metadata element definitions that have never been applied to an entity*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.54 Contextual Metadata Element Definition – Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | 23ab0db8-2a0a-495b-b7a2-211d577b2e00 |
| **Title** | **Contextual Metadata Element Definition – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active contextual metadata element definition |
| **Entity Type** | Contextual Metadata Element Definition (**E14.2.4**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.55 Disposal Hold – Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | d3110710-a391-4b15-b6c2-8e3f904549e2 |
| **Title** | **Disposal Hold – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |

| | |
|---|---|
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.56 Disposal Hold – Add Entity

| | |
|---|---|
| **System Identifier** | 3fa1c42b-1d1a-4888-8e26-f57a8d76df27 |
| **Title** | **Disposal Hold – Add Entity** |
| **Description** | Add an active class, aggregation or record to the active disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata** | • Held Record Identifier (**M14.4.39**)<br>• Held Aggregation Identifier (**M14.4.37**)<br>• Held Class Identifier (**M14.4.38**)<br>*Performing this function may also modify the following metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R9.4.3** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Event Comment (**M14.4.25**) |

### F14.5.57 Disposal Hold - Create

| | |
|---|---|
| **System Identifier** | 45e638b2-3eda-4a2d-b320-3b156ed82897 |
| **Title** | **Disposal Hold – Create** |
| **Description** | Create a disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Mandate (**M14.4.51**)<br>• Scope Notes (**M14.4.97**)<br>• *Contextual metadata elements*<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R2.4.25, R7.5.18, R9.4.1** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *The disposal hold may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event*<br>• *Where the disposal hold's inherited access controls are modified on* |

| | |
|---|---|
| | *creation then separate **F14.5.66 Disposal Hold – Modify ACL** events must be generated for each change made to the access control list* |

### F14.5.58 Disposal Hold - Delete

| | |
|---|---|
| **System Identifier** | 52e2be2e-3aa6-4854-8b7d-d58141cec8a5 |
| **Title** | **Disposal Hold – Delete** |
| **Description** | Delete the unused disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | *The unused disposal hold is deleted along with its metadata and event history* |
| ***From functional requirement(s)*** | **R9.4.6** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | *No event is generated* |

### F14.5.59 Disposal Hold - Delete Residual Event

| | |
|---|---|
| **System Identifier** | e6b9b61f-7b7e-4a30-84bc-0908b44d21af |
| **Title** | **Disposal Hold – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.60 Disposal Hold - Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | 462a20ad-18d6-4875-b47e-b2ec992c612a |
| **Title** | **Disposal Hold – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the residual disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R7.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R7.5.7**)* |

### F14.5.61 Disposal Hold - Destroy

| | |
|---|---|
| **System Identifier** | 4b02e580-7fdb-4780-85ce-fdaa88fff88d |
| **Title** | **Disposal Hold – Destroy** |
| **Description** | Destroy the active disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R8.4.11** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Event Comment (**M14.4.25**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the disposal hold on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.62 Disposal Hold - Exported

| | |
|---|---|
| **System Identifier** | 1b60af30-8fb7-44af-8441-425004fc2780 |
| **Title** | **Disposal Hold – Exported** |
| **Description** | The disposal hold has been exported in full or as a placeholder |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.63 Disposal Hold - Inspect

| | |
|---|---|
| **System Identifier** | cc102e9a-953f-410f-9cf2-e527ba70e49b |

| | |
|---|---|
| **Title** | **Disposal Hold – Inspect** |
| **Description** | Browse to the disposal hold, or discover it by searching, and inspect its metadata |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R5.4.7**, **R6.5.9**, **R6.5.17**, **R9.4.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the disposal hold and not when it is identified while browsing or included in search results* |

### F14.5.64 Disposal Hold - Inspect ACL

| | |
|---|---|
| **System Identifier** | 971b7480-cb7e-43de-9a37-41c65cd65c7a |
| **Title** | **Disposal Hold – Inspect ACL** |
| **Description** | Inspect the access control list of the disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**) |

### F14.5.65 Disposal Hold - Inspect Event

| | |
|---|---|
| **System Identifier** | a729cbb2-ae32-46ca-a31b-de9a1b010100 |
| **Title** | **Disposal Hold – Inspect Event** |
| **Description** | Browse the event history of the disposal hold and inspect its events |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.66 Disposal Hold - Modify ACL

| | |
|---|---|
| **System Identifier** | f308349e-fd2a-46ed-a3bc-abfd458f30bd |
| **Title** | **Disposal Hold – Modify ACL** |
| **Description** | Modify the access control list for the disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |

| | |
|---|---|
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the disposal hold is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.67 Disposal Hold – Modify Metadata

| | |
|---|---|
| **System Identifier** | 1ff0d40d-2a88-4b31-88f9-c1efc135c618 |
| **Title** | **Disposal Hold – Modify Metadata** |
| **Description** | Modify the metadata of the active disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Mandate (**M14.4.51**)<br>• Scope Notes (**M14.4.97**)<br>• *Contextual metadata elements* |
| ***From functional requirement(s)*** | **R9.4.2** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the record*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.68 Disposal Hold – Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | 812a29d1-a7ad-44d8-a7aa-4bab741e1e5b |

| | |
|---|---|
| **Title** | **Disposal Hold – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.69 Disposal Hold - Remove Entity

| | |
|---|---|
| **System Identifier** | dcde1a11-f6e8-44f6-b48b-d3e61e53b9e2 |
| **Title** | **Disposal Hold – Remove Entity** |
| **Description** | Remove a class, aggregation or record from the active disposal hold |
| **Entity Type** | Disposal Hold (**E14.2.5**) |
| **Entity metadata** | • Held Record Identifier (**M14.4.39**)<br>• Held Aggregation Identifier (**M14.4.37**)<br>• Held Class Identifier (**M14.4.38**) |
| ***From functional requirement(s)*** | **R9.4.3** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Hold Identifier (**M14.4.67**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Event Comment (**M14.4.25**) |

### F14.5.70 Disposal Schedule - Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | 88fed471-3f1a-4171-8099-b35304d9aee5 |
| **Title** | **Disposal Schedule – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the disposal schedule |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.71 Disposal Schedule - Create

| | |
|---|---|
| **System Identifier** | 25556d43-6aa9-41e5-b146-e98473e14024 |
| **Title** | **Disposal Schedule – Create** |
| **Description** | Create a disposal schedule |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Mandate (**M14.4.51**) |

| | |
|---|---|
| | • Scope Notes (**M14.4.97**)<br>• Disposal Action Code (**M14.4.18**)<br>• Retention Trigger Code (**M14.4.94**)<br>• Retention Trigger Element Identifier (**M14.4.95**)<br>• Retention Period Interval Code (**M14.4.90**)<br>• Retention Period Duration Number (**M14.4.89**)<br>• Retention Period Offset Code (**M14.4.91**)<br>• Retention Period Offset Month Code (**M14.4.92**)<br>• Confirmation Period Interval Code (**M14.4.7**)<br>• Confirmation Period Duration Number (**M14.4.6**)<br>• *Contextual metadata elements*<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R2.4.25, R7.5.18, R8.4.1** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *The disposal schedule may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event*<br>• *Where the disposal schedule's inherited access controls are modified on creation then separate **F14.5.80 Disposal Schedule – Modify ACL** events must be generated for each change made to the access control list* |

### F14.5.72 Disposal Schedule - Delete

| | |
|---|---|
| **System Identifier** | bac5ebc1-c4c3-4ec0-ba97-7421d17ca968 |
| **Title** | **Disposal Schedule – Delete** |
| **Description** | Delete the unused disposal schedule |

| | |
|---|---|
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | *The unused disposal schedule is deleted along with its metadata and event history* |
| ***From functional requirement(s)*** | **R8.4.10** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | *No event is generated* |

### F14.5.73 Disposal Schedule - Delete Residual Event

| | |
|---|---|
| **System Identifier** | 8b26a551-b2c9-4940-ade6-ddf9ce771e62 |
| **Title** | **Disposal Schedule – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual disposal schedule |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.74 Disposal Schedule - Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | 2ef9758c-9143-4859-bf7f-f7e3384c8750 |
| **Title** | **Disposal Schedule – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the residual disposal schedule |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |

| | |
|---|---|
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R7.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R7.5.7**)* |

### F14.5.75 Disposal Schedule – Destroy

| | |
|---|---|
| **System Identifier** | 20f4d0e0-98c7-45b5-81d2-a14e67325f8e |
| **Title** | **Disposal Schedule – Destroy** |
| **Description** | Destroy the active disposal schedule |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R8.4.11** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Event Comment (**M14.4.25**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the entity on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.76 Disposal Schedule - Exported

| | |
|---|---|
| **System Identifier** | 48f27df0-580a-4804-ab94-425dda402347 |
| **Title** | **Disposal Schedule – Exported** |
| **Description** | The disposal schedule has been exported in full or as a placeholder |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.77 Disposal Schedule - Inspect

| | |
|---|---|
| **System Identifier** | a87eaf88-b537-4061-b57a-2d50e6fb5f9d |
| **Title** | **Disposal Schedule – Inspect** |
| **Description** | Browse to the disposal schedule, or discover it by searching, and inspect its metadata |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R5.4.7**, **R6.5.17**, **R8.4.12** |

| | |
|---|---|
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the disposal schedule and not when it is identified while browsing or included in search results* |

### F14.5.78 Disposal Schedule – Inspect ACL

| | |
|---|---|
| **System Identifier** | 624f62eb-7e01-4e98-b886-291a1166698d |
| **Title** | **Disposal Schedule – Inspect ACL** |
| **Description** | Inspect the access control list of the disposal schedule |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**) |

### F14.5.79 Disposal Schedule – Inspect Event

| | |
|---|---|
| **System Identifier** | 028167e8-309a-458f-8ae3-5dfbab73451d |
| **Title** | **Disposal Schedule – Inspect Event** |
| **Description** | Browse the event history of the disposal schedule and inspect its events |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | *No metadata elements are modified* |

| | |
|---|---|
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.80 Disposal Schedule – Modify ACL

| | |
|---|---|
| **System Identifier** | 31b963f1-392c-4bd6-a696-934a53142632 |
| **Title** | **Disposal Schedule – Modify ACL** |
| **Description** | Modify the access control list for the disposal schedule |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the disposal schedule is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.81 Disposal Schedule – Modify Metadata

| | |
|---|---|
| **System Identifier** | 8ec42472-e351-4c7e-8c02-9da97677d9ac |
| **Title** | **Disposal Schedule – Modify Metadata** |
| **Description** | Modify the metadata of the active disposal schedule |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata** | • Title (**M14.4.104)**<br>• Description (**M14.4.16**)<br>• Mandate (**M14.4.51**)<br>• Scope Notes (**M14.4.97**)<br>• *Contextual metadata elements*<br>*In accordance with **R8.4.9**, the following metadata may only be modified prior to the disposal schedule being applied to a record:*<br>• Disposal Action Code (**M14.4.18**)<br>• Retention Trigger Code (**M14.4.94**)<br>• Retention Trigger Element Identifier (**M14.4.95**)<br>• Retention Period Interval Code (**M14.4.90**)<br>• Retention Period Duration Number (**M14.4.89**)<br>• Retention Period Offset Code (**M14.4.91**)<br>• Retention Period Offset Month Code (**M14.4.92**)<br>• Confirmation Period Interval Code (**M14.4.7**)<br>• Confirmation Period Duration Number (**M14.4.6**) |
| ***From functional requirement(s)*** | **R8.4.8, R8.4.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the record*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.82 Disposal Schedule – Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | d98a38e4-8a18-4141-a245-58043ada13c1 |

| | |
|---|---|
| **Title** | **Disposal Schedule – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active disposal schedule |
| **Entity Type** | Disposal Schedule (**E14.2.6**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.83 Entity Type - Inspect

| | |
|---|---|
| **System Identifier** | e9975a09-acdf-4ef7-8ae8-eae72c66cb03 |
| **Title** | **Entity Type – Inspect** |
| **Description** | Browse to the entity type, or discover it by searching, and inspect its metadata |
| **Entity Type** | Entity Type (**E14.2.7**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.9**, **R7.5.12** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Entity Type Identifier (**M14.4.70**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the entity type and not when it is identified while browsing or included in search results* |

### F14.5.84 Entity Type - Inspect ACL

| | |
|---|---|
| **System Identifier** | a500ff2b-0518-459e-94d6-1f2f281b30d8 |
| **Title** | **Entity Type – Inspect ACL** |
| **Description** | Inspect the access control list of the entity type |
| **Entity Type** | Entity Type (**E14.2.7**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Entity Type Identifier (**M14.4.70**) |

### F14.5.85 Entity Type - Inspect Event

| | |
|---|---|
| **System Identifier** | e6d3f367-ec8f-4cac-b65b-7285dd4b3358 |
| **Title** | **Entity Type – Inspect Event** |
| **Description** | Browse the event history of the entity type and inspect its events |
| **Entity Type** | Entity Type (**E14.2.7**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Entity Type Identifier (**M14.4.70**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.86 Entity Type - Modify ACL

| | |
|---|---|
| **System Identifier** | 0704069a-0129-4978-9e69-401da1dbdc0a |

| | |
|---|---|
| **Title** | **Entity Type – Modify ACL** |
| **Description** | Modify the access control list for the entity type |
| **Entity Type** | Entity Type (**E14.2.7**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Entity Type Identifier (**M14.4.70**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the entity type is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.87 Function Definition - Inspect

| | |
|---|---|
| **System Identifier** | eedac611-18b3-46ab-9f78-ce5d105343f3 |
| **Title** | **Function Definition – Inspect** |
| **Description** | Browse to the function definition, or discover it by searching, and inspect its metadata |
| **Entity Type** | Function Definition (**E14.2.9**) |
| **Entity metadata modified** | *No metadata elements are modified* |

| | |
|---|---|
| ***From functional requirement(s)*** | **R2.4.11, R4.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Function Definition Identifier (**M14.4.72**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the function definition and not when it is identified while browsing or included in search results* |

### F14.5.88 Function Definition - Inspect ACL

| | |
|---|---|
| **System Identifier** | eceb7200-db40-40cd-8140-a8a2fb2adfea |
| **Title** | **Function Definition – Inspect ACL** |
| **Description** | Inspect the access control list of the function definition |
| **Entity Type** | Function Definition (**E14.2.9**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Function Definition Identifier (**M14.4.72**) |

### F14.5.89 Function Definition - Inspect Event

| | |
|---|---|
| **System Identifier** | 4ba33648-4e99-421a-acc3-92f1579839e4 |
| **Title** | **Function Definition – Inspect Event** |
| **Description** | Browse the event history of the function definition and inspect its events |
| **Entity Type** | Function Definition (**E14.2.9**) |

| | |
|---|---|
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Function Definition Identifier (**M14.4.72**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.90 Function Definition – Modify ACL

| | |
|---|---|
| **System Identifier** | c1c23be7-1bfe-49bb-a9e6-648a947926af |
| **Title** | **Function Definition – Modify ACL** |
| **Description** | Modify the access control list for the function definition |
| **Entity Type** | Function Definition (**E14.2.9**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Function Definition Identifier (**M14.4.72**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the function definition is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded* |

| | |
|---|---|
| | *through the adding, modifying and deleting of access control entries* |

### F14.5.91 Function Definition - Modify Event Generation

| | |
|---|---|
| **System Identifier** | e2cf111d-faa8-4dd4-ad4c-36e15ad603f7 |
| **Title** | **Function Definition – Modify Event Generation** |
| **Description** | Change whether an event is generated when the function is performed |
| **Entity Type** | Function Definition (**E14.2.9**) |
| **Entity metadata modified** | • Generate Event Flag (**M14.4.34**) |
| ***From functional requirement(s)*** | **R2.4.13** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Function Definition Identifier (**M14.4.72**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.92 Function Definition - Modify Retain Event On Destruction

| | |
|---|---|
| **System Identifier** | 712df19b-80e1-4dda-b3a4-1971b8ebcf67 |
| **Title** | **Function Definition – Modify Retain Event On Destruction** |
| **Description** | Change whether an event is retained or deleted when the participating entity is destroyed |
| **Entity Type** | Function Definition (**E14.2.9**) |
| **Entity metadata modified** | • Retain On Destruction Flag (**M14.4.88**) |
| ***From functional requirement(s)*** | **R2.4.20** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Function Definition Identifier (**M14.4.72**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |

### F14.5.93 Group – Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | 60d37b06-55e5-44d9-998b-8b866afbecb9 |
| **Title** | **Group – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.94 Group – Add User

| | |
|---|---|
| **System Identifier** | 5327775f-2899-4268-aa59-92f5f6ee2f4f |
| **Title** | **Group – Add User** |
| **Description** | Add the active user to the active group |
| **Entity Type** | Group (**E14.2.10**) |

| | |
|---|---|
| **Entity metadata** | *The following metadata element belonging to the participating group may be modified (if it has not been set previously):*<br>• First Used Timestamp (**M14.4.32**)<br>*Performing this group function adds the following metadata element to the participating user entity:*<br>• Group Identifier (**M14.4.36**) |
| ***From functional requirement(s)*** | **R3.4.4** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Participating User Identifier (**M14.4.81**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function may occur in another system external to the MCRS and be synchronised to it*<br>• *Although this function (and the corresponding authority to perform it) is associated with the group, performing it has the effect of modifying the Group Identifier for the user entity* |

### F14.5.95 Group - Create

| | |
|---|---|
| **System Identifier** | 97039415-a9c2-4cf7-999d-f4135304d97a |
| **Title** | **Group – Create** |
| **Description** | Create a group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• *Contextual metadata elements*<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |

| | |
|---|---|
| ***From functional requirement(s)*** | **R2.4.25, R3.4.9, R7.5.18** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *This function may occur in another system external to the MCRS and be synchronised to it*<br>• *The group entity may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event*<br>• *Where the group's inherited access controls are modified on creation then separate **F14.5.104 Group – Modify ACL** events must be generated for each change made to the access control list* |

### F14.5.96 Group - Delete

| | |
|---|---|
| **System Identifier** | 30e6e15b-f889-4ff7-8a66-a695498b5565 |
| **Title** | **Group – Delete** |
| **Description** | Delete the unused group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | *The unused group entity is deleted along with its metadata and event history* |
| ***From functional requirement(s)*** | **R3.4.11** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | *No event is generated* |

### F14.5.97 Group - Delete Residual Event

| | |
|---|---|
| **System Identifier** | ac82726d-804e-491d-a5ec-a27ffb986a91 |
| **Title** | **Group – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.98 Group - Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | 41d00f75-2403-4d16-b825-f85acf6ee746 |
| **Title** | **Group – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the residual group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R7.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |

| | |
|---|---|
| ***Usage notes*** | *This function always generates an event (see **R7.5.7**)* |

### F14.5.99 Group – Destroy

| | |
|---|---|
| **System Identifier** | 0201d5d3-e201-4c5a-90b1-1a98913ba09a |
| **Title** | **Group – Destroy** |
| **Description** | Destroy the active group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R3.4.12** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Event Comment (**M14.4.25**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the group on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.100 Group – Exported

| | |
|---|---|
| **System Identifier** | 6978b25b-0036-44f9-ae99-b80e10027fe6 |
| **Title** | **Group – Exported** |
| **Description** | The group has been exported in full or as a placeholder |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |

| | |
|---|---|
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.101 Group - Inspect

| | |
|---|---|
| **System Identifier** | 642ce23c-542f-40c1-912d-07897a1415de |
| **Title** | **Group – Inspect** |
| **Description** | Browse to the group, or discover it by searching, and inspect its metadata |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R3.4.14** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the group and not when it is identified while browsing or included in search results* |

### F14.5.102 Group - Inspect ACL

| | |
|---|---|
| **System Identifier** | d52f4721-d55b-4eee-aaab-f77d83e0dd55 |
| **Title** | **Group – Inspect ACL** |
| **Description** | Inspect the access control list of the group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**) |

### F14.5.103 Group - Inspect Event

| | |
|---|---|
| **System Identifier** | bcadb823-13a1-4ba5-ae36-62b106132c7a |
| **Title** | **Group – Inspect Event** |
| **Description** | Browse the event history of the group and inspect its events |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.104 Group - Modify ACL

| | |
|---|---|
| **System Identifier** | f48f825f-2afb-427e-bdaa-056bb025521c |

| | |
|---|---|
| **Title** | **Group – Modify ACL** |
| **Description** | Modify the access control list for the group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry and should not be confused with the Participating Group Identifier which indicates the group entity that has the access control list*<br>• *If more than one access control entry belonging to the group is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.105 Group - Modify Metadata

| | |
|---|---|
| **System Identifier** | b6d7687f-d94e-4281-8575-1302556ca6ba |
| **Title** | **Group – Modify Metadata** |
| **Description** | Modify the metadata of the active group |
| **Entity Type** | Group (**E14.2.10**) |

| | |
|---|---|
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• *Contextual metadata elements* |
| ***From functional requirement(s)*** | **R3.4.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *This function may occur in another system external to the MCRS and be synchronised to it*<br>• *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the group*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.106 Group – Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | b5ed9497-7ac2-4f95-8728-fd23d30d3b86 |
| **Title** | **Group – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.107 Group - Remove User

| | |
|---|---|
| **System Identifier** | c3713f12-feb6-459e-a21a-7e63aaeeea6c |
| **Title** | **Group – Remove User** |
| **Description** | Remove the active user from the active group |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata** | *Performing this group function removes the following metadata element belonging to the participating user entity:*<br>• Group Identifier (**M14.4.36**) |
| ***From functional requirement(s)*** | **R3.4.4** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Participating User Identifier (**M14.4.81**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function may occur in another system external to the MCRS and be synchronised to it*<br>• *This function (and the corresponding authority to perform it) is associated with the group, not the participating user entity* |

### F14.5.108 Group - Report User Membership

| | |
|---|---|
| **System Identifier** | 8b8c09d4-30aa-41c4-b0fc-1a0ecf4f9d88 |
| **Title** | **Group – Report User Membership** |
| **Description** | Report the active users that belonged to the group at a specified historical date and time |
| **Entity Type** | Group (**E14.2.10**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R3.4.13** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Group Identifier (**M14.4.73**)<br>• Historical Date/Time (**M14.4.40**) |

### F14.5.109 Metadata Element Definition – Inspect

| | |
|---|---|
| **System Identifier** | a5423f4d-0f15-4376-8d22-b10affe3ae3a |
| **Title** | **Metadata Element Definition – Inspect** |
| **Description** | Browse to the metadata element definition, or discover it by searching, and inspect its metadata |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R7.5.12** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the metadata element definition and not when it is identified while browsing or included in search results* |

