

Rave Web Services

Exported on: 20-Oct-2025

Medidata Solutions, Inc., Dassault Systèmes company.

Corporate Office
350 Hudson Street, Floor 9
New York, NY 10014
+1 212 918 1800

Medidata Solutions, Inc. Proprietary and Confidential Information - Medidata and Authorized Client Personnel Only. This document contains proprietary and confidential information that shall be distributed, routed, or made available only within Medidata and specifically authorized client personnel. Further distribution, including distribution to a third-party, requires Medidata's advance, written permission. Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including, but not limited to, photocopying, photographing and recording, for any purpose without the express permission of Medidata Solutions, Inc.

© 2025 Medidata Solutions, Inc., a Dassault Systèmes company. All rights reserved. All Medidata trademarks or logos are the property of Medidata Solutions, Inc. All other brands or product names used in this document are trademarks, or registered trademarks, of their respective owners.

Table of Contents

1 Rave Web Services	8
1.1 Get Started with Rave Web Services	9
1.1.1 About Rave Web Services (RWS)	15
1.1.2 RWS Authentication Methods	23
1.1.3 RESTClient Browser Add-On	31
1.1.4 Sample Code for RWS	32
1.1.4.1 Sample Android App and RWS PostMan Code	45
1.1.4.2 Android POST RWS Code	50
1.2 Quick Reference	54
1.3 Cookbook	64
1.3.1 Create a Subject	67
1.3.2 Retrieve a Clinical View CSV-formatted Version	67
1.3.3 Retrieve ODM Format Clinical Views	68
1.3.4 Retrieve Rave Audit Data Incrementally	68
1.3.5 Retrieve List of Subjects in a Study	68
1.3.6 Update Fields	69
1.3.7 Update Item Groups	69
1.3.8 Upsert Forms	69
1.3.9 Use Biostat Adapter to View Comments and Protocol Deviations	69
1.3.10 Use Biostat Adapter to View Options in Data Dictionaries	70
1.3.11 Add External Audits to a Field	70
1.3.12 Update Field Statuses	71
1.4 Solutions	71
1.4.1 Biostat Adapter	72
1.4.2 ODM Adapter	73
1.5 Request Services	78
1.5.1 Clinical View Extracts	80
1.5.1.1 Retrieve Clinical Datasets Metadata as ODM	80
1.5.1.2 Retrieve Clinical View Form Datasets as CSV	90

1.5.1.3 Retrieve CSV Format Clinical Views.	99
1.5.1.4 Retrieve Incremental CSV Datasets.	100
1.5.1.5 Retrieve Incremental ODM Datasets.	105
1.5.1.6 Retrieve Clinical View Datasets as ODM.	109
1.5.1.7 Include CRF Version.	125
1.5.2 Retrieve Raw or Regular Data.	127
1.5.2.1 Include Raw Values in a Regular Service.	129
1.5.2.2 Include Standard and Unit Values in a Regular Service.	132
1.5.2.3 Include Standard Coded and Decode Values for Data Dictionaries.	139
1.5.2.4 Include Code and Decode Values for Data Dictionaries.	144
1.5.2.5 Retrieve Metadata for CV ODM Extensions.	147
1.5.3 Configurable Datasets.	152
1.5.3.1 Retrieve Results in Default XML Format.	152
1.5.3.2 Retrieve Paged Results.	154
1.5.3.3 Retrieve Lab Data with the Lab Analyte Ranges CSV Dataset.	162
1.5.3.4 Retrieve Clinical Data with the Clinical Audit Records Dataset.	168
1.5.3.4.1 Example - Retrieve Clinical Data with the Clinical Audit Records Dataset.	190
1.5.3.5 Retrieve Admin Data with the Version Folders Dataset.	197
1.5.3.5.1 Example - Retrieve Admin Data with the Version Folders Dataset.	208
1.5.3.6 Retrieve Admin Data with the Users Dataset.	212
1.5.3.6.1 Example - Retrieve Admin Data with the Users Dataset.	231
1.5.3.7 Retrieve Admin Data with the Sites Dataset.	233
1.5.3.7.1 Example - Retrieve Admin Data with the Sites Dataset.	242
1.5.3.8 Retrieve Admin Data with the Signatures Dataset.	244
1.5.3.8.1 Example - Retrieve Admin Data with the Signatures Dataset.	252
1.5.3.9 Retrieve the Protocol Deviations Dataset.	253
1.5.3.10 Retrieve the Comments Dataset.	255
1.5.3.11 Retrieve Custom Dataset Log Messages.	257
1.5.3.12 Retrieve Metadata with the Version Folders with Forms Dataset.	264
1.5.3.13 Retrieve Clinical View Metadata.	271
1.5.3.14 Retrieve Subject Calendar Visit Dates.	273

