
**Information technology — Multimedia
framework (MPEG-21) —**

**Part 8:
Reference software**

**AMENDMENT 3: Contract Expression
Language (CEL) and Media Contract
Ontology (MCO) Reference Software**

iTeh STANDARD PREVIEW
(standards.itih.ai)

ISO/IEC 21000-8:2008/Amd.3:2015
<https://standards.itih.ai/catalog/standards/sist/8ad20923-a566-40e0-8841-a1604f728658/iso-iec-21000-8-2008-amd-3-2015>
*Technologies de l'information — Cadre multimédia (MPEG-21) —
Partie 8: Logiciel de référence*

*AMENDEMENT 3: Langage d'expression des contrats (CEL) et
ontologie pour contrats de médias (MCO): logiciels de référence*

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC 21000-8:2008/Amd 3:2015](https://standards.iteh.ai/catalog/standards/sist/8ad20923-a566-40e0-8841-a1604f728638/iso-iec-21000-8-2008-amd-3-2015)
<https://standards.iteh.ai/catalog/standards/sist/8ad20923-a566-40e0-8841-a1604f728638/iso-iec-21000-8-2008-amd-3-2015>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: [Foreword - Supplementary information](http://www.iso.org/foreword)

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This Amendment describes extra reference software for ISO/IEC 21000-20:2013 and ISO/IEC 21000-21:2013.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 21000-8:2008/Amd 3:2015

<https://standards.iteh.ai/catalog/standards/sist/8ad20923-a566-40e0-8841-a1604f728638/iso-iec-21000-8-2008-amd-3-2015>

Information technology — Multimedia framework (MPEG-21) — Part 8: Reference Software, AMENDMENT 3: Contract Expression Language (CEL) and Media Contract Ontology (MCO) Reference Software

In 3.2 Abbreviated terms, insert the following acronyms:

CEL Contract Expression Language

MCO Media Contract Ontology

At the end of 4.2 Overview of ISO/IEC 21000 reference software, add:

— ISO/IEC 21000-20:2013, Information technology — Multimedia framework (MPEG-21) — Part 20: Contract Expression Language (CEL): The corresponding reference software modules provide an environment to manage CEL based documents. A reference to the reference software for ISO/IEC 21000-20:2013 is described in 5.23

— ISO/IEC 21000-21:2013, Information technology — Multimedia framework (MPEG-21) — Part 21: Media Contract Ontology (MCO): The corresponding reference software modules provide an environment to create and edit MCO documents. A reference to the reference software for ISO/IEC 21000-21:2013 is described in 5.24

<https://standards.iteh.ai/catalog/standards/sist/8ad20923-a566-40e0-8841-41664728058/iso-iec-21000-8-2008-amd-3-2015>
In Clause 5 Reference software for the ISO/IEC 21000 parts, insert 5.23 and 5.24:

5.23 ISO/IEC 21000-20:2013

5.23.1 Introduction

This Clause describes the reference software for the ISO/IEC 21000-20 Contract Expression Language (CEL).

The CEL Reference Software consists of a number of modules implementing operations and functionalities on Contract documents according to ISO/IEC 21000-20.

5.23.2 CEL Reference Software Java implementation

The Java CEL Reference Software provided implements the following modules:

- *Contract Identification*: This module uniquely identifies an MPEG-21 CEL contract.
- *Contract Validation*: This module syntactically validates an MPEG-21 CEL contract.
- *Contract Check-with*: This module verifies if a request usage matches with the content expressed in a CEL contract.

A test class for all the modules is provided as well.

5.23.3 Description of CEL Services

The description of the implemented CEL Services is detailed below:

Name or link: Contract Identification	
Description	
This module allows a user to obtain a uniquely identifier for a contract. The code implemented generates a new unique identifier ID and returns it.	
Input	
— A CEL contract.	
Output	
— The unique identifier for the contract.	