### F14.5.110 Metadata Element Definition – Inspect ACL

| | |
|---|---|
| **System Identifier** | 35f8d31b-9db2-4d37-96a2-d0ad36533a92 |
| **Title** | **Metadata Element Definition – Inspect ACL** |
| **Description** | Inspect the access control list of the metadata element definition |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |

| | |
|---|---|
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**) |

### F14.5.111 Metadata Element Definition – Inspect Event

| | |
|---|---|
| **System Identifier** | 72459e48-ea29-4b05-a125-75a36a0ea74f |
| **Title** | **Metadata Element Definition – Inspect Event** |
| **Description** | Browse the event history of the metadata element definition and inspect its events |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.112 Metadata Element Definition – Modify ACL

| | |
|---|---|
| **System Identifier** | 2f04f823-c37a-4c68-b6fa-76be4a76cf7d |
| **Title** | **Metadata Element Definition – Modify ACL** |
| **Description** | Modify the access control list for the metadata element definition |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |

| | |
|---|---|
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the metadata element definition is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.113 Metadata Element Definition – Modify Metadata

| | |
|---|---|
| **System Identifier** | 2a70d3d6-4f38-4666-82ac-82d0ceb5e608 |
| **Title** | **Metadata Element Definition – Modify Metadata** |
| **Description** | Modify the metadata of the active metadata element definition |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• Presentation Order (**M14.4.84**)<br>• Default Value (**M14.4.13**)<br>• Default Language Identifier (**M14.4.12**) |
| ***From functional requirement(s)*** | **R7.5.8** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.114 Metadata Element Definition – Modify Retain On Destruction

| | |
|---|---|
| **System Identifier** | f8f0cbb3-8f9e-4dc9-807e-36e351b85960 |
| **Title** | **Metadata Element Definition – Modify Retain On Destruction** |
| **Description** | Change whether the value of a metadata element is retained or deleted when the entity it belongs to is destroyed |
| **Entity Type** | Metadata Element Definition (**E14.2.11**) |
| **Entity metadata modified** | • Retain On Destruction Flag (**M14.4.88**) |
| ***From functional requirement(s)*** | **R7.5.6** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |

### F14.5.115 Record – Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | 83fa3a37-8bee-494e-af7a-8c23e56bb106 |
| **Title** | **Record – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |

| | |
|---|---|
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.116 Record – Cancel Destruction

| | |
|---|---|
| **System Identifier** | 09cb99bb-bcd8-45cd-b0a3-436aab6a46b5 |
| **Title** | **Record – Cancel Destruction** |
| **Description** | Cancel the pending destruction of the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Disposal Schedule Identifier (**M14.4.22**)<br>• Retention Start Date (**M14.4.93**)<br>• Disposal Action Code (**M14.4.18**)<br>• Disposal Action Due Date (**M14.4.19**)<br>• Disposal Confirmation Due Date (**M14.4.20**)<br>• Disposal Overdue Alert Timestamp (**M14.4.21**) |
| ***From functional requirement(s)*** | **R8.4.18** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *A new disposal schedule must be applied to the record to replace the disposal schedule in force*<br>• *The cancellation comment is stored as the Event Comment and is required under **R8.4.18*** |

### F14.5.117 Record - Cancel Transfer

| | |
|---|---|
| **System Identifier** | 19e3483a-dded-4e5f-a812-15b5a06da197 |
| **Title** | **Record – Cancel Transfer** |
| **Description** | Cancel the pending transfer of the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Disposal Schedule Identifier (**M14.4.22**)<br>• Retention Start Date (**M14.4.93**)<br>• Disposal Action Code (**M14.4.18**)<br>• Disposal Action Due Date (**M14.4.19**)<br>• Disposal Confirmation Due Date (**M14.4.20**)<br>• Disposal Overdue Alert Timestamp (**M14.4.21**) |
| ***From functional requirement(s)*** | **R8.4.18** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *A new disposal schedule must be applied to the record to replace the disposal schedule in force*<br>• *The cancellation comment is stored as the Event Comment and is required under **R8.4.18*** |

### F14.5.118 Record - Complete Review

| | |
|---|---|
| **System Identifier** | 339c50b5-3e06-4507-80e5-946cfa9b3ba5 |
| **Title** | **Record – Complete Review** |
| **Description** | Complete a review of the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Disposal Schedule Identifier (**M14.4.22**)<br>• Last Review Comment (**M14.4.49**)<br>• Last Reviewed Timestamp (**M14.4.50**) |

| | |
|---|---|
| ***From functional requirement(s)*** | **R8.4.17** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *The review comment is stored as the Event Comment (see **R8.4.17**)* |

### F14.5.119 Record – Confirm Destruction

| | |
|---|---|
| **System Identifier** | a221b6c3-4b3e-4737-a4f6-def8bded9af2 |
| **Title** | **Record – Confirm Destruction** |
| **Description** | Confirm the destruction of the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R8.4.20, R8.4.23** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *Where the user closes aggregations as part of confirming the destruction of the record, under **R8.4.23**, separate **F14.5.4 Aggregation – Close** events should be generated*<br>• *Upon confirmation, the MCRS must immediately perform the following automatic functions **F14.5.124 Record – Destroy**, **F14.5.41 Component – Destroy**, and possibly **F14.5.9 Aggregation – Destroy** (under **R8.4.21**)* |

### F14.5.120 Record – Confirm Transfer

| | |
|---|---|
| **System Identifier** | 601253a3-723e-458b-be7d-2f4659c88eae |

| | |
|---|---|
| **Title** | **Record – Confirm Transfer** |
| **Description** | Confirm the transfer of the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Transferred Timestamp (**M14.4.106**)<br>• Disposal Action Code (**M14.4.18**)<br>• Disposal Action Due Date (**M14.4.19**)<br>• Disposal Confirmation Due Date (**M14.4.20**)<br>• Disposal Overdue Alert Timestamp (**M14.4.21**) |
| ***From functional requirement(s)*** | **R8.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Event Comment (**M14.4.25**) |

### F14.5.121 Record – Create

| | |
|---|---|
| **System Identifier** | 13d444bf-3ba2-4c38-adc5-b57ec9e86f74 |
| **Title** | **Record – Create** |
| **Description** | Create a record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Parent Aggregation Identifier (**M14.4.63**)<br>• Aggregated Timestamp (**M14.4.1**)<br>• Class Identifier (**M14.4.4**)<br>• Disposal Schedule Identifier (**M14.4.22**)<br>• *Contextual metadata elements*<br>*The following metadata element belonging to the record's parent aggregation will be modified:*<br>• Last Addition Timestamp (**M14.4.48**) |

| | |
|---|---|
| | *The following metadata element belonging to the record's parent aggregation may also be modified (if it has not been set previously):*<br>• First Used Timestamp (**M14.4.32**)<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R2.4.25, R6.5.10, R6.5.14, R7.5.18** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Participating New Parent Identifier (**M14.4.75**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *One or more components must be created simultaneously with the record (see **F14.5.38 Component – Create**)*<br>• *The Class Identifier applies only when overriding the record's inherited classification (from its parent aggregation) on creation*<br>• *The Disposal Schedule Identifier applies only when immediately overriding the record's default disposal schedule (inherited from its classification) on creation*<br>• *All records must have a Parent Aggregation Identifier and an Aggregated Timestamp*<br>• *The record may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event*<br>• *Where the record's inherited access controls are modified on creation then separate **F14.5.134 Record – Modify ACL** events must be generated for each change made to the access control list* |

### F14.5.122 Record - Delete Residual Event

| | |
|---|---|
| **System Identifier** | 9a91e950-48cd-4270-a592-dd41f4248ecc |
| **Title** | **Record – Delete Residual Event** |

| | |
|---|---|
| **Description** | Delete the event from the event history of the residual record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.123 Record - Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | 5812baba-fd3f-49c0-8560-2068d3f8c994 |
| **Title** | **Record – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the residual record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R7.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R7.5.7**)* |

### F14.5.124 Record - Destroy

| | |
|---|---|
| **System Identifier** | 508e5ad6-0c8a-4ece-9b46-b8b39b53c857 |
| **Title** | **Record – Destroy** |
| **Description** | Destroy the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**)<br>• Disposal Schedule Identifier (**M14.4.22**) |
| ***From functional requirement(s)*** | **R8.4.20, R8.4.24** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the update process (see **F14.5.140 Record – Update Disposal**) for records with components subject to automatic destruction, or otherwise by user confirmation (see **F14.5.119 Record – Confirm Destruction**)*<br>• *The Disposal Schedule Identifier is updated to ensure that the disposal schedule, under which the record was destroyed, stays with the residual record even if it was previously inherited*<br>• *The components of the record must be simultaneously automatically destroyed (see **F14.5.41 Component – Destroy**), and the record's aggregation may also be destroyed under **R8.4.22** (see **F14.5.9 Aggregation – Destroy**)*<br>• *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the record on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.125 Record - Disposal Alert

| | |
|---|---|
| **System Identifier** | fdb351ed-d8eb-41e9-9e15-8b1bf32a3798 |
| **Title** | **Record – Disposal Alert** |
| **Description** | Alert for the record sent when the disposal confirmation due date has elapsed |

| | |
|---|---|
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Disposal Overdue Alert Timestamp (**M14.4.21**) |
| ***From functional requirement(s)*** | **R8.4.15** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Participating User Identifier (**M14.4.81**)<br>• Overdue Disposal Action Code (**M14.4.58**)<br>• Overdue Disposal Action Due Date (**M14.4.59**)<br>• Overdue Disposal Confirmation Due Date (**M14.4.60**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *Users receive alerts by being granted roles that include this function*<br>• *The function itself is always automatically performed by the MCRS as part of the disposal process (see **F14.5.140 Record – Update Disposal**)*<br>• *The event should include a Participating User Identifier for each user who was sent the alert*<br>• *The Overdue Disposal Action Code should indicate the disposal action that was due on the record when the alert was sent*<br>• *The Overdue Disposal Confirmation Due Date should indicate the confirmation due date for the record when the alert was sent*<br>• *The Event Comment may be used by the MCRS to give additional details of the alert (alerts may be sent using many different mechanisms and technologies, see **R8.4.15**)* |

### F14.5.126 Record - Duplicate

| | |
|---|---|
| **System Identifier** | e6f97ac3-c9cc-4036-a471-a00c865db8a6 |
| **Title** | **Record – Duplicate** |
| **Description** | Duplicate a record |
| **Entity Type** | Record (**E14.2.12**) |

| | |
|---|---|
| **Entity metadata modified** | *The following entities will be duplicated:*<br>• *The record and all its metadata;*<br>• *Every event and all its metadata, in the event history of the record; and*<br>• *Every component of the record, and all its metadata.*<br>*For each record, component and event, including both the original and the duplicates, the following metadata will be added to identify the corresponding record, component or event in the MCRS:*<br>• Duplicate Identifier (**M14.4.23**) |
| ***From functional requirement(s)*** | **R6.5.16** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Participating Duplicate Identifier (**M14.4.69**)<br>• Event Comment (**M14.4.25**)<br>• Duplicate Identifier (**M14.4.23**) |
| ***Usage notes*** | • *Two duplicate events will be generated, one for the first record that identifies the second record as the duplicate using the Participating Duplicate Identifier, and one for the second record that identifies the first record as the duplicate using the Participating Duplicate Identifier*<br>• *The duplicate events will be linked by the Duplicate Identifier in the event entity* |

### F14.5.127 Record - Exported

| | |
|---|---|
| **System Identifier** | 1140f9cc-99e9-40e3-b3dd-be485170f718 |
| **Title** | **Record – Exported** |
| **Description** | The record has been exported in full or as a placeholder |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |
| ***Purpose*** | Event generation only |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.128 Record - Held

| | |
|---|---|
| **System Identifier** | 38f887ed-7021-460d-8820-d26af5ce63a1 |
| **Title** | **Record – Held** |
| **Description** | Indicate that the record is subject to a disposal hold |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Disposal Action Code (**M14.4.18**)<br>• Disposal Action Due Date (**M14.4.19**)<br>• Disposal Confirmation Due Date (**M14.4.20**) |
| ***From functional requirement(s)*** | **R8.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the update process (see **F14.5.140 Record – Update Disposal**)*<br>• *The purpose of the function is to generate an event in the event history of the record*<br>• *The Disposal Action Code must be changed from DESTROY to RETAIN ON HOLD*<br>• *The Disposal Action Due Date and Disposal Confirmation Due Date must be deleted* |

| | |
|---|---|
| | • *See also **F14.5.139 Record – Released*** |

### F14.5.129 Record - Inherit Default Class

| | |
|---|---|
| **System Identifier** | e62728b4-13d7-4ef4-9833-7d8931c748e2 |
| **Title** | **Record – Inherit Default Class** |
| **Description** | Inherit the default classification of the record's parent aggregation |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata** | • Class Identifier (**M14.4.4**) |
| ***From functional requirement(s)*** | **R6.5.12, R6.5.14** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *Performing this function removes the Class Identifier from the record (see **F14.5.137 Record – Override Class**) ensuring the record inherits its classification from its parent aggregation* |

### F14.5.130 Record - Inherit Default Disposal Schedule

| | |
|---|---|
| **System Identifier** | eb4f94b8-9c0d-4f44-8d7c-b73c45735e49 |
| **Title** | **Record – Inherit Default Disposal Schedule** |
| **Description** | Inherit the default disposal schedule from the active record's class |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata** | • Disposal Schedule Identifier (**M14.4.22**) |
| ***From functional requirement(s)*** | **R6.5.14, R6.5.15** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *Performing this function removes the Disposal Schedule Identifier from the record (see **F14.5.138 Record – Override Disposal Schedule**) ensuring the record inherits the default disposal schedule from its classification* |

### F14.5.131 Record - Inspect

| | |
|---|---|
| **System Identifier** | 4f5f638f-f6ef-4917-adc0-82f90d065ef6 |
| **Title** | **Record – Inspect** |
| **Description** | Browse to the record, or discover it by searching, and inspect its metadata |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R6.5.9**, **R6.5.17**, **R9.4.7**, **R8.4.16** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the record and not when it is identified while browsing or included in search results* |

### F14.5.132 Record - Inspect ACL

| | |
|---|---|
| **System Identifier** | 7e4d0e6b-e726-4e3b-87a4-d1f6df60137e |
| **Title** | **Record – Inspect ACL** |
| **Description** | Inspect the access control list of the record |
| **Entity Type** | Record (**E14.2.12**) |

| | |
|---|---|
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**) |

### F14.5.133 Record – Inspect Event

| | |
|---|---|
| **System Identifier** | 29ee9ad5-6f9d-4663-9f99-99dca188c70e |
| **Title** | **Record – Inspect Event** |
| **Description** | Browse the event history of the record and inspect its events |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.134 Record – Modify ACL

| | |
|---|---|
| **System Identifier** | 5e3380b4-da4a-4e1c-9387-5b83934ff1c2 |
| **Title** | **Record – Modify ACL** |
| **Description** | Modify the access control list for the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |

| | |
|---|---|
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the record is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.135 Record – Modify Metadata

| | |
|---|---|
| **System Identifier** | b793efb9-fa12-41e9-9327-784324368bad |
| **Title** | **Record – Modify Metadata** |
| **Description** | Modify the metadata of the active record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• *Contextual metadata elements* |
| ***From functional requirement(s)*** | **R6.5.11** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77)**<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the record*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.136 Record – Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | 2249c5a3-e760-4a82-89b5-47b804ff4c32 |
| **Title** | **Record – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.137 Record – Override Class

| | |
|---|---|
| **System Identifier** | 069c29c9-15d9-42d8-8de4-a46b0405552a |
| **Title** | **Record – Override Class** |
| **Description** | Override the previous classification of the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata** | • Class Identifier (**M14.4.4**) |

| | |
|---|---|
| ***From functional requirement(s)*** | **R5.4.8, R6.5.12, R6.5.14** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *Performing this function adds a Class Identifier directly to the record or replaces it if the record already has a Class Identifier* |

### F14.5.138 Record – Override Disposal Schedule

| | |
|---|---|
| **System Identifier** | c53ebf62-c69f-4e3d-b728-33252f4faa01 |
| **Title** | **Record – Override Disposal Schedule** |
| **Description** | Override the previous disposal schedule of the active record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata** | • Disposal Schedule Identifier (**M14.4.22**) |
| ***From functional requirement(s)*** | **R6.5.14, R6.5.15, R8.4.13** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *Performing this function adds a Disposal Schedule Identifier directly to the record or replaces it if the record already has a Disposal Schedule Identifier* |

### F14.5.139 Record – Released

| | |
|---|---|
| **System Identifier** | 185d46fa-22c8-4a65-904b-c51604df1189 |
| **Title** | **Record – Released** |
| **Description** | Indicate that the record is no longer subject to any disposal hold |

| | |
|---|---|
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Disposal Action Code (**M14.4.18**)<br>• Disposal Action Due Date (**M14.4.19**)<br>• Disposal Confirmation Due Date (**M14.4.20**) |
| ***From functional requirement(s)*** | **R8.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the update process (see **F14.5.140 Record – Update Disposal**)*<br>• *The purpose of the function is to generate an event in the event history of the record*<br>• *The Disposal Action Code must be changed from RETAIN ON HOLD back to DESTROY*<br>• *The Disposal Action Due Date and Disposal Confirmation Due Date must be calculated and applied to the record*<br>• *See also **F14.5.128 Record – Held*** |

### F14.5.140 Record - Update Disposal

| | |
|---|---|
| **System Identifier** | 08f23ade-4023-46af-add7-39e6d399a5c3 |
| **Title** | **Record – Update Disposal** |
| **Description** | Update the disposal progress of the record |
| **Entity Type** | Record (**E14.2.12**) |
| **Entity metadata modified** | • Retention Start Date (**M14.4.93**)<br>• Disposal Action Code (**M14.4.18**)<br>• Disposal Action Due Date (**M14.4.19**)<br>• Disposal Confirmation Due Date (**M14.4.20**)<br>• Disposal Overdue Alert Timestamp (**M14.4.21**) |
| ***From functional requirement(s)*** | **R8.4.14** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Record Identifier (**M14.4.77**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *This function may be performed by a user and is also performed automatically by the MCRS in real time or at regular intervals*<br>• *When the metadata of the record is updated a Metadata Change Entry must be added to the event for each change*<br>• *An outcome of this function may be to trigger an alert (see **F14.5.125 Record – Disposal Alert**), the destruction of the record (see **F14.5.124 Record – Destroy**), to indicate a disposal schedule is in place (see **F14.5.128 Record – Held**) or has been lifted (see **F14.5.139 Record – Released**)* |

### F14.5.141 Role - Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | c10b0afa-87b6-42be-a45d-55a663f2df54 |
| **Title** | **Role – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.142 Role - Add Function Definition

| | |
|---|---|
| **System Identifier** | b5782d06-9597-4c2b-aa18-1845e650ce05 |
| **Title** | **Role – Add Function Definition** |
| **Description** | Add the function definition to the active role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata** | • Function Definition Identifier (**M14.4.33**) |
| ***From functional requirement(s)*** | **R4.5.4** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Participating Function Definition Identifier (**M14.4.72**)<br>• Event Comment (**M14.4.25**) |

### F14.5.143 Role - Create

| | |
|---|---|
| **System Identifier** | 246af58d-72a1-4e03-9a1d-3fe7094e00af |
| **Title** | **Role – Create** |
| **Description** | Create a role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Is Administrative Role Flag (**M14.4.44**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• *Contextual metadata elements*<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R2.4.25, R4.5.1, R7.5.18** |

| | |
|---|---|
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *The role may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event*<br>• *Where the role is created with one or more active function definitions associated with it then separate **F14.5.142 Role – Add Function Definition** events must be generated for each function definition added to it on creation*<br>• *Where the role's inherited access controls are modified on creation then separate **F14.5.152 Role – Modify ACL** events must be generated for each change made to the access control list* |

### F14.5.144 Role - Delete

| | |
|---|---|
| **System Identifier** | 451fda23-8a3a-4a80-976f-9dd45a7eb1a5 |
| **Title** | **Role – Delete** |
| **Description** | Delete the unused role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | *The unused role is deleted along with its metadata and event history* |
| ***From functional requirement(s)*** | **R4.5.5** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | *No event is generated* |

### F14.5.145 Role - Delete Residual Event

| | |
|---|---|
| **System Identifier** | 8eec771e-6441-4b58-b771-cccbaa94c55f |
| **Title** | **Role – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.146 Role - Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | 55e02943-a58f-4f31-8207-dc44a390bd03 |
| **Title** | **Role – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the residual role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R7.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |

| | |
|---|---|
| ***Usage notes*** | *This function always generates an event (see **R7.5.7**)* |

### F14.5.147 Role - Destroy

| | |
|---|---|
| **System Identifier** | b969e173-ec5f-4046-95b5-f8e903d1e77b |
| **Title** | **Role – Destroy** |
| **Description** | Destroy the active role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R4.5.6** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Event Comment (**M14.4.25**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the role on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.148 Role - Exported

| | |
|---|---|
| **System Identifier** | 32673385-e87a-410c-ad54-0556ae1ed332 |
| **Title** | **Role – Exported** |
| **Description** | The role has been exported in full or as a placeholder |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |

| | |
|---|---|
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.149 Role - Inspect

| | |
|---|---|
| **System Identifier** | f5833cf6-4d4a-43f9-bea8-77f768ba4e72 |
| **Title** | **Role – Inspect** |
| **Description** | Browse to the role, or discover it by searching, and inspect its metadata |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the role and not when it is identified while browsing or included in search results* |

### F14.5.150 Role - Inspect ACL

| | |
|---|---|
| **System Identifier** | 5a336626-eab8-451e-8e7c-547b8933f5f4 |

| | |
|---|---|
| **Title** | **Role – Inspect ACL** |
| **Description** | Inspect the access control list of the role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**) |