1.5.4 Retrieve the Data Dictionaries Dataset	282
1.5.5 Retrieve Clinical Study Lists.	283
1.5.6 Retrieve Custom Datasets.	290
1.5.7 Retrieve Study and Library Metadata.	297
1.5.7.1 Example - List Global Library CRF Versions.	307
1.5.7.2 Example - List Global Library Drafts.	308
1.5.7.3 Example - List Global Library Studies.	309
1.5.7.4 Example - List Studies.	310
1.5.7.5 Example - List Study CRF Versions.	311
1.5.7.6 Example - List Study Drafts.	312
1.5.7.7 Example - Request a Complete CRF Version with Vendor Attributes.	313
1.5.7.8 Example - Request a Complete Global Library CRF Version.	315
1.5.7.9 Example - Request a Complete Global Library CRF Version with Vendor Attributes.	317
1.5.7.10 Example - Request a Complete Study CRF Version.	318
1.5.7.11 Example - Request Form Labels as Independent Items.	320
1.5.7.12 Example - Request Form Labels as Independent Items and Vendor Attributes.	321
1.5.8 Retrieve Study and Library Metadata Lists.	322
1.5.9 Retrieve the List of Subjects in a Study.	331
1.5.10 Work With Study and Library Metadata.	345
1.5.10.1 Example - Import Forms with Labels and No Fields.	354
1.5.10.2 Example - Import Label Fields.	355
1.5.10.3 Example - Import Lab Settings.	356
1.5.10.4 Example - Importing Metadata (Study Design) in RWS.	358
1.5.11 Work With Subject Data.	359
1.5.11.1 Best Practices - Work With Subject Data.	374
1.5.11.2 Example - Assign a Lab to a Form.	375
1.5.11.3 Example - Change the Data Point Status.	376
1.5.11.4 Example - Identify Repeated Objects.	378
1.5.11.5 Example - Insert and Remove Log Lines Across Multiple Forms.	379
1.5.11.6 Example - Make a Record Inactive.	381
1.5.11.7 Example - Reactivate an Inactive Log Line.	382

1.5.11.8 Example - Use a Secondary Subject Name.	384
1.6 Resources.	386
1.6.1 Create a Subject in a Study and Site.	388
1.6.1.1 Example - Insert a Subject.	392
1.6.2 Locate a Folder.	393
1.6.2.1 Example - Locate Repeating Folders by Context.	408
1.6.2.2 Example - Locate My Folder.	410
1.6.2.3 Example - Update a Form in a Nested Folder.	437
1.6.3 Locate a Form.	438
1.6.3.1 Example - Insert a Form.	445
1.6.3.2 Example - Locate Repeating Forms by Context.	446
1.6.3.3 Example - Locate Repeating Log Lines and Repeating Forms by Context.	448
1.6.3.4 Example - Update a Form.	449
1.6.4 Locate a Log Line.	450
1.6.4.1 Example - Identify Repeated Log Lines.	454
1.6.4.2 Example - Locate Log Lines by Context.	456
1.6.4.3 Example - Update or Create a Logline Record.	457
1.6.5 Locate a Site.	460
1.6.6 Locate a Study.	465
1.6.7 Locate a Subject.	472
1.6.8 Review a Field.	479
1.6.9 Update a Field.	482
1.6.10 Update a Data Dictionary Drop Down List Field.	482
1.6.11 Update a Data Dictionary Search List Field.	485
1.6.12 Update a Date Field.	491
1.6.13 Update a Dynamic Search List Field.	495
1.6.14 Update a Text Field.	500
1.6.15 Update a Unit Dictionary Field.	506
1.6.16 Update a Whole Item Group.	508
1.6.17 Upsert a Form in RWS.	518
1.6.17.1 Example - Update or Insert a Form Using TransactionType Upsert.	523