Name or link: Contract Validation	
Description	
This module syntactically validates an MPEG-21 CEL contract. The code implemented validates the given contract against the CEL schema.	
Input	
— A CEL contract.	
Output	
<p style="text-align: center;">iTech STANDARD PREVIEW (standards.itech.ai)</p> <p style="text-align: center;">ISO/IEC 21000-8:2008/Amd 3:2015 https://standards.itech.ai/catalog/standards/sist/8ad20923-a566-40e0-8841-a1004728058/iso-iec-21000-8-2008-amd-3-2015</p>	
— It returns if the contract is valid or not and the reasons why.	

Name or link: Contract Check With	
Description	
<p>This service allows users to verify if a request action against a contract matches with its content. The result of the operation includes one of these values:</p> <ul style="list-style-type: none"> • OK if the action can be performed • USER_NOT_FOUND if the service invoker is different from one of the contract parties • DEONTIC_EXPRESSION_NOT_FOUND if a deontic expression is missing or incomplete • CONDITION_NOT_SATISFIED if the deontic expression is present but exists at least one condition that is not satisfied • UNKNOWN_ERROR if any unknown error occurs <p>The result of the operation also contains a message detailing more information about the result of the operation.</p>	
Input	
— A CEL contract.	
Output	

- The result of the contract based authorisation, detailing the reasons why if the authorization has been negative.

5.24 ISO/IEC 21000-21:2013

5.24.1 Introduction

This Clause describes the reference software for the ISO/IEC 21000-21 Media Contract Ontology (MCO).

The MCO Reference Software consists of a number of web applications or services (mco-refsw services) implementing operations and functionalities on Contract documents according to ISO/IEC 21000-21.

5.24.2 mco-refsw services

The mco-refsw set of services can be deployed on an Apache http server, for providing an MCO working environment to a number of users.

Each user can create and operate on MCO documents stored on a dedicated server repository area.

The services are grouped into contexts:

- “MCO CRUD” context - for includes the functionalities for
 - creating, presenting, and editing MCO contract documents
 - exporting MCO documents to the client side
 - deleting MCO documents from the server repository
 - delivering MCO document to a remote server for further delivery to a user
 - loading MCO document to the user working area on server repository
- “Check with and search” context – which includes the functionalities for
 - Check with - for identifying which contracts, in the user repository area, contain Permissions compatible with a user defined target exploitation. A target exploitation is defined by a Permission, for a given user, to act a given action on a given object, under a given set of conditions.
 - Search – for identifying which contracts, in the user repository area, contain the contract elements given in input.
- Help, Info, and Docs context – it is a utility context for making available to the user documentation information on the services.
- Setting context – it is a utility context for allowing the user to set his/her preferences on the behavior of the services.