### F14.5.151 Role - Inspect Event

| | |
|---|---|
| **System Identifier** | bd6c020f-9ef6-4263-880f-99f28b75ac13 |
| **Title** | **Role – Inspect Event** |
| **Description** | Browse the event history of the role and inspect its events |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.152 Role - Modify ACL

| | |
|---|---|
| **System Identifier** | 2803e039-8e4c-44a3-9e50-aebf5c2649e7 |
| **Title** | **Role – Modify ACL** |

| | |
|---|---|
| **Description** | Modify the access control list for the role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the role is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.153 Role - Modify Metadata

| | |
|---|---|
| **System Identifier** | 59c2b665-aa89-4e87-8666-3d294f35a9e8 |
| **Title** | **Role – Modify Metadata** |
| **Description** | Modify the metadata of the active role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• *Contextual metadata elements*<br>*In accordance with **R4.5.3**, the following metadata may only be modified prior* |

| | |
|---|---|
| | *to the role being used:*<br>• Is Administrative Role Flag (**M14.4.44**) |
| ***From functional requirement(s)*** | **R4.5.2**, **R4.5.3** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the role*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.154 Role – Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | 4cd52e0d-7972-499a-bc32-d7c81540094c |
| **Title** | **Role – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.155 Role – Remove Function Definition

| | |
|---|---|
| **System Identifier** | d0df9562-d560-4374-8ed8-07e4c3572d2a |

| | |
|---|---|
| **Title** | **Role – Remove Function Definition** |
| **Description** | Remove the function definition from the active role |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata** | • Function Definition Identifier (**M14.4.33**) |
| ***From functional requirement(s)*** | **R4.5.4** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Participating Function Definition Identifier (**M14.4.72**)<br>• Event Comment (**M14.4.25**) |

### F14.5.156 Role - Report Function Definitions

| | |
|---|---|
| **System Identifier** | 00a53f17-9866-4f0f-a387-16b549249284 |
| **Title** | **Role – Report Function Definitions** |
| **Description** | Report the function definitions that belonged to the role at a specified historical date and time |
| **Entity Type** | Role (**E14.2.13**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.14** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Role Identifier (**M14.4.78**)<br>• Historical Date/Time (**M14.4.40**) |

### F14.5.157 Service - Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | e2a1ab27-0a63-4abe-a2c7-a08c8027824c |
| **Title** | **Service – Add Contextual Metadata** |

| | |
|---|---|
| **Description** | Add one or more contextual metadata element definitions to the service |
| **Entity Type** | Service (**E14.2.14**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Service Identifier (**M14.4.79**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.158 Service - Inspect

| | |
|---|---|
| **System Identifier** | e9a0bfa5-6292-4ba2-80c4-eeab1a4c9a7f |
| **Title** | **Service – Inspect** |
| **Description** | Browse to the service, or discover it by searching, and inspect its metadata |
| **Entity Type** | Service (**E14.2.14**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.3** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Service Identifier (**M14.4.79**) |

| | |
|---|---|
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the service and not when it is identified while browsing or included in search results* |

### F14.5.159 Service - Inspect ACL

| | |
|---|---|
| **System Identifier** | ad683cd9-9f6a-45f7-a00d-a34550196b27 |
| **Title** | **Service – Inspect ACL** |
| **Description** | Inspect the access control list of the service |
| **Entity Type** | Service (**E14.2.14**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Service Identifier (**M14.4.79**) |

### F14.5.160 Service - Inspect Event

| | |
|---|---|
| **System Identifier** | 40015ae8-17d7-4be4-a381-ae79066351ac |
| **Title** | **Service – Inspect Event** |
| **Description** | Browse the event history of the service and inspect its events |
| **Entity Type** | Service (**E14.2.14**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating Service Identifier (**M14.4.79**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.161 Service – Modify ACL

| | |
|---|---|
| **System Identifier** | 537292ac-e4c4-45c0-8188-5597ffa58997 |
| **Title** | **Service – Modify ACL** |
| **Description** | Modify the access control list for the service |
| **Entity Type** | Service (**E14.2.14**) |
| **Entity metadata modified** | • Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Service Identifier (**M14.4.79**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the service is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.162 Service – Modify Metadata

| | |
|---|---|
| **System Identifier** | d5be380f-b172-402a-898b-117d968c0ce3 |
| **Title** | **Service – Modify Metadata** |

| | |
|---|---|
| **Description** | Modify the metadata of the service |
| **Entity Type** | Service (**E14.2.14**) |
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Owner Information (**M14.4.62**)<br>• *Contextual metadata elements* |
| ***From functional requirement(s)*** | **R2.4.4** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Service Identifier (**M14.4.79**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the service*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.163 Service - Report Compliance

| | |
|---|---|
| **System Identifier** | 8dbe7172-8919-4c42-a989-3d114a0812d4 |
| **Title** | **Service – Report Compliance** |
| **Description** | Report the compliance status of the service |
| **Entity Type** | Service (**E14.2.14**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.5** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Service Identifier (**M14.4.79**) |

| | |
|---|---|
| ***Usage notes*** | *The report should list only those services, or service bundles under **R2.4.1**, for which the user has a role that includes this function* |

### F14.5.164 Template - Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | 9540ba23-531f-45cf-9d9c-99150890bd2f |
| **Title** | **Template – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the template |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.165 Template - Create

| | |
|---|---|
| **System Identifier** | 92634d40-ac65-4f87-ae90-fbe914dfe318 |
| **Title** | **Template – Create** |
| **Description** | Create a template |
| **Entity Type** | Template (**E14.2.15**) |

| | |
|---|---|
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Template Entity Type Identifier (**M14.4.102**)<br>• Template Service Identifier (**M14.4.103**)<br>• Template Class Identifier (**M14.4.101**)<br>• Contextual Metadata Element Definition Identifier (**M14.4.8**)<br>• *Contextual metadata elements*<br>*If contextual metadata elements are applied from a template the following metadata element may be modified belonging to the template from which the contextual metadata elements were applied (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R2.4.25, R7.5.14, R7.5.18** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *The template may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event*<br>• *Where the template's inherited access controls are modified on creation then separate **F14.5.174 Template – Modify ACL** events must be generated for each change made to the access control list* |

### F14.5.166 Template - Delete

| | |
|---|---|
| **System Identifier** | 5643fb8d-380b-4b0a-a30c-e0f09d3c2b0e |
| **Title** | **Template – Delete** |
| **Description** | Delete the unused template |

| | |
|---|---|
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | *The unused template is deleted along with its metadata and event history* |
| ***From functional requirement(s)*** | **R7.5.16** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | *No event is generated* |

### F14.5.167 Template - Delete Residual Event

| | |
|---|---|
| **System Identifier** | 5a95ef31-8881-424e-adfa-e6a4bfd3456e |
| **Title** | **Template – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual template |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.168 Template - Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | aa439637-6d4f-43e0-a63f-90b96f6be54f |
| **Title** | **Template – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the residual template |
| **Entity Type** | Template (**E14.2.15**) |

| | |
|---|---|
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R7.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R7.5.7**)* |

### F14.5.169 Template - Destroy

| | |
|---|---|
| **System Identifier** | 37ff38c4-e14e-442c-aca6-1f1ad94ed41f |
| **Title** | **Template – Destroy** |
| **Description** | Destroy the active template |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R7.5.17** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Event Comment (**M14.4.25**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the template on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.170 Template - Exported

| | |
|---|---|
| **System Identifier** | 376e669e-9c7a-455c-bc2c-a46f344ad6da |
| **Title** | **Template – Exported** |
| **Description** | The template has been exported in full or as a placeholder |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.171 Template - Inspect

| | |
|---|---|
| **System Identifier** | 1d699116-21ca-4318-8b2c-5a72851de157 |
| **Title** | **Template – Inspect** |
| **Description** | Browse to the template, or discover it by searching, and inspect its metadata |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R7.5.12** |

| | |
|---|---|
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the template and not when it is identified while browsing or included in search results* |

### F14.5.172 Template - Inspect ACL

| | |
|---|---|
| **System Identifier** | 9c75f837-16aa-4e1c-b8d7-705376cd800b |
| **Title** | **Template – Inspect ACL** |
| **Description** | Inspect the access control list of the template |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**) |

### F14.5.173 Template - Inspect Event

| | |
|---|---|
| **System Identifier** | 142ba8e5-1cdf-4ca7-a6d6-2f8db1e4b08a |
| **Title** | **Template – Inspect Event** |
| **Description** | Browse the event history of the template and inspect its events |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | *No metadata elements are modified* |

| | |
|---|---|
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.174 Template - Modify ACL

| | |
|---|---|
| **System Identifier** | f5b7c65a-9b73-4769-969a-fde91d197381 |
| **Title** | **Template – Modify ACL** |
| **Description** | Modify the access control list for the template |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry*<br>• *If more than one access control entry belonging to the template is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.175 Template - Modify Metadata

| | |
|---|---|
| **System Identifier** | d49046d1-984a-4ff0-b676-8598c2577466 |
| **Title** | **Template – Modify Metadata** |
| **Description** | Modify the metadata of the template |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Template Entity Type Identifier (**M14.4.102**)<br>• Template Service Identifier (**M14.4.103**)<br>• Template Class Identifier (**M14.4.101**)<br>• Contextual Metadata Element Definition Identifier (**M14.4.8**)<br>• *Contextual metadata elements* |
| ***From functional requirement(s)*** | **R7.5.15** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the template*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.176 Template - Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | acf69ef4-7c80-48de-a11a-896c09d58b05 |
| **Title** | **Template – Modify Originated Date/Time** |
| **Description** | Modify the Originated Date/Time of the active template |
| **Entity Type** | Template (**E14.2.15**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |

| | |
|---|---|
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Template Identifier (**M14.4.80**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.177 User - Add Contextual Metadata

| | |
|---|---|
| **System Identifier** | 2601f4d6-79df-4978-b860-b8e17b6520ce |
| **Title** | **User – Add Contextual Metadata** |
| **Description** | Add one or more contextual metadata element definitions to the user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | • *Additional contextual metadata elements, as specified*<br>*Applying contextual metadata elements from a template may also modify the following template metadata element (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R7.5.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata (the template is not considered a participating entity)* |

### F14.5.178 User - Browse Records Due For Disposal

| | |
|---|---|
| **System Identifier** | 2c3722d2-bc17-4008-9c60-58d3d7d72f83 |
| **Title** | **User – Browse Records Due For Disposal** |

| | |
|---|---|
| **Description** | Access the records currently due for disposal by review, transfer and destruction |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R8.4.16** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**) |
| ***Usage notes*** | *This function enables the user to browse the records currently due for disposal, to inspect any of these records individually requires **F14.5.131 Record – Inspect*** |

### F14.5.179 User - Create

| | |
|---|---|
| **System Identifier** | 2cde7448-6c71-4cff-988a-973e0701a824 |
| **Title** | **User – Create** |
| **Description** | Create a user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• *Contextual metadata elements*<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R2.4.25, R3.4.2, R7.5.18** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *This function may occur in another system external to the MCRS and be synchronised to it*<br>• *The user entity may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event*<br>• *Where the user is created as a member of one or more active groups then separate events for **F14.5.94 Group – Add User** must be automatically for each group the user is created in, on creation*<br>• *Where the user's inherited access controls are modified on creation then separate **F14.5.190 User – Modify ACL** events must be generated for each change made to the access control list* |

### F14.5.180 User - Delete

| | |
|---|---|
| **System Identifier** | 3be51e0b-efd1-49e9-be3d-419ef6a60660 |
| **Title** | **User – Delete** |
| **Description** | Delete the unused user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *The unused user entity is deleted along with its metadata and event history* |
| ***From functional requirement(s)*** | **R3.4.5** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | *No event is generated* |

### F14.5.181 User - Delete Residual Event

| | |
|---|---|
| **System Identifier** | b2593eb6-a542-4d20-8942-3f2cd0218edf |

| | |
|---|---|
| **Title** | **User – Delete Residual Event** |
| **Description** | Delete the event from the event history of the residual user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | • *No metadata elements are modified*<br>• *The event entity is deleted* |
| ***From functional requirement(s)*** | **R2.4.21** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R2.4.14**)* |

### F14.5.182 User - Delete Residual Metadata

| | |
|---|---|
| **System Identifier** | a8783b53-6557-4885-88c5-3781508809cd |
| **Title** | **User – Delete Residual Metadata** |
| **Description** | Delete the element from the metadata of the residual user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *Any metadata element may be deleted, including both system and contextual metadata elements, except a system identifier or a timestamp* |
| ***From functional requirement(s)*** | **R7.5.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | *This function always generates an event (see **R7.5.7**)* |

### F14.5.183 User - Destroy

| | |
|---|---|
| **System Identifier** | c33e48b3-2bc6-4851-b2e7-a40d602bfe1c |
| **Title** | **User – Destroy** |
| **Description** | Destroy the active user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | • Destroyed Timestamp (**M14.4.17**) |
| ***From functional requirement(s)*** | **R3.4.6** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Event Comment (**M14.4.25**)<br>• Deleted Metadata Element Definition Identifier (**M14.4.15**)<br>• Deleted Event Function Definition Identifier (**M14.4.14**) |
| ***Usage notes*** | *The Deleted Metadata Element Definition Identifier and Deleted Event Function Definition Identifier are used to show what metadata elements and which types of events were pruned from the event history of the user entity on its destruction, under **R2.4.20** and **R7.5.6*** |

### F14.5.184 User - Detailed Report

| | |
|---|---|
| **System Identifier** | 428eb316-5eb8-4bda-87d6-fa4cc42331a3 |
| **Title** | **User – Detailed Report** |
| **Description** | Generate a detailed report based on a search query |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R10.4.24, R10.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Search Query (**M14.4.98**)<br>• Total Entities (**M14.4.105**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *The detailed report is based on performing a search which should be described by the Search Query, see **R10.4.22** and **R10.4.24***<br>• *The Total Entities should indicate the number of entities included in the detailed report, see **R10.4.20** and **R10.4.24**, this number may be an approximation*<br>• *Further supplementary information can be placed into the Event Comment, as required* |

### F14.5.185 User - Export

| | |
|---|---|
| **System Identifier** | 1262769d-024d-430f-a917-48b6a8e58d21 |
| **Title** | **User – Export** |
| **Description** | Export entities from the MCRS |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.1**, **R11.4.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Export Identifier (**M14.4.30**)<br>• Export Commencing Timestamp (**M14.4.28**)<br>• Export Completed Timestamp (**M14.4.29**)<br>• Total Entities (**M14.4.105**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Total Entities should indicate the total number of entities that were exported, including entities that were exported in full, entities that were exported as placeholders, and events*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

| | |
|---|---|
| | • *This is not the event which is appended to the event history of the entity following its export under **R11.4.10**, the event for this function becomes part of the event history of the user*<br>• *See also **F14.5.10**, **F14.5.29**, **F14.5.43**, **F14.5.52**, **F14.5.62**, **F14.5.76**, **F14.5.100**, **F14.5.127**, **F14.5.148**, **F14.5.170**, and **F14.5.186*** |

### F14.5.186 User - Exported

| | |
|---|---|
| **System Identifier** | 6da05d24-5cbb-4b2b-9eb8-45d75564c6ab |
| **Title** | **User – Exported** |
| **Description** | The user has been exported in full or as a placeholder |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R11.4.10** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Export Identifier (**M14.4.30**)<br>• Exported In Full Flag (**M14.4.31**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is performed automatically by the MCRS as a result of the export process (see **F14.5.185 User - Export**) for all entities which are exported whenever a user conducts an export under **R11.4.1***<br>• *The Export Identifier is the system identifier generated by the MCRS for the export under **R11.4.4***<br>• *The Exported In Full Flag should be set if the entity was exported in full and cleared if the entity was exported as a placeholder*<br>• *The Event Comment contains the export comment under **R11.4.5*** |

### F14.5.187 User - Inspect

| | |
|---|---|
| **System Identifier** | 246c3b2f-66cb-4585-8b6f-d38162ddaf99 |
| **Title** | **User – Inspect** |
| **Description** | Browse to the user, or discover it by searching, and inspect its metadata |

| | |
|---|---|
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R3.4.14** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**) |
| ***Usage notes*** | *An event should only be generated for this function when the user examines the metadata of the user and not when it is identified while browsing or included in search results* |

### F14.5.188 User - Inspect ACL

| | |
|---|---|
| **System Identifier** | 93dde507-4b84-4231-9641-8572f7b14c13 |
| **Title** | **User – Inspect ACL** |
| **Description** | Inspect the access control list of the user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.9** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**) |

### F14.5.189 User - Inspect Event

| | |
|---|---|
| **System Identifier** | 8fe366ae-0e72-4042-8a8a-57e296927ee5 |
| **Title** | **User – Inspect Event** |

| | |
|---|---|
| **Description** | Browse the event history of the user and inspect its events |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R2.4.19** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Participating Event Identifier (**M14.4.71**) |

### F14.5.190 User - Modify ACL

| | |
|---|---|
| **System Identifier** | ef1deffd-68a5-4f19-ac54-ae3184d991d3 |
| **Title** | **User – Modify ACL** |
| **Description** | Modify the access control list for the user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | • Include Inherited Roles Flag (**M14.4.43**)<br>• Access Control Entry (**D14.3.1**) |
| ***From functional requirement(s)*** | **R4.5.10** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Participating User Or Group Identifier (**M14.4.82**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Granted Role Identifier (**M14.4.35**)<br>• Rescinded Role Identifier (**M14.4.87**) |
| ***Usage notes*** | • *If the value of the Include Inherited Roles Flag is modified then a Metadata Change Entry must be added to the corresponding event*<br>• *The Participating User Or Group Identifier refers to the user or group which is associated with the access control entry and should not be confused with the Participating User Identifier which indicates the user* |

| | |
|---|---|
| | *entity that has the access control list*<br>• *If more than one access control entry belonging to the user is modified simultaneously then one event must be generated for each access control entry that is added, removed or modified*<br>• *The event metadata shows which new roles were granted to the participating user or group and which existing roles were rescinded through the adding, modifying and deleting of access control entries* |

### F14.5.191 User - Modify Metadata

| | |
|---|---|
| **System Identifier** | 53b191b5-def2-4ab3-a110-3959c52784dc |
| **Title** | **User – Modify Metadata** |
| **Description** | Modify the metadata of the active user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata** | • Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• *Contextual metadata elements* |
| ***From functional requirement(s)*** | **R3.4.3** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | • *This function may occur in another system external to the MCRS and be synchronised to it*<br>• *Any of the system metadata elements listed may be modified and any modifiable contextual metadata elements belonging to the user entity*<br>• *For each metadata element modified a Metadata Change Entry must be added to the event generated by performing the function* |

### F14.5.192 User - Modify Originated Date/Time

| | |
|---|---|
| **System Identifier** | e3e88cce-a4b1-4860-9868-d9561f7417b3 |
| **Title** | **User – Modify Originated Date/Time** |

| | |
|---|---|
| **Description** | Modify the Originated Date/Time of the active user |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | • Originated Date/Time (**M14.4.61**) |
| ***From functional requirement(s)*** | **R2.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**) |
| ***Usage notes*** | *The event for this function must always have an Event Comment (see **R2.4.26**)* |

### F14.5.193 User – Report Authorisation

| | |
|---|---|
| **System Identifier** | 25315f00-835e-4dfb-9894-80688eadbf2b |
| **Title** | **User – Report Authorisation** |
| **Description** | Report the functions and the active roles containing those functions that the user may perform on a nominated entity |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.13** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>*And one of:*<br>• Participating Aggregation Identifier (**M14.4.64**)<br>• Participating Class Identifier (**M14.4.65**)<br>• Participating Component Identifier (**M14.4.66**)<br>• Participating Disposal Hold Identifier (**M14.4.67**)<br>• Participating Disposal Schedule Identifier (**M14.4.68**)<br>• Participating Entity Type Identifier (**M14.4.70**) |

| | |
|---|---|
| | • Participating Function Definition Identifier (**M14.4.72**)<br>• Participating Group Identifier (**M14.4.73**)<br>• Participating Metadata Element Definition Identifier (**M14.4.74**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Participating Role Identifier (**M14.4.78**)<br>• Participating Service Identifier (**M14.4.79**)<br>• Participating Template Identifier (**M14.4.80**) |
| ***Usage notes*** | • *This function is run against the user entity nominated by the authorised user*<br>• *Therefore the Participating User Identifier refers to the nominated user entity*<br>• *A maximum of one other participating entity, representing the nominated entity, should be included in the event in addition to the Participating User Identifier*<br>• *If no other participating entities are included in the event then the nominated user's own user entity is also the nominated entity* |

### F14.5.194 User – Report Group Membership

| | |
|---|---|
| **System Identifier** | 868fd38a-74d4-4534-b3ec-37b0ec4d65aa |
| **Title** | **User – Report Group Membership** |
| **Description** | Report the active groups that a nominated user belonged to at a specified historical date/time |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R3.4.7** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Historical Date/Time (**M14.4.40**) |
| ***Usage notes*** | • *This function is run against the user entity nominated by the authorised user*<br>• *Therefore the Participating User Identifier refers to the nominated user entity* |

### F14.5.195 User - Search

| | |
|---|---|
| **System Identifier** | 34dcf951-8608-46b5-808f-84fe8c378e7a |
| **Title** | **User – Search** |
| **Description** | Search for entities |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R4.5.15**, **R6.5.18**, **R10.4.1**, **R10.4.22** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Search Query (**M14.4.98**)<br>• Total Entities (**M14.4.105**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *The Search Query should contain a description of the search query performed, see **R10.4.22***<br>• *The Total Entities should indicate the number of entities found by the search, see **R10.4.20**, this number may be an approximation*<br>• *Further supplementary information can be placed into the Event Comment, as required* |

### F14.5.196 User - Summary Report

| | |
|---|---|
| **System Identifier** | aafece46-99dd-4e55-b0b2-81e5c0f1cf23 |
| **Title** | **User – Summary Report** |
| **Description** | Generate a summary report based on multiple search queries |
| **Entity Type** | User (**E14.2.16**) |
| **Entity metadata modified** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R10.4.25**, **R10.4.26** |
| ***Purpose*** | • Access control<br>• Event generation |

| | |
|---|---|
| **Additional event metadata (see R2.4.16)** | • Participating User Identifier (**M14.4.81**)<br>• Search Query (**M14.4.98**)<br>• Total Entities (**M14.4.105**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *One Search Query should be added to the event for each search included in the summary report, see* ***R10.4.25***<br>• *The Total Entities should indicate the grand total of all entities found across all search queries, see* ***R10.4.20*** *and* ***R10.4.25****, this number may be an approximation*<br>• *Further supplementary information can be placed into the Event Comment, as required* |

# 15. Acknowledgements

Developing a specification like MoReq2010® requires a large amount of energy and commitment from a great many people. Because records management is a technical and highly specialised field, these contributors are usually leading national and international experts in the discipline who volunteer their time and effort on an unpaid basis.

The DLM Forum® would like to thank everyone who contributed to the development of MoReq2010®.