1.6.18 Upsert a Log Line.	524
1.6.19 Add Coding Decisions to Fields.	533
1.6.19.1 Example - Code a Data Point Using a Third-Party Coding System.	544
1.6.20 Add External Audits to Fields.	546
1.6.20.1 Example - Add an External Audit Record.	551
1.6.21 Use Field Queries.	552
1.6.21.1 Example - Open a Rave Query Using TransactionType Context.	573
1.6.21.2 Example - Open a Rave Query Using ItemData SubmitMarkingsOnly.	574
1.6.22 Add Translations to Fields.	575
1.6.22.1 Example - Import Translated Data.	577
1.6.23 Update Field Workflow Properties.	578
1.6.24 Assign Labs to Forms.	588
1.7 References.	591
1.7.1 Operational Data Model (ODM) Schema.	592
1.7.1.1 How to Use the Operational Data Model (ODM).	593
1.7.1.2 ODM Schema Downloads and Releases History.	597
1.7.1.3 RWS ODM Header Requirements.	604
1.7.1.4 ODM Study Metadata Schema Support.	606
1.7.1.5 ODM Subject Data Schema Support.	678
1.7.1.6 ODM Implementation.	721
1.7.1.7 ODM Notes.	723
1.7.1.8 A Valid ODM 1.3 Snapshot Document.	729
1.7.1.9 A Valid ODM 1.3 Transactional Document.	732
1.7.1.10 ODM Unicode Character Conversion List.	734
1.7.2 ODM Adapter Guidelines.	751
1.7.3 ODM Adapter Included Audit Subcategories.	754
1.7.4 Clinical View Service Limitations.	766
1.7.5 Configurable Datasets Authorization Best Practices.	767
1.7.6 Navigating Repeated, Nested Folders.	771
1.7.7 Definitions.	773
1.7.7.1 Code List Definitions.	774

1.7.7.2 Confirmation Message Definitions	778
1.7.7.3 Field Definitions	781
1.7.7.4 Field Edit Check Definitions	787
1.7.7.5 Folder Definitions	795
1.7.7.6 Form Definitions	798
1.7.7.7 Label Definitions	802
1.7.7.8 Log Line Definitions	807
1.7.7.9 Measurement Unit Definitions	810
1.7.7.10 Medidata Coder Definitions	813
1.7.8 Error Responses - Complete List	823
1.7.9 Specify a Study OID	841
1.7.10 URL Special Character Escaping	841
1.7.11 Product Mapping Terminology	847
1.8 Troubleshooting	872
1.9 FAQs	878

1 Rave Web Services

Rave Web Services (RWS) integrates Medidata Rave with third-party systems to exchange CDISC ODM standard clinical data or metadata synchronously and with immediate confirmation.

Get Started	What's New	...And More
Get Started with Rave Web Services	Medidata Rave Release Notes	RWS Request Services
RWS Authentication Methods	Medidata Rave ODM Adapter	RWS Resources
ODM Adapter	Training Workshop	References
Biostat Adapter	Subscribe to Rave EDC and Classic Rave Release News	

1.1 Get Started with Rave Web Services

Rave Web Services (RWS) integrates Medidata Rave with third-party systems to exchange CDISC ODM standard clinical data or metadata synchronously and with immediate confirmation.

The examples below are from [rwslib](#). rwslib is an open source Python library that provides a simple interface to RWS.

 **Important:** For any API calls that require authentication, contact your Medidata representative to get access to a Rave environment.

Installation

We strongly recommend working within a virtual environment like [Virtualenv](#).

Install with [pip](#):

```
> pip install rwslib
```

Or directly from GitHub with:

```
> pip install git+https://github.com/mdsol/rwslib.git
```

This also installs all required dependencies.

 **Note:** On Windows, lxml requires a binary installation.

Creating a Connection to RWS

Before you can work with [rwslib](#), you must create a connection to a Rave instance through the `RWSConnection` object:

```
> from rwslib import RWSConnection  
> rws = RWSConnection('innovate')
```

 **Note:** The first parameter to the `RWSConnection` is the name of the URL required to connect to. A URL that does not start with "http" is assumed to be the subdomain of mdsol.com. In the example above, "innovate" is <https://innovate.mdsol.com>.