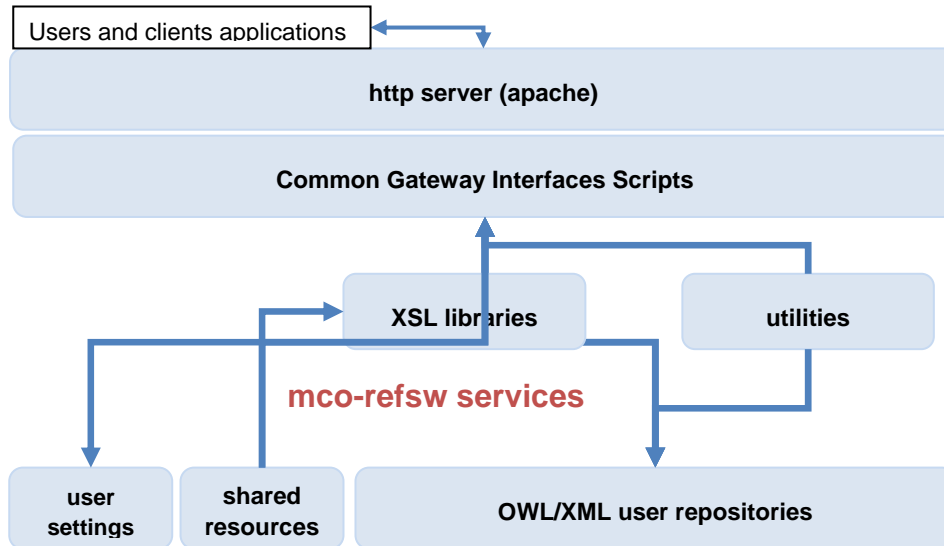
5.24.2.1 Architecture

The technical implementation of “mco-refsw services” is depicted in the Figure below. Services are provided to the users by a normal HTTP server, so that a simple browser can be used, although support of Scalable Vector Graphics (SVG) is required for the Contract Presentation part. The scripts deployed under the common gateway interface parse the user requests and run the basic operations consistently. The actual operations on

the MCO/OWL files, serialized according to OWL/XML specification, are executed by using the provided XSL libraries.

Each User can define his/her settings and owns a part of the document repository.

The “mco-refsw services” can also be used by Client Applications.



STANDARD PREVIEW
(standards.iteh.ai)

5.24.2.2 System requirements

The “mco-refsw” services can be run on any Linux system, although they have been tested specifically on Ubuntu. To install “mco-refsw” on a system with a Linux distribution different from Ubuntu, it is necessary to modify the dependency check part of the configuration and might be useful create aliases to some commands.

The configuration script requires the *tcsh* interpreter.

Before actually installing the software it is required to run the configuration script, which will set a number of properties, the default values of which can be overwritten, by indicating the desired option from the command line.

To run the configuration, execute the following commands from a terminal:

```
$cd /<yourpath>/mco-refsw
```

```
$./configure [OPTIONS]
```

For knowing the configuration options and their default values, just run:

```
$./configure -h
```

The configuration script will first run a dependency check. In case some package will result missing, the script will notify the problem and exit.

On success, the configuration will create the scripts for completing the installation and for the users administration.

The installation command must be given with administrative privileges.

```
$cd /<yourpath>/mco-refsw
```

```
$sudo ./install
```


Within the installation, a user named "*default*" is created. You will be required to set a password for this user and confirm it.

The user "*default*" is used also as a template for the creation of other users.

To uninstall it is sufficient to run, as root:

```
$cd /<yourpath>/mco-refsw1
```

```
$sudo ./uninstall
```

The uninstall script just makes inactive the services to the http server but it does not remove the users files and the mco-refsw working directories.

5.24.2.2.1 Dependencies for Ubuntu Server 12.04

Dependencies in the case of Ubuntu Server 12.04, out of the box, are:

- tcsh, apache2, xsltproc, graphviz, imagemagic, curl, tofrodos, gridside-clients, libxml2-utils
- Java virtual machine (version 7 or higher) is required for running the java conversion components possibly required by "load" and "export" services. One component, which makes use of OWL-API library (available from <http://owlapi.sourceforge.net/>), implements OWL serialization conversion from/to OWL/XML to/from RDF/XML, while the other component uses the MCO-to-CEL and CEL-to-MCO conversion modules described in A.15.

5.24.2.3 Description of MCO CRUD

This description provides all the services for "Create, Read, Update, and Delete (CRUD)" MCO documents in the user repository area.

For initiating a new contract document no input is required. A new MCO document, with an automatically generated filename, will be created with an empty contract individual.

By clicking on the ellipse representing the contract individual in graphical representation, the user will get forms for adding other contract elements, such as:

- Contract metadata, such as the contract date;
- Contract Parties, either Organizations or Persons;
- Deontic Expressions (Permissions, Obligations, Prohibitions)
- The textual version of the contract (loading a TXT file)

The refresh button allows the user to re-call the "viewedit" service and get the graphical representations updated with all the added contract elements. The user can get an appropriate set of forms, for each contract element, represented in the diagram with an ellipse, for adding further elements or metadata, or for removing them.

5.24.2.3.1 Main services

Name or link: <a href="http://<host>/cgi-bin/mco-refsw/<user>/mco.crud/create">http://<host>/cgi-bin/mco-refsw/<user>/mco.crud/create
Description
Creates a new OWL document, including an Individual of the mco-core:Contract Class.