## 15.1 Project Team

*The MoReq2010® work programme took a year. It was launched at the DLM Forum AGM in Madrid in May 2010, and MoReq2010® was officially launched at the DLM Forum AGM in Budapest in May 2011.*

### 15.1.1 Author

Jon Garde, Journal IT

### 15.1.2 Editor

Richard Blake, The National Archives, UK

### 15.1.3 Programme management

Simon Cole, Automated Intelligence

Martin Waldron, Weathervane Consult

### 15.1.4 Technical team

Hans Fredrik Berg, National Archives of Norway

Richard Jeffrey-Cook, InForm Consult

### 15.1.5 Proofreader

Paul Sutcliffe, Consultant

## 15.2 Experts Review Group

*The DLM Forum® acknowledges the generous assistance of the European Commission in sponsoring the MoReq2010® Experts Review Group, facilitating communications and coordinating meetings.*

### 15.2.1 Chair

Malcolm Todd, The National Archives, UK

### 15.2.2 European Commission

Jef Schram, European Commission

### 15.2.2 Experts

Francisco Barbedo, National Archives of Portugal

Diane Carlisle, ARMA International

Tracy Caughell, OpenText Corporation

Marie-Anne Chabin, Archive 17

Elena Cortés Ruiz, National Archives of Spain

Andrew Ewing, Hewlett Packard Software

Maria Guercio, University of Urbino

Andrea Hänger, Bundesarchiv

Hans Hofman, National Archives of the Netherlands/International Council on Archives

Toivo Jullinen, National Archives of Estonia

Julie McLeod, University of Northumbria

Bogdan-Florin Popovici, National Archives of Romania

Barbara Reed, Recordkeeping Innovation

Jože Škofljanec, National Archives of Slovenia

## 15.3 Consultees

*The following people contributed to the formal public review of the specification during the MoReq2010® concept consultation (held in the summer of 2010) and the MoReq2010® draft consultation (held in the winter of 2010/11).*

Malcolm Beach, In-Form Consult

Steve Bertone, OpenText Corporation

Kris Brown, HP

Stephen Clarke, ISO Committee TC46/SC11 Archives & Records Management

Lucas Colet, Public Research Centre Henri Tudor

Julian Croker, Steria

Christian Dubourg, Ever-Team Software

Laznik Dušan, Mentis d.o.o.

Marc Fresko, Inforesight

Rita Gago, Arquivo Municipal de Lisboa (Lisbon Municipal Archive)

Kirsten Glenwright, Objective Corporation

Mariella Guercio, University of Urbino

Paul Hampton, Alfresco

Katrina Hinton, Objective Corporation

Vincent Caper Hoolt, European Central Bank

Gabor Hornyak, MATRIX Auditing Evaluating and Certification

Gregor Joeris, SER Software Technology

Ben Johnson, Objective Corporation

Ulrich Kampffmeyer, PROJECT CONSULT Unter

Hanns Köhler-Krüner, HKK Consulting

Robert Lentz, cBrain A/S

Marko Lukičić, Ericsson Nikola Tesla

Karl Mayrhofer, Fabasoft

Leo Merman, Celt Consultancy

Christoph Mueller, HTW Chur

Giovanni Michetti, University of Rome "La Sapienza"

Tony Ogilvie, Automated Intelligence

Maria Palma, AedocDigital

Osmo Polonenn, Finnish MoReq Working Group

Bogdan-Florin Popovici, National Archives of Romania

Professor Roberta Raimondi, Bocconi School of Management Milan

Shaheen Ramdiane, OpenText Corporation

Susana. B. Rodriguez, Ministry Defense Spain

John Seeley, IKM Solutions

Deirdre Sharp, Norfolk County Council

María José Aldaz Sola, Archivistica.Net

Lucia Stefan, Archiva

Ricardo Vieira, INESC-ID

Naina Visani, IKM Solutions

Chris Walker

Robert Whiter, Oracle

Anthony Woodward, Record Point Software

Sherry Xie, University of British Columbia


# PART TWO – PLUG-IN MODULES

# 100. INTERFACE SERIES

# 101. Graphical User Interface (GUI)

## 101.1 Module Information

| | |
|---|---|
| **Module Name** | **Graphical User Interface (GUI)** |
| **Module Version** | **1.0** |
| **Implements Module Identifier *(see M14.4.41)*** | 0f9584e5-552a-4a79-a8ea-3c2801765255 |
| **Prerequisites** | MoReq2010® Core Services |
| **Co-requisites** | *none* |

## 101.2 Key Concepts

### 101.2.1 Main features of a graphical user interface

A graphical user interface or GUI is the most common way for users to interact with modern computer technologies. A graphical user interface may be provided on any information system that has a screen and one or more input devices. Information is typically transmitted to a GUI through input devices such as keyboards, mice, tablets, pens/styli, touch screens and microphones that transmit voice commands. Feedback from a GUI is primarily visual through a screen or display, however some audio and other supplementary feedback is sometimes provided.

The key features of most graphical user interfaces are:

- They allow users to run more than one application at a time and to follow links and to switch between applications, for example, a user reading an email in an email client may click on a link that opens in a web browser;
- They allow users to see more than one item or piece of information at a time and to see some information as text and other information as images, icons or graphics;
- They allow users to manipulate and perform actions on items by clicking with a mouse, tapping with a finger or typing on a keyboard;
- They allow users to navigate visually between items by providing multiple item displays such as list views and tree views;
- They allow users to see the relationships between items;
- They allow users to select multiple items at once by lassoing them, or similar technique, and then perform the same actions on more than one item simultaneously;
- They allow users to manipulate items by dragging and dropping them onto other items;
- They allow users to switch between views, or windows, of the same dataset;

- They allow users to magnify entities, or see an alternative view, by zooming in and out;
- They are able to display information sorted in different ways and to indicate the status of items with various visual effects;
- They are able to overlay different information from parts of a system in a single coherent display;
- They can provide pop-up notifications and dialogue boxes for the user to interact with;
- They can simulate a form for the entry of metadata about an item;
- They can guide the user through complex processes through techniques such as a wizard;
- They can limit the operations that a user can choose from through pick lists and menus;
- They can disable functions that are not appropriate for a particular item;
- They can display search results, reports and the content of items directly to the screen or display; and
- They can be tailored and arranged to suit individual users and particular roles.

This module contains requirements for MCRS solutions that implement a GUI.

### 101.2.2 Graphical user interfaces and records systems

Where a records system has a GUI, all of the disparate functionality of the records system is brought together into that interface so that users may interact with it. The processes themselves may be running in different places, both locally and on remote servers and web servers. The different parts of the records system may connect with one another over local area networks, wide area networks, private intranets, or the public Internet. The GUI, however, should seek to give a unified and coherent view of the records system that hides the technical complexity of its underlying infrastructure.

For most records systems the GUI will either belong to a native application or to a web application hosted by a web browser. Where the GUI belongs to a native application it should seek, as much as possible, to follow the conventions and user interface guidelines for the operating system on which the application is being hosted. Where the GUI is part of a web application it should seek, as much as possible, to follow the conventions and accessibility guidelines of the World Wide Web Consortium (W3C). In both cases the GUI should seek to be easy to use and consistent with users' expectations, so as to afford users the greatest chance of being able to interact with and operate the records system with minimal previous experience, training and ergonomic effort.

In particular an MCRS must use its graphical user interface to:

- Allow users to interact with the contents of the records system by presenting entities grouped into logical structures, such as list views, tree views and network maps;
- Represent the status of entities through graphical means;
- By default show only active entities, but allow switching of the display to show residual entities as well;
- Facilitate browsing, as it is defined by MoReq2010®, for example from a parent aggregation, being able to easily access and inspect its children;

- Allow users to inspect and, where they have authorisation, modify the metadata of entities, for example through a form view;
- Facilitate the completion of tasks, such as the population of metadata elements, by providing aids such as default values, selection lists, spelling checkers, look up panels, and so on;
- Allow users to easily create entities in the records system, for example, possibly initiating the creation a record by dragging and dropping a document or link from the operating system or another application;
- Enable users to build search queries and specify search criteria visually, rather than through a scripted search expression language;
- Display the results of searches in a way that allows users to directly interact with and inspect the entities returned;
- Allow graphical interaction with search results, such as reordering columns, and so on;
- Possibly allow users to directly view the content of records through the graphical user interface, or to access and download record content;
- Show users only those operations that they are authorised to perform on an entity before they attempt them;
- Provide role-specific views for users with those responsibilities, for example, to allow records managers to monitor and carry out the disposal process;
- Enable users to perform operations on entities easily with the fewest possible number of steps by providing aids such as contextual and application menus, tool bars, keyboard key combinations, and so on;
- Provide feedback when operations take longer than a second or so, for example, through the use of busy icons and progress bars;
- Provide visual and possibly auditory feedback on the results of all operations, clearly indicating where they are successful and where they have failed;
- Where operations fail, provide meaningful messages to the user that indicate what has failed, preferably why it failed, and what to do about it;
- Cater for users with special needs by adhering to accessibility guidelines and, for example, supporting screen magnification, presenting the same information in different ways such as using a colour and an underline for hyperlinks, and so on;
- Allow users to personalise their own experience of the GUI, for example, by selecting the default colours, fonts and sizes, and arranging elements of the GUI in different places; and
- Provide immediately available assistance to users, such as context sensitive help, on line user guides, frequently asked questions (FAQ), wizards, tutorials, access to help desk support, and so on.

An MCRS may also use its graphical user interface to facilitate collaborative working by:

- Showing which users are currently accessing the records system;
- Facilitating communication, chats, screen sharing and electronic whiteboards, or their equivalent, between users who are using the records system simultaneously;
- Allowing users to post messages to individuals or groups of other users;
- Allowing users to place additional notes against entities in the records system for the benefit of other users;

- Promoting the use of and possibly hosting blogs and on line discussions;
- Allowing users to create task lists that refer to entities in the records system;
- Extending the use of task lists to whole groups rather than just individual users;
- Promoting the use of and possibly hosting individual and group calendars that show upcoming events and tasks;
- Allowing users to refer important tasks, such as complex disposal review decisions to other more expert users; and
- Provision of dashboards to monitor records system usage and status, such as a summary of records due and overdue for disposal action.

An MCRS must not allow its graphical user interface to:

- Access the records system if they are not authorised to access it;
- Show users that entities exist that they are not authorised to inspect;
- Allow users to perform operations that they are not authorised to perform;

### 101.2.3 Organisational considerations

An organisation using an MCRS with a graphical user interface may wish to brand it in particular ways, for example, so that it:

- Displays a corporate logo;
- Uses corporate colour schemes and fonts;
- Uses corporate defaults.
- Incorporates the organisation's own terms and conditions for access and use;
- Refers support enquires to a corporate help desk rather than supplier level support;
- Has customised user guides, FAQ, tutorials and wizards; and
- Has a default layout and presentation decided at an organisational level.

The organisation must also be aware of the security implications of having some or part of the records system installed on its users' personal computers and workstations, by asking such questions as:

- Does it cache records, record content or other elements locally?
- Is the cache encrypted?
- How easy is it to bypass or hijack its security controls?
- How secure is the communication between the client application and other server-based parts of the records system?
- Is this communication encrypted?
- Is the records system remotely accessible, for example, through third party computers at Internet cafés, etc.?

These considerations may also be necessary for other types of user interface, other than graphical user interfaces.

## 101.4 Functional Requirements

#### R101.4.1

In accordance with core services requirement **R2.4.6**, the MCRS must implement a graphical user interface (GUI) that allows authorised users access to the full functionality specified by the service or bundle of services.

*The term "GUI" describes any primarily visual interface that is controlled by a human operator. GUIs may be contrasted with earlier command line interfaces where systems are controlled by typing instructions or commands into a terminal.*

*To be certified as MoReq2010® compliant, the GUI must not implement only a part of the functionality specified by MoReq2010® for the service or bundle of services covered.*

**R101.4.2**

The MCRS must display entities in a consistent manner that enables users to recognise them by entity type, to identify them by Title or other metadata, and to see immediately their status.

*For example, a user should be able to immediately recognise a class from an aggregation or a disposal schedule.*

**R101.4.3**

The MCRS must show, by default, only active entities, but must allow users to display both active and residual entities.

*By default, residual entities should not be visible in the GUI. Where a user elects to display both active and residual entities, then their status should be clearly indicated, under* ***R101.4.2****.*

*See also* ***R2.4.22*** *and* ***R10.4.17****.*

**R101.4.4**

The MCRS must allow authorised users to create new entities and add them to the records system, specifying values for their metadata, including contextual metadata.

*For example, authorised users must be able to add new classes to the classification service, new aggregations to the record service, new disposal schedules to the disposal scheduling service, new disposal holds to the disposal holding service, etc.*

*Depending on the nature of the MCRS it may also allow users to create new records on an ad hoc basis, by dragging and dropping documents and links from the operating system or other applications. For other types of MCRS the creation of records will be more structured.*

*Metadata values may be provided by the user, or the MCRS may generate values (for example the System Identifier), or provide default values, or the MCRS may be able to automatically extract metadata values from the content of new records. In all cases, all mandatory metadata values must be present before the new entity is created in the MCRS.*

**R101.4.5**

The MCRS must facilitate the inspection of entities by displaying their metadata, including any contextual metadata, on request and allowing their values to be modified by an authorised user in accordance with the other requirements of MoReq2010®.

*MoReq2010® does not specify how a user might request a view of an entity's metadata through a GUI.*

#### R101.4.6

The MCRS must provide an authorised user creating an entity, under **R101.4.4**, or adding or modifying metadata, under **R101.4.5**, with automated assistance in completing the task, as appropriate, and must allow the user to search for and find entities to add to the metadata while modifying it.

*Assistance may be provided by pop up selection lists, autocompletion, spelling checkers, formatting masks, context sensitive help, etc. Searching and adding entities is necessary for many different tasks, for example, specifying the classification of an aggregation, or overriding the default disposal schedule of a record.*

#### R101.4.7

The MCRS must ensure that authorised users cannot modify read only metadata values and cannot enter values for metadata contrary to the datatype of the element.

*The MCRS must enforce and support the entry of correct formatted metadata including any constraints specified by the metadata element definition and its specified datatype.*

#### R101.4.8

In addition to metadata, under **R101.4.5** the MCRS must also allow an authorised user to view and, if authorised, modify the access control list (or equivalent) for an entity.

*The MoReq2010® model role service does not specify any particular way in which the access to entities should be controlled and managed, provided it meets the general principles of access control specified by MoReq2010® and is able to be exported as a MoReq2010® compliant access control list.*

#### R101.4.9

The MCRS must visually show the relationships between entities and allow authorised users to browse these relationships.

*For example, the MCRS must allow a user to inspect an aggregation, browse to and inspect its classification, browse to and inspect its event history, browse to and inspect its child records, browse to and inspect their disposal schedules, browse to and inspect their components, etc.*

*MoReq2010® does not specify how relationships between entities in a GUI should be shown, or the type of view required.*

#### R101.4.10

The MCRS must guide the user by showing what operations are available to perform on an entity and allowing the user to choose from the operations available.

*The GUI must responsively adapt its display to the authorisation available to the current user. It must not show the same palette of operations available to all users without providing some visual guidance as to which sub-set of operations are available to the current user.*

*For example, a context menu contains a full list of operations that can be performed on an entity of a particular type. If the current user does not have the relevant access controls to perform one of these operations for the selected entity then that option should be greyed out or removed from the menu. It should not be necessary for a user to initiate an operation before being informed that it is not available because the user does not have sufficient access control.*

**R101.4.11**

The MCRS must make the search function available to users from any view or screen, with the exception of configuration screens and modal dialogues.

*The user should not have to navigate to a special screen or view in order to conduct a search.*

**R101.4.12**

The MCRS must allow users to graphically construct search queries and complex search criteria, by selecting search terms and looking up entities, as required.

*Users of records systems with a GUI should not be required to enter search scripts using the keyboard or to use a search expression language.*

*For example, to build the search criterion, "Originated Date/Time is earlier than 1st April 2010", the GUI might allow a user to choose the metadata element definition, "Originated Date/Time" from a list, select the search term, "is earlier than" from another context sensitive control, and then enter "1st April 2010" by finding and picking it from a pop-up calendar.*

**R101.4.13**

The MCRS must allow users to graphically configure search results, including the metadata elements to be displayed, in what order they are to be displayed, and how the results are to be sorted.

*See also R10.4.18. Graphically configuring search results may include dragging and dropping columns to reorder them or clicking on column headings to sort by that metadata element. The GUI should also graphically support pagination (or the equivalent) of search results under* ***R10.4.19****.*

**R101.4.14**

The MCRS must allow authorised users to save sets of search results into different document formats, and potentially capture them as records.

*The documentary form of search results should be similar to a detailed report, under* ***R10.4.24****. For example, saving the results to a CSV datafile or a PDF datafile. The MCRS may also be able to directly capture the search results as a record, but this level of functionality is not included in the MoReq2010® core services.*

**R101.4.15**

The MCRS must allow authorised users to inspect and perform operations on entities directly from search results.

*When the GUI displays the results of a search then the entities shown in the list must have the same level of functionality as they do in other parts of the interface. In other words, the user should be able to inspect their metadata, browse them, perform operations, and so on; all without having to switch out of the search results to a different view. See also* ***R101.4.17****.*

**R101.4.16**

The MCRS must allow users to make shortcuts to entities that can be shared with other users.

*A shortcut is an external reference to an entity in the MCRS that can be used to initiate opening the GUI at a particular view and displaying the entity referred to by the shortcut.*

*An example of a shortcut is a URI hyperlink. Users should be able to save shortcuts to their local environment and share them with other GUI users, for example by sending them in emails.*

#### R101.4.17

The MCRS must allow a user to select a set of entities of the same type both from the display generally, and from search results, under **R101.4.13**, so as to collectively:

- Reclassify them with the same class, under **R6.5.4** or **R6.5.12**;
- Move them to the same parent, under **R6.5.8** or **R6.5.13**;
- Change their disposal schedule to the same disposal schedule, under **R6.5.15**;
- Carry out a review of them, under **R8.4.17**, cancel their transfer or destruction, under **R8.4.18**, or confirm their transfer or destruction, under **R8.4.19** and **R8.4.20**;
- Associate them with the same disposal hold, under **R9.4.3**;
- Export them, under **R11.4.3** and **R101.4.19**; or
- Create a single shortcut to a set of multiple entities rather than to an individual entity, under **R101.4.16**.

*MoReq2010® does not specify how multiple entities should be selected from search results.*

#### R101.4.18

The MCRS must provide a specialised interface for authorised users to engage with and complete the disposal process described in **8. Disposal Scheduling Service**.

*In particular, the interface must allow users to browse and inspect all active records that are due for disposal, without having to conduct a search query, under **R8.4.16**, including allowing them to be grouped in various ways as described. The interface must also facilitate the various disposal actions described in **R8.4.17**, **R8.4.18**, **R8.4.19** and **R8.4.20**, especially allowing them to be carried out across nominated groupings of records.*

*Desirable behaviour for this interface, that is not required for compliance with MoReq2010® includes:*

- *Being able to plan ahead and anticipate records coming due for disposal by calendar day, week and month;*
- *Collect, display and chart graphically statistics on disposal action throughput and average time taken to carry out various disposal actions; and*
- *Manage disposal holds, including which records are currently being held and determine how long overdue for destruction they are.*

#### R101.4.19

The MCRS must provide a specialised interface for authorised users to perform the export process described in **11. Export Service**.

*In particular the interface should facilitate the following:*

- *Enable a user to put together a list of entities for exporting, under R11.4.3, possibly by selecting them and moving them to a list, under **R101.4.17**;*
- *Enable a user to select options for exporting, such as **R11.4.2**;*

- *Enable the user to initiate the export, to monitor its progress and to pause and resume it, under **R11.4.7**; and*
- *Give the user immediate feedback on any errors encountered and the success or failure of the export.*

**R101.4.20**

The MCRS must provide feedback to users who have initiated lengthy operations and make provision for users to interrupt and cancel lengthy operations before they have completed.

*Specifically the interface must provide the following:*

- *Provide feedback to the user that the command has been received in under a second, preferably around 0.1 second to make the user feel that the interface is responsive;*
- *Provide feedback that the application is processing the command if it has not been completed in under 2 seconds, such as a spinning ball or hourglass cursor; and*
- *Show a progress indicator and allow the user to cancel the command if it has not been completed in under 10 seconds.*

*An example of an operation that might be lengthy and require these measures is when the user initiates a search.*

**R101.4.21**

Where operations are not successful, the MCRS must provide useful error messages that explain to the user the nature of the error, what happened and how to proceed.

*The MCRS must not allow user initiated operations to fail without informing the user. Similarly, error messages must be worded so as to be clearly understood by users with little or no technical background and they must indicate to the user what to do about the error so that the user can carry on in confidence.*

*Note that messages that are purely informational or that are warnings but not errors should be clearly differentiated from error messages so as not to inadvertently alarm users.*

*Error messages must be provided in a GUI in addition to logging failed operations, under **R2.4.8**, and are not a substitute for this requirement.*

**R101.4.22**

The MCRS must provide easily accessible help from its screens and views, including from its modal dialogues, for users who:

- Do not know how to navigate its graphical user interface;
- Do not understand the consequence of a particular operation;
- Do not know how to create entities;
- Do not know how to modify metadata;
- Do not know how to construct search queries and search criteria, particularly complex ones,
- Do not know where to find the controls for actions and operations,
- Do not know what to do about system errors, and
- Want to learn more about the system.

*Help may be provided in many forms, including instructions, manuals, FAQ, wizards and tutorials. The help must be relevant to the process the user is performing. Further detail is provided in the non-functional requirement* ***N101.5.11****.*

## 101.5 Non-functional Requirements

### N101.5.1

*The type of graphical user interface used will depend on the operating environment and platform.*

Further to the responses to **N12.3.1** and **N12.6.1**, what types of GUI does the records system support and what interfaces and operating systems are these interfaces available on?

### N101.5.2

*Good GUIs have an underlying rationale that provides consistency across the interface. The user learns to expect the GUI to respond in certain ways. The GUI may use icons consistently to represent different entity types and states. The GUI will also make it easy to see the difference between active and inactive entities, or functions that can be performed and those that can't.*

For each of the GUIs listed under **N101.5.1**, what is the design approach taken by the GUI, including:

- How does it make use of graphical elements such as icons and images?
- How does it maintain consistency across the interface?
- How does it assist the user to perform functions?

Provide examples where necessary.

### N101.5.3

*Ergonomics is an important albeit subjective judgment of all GUIs. For example, how many mouse clicks and movements does it take to perform a commonly used function? The best GUIs are often described as user friendly and intuitive to use.*

What is the average and the most number of user actions (for example, mouse clicks or finger touches) required to access any screen or dialogue in the records system interface, or to complete any of the following MoReq2010® functions from the default view or home screen of the application:

- Create a record in an aggregation?
- Search for an entity by its entity type using a single search criterion?
- Inspect the metadata of an entity in a set of search results?
- Browse from the first record to the last record in an aggregation of 100 records?