If you require to override this behavior, supply a base URL that includes http or https at the start of the URL, for example:

```
> rws = RWSConnection('http://192.168.1.99')
```

It is important to understand that an `RWSConnection` is not a persistent connection to Rave, just a convenience class for making calls to RWS endpoints.

Making an RWS Request

Once you have an `RWSConnection`, you can use it to send messages to Rave and receive results back.

rwslib provides a set of request classes. To make a request, create an instance of that request type and pass it to the `RWSConnection` `send_request` method:

```
> from rwslib import RWSConnection  
> rws = RWSConnection('innovate')  
> from rwslib.rws_requests import VersionRequest  
> rws.send_request(VersionRequest())  
u'1.8.0'
```

`Rave.VersionRequest()` The result you get back from the `send_request` depends on the request type, since the Request objects can process the text values returned from `Rave.VersionRequest()` returns a string value, but other request types may return objects or python lists of objects.

All Request classes are descendants of the `RWSRequest` class. The rwslib API allows you to add new or custom requests to the library. This allows for easier versioning of requests and also subclassing of existing request types. For instance, you can subclass `VersionRequest()` to return major, minor, and patch-level

values as a tuple of integers, rather than as a string. In this way, you can make specialized request classes for your integration.

rwslib provides several sets of standard request types arranged into python units, as follows:

- `rws_requests.py` contains the RWSRequest class and standard requests like `VersionRequest()`.
- `rws_cv_requests.py` contains requests related to Rave Clinical Views and BioStat Gateway data extracts for Comments and Protocol Violations.
- `odm_adapter_requests.py` contains requests related to the ODM Adapter datasets added in Rave 2013.3.0.

Overriding Default Domain Name and Virtual Directory

For convenience, rwslib defaults the domain name to end with '[mdsol.com](#)' and the virtual directory to be 'RaveWebServices', for example:

```
> from rwslib import RWSConnection
> rws = RWSConnection('innovate')
> rws.base_url
'https://innovate.mdsol.com/RaveWebServices'
```

The default values work for most Rave URLs but you can override them, if required:

```
> from rwslib import RWSConnection
> rws = RWSConnection('http://10.0.1.20', virtual_dir='RWS')
> rws.base_url
'http://10.0.1.20/RWS'
```

Authentication

Most requests require authentication. You can authenticate requests through [Basic Authentication](#) by providing a Rave (not iMedidata) username and password:

```
> from rwslib import RWSConnection
> from rwslib.rws_requests import MetadataStudiesRequest
> rws = RWSConnection('https://innovate.mdsol.com', "raveusername", "ravepassword")
>
> # Make an authenticated request to Rave
> rws.send_request(MetadataStudiesRequest())
```

Alternatively, use [MAuth](#) credentials to make a request. MAuth is Medidata's API authentication mechanism. MAuth credentials consist of an App UUID representing the application making the request and a Private Key, representing its proof. You can use them with MAuth to sign requests.

Medidata provides the `requests_mauth` library, which provides MAuth signing capabilities for accessing Medidata APIs through MAuth:

```
> from requests_mauth import MAuth
> from rwslib import RWSConnection
> from rwslib.rws_requests import MetadataStudiesRequest
>
> app_id = '635r8aib-21e9-6b5f-867e-bk2358ub2784'
> key = open('private_key_file','r').read()
>
> rws = RWSConnection('https://innovate.mdsol.com', auth=MAuth(app_id, key))
>
> # Make an authenticated request to Rave
> rws.send_request(MetadataStudiesRequest())
```

A set of MAuth credentials are associated with a user in Rave just as with Basic Authentication. Requests are performed in the context of a user's rights and permissions. However, a user account associated with MAuth App ID does not have a password expiry. Hence, MAuth is a better approach to long-term integration with Rave URLs.



Note: You can associate an MAuth AppID with multiple Rave URLs, but only one user per URL.

Timeouts

By default, rwslib does not time out. You can set a timeout limit—in seconds—on `send_request`, after which a `Timeout` exception is thrown:

```
> from rwslib import RWSConnection
> rws = RWSConnection('innovate', 'my_username', 'my_password')
> # Get the rave version from rws
> rws.send_request(VersionRequest(),timeout=1)
```

In practice, set the timeout to a value greater than any expected valid response time, depending on the request types and volumes of data sent or received.