The default behaviour is that, after successful creation, the OWL document is immediately opened for editing, by re-direction to “viewedit”

The optional input “identifier” is used to name the OWL document and as a suffix to the OntologyIRI prefix, defined in the user settings, for setting the OntologyIRI of the OWL document.

If not given the module will create a default identifier based on the timestamp.

Method

- GET

Input

- identifier - if given it will be the basename of the OWL file, which is otherwise created automatically.
- goedit – Boolean, optional, default=”true”. If true, on Success, the service will directly re-direct to the utility service “viewedit”.

Output

- If (goedit = false)
Text with Resulting code and message for operation; e.g. “200\nSuccess\n”
- If (goedit = true) on Success
“200\nSuccess\n” and the TEXT/HTML output of “viewedit”
- On failure
Text with Resulting code and message for operation: e.g. “400\nBad request\n”

Error codes and message

- 400 Bad request – Method must be GET. Input syntax error
- 403 Forbidden – The user repository area already contains OWL document with the same name
- 500 Server Error – The service could not complete the task successfully

Name or link: <http://<host>/cgi-bin/mco-refsw/<user>/mco.crud/export>

Description

By default it serves the requested OWL file to User, serialized in OWL/XML. If the optional parameter “format” is given, the returned document can also be:

- OWL file serialized in RDF/XML
- CEL / XML Contract, as resulting from the use of the MCO-to-CEL conversion module

Method
— GET
Input
— instance - name of OWL file
— format – one among OWL/XML (default), RDF/XML, CEL/XML
Output
— OWL/XML, or RDF/XML, or CEL/XML
Error codes and message
400 Bad request – Method must be GET. Input syntax error
404 Not Found –OWL document not found
500 Server Error – The service could not complete the task successfully

Name or link: <a href="http://<host>/cgi-bin/mco-refsw/<user>/mco.crud/delete">http://<host>/cgi-bin/mco-refsw/<user>/mco.crud/delete
Description
Deletes the given OWL file from the User's repository/Amd 3:2015 https://standards.iteh.ai/catalog/standards/sist/8ad20923-a566-40e0-8841-a1604f728638/iso-iec-21000-8-2008-amd-3-2015
Method
— GET
Input
— instance - name of OWL file
Output
— Text with Resulting code and message for operation: e.g. "200\nSuccess\n"
Error codes and message
400 Bad request – Method must be GET. Input syntax error
404 Not Found –OWL document not found
500 Server Error – The service could not complete the task successfully

Name or link: <a href="http://<host>/cgi-bin/mco-refsw/<user>/mco.crud/deliver">http://<host>/cgi-bin/mco-refsw/<user>/mco.crud/deliver	
Description	
<p>Delivers the given Contract document, stored in User repository area, to a remote service, in order to be further delivered to another User given in input.</p> <p>This requires the configuration of the URL of the remote service and the account for accessing it, through the settings context.</p>	
Method	
— GET	
Input	
<p>— instance - name of OWL file</p> <p>— deliveruser – User to whom the remote service has to forward the contract</p>	
Output	
— Text with Resulting code and message for operation: e.g. “200\nSuccess\n”	
Error codes and message	
<p>400 Bad request – Method must be GET. Input syntax error</p> <p>404 Not Found – OWL document not found</p> <p>500 Server Error – The service could not complete the task successfully</p>	

Name or link: <a href="http://<host>/cgi-bin/mco-refsw/<user>/mco.crud/load">http://<host>/cgi-bin/mco-refsw/<user>/mco.crud/load	
Description	
<p>Receives the MCO file in input and stores it in the User repository area.</p> <p>The default behaviour is that, on success, the OWL document is immediately open for editing, by redirection to “viewedit”</p> <p>If input file is not an MCO document with OWL/XML serialization, then conversion is tried from MCO in RDF/XML serialization and from CEL/XML.</p>	
Method	
— POST	
Input	
<p>— instance - name of OWL file</p> <p>— goedit - Boolean, optional, default=“true”. If true, on Success, the service will directly redirect to the utility service “viewedit”</p>	