(The answer should outline what the default view or home screen is, and what each of the user actions are, that are required to perform a typical function and describe a function that takes n average number of user actions, as well as the function which takes the most user actions.)

### N101.5.4

*A GUI may use common gestures appropriate to the interface to allow the user to perform functions. For example, it may allow records to be captured by dragging and dropping datafiles from the user's desktop.*

How does the GUI use common gestures, such as scrolling and zooming, in appropriate ways to help make the records system more usable and useful?

**N101.5.5**

*A GUI will often support user personalisation. Alternative GUI elements, such as colours, layout, fonts and font sizes may also be available on different devices, such as desktops, notebooks, tablets and smartphones. The user may wish to set these personalisation options differently for each form factor. Personalisation requires that these elements, once changed are saved and stay with the particular user from session to session and from device to device, as the user so selects.*

*There are many ways in which users may personalise a GUI, it may involve some or all of:*

- *Inclusion in the interface of lists of recently used entities, see **N101.5.8**;*
- *Inclusion in the interface of a list of favourite entities, see **N101.5.9**;*
- *Ability to set display colours, fonts and font sizes;*
- *Ability to rearrange screen elements, views, and columns;*
- *Ability to set a personal default sort order for lists and views;*
- *Ability to specify the view where the application will open or have the application always open at the most recent view;*
- *Keep track of the user's location in the application using a breadcrumb trail or equivalent;*
- *Allow users to browse back to previous screens and views, see **N101.5.8**; and*
- *Provide an ability to create personal saved searches and saved report definitions, see **R10.4.23** and **R10.4.27**.*

In what ways can the GUI be personalised to suit individual users, and can these settings be saved as a user's personal default?

**N101.5.6**

*Audio alerts and notifications may also be configurable.*

Does the GUI make use of audio alerts and notifications, and can audio alerts, notifications, and volume settings be individually adjusted and personalised under **N101.5.5**?

**N101.5.7**

*It may be possible to navigate the GUI and perform functions using voice commands only.*

Does the GUI respond to voice commands and, if so, how comprehensive is the coverage of voice commands compared to the functionality required by MoReq2010®?

**N101.5.8**

*Many GUIs store up the places that a user visits and the entities that a user accesses. This list then provides a means for users to return to entities that they created, inspected or modified recently without having to search or browse to find them.*

Does the records system provide easy access to recently accessed entities through the GUI so that the user can find them and access them later without searching?

**N101.5.9**

*Some GUIs allow users to add entities to a list of favourites. This allows users to find and access these favourite entities more easily without having to browse or search to find them. This is a potentially*

*important feature because users may often need to have access to only a small number of aggregations to do most of their daily work capturing, declaring and referring to records. Many users will wish to keep these aggregations close at hand, rather than constantly search for them. Another good use of the favourites list is to keep saved searches and reports that are regularly performed by the user, see* ***R10.4.23*** *and* ***R10.4.27****.*

Does the records system allow users to keep a list of favourite entities and access them easily through the GUI?

#### N101.5.10

*It is important that informational notifications and error messages to users be intelligible, informative and friendly. Each notification, message or dialogue should also indicate the actions available to users and, where possible, suggest what to do next.*

How does the GUI handle user notification, error conditions and failed functions; and are the messages to users helpful in suggesting a remedy or course of action?

#### N101.5.11

*The degree of help and assistance provided by the GUI under* ***R101.4.22*** *is extremely important. Wherever possible, help should be context sensitive. Help is not necessarily restricted to text based information, it may contain images and diagrams, embedded videos, on line tutorials, and other e-learning techniques.*

Further to **R101.4.22**, what types of online help are available through the GUI, and in different parts of the GUI, and how is this useful in assisting the users to perform functions and learn about the records system?

## 101.6 Glossary of Terms

| Term | Explanation and relationship to general concepts |
|---|---|
| **Configuration screen** | (*noun*) A "settings" screen or view intended primarily to help configure the MCRS, but not used in day to day operations. This might include options that users do not regularly encounter.<br>See also **screen**. |
| **Dialogue** | (*noun*) See **modal dialogue**. |
| **Display** | (*verb*) To show to a **user** of a **GUI**, especially by presenting information on a **screen**. A GUI may display various **entities** as text, images or icons, or a combination of any of these. **Relationships** between entities may be displayed using graphical elements such as lines and arrows, or they may be evident by their arrangement and proximity. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Error message** | (*noun*) A graphical indicator to a **user** that a particular **function** or **operation** has been unsuccessful, typically a function or operation that has been initiated by the user. There are many different ways to provide an error message to the user of a **GUI**, including a **modal dialogue**. |
| **Feedback** | (*noun*) See **visual feedback**. |
| **Graphical** | (*adjective*) Primarily visually, using the natural attributes and advantages of a **graphical user interface**. |
| **Graphical user interface** | (*noun*) An interface to an **information system**, usually abbreviated to "**GUI**", that presents **entities** and their **relationships** as images, icons and visual indicators as well as using text. A GUI may present information in different **windows** or **screens**. **Users** are able to manipulate the various objects that are displayed in the GUI using a variety of input methods including keyboards, mice and other pointing devices, as well as through touch and gestures. A GUI typically allows users to select an entity, or entities, and **perform functions** on it. |
| **GUI** | (*acronym*) A **graphical user interface**. |
| **Home screen** | (*noun*) The landing page or standard **view** of the MCRS that a non-specialised **user** sees by **default**. From the home screen a user might expect to immediately **browse**, **search** and **inspect** the **entities** in the **MCRS**. Depending on the purpose of the MCRS, the user may also expect to create records and access the content of records from this view.<br>See also **screen**. |
| **Modal dialogue** | (*noun*) A notification, usually in the form of a popup **window**, that must be acknowledged, or dismissed, or for which some action must be taken before the **user** can continue using the **GUI**. |
| **Navigate** | (*verb*) To change **screens** or views, **browse** the **relationships** between **entities**, **inspect** entities, and change the entity that is the focus of the **GUI** by direct manipulation of screen objects. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Personalisation** | (*concept*) Changing the character and experience of a standard **user** interface and customising it to the user's personal taste. Many **applications** provide options for rearranging a **screen** in a **GUI** to reflect personal preference, for example, by changing the size, colour, position and fonts used by screen objects. Where the user invests time and effort in personalisation it is important that the **MCRS** should "remember" these changes so that they can be reused in the next session. |
| **Screen** | (*noun*) A screen, a **view**, a **window** or an interface, depending on the nature of the **GUI**. Each GUI **application** is typically made up of several different screens with different functions. In the **functional** and **non-functional requirements** at least four different types of screen or interface are envisaged:<br>• **Configuration screen**,<br>• **Home screen**,<br>• **Specialised disposal interface**, and<br>• **Specialised export interface**.<br>It should be noted that these are design assumptions based on grouping the different types of functionality the GUI should support. They are not intended to place unnecessary restrictions on the GUI designer nor on the **MCRS supplier**, provided the application does in some way coherently facilitate these different aspects to managing and using a **records system**. |
| **Shortcut** | (*noun*) A **URI** hyperlink or other external reference to an **entity** or entities in an **MCRS** that can be shared with other **users**. For example, a user may send an email to another user, containing a shortcut to a **record**, instead of sending the content of the record as an attachment. When initiated a shortcut accesses the **MCRS** and allows the user to immediately discover the entity. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Specialised disposal interface** | (*noun*) An interface that facilitates the **disposal process**. Typically records managers need to engage with this **process** to do the following activities on a regular basis:<br>• Anticipate, monitor and manage the **records** coming due for **disposal**;<br>• Conduct **reviews** and enter **review decisions**;<br>• Ensure **transfers** are completed successfully and confirmed;<br>• Execute and confirm the **destruction** of record **content** where **manual** confirmation is required;<br>• Close **aggregations** where appropriate; and<br>• Respond to **alerts**.<br>There are many other peripheral activities that could also be included in this specialised activity, such as raising and lifting **disposal holds**, managing **disposal schedules**, and so on.<br>See also **screen**. |
| **Specialised export interface** | (*noun*) An interface that facilitates the **export process**. Typically **authorised users** will need to do the following:<br>• Select **records** and other **entities** for export, possibly based on previous exports or on whether entities have been updated since they were last exported;<br>• Provide an **export comment** explaining the export, and possibly a name for the export **datafile** and other details;<br>• Determine where to export the datafile to, this could be a network storage location, another **business system**, across an encrypted internet channel, or similar;<br>• Initiate and monitor the export;<br>• Following successful completion, possibly continue to manage the export, and any previous exports, in a secure location.<br>The exact tasks and the nature of the interface will depend on the individual **MCRS solution**.<br>See also **screen**. |
| **View** | (*noun*) See **screen**.<br>(*verb*) Visually **inspect**.<br>See also **inspect**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Visual feedback** | (*noun*) One of the attractions of a **GUI** is that **users** feel that they are actually manipulating real objects rather than simply two dimensional graphical representations of **logical entities**. As a result, it is important that the GUI is responsive and visual feedback is essential to the user experience. If users **perform** a **function** and nothing appears to happen, even if it is only for a few seconds, then the user experience can be seriously degraded. |
| **Window** | (*noun*) See **screen**. |

# 102. Application Programming Interface (API)

## 102.1 Module Information

| | |
|---|---|
| **Module Name** | **Application Programming Interface (API)** |
| **Module Version** | **1.0** |
| **Implements Module Identifier *(see M14.4.41)*** | 654633ec-8b17-4a3c-a483-436ee2bd506a |
| **Prerequisites** | MoReq2010® Core Services |
| **Co-requisites** | *none* |

## 102.2 Key Concepts

### 102.2.1 About application programming interfaces

An application programming interface or API is a published and standardised interface to an information system, such as a records system, that allows other software to interact with it. There are many different types of APIs for different programming languages, platforms and frameworks. An example of an API is a set of one or more web service definitions.

The provision of an API obviates the need for a records system to directly provide for any human user. The "user" of the MCRS is effectively the external system. A human user may interact with the MCRS indirectly through the external system or the interaction may be entirely automated. For example, a warehousing system may interface with a records system through an API so as to automatically store invoices, delivery dockets and purchase orders as they are raised or received. In this example, aggregations may be automatically created whenever new customer accounts are entered in the warehousing system and closed when customer accounts are deleted; classes and disposal schedules may be preconfigured and automatically applied; and there may be little or no direct interaction with the records system, except by or through the warehousing system.

This module contains requirements for an MCRS which implements an API.

### 102.2.2 Technical compliance

MoReq2010® defines an API set as consisting of a collection of "methods" which can be evoked by an external application. For an MCRS, these methods then implement the functionality described by the MoReq2010® specification. It is important that an MCRS has a complete API set if it is to be certified as compliant under this module.

Usually, each MoReq2010® function will correspond to a single method call in the API set, and it should not be necessary through the API to call more than one method to perform a single function defined by the specification.

Sufficient API methods must be included in the API set to provide a full suite of functionality for at least one service of MoReq2010®, or a bundle of services, see **R2.4.6**.

Some records systems may provide only partial API sets which include only some of the functionality specified by MoReq2010®. Where this occurs the requirements listed in this module may provide a useful guide to the expectations of MoReq2010® around the provision of an API, but a records system that is partially reliant for a given service, or bundle of services, on another type of interface in addition to its API, cannot be certified for compliance under the provisions of this module.

In order to evaluate the quality and completeness of an API set, the supplier must provide an accredited test centre with a manual to its API describing:

- How third-party applications discover and connect to the API; and
- The methods of the API set including their pre- and post-conditions.

In addition to the above, suppliers wishing to have their API sets tested for compliance must provide to the test centre a test harness or shell to enable testing to take place. Accredited test centres are not expected to write their own third-party applications so as to facilitate testing of an API set.

### 102.2.3 API set considerations

API sets are written for different purposes. Sometimes they are intended by suppliers to be general purpose interfaces and to encourage third party applications to be written supporting them. On other occasions they are intended to provide tight integration with a single third party product, perhaps one that does not have its own built in records management functionality. In the latter case the API is generally proprietary in nature and third party access is not encouraged by suppliers.

In either case, the presence of interaction via an API must not exempt an MCRS from its responsibility to ensure that:

- Records are given an appropriate business context through proper classification;
- That event histories are populated with appropriate audit information about which user performed a particular function;
- That access controls are set and maintained, and unauthorised users are prevented from accessing and manipulating the entities in the MCRS;
- That the disposal process described by **9. Disposal Schedule Service** is properly managed and that considered disposal decisions are made at appropriate times; and
- That records and other entities can be exported in full from the MCRS, as specified under **13. Export Service**.

In particular, API sets can be problematic in attributing actions in the records system to particular users. For this reason, although the external application is technically the user of the records system, care must be taken that the external system provides information via the API related to who the real user was who initiated the action that led to the API call being initiated.

## 102.4 Functional Requirements

#### R102.4.1

In accordance with core services requirement **R2.4.6**, the MCRS must implement an application programming interface (API) containing a set of methods that allow authorised

users access to the full functionality specified by the service or bundle of services, under **R2.4.1**.

*The term "API" describes any interface that is accessed by a client application or another business system, rather than by a human operator. A human operator may interface with the business system and indirectly manipulate entities in the MCRS, but there is no direct interaction with people.*

*To be certified as MoReq2010® compliant, the API must not implement only a part of the functionality specified by MoReq2010® for the service or bundle of services covered.*

#### R102.4.2

The MCRS must support multi-user, asynchronous calls to the methods in its API set.

*One user, application or thread accessing the MCRS via a method call must not block other users, applications or threads from simultaneously accessing the MCRS and making method calls.*

#### R102.4.3

The MCRS must ensure that each method call, under **R102.4.1**, is made by an authenticated user and that all functions and events are attributed to an individual user of the MCRS authorised to perform that function for the entity it is performed on.

*It is not sufficient for all functions performed in an MCRS to be attributed to a single user, that user being the external service that calls the API. This does not provide sufficient context for effective accountability when constructing or later interpreting event histories.*

#### R102.4.4

The MCRS must ensure that every method call, under **R102.4.1**, returns an error code that indicates the success or failure of the call.

*The calling process, application or system should not have to make a second method call to discover the outcome of a previous method call.*

*MoReq2010® does not specify what method calls or error codes should be used by an API.*

#### R102.4.5

The MCRS must provide a method that returns extended error information for the error code returned under **R102.4.4**.

*The extended error information should be the same, or equivalent, as that retrieved under **R2.4.8**. Note that this does not replace the requirement for an MCRS to maintain an error log under **R2.4.7**.*

#### R102.4.6

The MCRS must provide a method or methods that return, for any given entity, the current user's allowable operations in respect of that entity.

*MoReq2010® does not specify how such a method or methods must be implemented. Finding out about allowable operations for any given entity enables the current API user to discover what methods can be called for the entity without specifically having to call each operation.*

## 102.5 Non-functional Requirements

### N102.5.1

*The records system may support a number of different types of API and architectures. For example, web services, REST, programming languages (Java, C++, Python, etc.), Microsoft .NET framework, SOA, etc.*

What type(s) of API does the records system support?

### N102.5.2

*The records system may run on different platforms as listed under,* ***N12.6.1****. The records system API, however, may be made available on the same platforms, or it may be supported on separate platforms. For example, the records system may run on a Linux server but provide a Windows API set.*

*Examples of different API platforms include, Google App Engine, Amazon EC2, Microsoft Windows Azure, Microsoft Windows, Linux, Mac OS X, Google Android, Apple iOS, etc.*

What platforms does the records system provide API sets for?

### N102.5.3

*The API set may use a specific protocol, such as, HTML, SOAP, RPC, TCP/IP, XML, etc.*

What industry standard protocols does the API use?

### N102.5.4

*There is generally a learning curve associated with using an API set. Documentation must be provided and a development environment acquired. The skills necessary to develop to the API may be obtained from the supplier or from a third-party organisation providing API training, education and support. Alternatively, the organisation may choose to have API based applications developed for it by a third-party developer.*

Further to **N12.9.1** and **N12.9.2**, what skills, training and education are required to become an API developer for the records system, and where can these be obtained from?

### N102.5.5

*It may not necessarily be possible for any developer to licence and use the API, and freely develop new API based applications, not even the organisation that purchases the records system.*

Is it possible for any developer to develop for the records system API or must developers be curated in some way?

### N102.5.6

*The records system API set may have different versioning to the main product versioning, or it may even be a different product set, and may be managed semi-independently.*

Separate to **N12.14.1**, what versioning is used by the records system API set?

### N102.5.7

*Each version of the records system API may add new methods and deprecate older methods. This must be managed to ensure that API applications are no longer able to execute successfully.*

For each of the release types under **N102.5.6**, how are new method calls introduced into the records system API set, and old ones deprecated, and how are developers notified of these changes?

#### N102.5.8

*An important aspect of any API set is backwards compatibility.*

In relation to **N102.5.6** and **N102.5.7**, can code written to interface with one version of the API set run against previous versions, and under what compatibility constraints?

#### N102.5.9

*In addition to the support programme for users, API developers may be given additional assistance and separate support provision by the supplier.*

In addition to **N12.15.3**, **N12.15.4**, **N12.15.5**, **N12.15.6** and **N12.15.7**, how is the API set supported and is there a separate provision for the support programme?

#### N102.5.10

*Use of the API set may require additional warranties and may be subject to additional terms and conditions.*

What additional warranties under **N12.16.1**, additional service level agreements under **N12.16.2**, or additional terms and conditions under **N12.16.3**, apply to the records system API set?

## 102.6 Glossary of Terms

| Term | Explanation and relationship to general concepts |
|---|---|
| **Allowable operation** | (*noun*) A **method** that can be called in respect to a particular **entity** by the **user**. **MoReq2010®** does not require that there be a direct one to one mapping between a method in the **API** and a **function** that may be **performed** on an **entity** by an **authorised user**. For this reason, the term "allowable operation" is used as a broader and more inclusive description of the methods that may be invoked by a particular user, on a particular entity, at a particular moment in time. The reason for requiring that allowable operations be discoverable is to enable the user of an API to learn which methods may be called without having to call each one. |
| **API** | (*acronym*) **Application programming interface**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Application programming interface** | (*noun*) A software interface that may be used by an **application**. An **API** may be described as a "computer-computer interface" in contrast to a "human-computer interface". Each API has a number of **methods** which an application may **call**. The complete set of methods included in an API is sometimes referred to as an "API set". There are many different technologies and protocols that may be used to provide an API for an **MCRS**. **MoReq2010®** does not require any particular technology or protocol, provided the API includes full coverage of the functionality of the **service**, or bundle of services, to which it is applied. |
| **Asynchronous** | (*concept*) The principle that a **call** to an **API** will not block other calls from being received and processed simultaneously. Synchronous use of an API forces **applications** to wait for each call to complete before the next call can be made. Asynchronous use of an API means that applications cannot predict which of two overlapping calls will complete first. |
| **Call** | (*verb*) Invoke a **method** which sends input data to an **API** which is then processed by the **MCRS**. Each call to an API will typically return an **error code** and may also return other output data.<br>See also **method**. |
| **Error code** | (*noun*) An indication of the success or failure of a **call** to a **method**. In the event of failure or of only partial success, the error code may also be a value that indicates the reason the call was not successful. Following an unsuccessful call, and the return of an error code, an **API** must provide **extended error information** if requested. |
| **Method** | (*noun*) A routine or sub-routine in the **API** which can be discovered programmatically and subsequently invoked by a **call**. Each method called will, if successful, execute an action, input or retrieve data, or change some property or value within the **MCRS**.<br>See also **call**. |
| **Multi-user** | (*concept*) Support for multiple simultaneous **users**. An **MCRS** should not allow access by only one user at a time. |

# 200. CLASSIFICATION SERIES

## 201. Hierarchical Classification

### 201.1 Module Information

| | |
|---|---|
| **Module Name** | **Hierarchical Classification** |
| **Module Version** | **1.0** |
| **Implements Module Identifier *(see M14.4.41)*** | 5c772478-0a49-4391-a1d4-a5cd142a72d1 |
| **Prerequisites** | MoReq2010® Core Services |
| **Co-requisites** | *none* |

### 201.2 Key Concepts

#### 201.2.1 Hierarchical classification structure

Many organisations use hierarchical classification as a simple yet powerful way of developing a classification scheme for their business. Hierarchical structures are easy to navigate making the classes they contain easy to browse to.

In a hierarchical classification scheme, classes are organised into a multi-tiered tree structure as shown in **Figure 201a**.

A typical three tiered classification scheme might be organised by business function, and then within each function by business activity, and finally within each activity by transaction. Although three levels are commonly used for functional classification, MoReq2010® does not limit the maximum depth of a hierarchical classification scheme, and MoReq2010® also allows the number of levels to vary across a single hierarchical classification structure.

As **Figure 201a** shows, a hierarchical classification scheme has one or more top level classes representing the broadest, or most general, classifications, such as organisational functions. A class, such as a top level class, may be the parent of several lower level child classes. Lower level classes represent narrower and more specific classifications, such as the particular business activities and individual transactions carried out within each organisational function. A child class must have only one parent class, and every class except a top level class must have a parent. In this way, the hierarchical structure can fan out and grow to the required depth.

A parent class in a hierarchical classification scheme is not used for classifying aggregations or records. Only a class with no child classes beneath it, representing the most specific available classification may be associated with a record or aggregation. Once a class has been used in this way, for classifying aggregations or records, it can no longer be made a parent class by adding child classes to it.

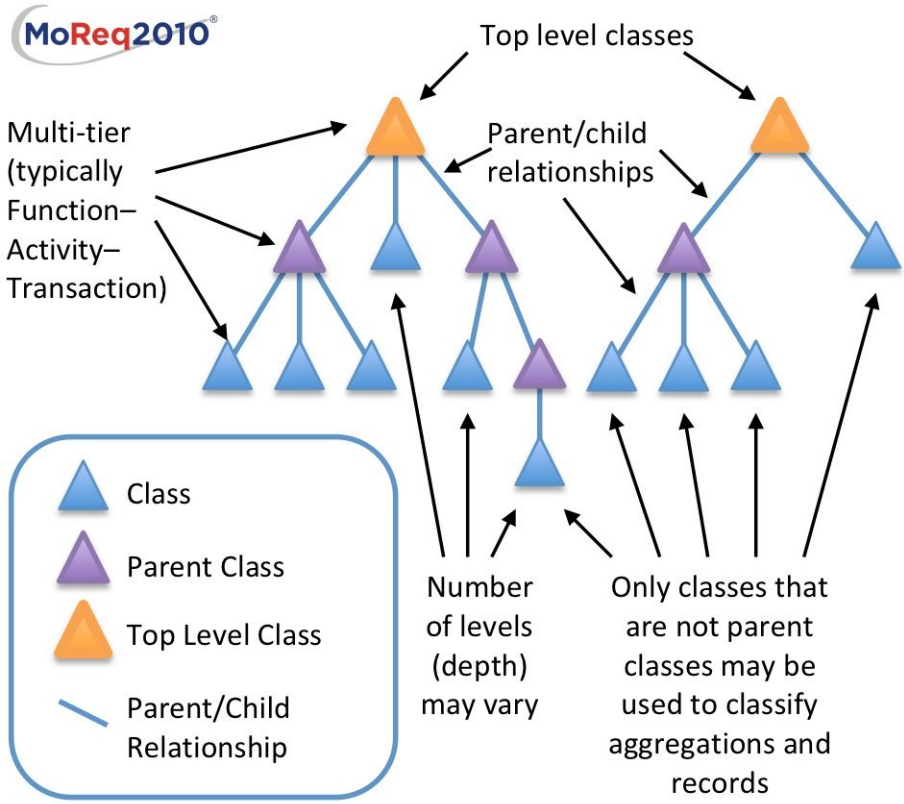


***Figure 201a** – The main features of hierarchical classification are top level classes, parent classes and child classes; the hierarchy can extend to any depth but most traditional hierarchical classification schemes adopt a three level hierarchy*

### 201.2.2 Inheritance

One of the primary advantages of hierarchical classification is that it allows child classes to inherit the attributes of their parents. MoReq2010® takes advantage of this, allowing the inheritance of a parent class's:

- Default disposal schedule, and
- Associated disposal holds.

If the MCRS implements the model role service, child classes also inherit:

- Access control lists.

If the MCRS implements the model metadata service, child classes inherit:

- Aggregation templates, and
- Record templates associated with the parent class (if any).

### 201.2.3 Traditional practice with hierarchical classification

According to previous specifications, such as MoReq2®, classification was always hierarchical and was always applied to root aggregations, only. This approach requires that every child aggregation and every record carry the same classification as their root aggregation. This traditional approach is shown in **Figure 201b**.

The traditional approach implies that, throughout a record service, aggregations of records will always be homogenous or, in other words, every record in a particular aggregation will always have been generated by the same business function, activity and transaction.

As the illustration shows, MoReq2010® continues to support this approach for those organisations that require it.

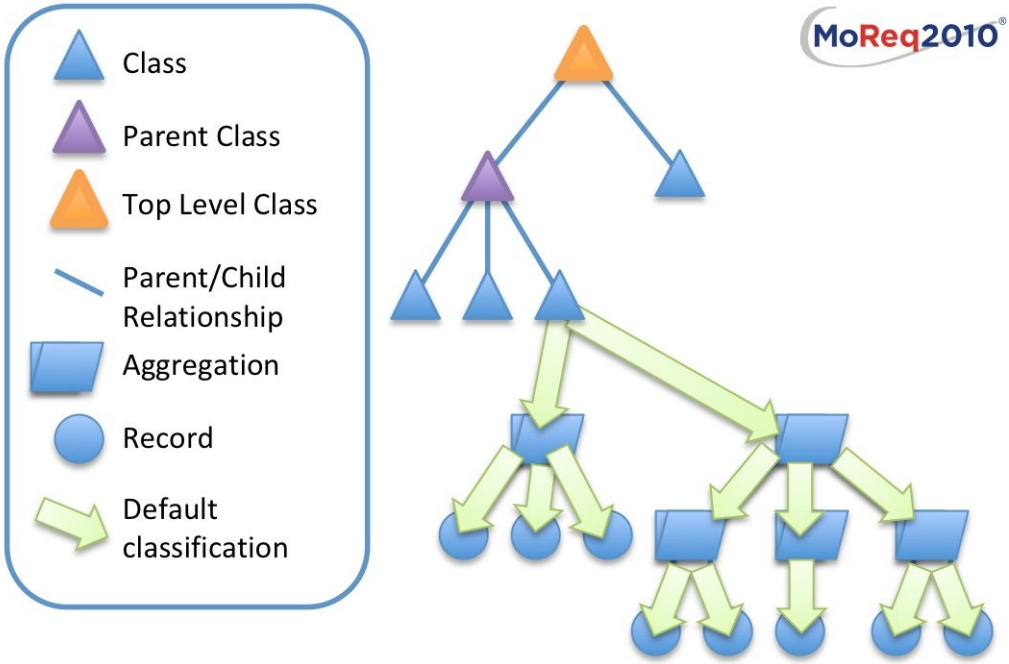


***Figure 201b** – Hierarchical classification when applied to a root aggregation is inherited as the default classification for all descendants of that aggregation; this mirrors the traditional approach of combined classification/aggregation hierarchies*

Where hierarchical classification is applied only at the level of the root aggregation, the whole structure combining classes, aggregations and records can be conceptualised as a single hierarchy. In this way, the structure described above by **Figure 201b** is logically identical to that illustrated by **Figure 1i** (see **1.4.5 Classification and aggregation** in the MoReq2010® core services), and shows how MoReq2010® can be used for the hierarchical conjoining of classification and aggregation, if desired.

### 201.2.4 Alternative uses of hierarchical classification

Hierarchical classification in MoReq2010® is not restricted, however, to representing traditional classification/aggregation approaches. In common with other types of classification, hierarchical classes can be used to override the default classification inherited from the parent aggregation of a child aggregation, or record.

This happens when a class is applied directly to a particular child aggregation or record, thereby breaking the default chain of inheritance of the root aggregation's classification. This is illustrated in **Figure 201c**.

By allowing the overriding of the default classification of a child aggregation or individual record, MoReq2010® supports heterogeneous aggregations containing records with different

classes, such as those generated as a result of separate business transactions, activities or even functions.

This allows, for example, the creation of project aggregations, where all the records relating to a particular project undertaken by an organisation are aggregated together, regardless of which transaction produces them.

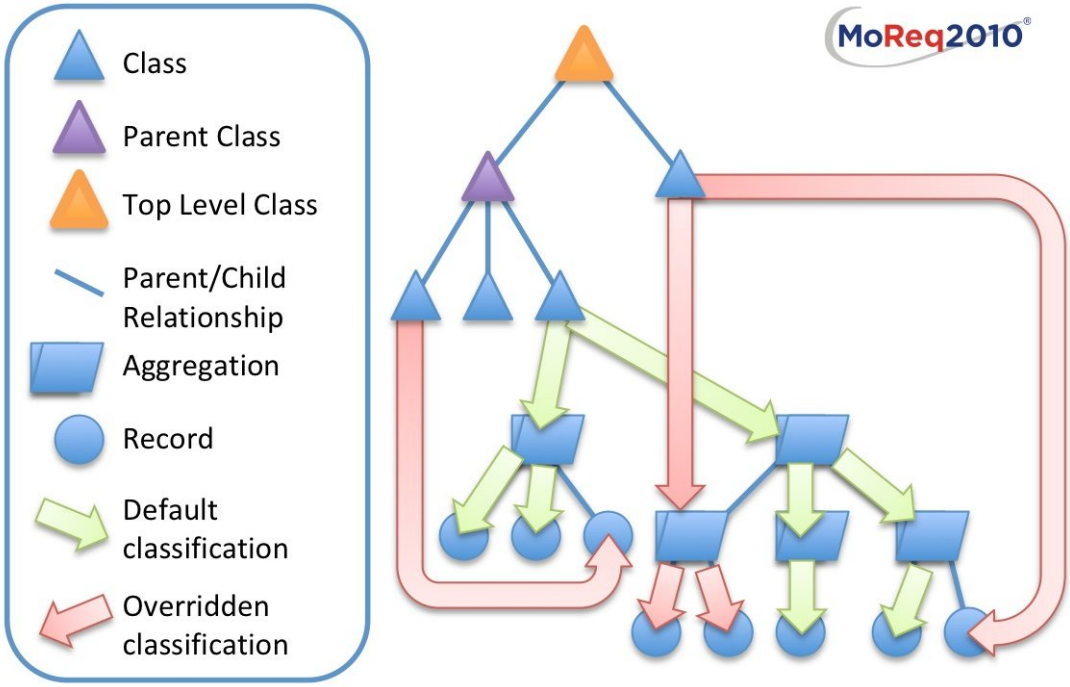


***Figure 201c** – Hierarchical classification can also be used in non-traditional ways; to override default classification at any level by applying it to child aggregations or directly to records*

Taken together, **Figures 201b** and **201c** demonstrate the utility of hierarchical classification and show how it can be adopted and used within both traditional and non-traditional record services.

### 201.2.5 Hierarchical class entities

Class entities and their attributes are defined in **5. Classification Service**. Classification services that use a hierarchical classification scheme implement a special type of class entity called a "hierarchical class". Hierarchical class entities are considered to be the same in all respects as other class entities, except that they have additional system metadata and are constrained by additional rules of behaviour that other classes are not subject to.

For example, hierarchical classes, other than top level classes, have parent classes. This means that a hierarchical class entity will include its parent class identifier as part of its system metadata. Classes belonging to other types of classification scheme do not have parent class identifiers. The additional rules of behaviour for hierarchical classes are defined by the functional requirements contained in this module.

### 201.2.6 Exporting hierarchical classification schemes

When classes are organised into a hierarchy each class represents a more specialised classification than its parent class. The parent class also provides the broader context for the child class. The full context is only provided by maintaining the relationships between the levels of hierarchy stemming from the top level class. These relationships must be preserved when hierarchical classes are exported.

The export rules described in **11. Export Service** must be extended when applied to hierarchical classes. Specifically hierarchical classes extend the definitions of which entities are significant to other entities under **11.2.9 Exporting significant entities** and which entities are included entities under **11.2.10 Exporting included entities**.

For hierarchical classes, the following entities are significant:

- The hierarchical class's disposal schedule;
- Any disposal holds associated with the hierarchical class; and
- The hierarchical class's parent class, and any ancestor classes, up to and including the top level class.

In addition, for hierarchical classes:

- The included entities of hierarchical classes are child classes.

These extended export rules mean that when a hierarchical class is exported in full, its descendant classes must be exported with it in full. Equally, the hierarchical class's ancestor classes, from its parent up to the top level class above it, must also be exported as placeholders to provide it with the necessary context. This is shown in **Figure 201d**.

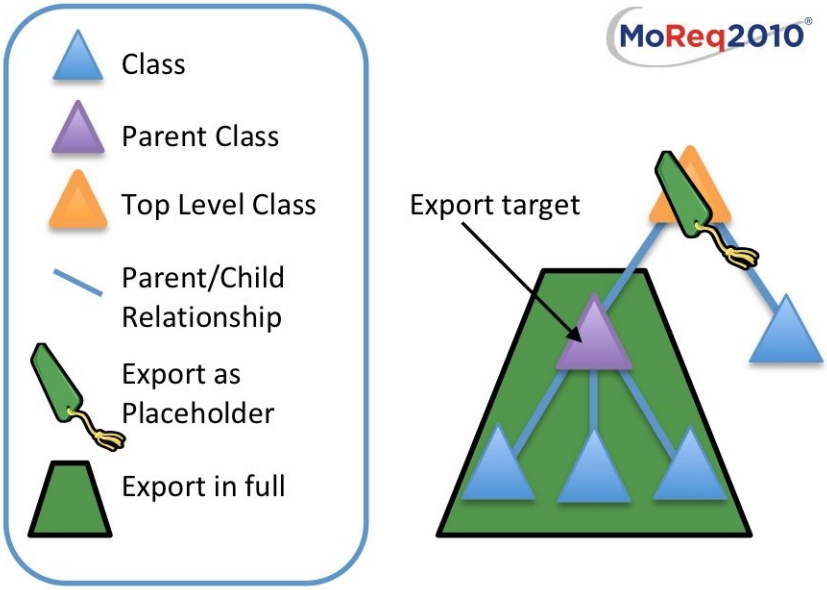


***Figure 201d** – The descendant classes of hierarchical classes that are exported must be exported in full, while the ancestor classes of hierarchical classes must be exported as placeholders*

These extended export rules apply also when an aggregation or record is exported in full and it is classified with a hierarchical class. In accordance with **11. Export Service**, the

hierarchical class is a significant entity and will be exported along with the aggregation or record as a placeholder. However, exporting a single hierarchical class as a placeholder is not sufficient to provide the full context for the classification of the aggregation or record. It is necessary to export all levels of the hierarchy that support that hierarchical class, from its parent up to the top level class.

**Figure 201e** gives an example of this. In this example a child aggregation is exported from the MCRS and its parent aggregation is exported as a placeholder. As required by **11. Export Service**, the aggregation's class must also be exported as a placeholder, regardless of whether it is inherited or directly applied to the aggregation being exported.

However, as **Figure 201e** shows, the class used to classify the aggregation is a hierarchical class with its own parent class under a top level class. These are all significant classes. Instead of simply exporting the aggregation's class, it is necessary in this example to export all of the levels of classification, including the class, its parent class and the top level class, as placeholders, to ensure the full context of the aggregation is provided.

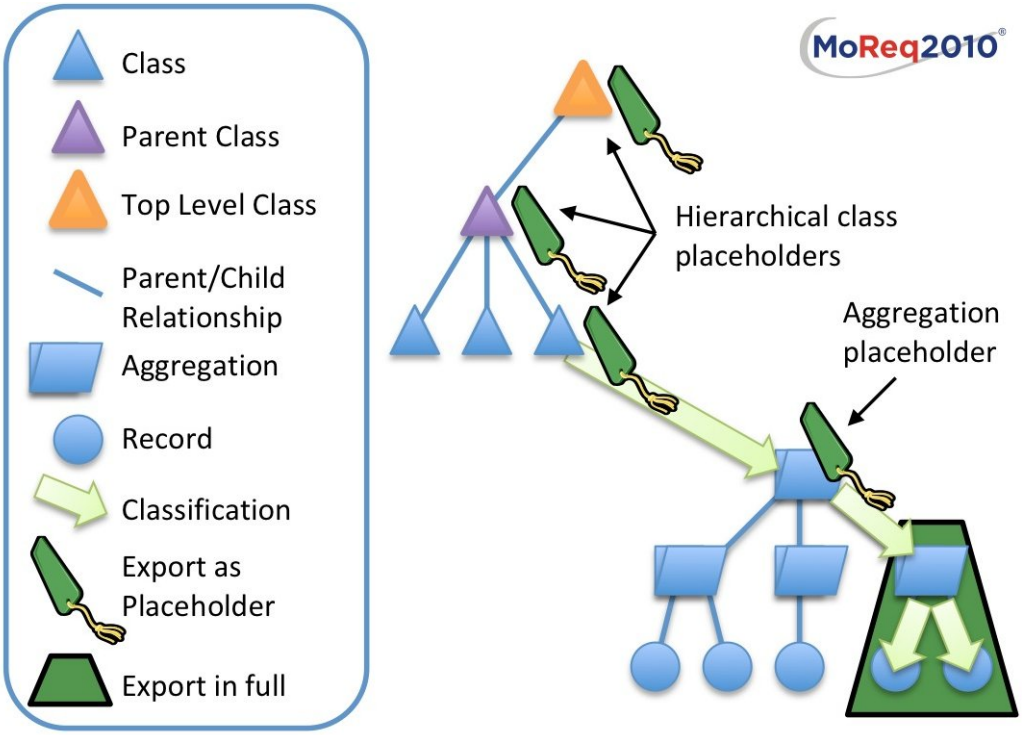


***Figure 201e** – Placeholders must be exported for all hierarchical classes up to the top level class that are ancestors of the class used to classify aggregations and records that are being exported in full*

## 201.4 Functional Requirements

### R201.4.1

A classification service that implements hierarchical classes must not implement any of the other MoReq2010®, **200. Classification Series** modules.

*Note that an MCRS may utilise more than one classification service, but each separate classification service must contain entities belonging to only one classification entity sub-type.*

#### R201.4.2

The MCRS must ensure that in addition to the metadata listed under **R5.4.2**, hierarchical classes (**E201.7.1**) are also created with the following system metadata:

- Hierarchical Parent Class Identifier (**M201.7.2**).

*Only top level hierarchical classes will not have a Parent Class Identifier. Child classes may inherit their disposal schedule and templates from their parent classes.*

*Hierarchical classification schemes should support any number of levels of hierarchy and any number of child classes belonging to a single parent class.*

*Function reference:* ***F201.7.4***

#### R201.4.3

The MCRS must ensure that all hierarchical classes created under **R5.4.2** and **R201.4.2**, either:

- Are created with no parent class as a top level class, provided they are associated with an active default disposal schedule; or
- Are created within an active parent class that has never been used for classification.

*Hierarchical classes that are used for classifying aggregations and records may not become parent classes. The First Used Timestamp determines whether a class has been used for classification.*

*A hierarchical class created within a parent class may inherit its parent's default disposal schedule identifier and templates. All top level classes must have a Default Disposal Schedule Identifier.*

*Function reference:* ***F201.7.4***

#### R201.4.4

Further to **R5.4.4**, the MCRS must allow a hierarchical class, except a top level class, to inherit its default disposal schedule from its parent class, instead of requiring an authorised user to provide it.

*The metadata element, Default Disposal Schedule Identifier is not required for hierarchical classes, apart from top level classes with no parent class. See also* ***R201.4.3*** *and* ***R201.4.5****.*

#### R201.4.5

For any active hierarchical class, including a top level class, the MCRS must allow an authorised user to move it, either:

- To an active parent class that has never been used for classification so that it retains its original disposal schedule;
- To an active parent class that has never been used for classification so that it inherits the disposal schedule of its new parent; or
- So that it becomes a top level class while retaining its previous disposal schedule and templates.

*To retain the hierarchical class's previous disposal schedule, the MCRS must ensure that a Default Disposal Schedule Identifier is added to the metadata of the class during the move operation.*

*To inherit the disposal schedule of its new parent, the MCRS must remove any Default Disposal Schedule Identifier from the class during the move operation.*

*Note that changing the disposal schedule of a class will have a cascading effect on all active records classified by that class, as described in* ***R5.4.4****.*

*Function references:* ***F201.7.3****,* ***F201.7.5***

#### R201.4.6

The MCRS must only allow a hierarchical class with no child classes to be used for classifying aggregations or records.

*A parent class, which has one or more child classes, may not be applied to an aggregation or record.*

*Whenever a hierarchical class is first applied to any aggregation or record its First Used Timestamp, see* ***R5.4.2****, must be set by the MCRS and it may no longer become a parent class under* ***R201.4.3*** *or* ***R201.4.5****.*

*Function references:* ***F14.5.20****,* ***F14.5.138***

#### R201.4.7

The MCRS must not allow a hierarchical class to be deleted under **R5.4.5** if it is a parent class with one or more child classes.

*The children of the hierarchical class must first be deleted or moved before the hierarchical class can be deleted.*

*Function reference:* ***F14.5.25***

#### R201.4.8

The MCRS must not allow a hierarchical class to be destroyed under **R5.4.6** if it has one or more active child classes.

*An active hierarchical class cannot be the child or descendant of a residual class. The child classes must either be destroyed first, or concurrently, with the hierarchical class.*

*Function reference:* ***F14.5.28***

#### R201.4.9

Subject to **R2.4.22**, and further to **R5.4.7**, the MCRS must allow an authorised user to browse and inspect hierarchical classes in at least the following ways:

- Browse from a parent class to its child classes and inspect their metadata, and
- Browse from a child class to its parent class and inspect its metadata.

*The terms "browse" and "inspect" are defined in* ***13. Glossary of Terms****.*

*Function reference:* ***F14.5.30***

#### R201.4.10

Where the MCRS implements the model role service (see **4. Model Role Service**) then, further to **R4.5.11** and subject to **R4.5.10**, the MCRS must allow a child hierarchical class to inherit active roles that have been granted to active users and groups from its parent class.

*The specific rules of inheritance used by the model role service, listed in the rationale to **R4.5.11**, should be extended as follows:*

- *Child hierarchical classes inherit from their parent classes.*

*Where the MCRS does not implement the model role service it must demonstrate similar or equivalent functionality under **4.2.4 How to meet the alternative (type B) requirements**.*

**R201.4.11**

Where the MCRS implements the model metadata service (see **7. Model Metadata Service**), and a template has been associated with a hierarchical class using the Template Class Identifier, under **R7.5.14** or **R7.5.15**, then the template must be applied automatically, under **R7.5.18**, to any aggregation or record created with that hierarchical class or with any of its child classes or descendants.

*An aggregation or record template may be associated with a top level class, or a parent class, and will then be automatically applied to all aggregations and records classified by any descendant of that class.*

*Where the MCRS does not implement the model metadata service it must demonstrate similar or equivalent functionality under **7.2.4 How to meet the alternative (type B) requirements**.*

*Function references: **F14.5.5**, **F14.5.121***

**R201.4.12**

The MCRS must allow an authorised user to group active records under **R8.4.16** and perform disposal related functions under **R8.4.17**, **R8.4.18**, **R8.4.19** and **R8.4.20** for any hierarchical class.

*Where the hierarchical class is a parent class then the function should apply to all of the records that are classified by any of the descendants of the hierarchical class.*

*Function references: **F14.5.41**, **F14.5.116**, **F14.5.117**, **F14.5.118**, **F14.5.119**, **F14.5.120**, **F14.5.124**, **F14.5.131**, **F14.5.177***

**R201.4.13**

When an active disposal hold is associated with a hierarchical class, under **R9.4.3**, the MCRS must, in addition to the other clauses of **R9.4.4**, prevent the destruction of any record that:

- Has been classified with a hierarchical class that is a child or descendant of a hierarchical class associated with the disposal schedule.

*The association of a disposal hold with a hierarchical class that is a top level class, or a parent class, should have the same effect as associating the disposal hold individually with each child class or descendant of that class.*

**R201.4.14**

The MCRS must allow an authorised user to search for and find aggregations and/or records that are classified by any descendant of a nominated hierarchical class.

*This requirement extends* ***R6.5.18*** *to allow a search by a nominated class to find an aggregation or record where the nominated class is a hierarchical class that is not the entity's class but is the ancestor of the entity's class.*

*Function reference:* ***F14.5.195***

#### R201.4.15

When hierarchical classes are exported in full under **R11.4.3**, then the MCRS must also export their child and descendant classes in full, as included entities.

*See* ***11.2.10 Exporting included entities*** *and* ***201.2.6 Exporting hierarchical classification schemes*** *for an explanation of included entities in export and their relationship to hierarchical classification.*

*Function reference:* ***F14.5.185***

#### R201.4.16

When hierarchical classes are exported in full, or as placeholders, under **R11.4.3**, then the MCRS must also export placeholders for their parent and ancestor classes, as significant entities.

*See* ***11.2.9 Exporting significant entities*** *and* ***201.2.6 Exporting hierarchical classification schemes*** *for an explanation of significant entities in export and their relationship to hierarchical classification.*

*Function reference:* ***F14.5.185***

## 201.5 Non-functional Requirements

#### N201.5.1

*Hierarchical classification schemes may impose internal limitations on the number of hierarchical classes or levels of hierarchy supported.*

What are the technical limits in the classification service to each of the following:

- The number of hierarchical classes that can be managed?
- The number of top level classes that can be added to the classification service?
- The number of child classes that can be added to parent class?
- The depth or number of levels of classes under a top level class?

## 201.6 Glossary of Terms

| Term | Explanation and relationship to general concepts |
|---|---|
| **Child class** | (*noun*) Any **hierarchical class** that is not a **top level class**.<br>See also **child**, **hierarchical class**, **parent**, **parent/child relationship**, **parent class** and **top level class**. |

| Term | Explanation and relationship to general concepts |
|---|---|
| **Hierarchical class** | (*entity*) A **sub-type** of **class** that allows a **classification scheme** to be arranged in a **hierarchical** structure. Each class, except a **top level class**, has a **parent class** and may have **child classes**.<br>See also **class** and **hierarchical**. |
| **Multi-tiered** | (*concept*) Especially in relation to a hierarchical **classification scheme**, a **hierarchical** structure where each tier, or **level**, under a **top level class** represents a particular concept. For example, in a typical three tiered classification scheme, the top level classes might represent **business functions**, the **child classes** of the top level classes would then represent **business activities**, while the grandchild classes of the top level classes represent **business transactions**. This arrangement of classes is also sometimes known as a "functional classification scheme".<br>See also **level**. |
| **Parent class** | (*noun*) Any **hierarchical class** that contains **child classes**. In **MoReq2010®** a parent class cannot be used to **classify aggregations** or **records**.<br>See also **child**, **child class**, **hierarchical class**, **parent** and **parent/child relationship**. |
| **Top level class** | (*noun*) A **hierarchical class** that is not a **child** of another hierarchical class. Each top level class is created directly under the **classification service**.<br>See also **hierarchical class**. |

## 201.7 Information Model



### E201.7.1 Hierarchical Class

| | |
|---|---|
| **System Identifier** | 8e98092d-e20b-48ea-b3d6-ca75375590ee |
| **Title** | **Hierarchical Class** |

| | |
|---|---|
| **Description** | Definition of a hierarchical class, as a sub-type of class used in hierarchically structured classification schemes |
| **Sub-type of** | Class (**E14.2.2**) |
| **Service** | Classification Service |
| **Additional system metadata** | *As for Class (**E14.2.2**) plus the following additional system metadata:*<br>• Hierarchical Parent Class Identifier (**M201.7.2**)<br>The following metadata element becomes optional, except for top level classes, under **R201.4.4**:<br>• Default Disposal Schedule Identifier (**M14.4.11**) |
| **Additional functions** | *As for Class (**E14.2.2**) plus the following additional functions:*<br>• Hierarchical Class – Add Class (**F201.7.3**)<br>• Hierarchical Class – Remove Class (**F201.7.5**)<br>*The following function replaces Class – Create (**F14.5.24**):*<br>• Hierarchical Class – Create (**F201.7.4**) |
| ***Usage notes*** | • *The new function Hierarchical Class – Create (**F201.7.4**) replaces the function Class – Create (**F14.5.24**) for hierarchical classes*<br>• *The Default Disposal Schedule Identifier (**M14.4.11**) is inherited from the parent class unless overridden and mandatory for top level classes*<br>• *A classification service for hierarchical classes may not simultaneously support other types of classes, under **R201.4.1*** |

### M201.7.2 Hierarchical Parent Class Identifier

| | |
|---|---|
| **System Identifier** | caa1ff78-8cf9-40ac-9e2f-6ca75b87637e |
| **Title** | **Hierarchical Parent Class Identifier** |
| **Description** | The parent class for a hierarchical class |
| **Entity Type** | Hierarchical Class (**E201.7.1**) |
| **Min Occurs** | 0 (for top level classes)<br>1 (for child classes) |
| **Max Occurs** | 0 (for top level classes)<br>1 (for child classes) |

| | |
|---|---|
| **Modifiable?** | Yes (by moving the class) |
| **Entity Reference?** | Yes |
| **Refers To Type** | Hierarchical Class (**E201.7.1**) |
| **Datatype** | UUID |

### F201.7.3 Hierarchical Class - Add Class

| | |
|---|---|
| **System Identifier** | 9ccb3ff0-b225-42bc-9edc-e05ed74547a5 |
| **Title** | **Hierarchical Class – Add Class** |
| **Description** | Add a child class to the hierarchical class by moving it from a top level class or its previous parent |
| **Entity Type** | Hierarchical Class (**E201.7.1**) |
| **Entity metadata** | *The following metadata element belonging to the participating child class will be modified:*<br>• Hierarchical Parent Class Identifier (**M201.7.2**) |
| ***From functional requirement(s)*** | **R201.4.5** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Previous Parent Identifier (**M14.4.76**)<br>• Participating New Parent Identifier (**M14.4.75**)<br>• Participating Class Identifier (**M14.4.65**)<br>• Event Comment (**M14.4.25**) |
| ***Usage notes*** | • *This function is always performed in conjunction with **F201.7.5 Hierarchical Class – Remove Class***<br>• *Before a user can move a class the user must have the authority to perform this function on its new parent, as well as the authority to remove the class from its previous parent or from being a top level class*<br>• *To move a hierarchical class so that it becomes a top level class, the user must have the authority to perform this function for the classification service as a whole*<br>• *This function only applies to adding child classes by moving them from elsewhere, it does not apply to adding child classes by creating them in the hierarchical class (see **F201.4.4 Hierarchical Class - Create**)* |

### F201.7.4 Hierarchical Class - Create

| | |
|---|---|
| **System Identifier** | a148a5ee-58ab-4b7b-a925-bb6f426b0d7a |
| **Title** | **Hierarchical Class – Create** |
| **Description** | Create a hierarchical class |
| **Entity Type** | Hierarchical Class (**E201.7.1**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Scope Notes (**M14.4.97**)<br>• Default Disposal Schedule Identifier (**M14.4.11**)<br>• Hierarchical Parent Class Identifier (**M201.7.2**)<br>• *Contextual metadata elements*<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R201.4.2**, **R201.4.3** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.65**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *The hierarchical class may be created as a top level class, in which case the Hierarchical Parent Class Identifier is omitted and the Default Disposal Schedule Identifier becomes mandatory*<br>• *If the hierarchical class is created as a child class then the Hierarchical Parent Class Identifier is mandatory and the Default Disposal Schedule Identifier becomes optional*<br>• *The hierarchical class may be created with contextual metadata elements as well as the system metadata elements listed*<br>• *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event* |

| | |
|---|---|
| | • *Where the class's inherited access controls are modified on creation then separate **F14.5.33 Class – Modify ACL** events must be generated for each change made to the access control list* |

### F201.7.5 Hierarchical Class - Remove Class

| | |
|---|---|
| **System Identifier** | 2b0c7c88-ee72-4e83-87dd-ad9d98567789 |
| **Title** | **Hierarchical Class – Remove Class** |
| **Description** | Remove a child class from the hierarchical class by moving it to a top level class or to another parent class |
| **Entity Type** | Hierarchical Class (**E201.7.1**) |
| **Entity metadata** | *See related function **F201.7.3 Hierarchical Class – Add Class*** |
| ***From functional requirement(s)*** | **R201.4.5** |
| ***Purpose*** | Access control only |
| ***Usage notes*** | • *Before a user can move a child class out of the hierarchical class, the user must have the authority to perform this function, as well as the authority to add a class to the new parent or to make it a top level class*<br>• *To move a top level class into a parent class, the user must have the authority to perform this function for the classification service as a whole*<br>• *This function is always performed in conjunction with **F201.7.3 Hierarchical Class – Add Class**, which describes the metadata that is modified and the event that is generated*<br>• *This function does not separately modify metadata or generate an event* |

# 300. COMPONENT SERIES

# 301. Electronic Components

## 301.1 Module Information

| | |
|---|---|
| **Module Name** | **Electronic Components** |
| **Module Version** | **1.0** |
| **Implements Module Identifier *(see M14.4.41)*** | 13b6976c-2409-48ff-a576-a6f6662c5044 |
| **Prerequisites** | MoReq2010® Core Services |
| **Co-requisites** | *none* |

## 301.2 Key Concepts

### 301.2.1 Features of an electronic component

As described in **6.2.9 Record components**, records can have physical or electronic components. The terms "physical" and "electronic" in each case refer to the content of the component.

The term "electronic component" refers to a component of a record whose content:

- Is entirely digital and contained in a series of binary digits or "bits";
- Is not tied to any physical object or medium, although it may require particular hardware and/or specialised software in order to be accessed or viewed;
- May be copied any number of times, where each copy will be indistinguishable from the original and other copies;
- May be transmitted electronically, including in a series of datagrams or packages that can be reassembled into an identical copy of the original content; and
- May be stored in a datafile on an electronic medium, such as a magnetic drive.

A distinction should be drawn between the component, which is an entity in the MCRS, and the content referenced by the component, which can be held in the records system's own data store (if it has an internal data store), but may equally be held externally in the system of origin or in a third-party repository. This distinction between component and content is shown in **Figure 301a**.

Some records may have electronic content that is discontinuous and naturally divided into discrete datafiles or chunks. In an MCRS, these records will be made up of several electronic components where each separate component will be associated with a single contiguous item of electronic content. The content of each component, belonging to the same record, may be stored in different data stores, as **Figure 301a** shows.

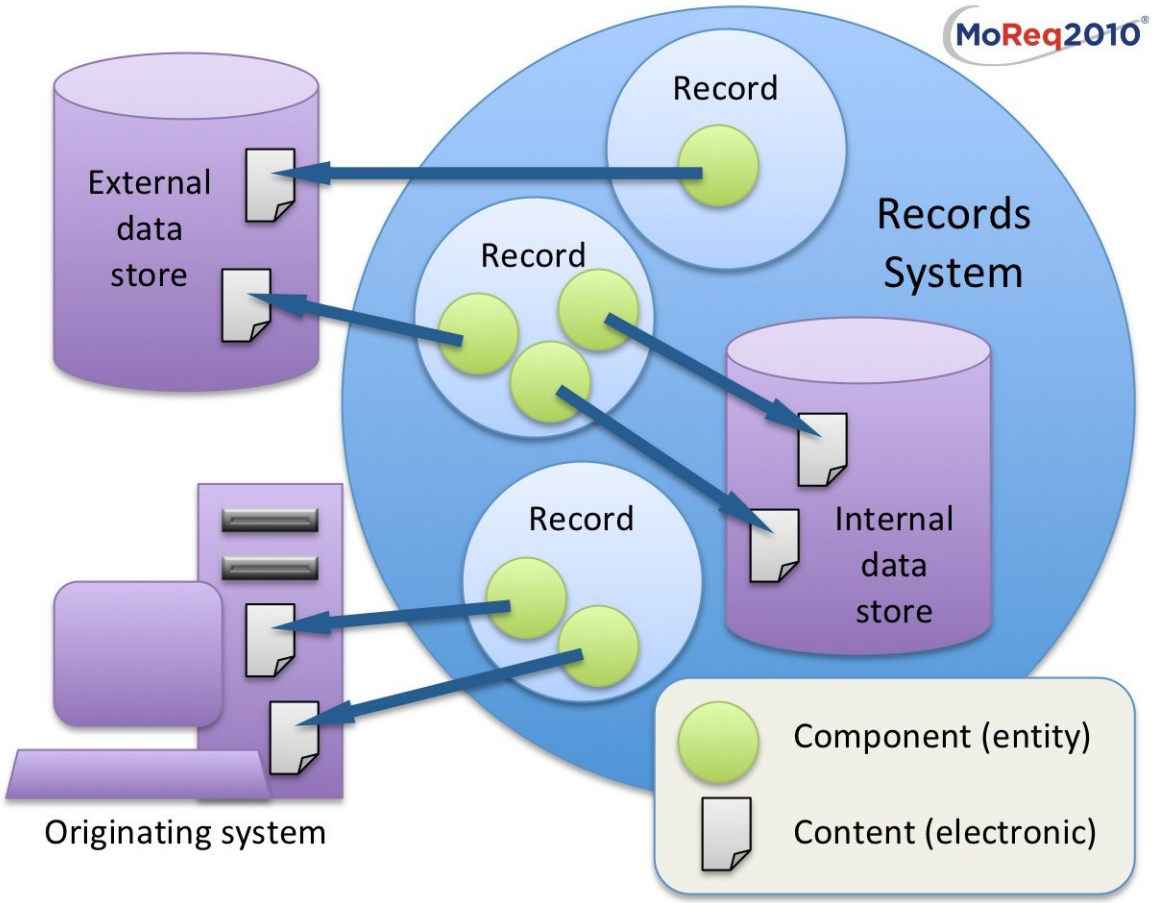


***Figure 301a** – While records and components are entities in the MCRS, the content of electronic components may be stored, by design, in any of a number of different data stores in different locations*

### 301.2.2 Revisiting the principles of managed components

The principles common to all forms of component, including electronic components, are detailed in **6.2.10 Principles of managing component content**. These principles include:

- **Discreteness** – the electronic component must have its own separately identifiable content;
- **Completeness** – the electronic component, in combination with the other components of the record, must comprise a whole atomic record which is not dependent for its evidential worth on external resources;
- **Immutability** – once created, the content of the electronic component must not change over time, or be able to be erased, until it is destroyed by the MCRS; and
- **Destructibility** – the content of the electronic component must be able to be deleted, either automatically or with confirmation, in response to its destruction in the MCRS.

In addition to these general principles, all electronic components must also be transportable so that they may be transferred from their originating system to another receiving system for interoperability and long term preservation. This principle of **transportability** is discussed below.

### 301.2.3 Transporting electronic content

All electronic content is data that originates within a business application or system as a whole or partial representation of its internal state at a particular moment in time. The information represented by an electronic component must be transportable, as content, to allow another separate application that implements the same processing rules as the system of origin, to read it, interpret it and process it correctly.

Information that is locked into its system of origin cannot be managed as a record as it cannot be independently captured, preserved, duplicated, secured, verified or accessed and it cannot survive the physical longevity or technological obsolescence of its host.

Provided electronic content meets the criterion of transportability, shown in **Figure 301b**, as well as the principles of discreteness, completeness, immutability and destructibility described above, then it is suitable as content for an electronic component of a record managed by an MCRS.

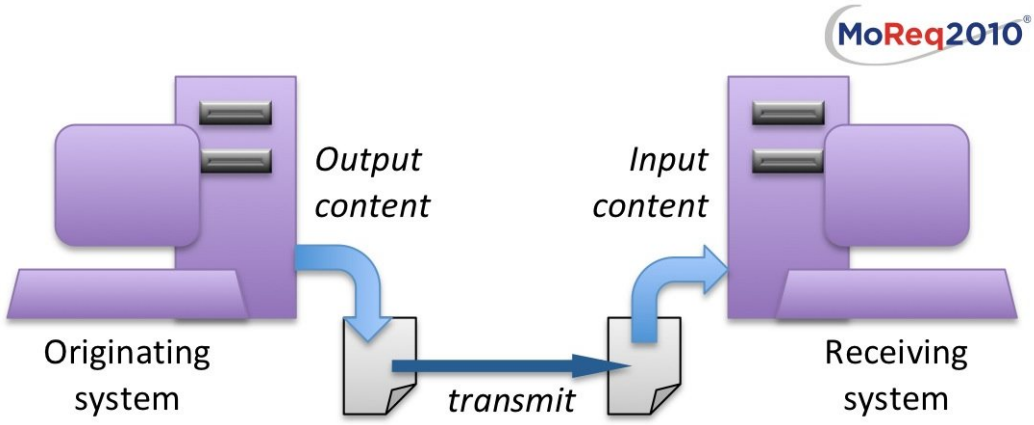


***Figure 301b** – Electronic content must be transportable; the originating system must be able to output it in a format that allows it to be transmitted to a receiving system that can input and understand it (neither system need necessarily be an MCRS)*

Although the MCRS must store the media type of the transportable form of the electronic content in the metadata of the component, MoReq2010® does not require that it be in any particular format. The transportable form may be a commonly used document format, like Rich Text Format, or a standardised format like PDF/A, or a mark-up language such as XML. These well known media are generally preferred for the long term preservation of records with electronic components and may therefore be part of the operational procedures of the organisation. However, MoReq2010® also allows electronic components to have content that is in a proprietary document format, or even compressed and/or encrypted.

It is also unnecessary for the originating system to keep the electronic content of a component internally in its transportable format, provided the transportable form of the content can be generated by the originating system on demand, and, if requested multiple times, is identical each time it is regenerated (to meet the requirement of immutability).

### 301.2.4 The function of the records system

An MCRS may fulfil many different functions in relation to the applications and business systems that produce electronic content.

- The MCRS may have its own data store and may only allow records to be created with electronic components where it receives the content of those components in transportable form, and stores them internally;
- The MCRS may manage records in their originating system without requiring them to be generated, except when they are accessed or eventually exported to another system;
- The MCRS may manage content generated by the originating application in an external third party repository that is also accessed by other systems; or
- The system of origin may itself be a dedicated MCRS, with built in records management capabilities that allow it to manage its own records, generating the transportable form as required and for export.

One notable outcome of this flexibility within MoReq2010® is that for an MCRS to manage records it is not necessary for it to directly handle the content of the components it manages. For example, apart from maintaining an external reference to the component content in the originating system, the MCRS may never be directly involved in the process depicted in **Figure 301b**.

The MCRS must be able to export its own entities as data, but when it exports electronic components then their content may be exported either as data that is included in the export XML, under **R11.4.8**, or the content may be exported as an external URI reference that can be subsequently used to separately retrieve the content from its data store. For example, an MCRS may manage as records, web pages on a web server (or in a content management system), for each separate component (or web resource) storing only a URI that then gives the actual location of the datafile.

Note that URI syntax is flexible enough to equally indicate datafiles in an operating system as it can web pages on a web server.

Most importantly, a records system does not need to be a general purpose system to be compliant with MoReq2010®. It does not have to be integrated with many different business systems or to manage many different content formats. It may be a small records system with a dedicated purpose and still be fully compliant with MoReq2010®, provided it can successfully manage at least one type of record containing at least one form of electronic content.

### 301.2.5 Examples of different approaches to managing electronic components

The following are all examples of compliant records systems that manage electronic components differently. They are indicative of the diverse possibilities allowed by MoReq2010®.

None of the following examples support all possible combinations and permutations of electronic component, yet they are all potentially compliant with the specification:

- An MCRS captures email messages from an email application as records into its own internal data store – records may be made up of one or several components representing the original email and its attachments (if any);
- An MCRS manages office documents as "in place" records, their content is held in an external document management system – records are made up of only one component (the office document) with content in one of a handful of well known document formats;
- An MCRS manages web pages on a web server as records, where each record can have a varying number of electronic components, including HTML, CSS, JavaScript and images in different web formats – many of the CSS and JavaScript datafiles are common resources shared by different records, and the MCRS uses a system of pointers to enable the same content to be included in the logically discrete electronic components of different records;
- An MCRS manages call centre records – these are made up of two components, one an audio recording of a call that is captured into the records system's internal data store, the other is a snapshot of a customer contact that is held externally in a dedicated customer relationship management system; and
- An MCRS manages financial transactions as records held externally in a dedicated finance system – records are always made up of three components that each represent a row in (respectively) the invoice database table, the purchase order database table and the supplier database table.

The final scenario outlined in the list above is of a database record drawn from rows held in different relational tables, a possible example of this is depicted in **Figure 301c**, below.

| INVOICES TABLE | | | | |
|---|---|---|---|---|
| **Invoice No.** | **Invoice Date** | **Supplier ID** | **Order ID** | **Amount** |
| **05266** | 2011 JAN 02 | S003452 | P0017035 | €298,16 |
| **R15523** | 2011 FEB 15 | S001636 | P0024773 | €99,99 |
| **09356** | 2011 FEB 21 | S003452 | P0019752 | €6363,00 |
| **KSU-0234-L** | 2011 MAR 16 | S002850 | P0009456 | €1900,02 |

| PURCHASE ORDERS TABLE | | | | | |
|---|---|---|---|---|---|
| **Order ID** | **Order Date** | **Supplier ID** | **Item** | **Quantity** | **Estimate** |
| **P0009456** | 2010 OCT 03 | S002850 | Sprockets | 1 only | €1900,00 |
| **P0017035** | 2010 DEC 26 | S003452 | Widgets | 3 off | €300,00 |
| **P0019752** | 2011 JAN 16 | S003452 | Grommets | 8 off | €6000,00 |
| **P0024773** | 2011 FEB 14 | S001636 | Gadgets | 1 only | €100,00 |

| SUPPLIERS TABLE | | | | |
|---|---|---|---|---|
| **Supplier ID** | **Supplier Name** | **Contact** | **City** | **Country** |
| **S001636** | Parafusos e Porcas | +351.210497985 | Lisbon | Portugal |
| **S002850** | Mécanisme Tech. | +32.23843331 | Brussels | Belgium |
| **S003452** | Keerulisi Masinaid | +372.57903924 | Tallinn | Estonia |

***Figure 301c** – An example of a record (in red) of the invoice numbered "09356" which is stored in a relational database; individual rows from three different tables collectively make up the complete record*

While an electronic component remains stored in its system of origin it need not be in a transportable format. In this example the internal state of the originating system is as a relational database.

However, the originating system must be capable of outputting the content of the record as one or several components that can be transmitted to and stored by another system (*principle of transportability*), and represent the complete content of the record (*principle of completeness*). Also, each time the content is requested it must able to be output in the same way.

This means that once a record has been created in the MCRS, the system of origin must keep the rows of the database locked, or keep a copy of the rows, to ensure that the data they contain is not overwritten or updated (*principle of immutability*). For example, in **Figure 301c** the entry in the suppliers table for supplier "S003452" cannot be later updated, to change a detail such the supplier's name or contact number, after the record has been created.

It also means that the system of origin must provide a method for outputting the data of an individual record that does not require the export of the whole relational database (*principle of discreteness*). Transportable formats for records from database tables may include:

- XML datafiles,
- Comma or tab separated values,
- Spreadsheet formats,

- Proprietary portable database formats, or even
- Unstructured document formats, such as PDF or HTML that might be generated, for example, when printing or viewing an invoice from a financial transaction system.

Regardless of the transportable format supported, the complete data relating to the record, stored in the original database table rows, and their relationship to one another must be included in the electronic content.

Finally, when the record is destroyed by the MCRS in in accordance with the disposal schedule applied to the record, then the appropriate rows must be able to be blanked or deleted from their respective tables in the database (*principle of destructibility*). If, for example, the row of the suppliers table containing supplier "S003452" is also referenced by another record, such as invoice "05266" in the example, then the database must ensure its deletion in response to the MCRS is managed so that neither record is compromised, for example by using a system of pointers.

### 301.2.6 Ensuring electronic components are really gone

Good business systems, including MoReq2010® compliant records systems, are designed with many levels of built in redundancy to ensure that the organisation's data can survive critical disasters such as hardware failures. For example, most organisations will regularly place system backups into remote offsite storage for extended periods of time.

The presence of these safeguards can make it difficult for an organisation to be finally rid of electronic content even when the disposal process has been followed. Ensuring that a record has been destroyed and its electronic content erased from the MCRS and/or the system of origin, does not necessarily mean that the same content cannot still be found within the organisation, on backups, in the email system, on shared file drives and with staff who have made personal copies for their own use.

The extent of this issue stretches beyond the records system and the scope of MoReq2010®, to the heart of the organisation's corporate policies and procedures around information governance. Each organisation must assess its own risk and put into place appropriate mitigation strategies.

Some of the measures that an organisation can put into place to mitigate the risk of retaining electronic content after it has been destroyed, include:

- Regularly recycling backups, so that no backup is older than (say) three months, this then sets a fixed time window from when records are destroyed in the MCRS to when their content is no longer recoverable from backup;
- Implementing a method for staff to send references to records in emails (see, for example, **R101.4.16**), rather than sending the record itself as an attachment (this will also reduce the amount of storage required for the corporate email system);
- Regularly capturing email into the records system, or purchasing an email archiving system that is also an MCRS;
- Locking down shared file drives and requesting that employees do not store records outside the records system;
- Automatically removing content stored outside the corporate business and records systems that has not been accessed for a period of (say) three months;

- Training and educating staff, raising awareness of the issues, and providing incentives around the proper use of corporate records systems; and
- Conducting regular information audits and compliance health checks.

The list above reinforces the knowledge that implementation of MoReq2010®, while an important contributing factor to good records management practice, is not in itself sufficient for ensuring that an organisation complies with its regulatory obligations, or puts into place a sound corporate-wide information governance framework.

## 301.4 Functional Requirements

### R301.4.1

In accordance with core services requirement **R6.5.19**, the MCRS must allow records to be created, under **R6.5.10**, with electronic components, that have immutable content held securely in either the originating system or a separate data store.

*The content of an electronic component must be immutable and secure, meaning it must be unchanging and unchangeable. The content must be identical each time it is accessed.*

### R301.4.2

The MCRS must ensure that all electronic components (**E301.7.1**) are created with the metadata listed under **R6.5.19**, as well as the following additional system metadata:

- Content Media Type (**M301.7.2**).

If the record has more than one electronic component then each electronic component belonging to the record must also have a:

- Presentation Order (**M14.4.84**).

Each electronic component will also have electronic content that is either:

- Directly available to an authorised user as electronic content, or
- Indirectly available to an authorised user as a URI.

*As explained in the rationale to **R301.4.3**, the MCRS may provide a URI to the content in either the originating system or data store. Whether directly or indirectly accessible, the Content Media Type represents the format of the electronic content of the component in its transportable form.*

*Note that an MCRS that keeps component content internally in its own data store may make it accessible as a URI, while equally, an MCRS that manages component content in an external data store may well provide users with direct access to the content. Some MCRS solutions may support both methods. This is dependent on their design and their degree of integration with the data store or originating system. Importantly, MoReq2010® does not require that an MCRS provide support for both approaches.*

*Function reference: **F301.7.3***

### R301.4.3

The MCRS must allow an authorised user to access the content of an electronic component as a datafile, either directly from the MCRS or by providing the user with a URI.

*While held in the originating system or data store, the content of an electronic component must be accessible as a datafile of a specified media type, stored in the Content Media Type metadata element.*

*This is known as the content's "transportable form". The MCRS must allow users to either access the datafile directly or must provide a URI to the datafile to allow the user to access it indirectly.*

*Note that where the MCRS provides access via a URI, then there may be additional external access controls to the datafile that are imposed and managed by the originating system or data store and not by the MCRS. This is outside the scope of MoReq2010®.*

*Function references:* ***F301.7.4***, ***F301.7.5***

#### R301.4.4

Where there is more than one component in a record, the MCRS must generate a unique Presentation Order for each component.

*The Presentation Order provides a simple way of logically ordering the components of a record, when there is more than one, so they may be listed in a relevant way. The Presentation Order of each component must be unique; meaning no two components of the same record may have the same Presentation Order.*

*For example, when capturing the contents of a web page as a series of components, the main HTML script may be listed first in presentation order, before the other components (CSS, JavaScript, images, etc.). Similarly, multipart datafiles may have an intrinsic order, an email message may be presented before its attachments, and so on.*

*MoReq2010® does not give specific guidance on the presentation order of components within a record for different content types.*

#### R301.4.5

The MCRS must provide a Content Media Type for the datafile for each electronic component created under **R301.4.1**.

*The Content Media Type for an electronic component must be the MIME media type for the datafile generated when the component is exported from the originating system. MIME media types are managed by IANA (see RFC 4288, RFC 4855 and http://www.iana.org/assignments/media-types/index.html).*

*The content of an electronic component may, or may not, be stored in the originating system or in a data store associated with the MCRS in the format indicated by the Content Media Type, but must be converted into that media type on export. The Content Media Type therefore represents the transportable format of the content.*

*For example, a record of a web page may have three components:*

- *An HTML script with the media type, "text/html";*
- *A cascading style sheet with the media type, "text/css"; and*
- *An image with the media type, "image/jpeg".*

#### R301.4.6

The MCRS must set a flag in the metadata of an electronic component when it is created, under **R6.5.19**, that describes whether the MCRS must wait for confirmation of the destruction of the content of a component when the component is destroyed.

*Under **R301.4.8**, following the decision to destroy a record under a disposal schedule, the content of all electronic components of a record must be deleted from the originating system or the appropriate data store(s). Depending on the implementation of the data store and the degree of its integration to the MCRS, this deletion operation may require confirmation if it cannot be carried out automatically.*

**R301.4.7**

Subject to **R2.4.22**, and further to **R6.5.17**, the MCRS must allow an authorised user to browse and inspect electronic components in at least the following ways:

- Browse the electronic components of a record, in presentation order, and inspect their metadata; and
- Browse from an electronic component to its content, subject to the content being accessible, under **R301.4.3**.

*See **R6.5.17**. The terms "browse" and "inspect" are defined in **13. Glossary of Terms**.*

*Function reference: **F14.5.44***

**R301.4.8**

When the MCRS exports an electronic component, under **R11.4.8**, then it must either include a URI to the content in the export data or else embed the datafile representing the content, under **R301.4.3**, in the export data.

*A URI to the content will always be exported whenever the MCRS does not embed the full content of the electronic component directly into the XML data generated on export. When the Component Content is embedded in the XML data generated on export, then it is included as Base64 encoded binary data.*

*Some more sophisticated MCRS solutions may relegate this choice to the user. In other words, an authorised user may be able to choose as part of an export operation, either to export component content embedded in the exported XML data, or to export components with embedded URIs that enable the importing service to separately download the content for each component.*

*Function reference: **F14.5.185***

**R301.4.9**

When a record containing electronic components is destroyed, under **R8.4.20**, the MCRS must ensure that the managed content belonging to each electronic component is deleted from the originating system and any data stores, under **R301.4.1**.

If the MCRS cannot automatically delete the managed content then the MCRS must wait for confirmation of its deletion, under **R8.4.20**, before the component is destroyed.

*This is determined by the metadata flag referred to in **R301.4.6**. Note that the record cannot be destroyed until all of its components have been destroyed, and electronic components cannot be destroyed until their content has been either automatically deleted, or manually confirmed as having been deleted, by an authorised user, under **R8.4.20**.*

*Function reference: **F14.5.41**, **F14.5.119***

## 301.5 Non-functional Requirements

### N301.5.1

*Each MCRS may come in a different form. Some MCRS solutions are general purpose applications, but others have a dedicated purpose, intended for a particular vertical market, and support integration with only specific types of business system. The MCRS may even be built into a particular business system and be indistinguishable from it.*

Is the records system part of a business system, or tightly integrated with a business system, or is it a standalone records system?

### N301.5.2

*The purpose of the MCRS informs the types of component content it manages as well as how this content is managed. As a consequence, the MCRS may provide wide ranging support for a number of different component content types, or it may support only a few content types. The content of electronic components can vary from traditional office documents to a number of rows in a database.*

In combination with **N301.5.1**, what types of electronic component content does the records system support?

### N301.5.3

*Records may have one component or many components, this too is a result of the types of record and their electronic components that the MCRS is designed to support.*

Further to **N301.5.2** do the records with electronic components supported by the records system have one or more than one component and is the number of components fixed or variable?

### N301.5.4

*Integral to the design of an MCRS is the location of the content that is managed. This may be in the system of origin, in an internal data store controlled by the MCRS or in an external data store.*

In relation to **N301.5.1**, where is the content of electronic components stored for records in the records system?

### N301.5.5

*An important characteristic of a component is immutability which means that the content of the component does not change once the record has been created. The content of record components must therefore be unalterable. If electronic components are located externally to the MCRS then the organisation must employ safeguards to ensure the integrity of the component content is preserved in its environment.*

How does the records system ensure the immutability of content in its storage location under **N301.5.4**?

### N301.5.6

***N12.13.6** describes synchronisation issues that may occur when the MCRS and the content of components is backed up and restored separately or at different times. Records systems which manage records in the system of origin or in external data stores can be particularly vulnerable.*

*For example, if the MCRS is restored from a backup, then it may contain as active records, records that were previously destroyed, prior to the discontinuity, and for which the electronic content was deleted and is no longer in the external data store or system. Alternatively, there may be content in the external data store or system which belongs to records that do not exist in the MCRS, because they were created after the last backup but before the discontinuity occurred.*

*The design of the MCRS must overcome these potential synchronisation issues, especially when it manages records in place.*

How does the records system ensure that it re-synchronises with the content location specified in **N301.5.4** if one or the other has to be restored from backup as part of disaster recovery?

#### N301.5.7

*In MoReq2010® the content of some components can support automatic deletion by the MCRS. This is a property of the component itself. The content of other components cannot be automatically deleted by the MCRS in response to a disposal schedule.*

Further to **R301.4.6** does the records system manage components that can be automatically deleted, or manually deleted, or does it allow both, and if so how what characterises these different types of component content in the records system?

#### N301.5.8

*Where the record content managed by the MCRS is not able to be automatically deleted, then it must be either manually deleted, or deleted by some external process, and then confirmed, so that the MCRS can continue with the destruction of the record.*

If, under **N301.5.7**, the content of electronic components must be manually deleted from its location in **N301.5.4**, then, further to **R301.4.9**, what is the process for doing this and how is it confirmed in the records system?

#### N301.5.9

*When managing records with electronic components the MCRS also has design choice about how users access the content of these components. The MCRS may be able to serve up the content to the user, or it may provide the user with a URI to the content and require the user to access the content directly from the data store. Some solutions may provide both options.*

Further to **R301.4.3**, does the records system allow authorised users to access the content of electronic components as datafiles, or using a URI, or both?

#### N301.5.10

*Export of electronic components is similar to user access described under **N301.5.9**. The MCRS may include the content in the export directly, or it may export a URI that points to the location of the content, which the importing system must then retrieve.*

Further to **R301.4.8**, does the records system allow authorised users to export the content of electronic components as datafiles, or using a URI, or both?

#### N301.5.11

*As described in the rationale to **R301.4.3**, where the MCRS provides a URI to the content of an electronic component, then the user may require additional access controls to inspect the content.*

Further to **N301.5.9** and **N301.5.10** what further authorisation does the user require when retrieving content using a URI provided by the records system, and how is this managed?

#### N301.5.12

*Where the content of the electronic component is stored externally and accessed using a URI, additional security measures may need to be put in place to ensure that unauthorised users do not access the content from outside the MCRS.*

How does the MCRS or supporting system prevent unauthorised access to the URI retrieved under **N301.5.11**?

#### N301.5.13

***R301.4.5** requires that each electronic component have a Content Media Type. The Content Media Type relates to the content of the component in its transportable form. Depending on the nature and purpose of the MCRS under **N301.5.1**, and the types of electronic component content supported under **N301.5.2**, the MCRS may provide in depth support for component content types based on particular document formats, such as the Microsoft Office document formats or ODF.*

*The degree of support provided for components based on particular content formats may vary, for example, the MCRS may be able to index and search over the content of these components when they are captured as records. The MCRS may also be able to capture some or all of the properties of these document types as additional component metadata.*

Further to the types of component content supported under **N301.5.2**, what common content types and formats does the records system support, and in which of the following ways:

- Identify the document type and index and search over the document content?, or
- Identify the document type and extract additional metadata from the document?

(Include the date the list was compiled as additional document formats may be added over time.)

#### N301.5.14

*A records system is designed to manage records and their components, however, the content of record components, especially electronic content, can proliferate in an organisation and copies may exist outside an MCRS as explained in **301.2.6 Ensuring electronic components are really gone**.*

*While the records system by itself cannot necessarily remove this proliferation it should do as much as it can to assist in resolving it. For example, when records are captured from a system, such as an email system or a shared drive, and placed into a data store, then the records system may be able to simultaneously delete the source content of the new record so that there remains only one copy of the content now in the data store, and not two.*

In what ways does the records system assist in controlling unmanaged record content within the organisation?

**N301.5.15**

*Many operating systems do not overwrite the binary data of a datafile when it is deleted from storage media. In sensitive and secure environments the footprint previously occupied by the datafile in the repository of the MCRS should be overwritten to ensure the datafile cannot be wholly or substantially recovered, for example, from close analysis of magnetic disc surfaces.*

Does the records system provide a mechanism for overwriting the content of electronic record components when they have been destroyed and are deleted under **R301.4.9**?

## 301.6 Glossary of Terms

| Term | Explanation and relationship to general concepts |
|---|---|
| **Content media type** | (noun) See **media type**. |
| **Data store** | (concept) A **secure** location for storing **electronic** data or **content** so that it is accessible by an **MCRS** or other **business system**.<br>See **internal data store** and **external data store**. |
| **Electronic component** | (entity) A **sub-type** of **component** whose **content** is **electronic**. |
| **External data store** | (concept) A **data store** that is not under the direct control of the **MCRS**. An external data store may be managed by another **business system** or it may even be an integral part of the other business system, similar to an **internal data store**. The external data store may be shared and used by several different business systems. |
| **Internal data store** | (concept) A **data store** that is fully controlled and managed by the **MCRS**. All access to an internal data store will be through the MCRS which can then use its normal **access control** methods to ensure that the **content** remains **immutable** (is not modified or deleted). |
| **Media type** | (noun) The **datafile** format of the **content** of an **electronic component** in its **transportable form**. **MoReq2010®** specifies the use of MIME media types. For example, a video that uses MPEG-4 Part 14 (otherwise known as MP4) has a MIME media type of "video/mp4", while a Microsoft Excel spreadsheet has the MIME media type "application/vnd.ms-excel".<br>See also **datafile** and **transportable form**. |
| **Originating system** | (concept) The **application** or **business system** where the **electronic content** that is made into a **record** is generated. The content generated by the originating system must be **transportable**, so that either on **creation** of the record, or later, it can be transmitted to a **receiving system** or **captured** by the |

| Term | Explanation and relationship to general concepts |
|---|---|
| | **MCRS** and placed into a **data store**.<br>See also **information system**, **receiving system** and **transportability**. |
| **Receiving system** | (concept) Any **application** or **business system** that receives and stores or uses **electronic record content**. For electronic content to be transmitted and received it must have a **transportable form**.<br>See also **information system**, **originating system** and **transportability**. |
| **Transportability, Principle of** | (concept) The principle that **electronic content** cannot be made into a **record** in an **MCRS** unless it can be retrieved or transmitted from its **originating system** in a **transportable form**. Without transportability, records cannot be **transferred** or **migrated** and **interoperability** requirements cannot be met. A specialised **business system** may keep records in **tables** in a **database** or other **proprietary** internal format. However, according to the principle of transportability there must be some provision made for exporting the **content** from the business system or it is not sufficient for a record.<br>See also **completeness**, **destructibility**, **discreteness** and **immutability**. |
| **Transportable form** | (noun) The **content** of an **electronic component** as a **datafile** that can be described by a **media format**, allowing it to be stored in a **data store**, transmitted or received by **applications** and other **business systems**, and retrieved by the **authorised users** of the **MCRS**.<br>See also **datafile**. |

## 301.7 Information Model

### E301.7.1 Electronic Component

| | |
|---|---|
| **System Identifier** | a2374646-3b29-4a30-93a8-96e369dc150c |
| **Title** | **Electronic Component** |
| **Description** | Definition of an electronic component, as a sub-type of component, used in MCRS solutions that manage electronic content |
| **Sub-type of** | Component (**E14.2.3**) |
| **Service** | Record Service |
| **Additional system metadata** | *As for Component (**E14.2.3**) plus the following additional system metadata:*<br>• Media Type (**M301.7.2**)<br>The following metadata element is necessary for records with more than one electronic component, under **R301.4.2** and **R301.4.4**:<br>• Presentation Order (**M14.4.84**) |
| **Additional functions** | *As for Component (**E14.2.3**) plus the following additional functions:*<br>• Electronic Component – Get Content (**F301.7.4**)<br>• Electronic Component – Get Content URI (**F301.7.5**)<br>*The following function replaces Component – Create (**F14.5.38**):*<br>• Electronic Component – Create (**F301.7.3**) |
| ***Usage notes*** | *The new function Electronic Component – Create (**F301.7.3**) replaces the function Component – Create (**F14.5.37**) for electronic components* |

### M301.7.2 Content Media Type

| | |
|---|---|
| **System Identifier** | 277c94bc-9cc8-4018-8cd9-dc2aa3b189b5 |
| **Title** | **Content Media Type** |
| **Description** | The media type of the content of the electronic component |
| **Entity Type** | Electronic Component (**E301.7.1**) |
| **Min Occurs** | 1 |
| **Max Occurs** | 1 |
| **Modifiable?** | No |
| **Entity Reference?** | No |

| | |
|---|---|
| **Datatype** | Valid MIME media type |

### F301.7.3 Electronic Component - Create

| | |
|---|---|
| **System Identifier** | ea33d749-92aa-421b-9eba-6fb90786d4b9 |
| **Title** | **Electronic Component – Create** |
| **Description** | Create the electronic component of a record |
| **Entity Type** | Electronic Component (**E301.7.1**) |
| **Entity metadata modified** | • System Identifier (**M14.4.100**)<br>• Created Timestamp (**M14.4.9**)<br>• Originated Date/Time (**M14.4.61**)<br>• Record Identifier (**M14.4.86**)<br>• Title (**M14.4.104**)<br>• Description (**M14.4.16**)<br>• Automatic Deletion Flag (**M14.4.3**)<br>• Content Media Type (**M301.7.2**)<br>• *Contextual metadata elements*<br>*If the record has more than one electronic component then it will also have:*<br>• Presentation Order (**M14.4.32**)<br>*If contextual metadata elements are applied from a template the following template metadata element may be modified (if it has not already been set):*<br>• First Used Timestamp (**M14.4.32**) |
| ***From functional requirement(s)*** | **R301.4.2** |
| ***Purpose*** | Event generation only |
| **Additional event metadata (see R2.4.16)** | • Participating Component Identifier (**M14.4.66**)<br>• Participating Record Identifier (**M14.4.77**)<br>• Event Comment (**M14.4.25**)<br>• Metadata Change Entry (**D14.3.3**)<br>• Applied Template Identifier (**M14.4.2**) |
| ***Usage notes*** | • *The electronic component is created simultaneously with its record (see **F14.5.121 Record – Create**)*<br>• *The Content Media Type must include a valid MIME media type for the transportable form of the electronic content*<br>• *The electronic component may be created with contextual metadata elements as well as the system metadata elements listed* |

| | |
|---|---|
| | • *If contextual metadata elements are added from a template then the Applied Template Identifier must be included in the event metadata*<br>• *For each metadata element set on creation, except System Identifier and Created Timestamp, a Metadata Change Entry must be added to the corresponding event* |

### F301.7.4 Electronic Component - Get Content

| | |
|---|---|
| **System Identifier** | c1a3d38b-72ee-48c2-983b-b028db002d7f |
| **Title** | **Electronic Component – Get Content** |
| **Description** | Retrieve the content of an electronic component from the system |
| **Entity Type** | Electronic Component (**E301.7.1**) |
| **Entity metadata** | *No metadata elements are modified* |
| ***From functional requirement(s)*** | **R301.4.3** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.76**) |
| ***Usage notes*** | • *The system must implement this function as well as, or instead of, the function Electronic Component – Get Content URI (**F301.7.5**)*<br>• *This function should generate an event, subject to **R2.4.13**, whenever the user retrieves the content of the electronic component directly from the system* |

### F301.7.5 Electronic Component - Get Content URL

| | |
|---|---|
| **System Identifier** | 1f0784ee-3825-4422-ad45-46c3dee59c74 |
| **Title** | **Electronic Component – Get Content URI** |
| **Description** | Retrieve a URI to the content of an electronic component from the system |
| **Entity Type** | Electronic Component (**E301.7.1**) |
| **Entity metadata** | *No metadata elements are modified* |

| | |
|---|---|
| ***From functional requirement(s)*** | **R301.4.3** |
| ***Purpose*** | • Access control<br>• Event generation |
| **Additional event metadata (see R2.4.16)** | • Participating Class Identifier (**M14.4.76**) |
| ***Usage notes*** | • *The system must implement this function as well as, or instead of, the function Electronic Component – Get Content (**F301.7.4**)*<br>• *This function should generate an event, subject to **R2.4.13**, whenever the user retrieves from the system a URI to the content of the electronic component* |