This timeout setting only applies to rwslib and does not alter timeouts in RWS itself or any other component in the network, such as load balancers.

Retries

By default, rwslib makes a request only once. You can adjust the number of retries by setting the retries parameter to the send_request:

```
> from rwslib import RWSConnection
> rws = RWSConnection('innovate', 'my_username', 'my_password')
> # Get the rave version from rws
> rws.send_request(VersionRequest(), retries=3)
```

 **Important:** When performing retries where a request changes data - for example, with POST requests
- Rave may incorrectly return an error for successful requests.

Getting More Information From Last_result

Each time RWSConnection sends a request and receives results, it keeps the result of the RWS call in its last_result attribute. This is very useful for debugging RWS calls, since it allows you to find out what headers were sent or what URL was called.

```
> from rwslib import RWSConnection
> from rwslib.rws_requests import VersionRequest
> rws = RWSConnection('innovate')
> # Get the rave version from rws
> rws.send_request(VersionRequest())
u'1.8.0'
> rws.last_result.url
https://innovate.mdsol.com/RaveWebServices/version
> rws.last_result.status_code
200
> rws.last_result.headers['content-type']
text/plain; charset=utf-8
> rws.last_result.text
u'1.8.0'
```

last_result is a Requests object - Please refer to the rwslib library for more information on all the returned properties.

Having access to last_result means that rwslib never hides its workings from you. You can use rwslib as a helper library to get your own integrations up and running.

Getting the Elapsed Time of the Request

Each time RWSCconnection sends a request and receives results, it keeps the elapsed time, in seconds, of the RWS call in its `request_time` attribute.

```
> from rwslib import RWSCconnection
> from rwslib.rws_requests import VersionRequest
> rws = RWSCconnection('innovate')
> # Get the rave version from rws
> rws.send_request(VersionRequest())
u'1.8.0'
> # Get the elapsed time in seconds to process the previous request
> rws.request_time
0.760736942291
```

Error Handling

RWS returns various error results, depending on the type of request. rwslib packages these error types into rwsobjects. RWSEexception exceptions have an `rws_error` property. The `rws_error` property is populated with a different object type, depending on the error type.

When RWS returns an XML error response, rwslib parses the error and returns it in an RWSError, rwsobjects, or rwsobjects.RWSErrorResponse.

RWSError instances have an `errordescription` attribute, while an RWSErrorResponse has an `errordescription` and a `reasoncode`.

You can use RWSError to parse ODM-formatted return messages:

```
<?xml version="1.0" encoding="utf-8"?>
<ODM xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata"
  FileType="Snapshot"
  CreationDateTime="2013-04-08T10:28:49.578-00:00"
  FileOID="4d1372a-ceb6-4419-a917-b6ad5d0bc30e"
  ODMVersion="1.3"
  mdsol:ErrorDescription="Incorrect login and password combination. [RWS00008]"
  xmlns="http://www.cdisc.org/ns/odm/v1.3" />
```

RWSErrorResponse parses simple XML return messages:

```
<Response
  ReferenceNumber="0b47fe86-542f-4070-9e7d-16396a5ef08a"
  InboundODMFileOID="Not Supplied"
  IsTransactionSuccessful="0"
  ReasonCode="RWS00092"
```

```
ErrorClientResponseMessage="CRF version not found">
</Response>
```

RWSEException also has a standard message attribute that copies the error description content from RWS. This allows rwslib to raise a standard exception type that surfaces the error message from the source RWS response, and at the same time provides full access to the content of the original RWS error message.

Handling XML Responses

- Addition of new XML elements or attributes will happen from time to time. Customer code should handle this gracefully. This should not be considered a breaking change.
- XML attributes order changes should be handled gracefully and should not be considered a breaking change.

1.1.1 About Rave Web Services (RWS)

Rave Web Services (RWS) integrates Medidata Rave with third-party systems to exchange CDISC ODM standard clinical data or metadata synchronously and with immediate confirmation.



Important:

- If you encounter intermittent problems when using RWS, then you must implement the appropriate retry logic.
- See [Rave Configuration FAQs](#) for more information on fields that exist in a form but do not appear in Clinical Views based reports and outputs. For example, as a SAS On-Demand report, Data Listing report, Standard Outputs, or RWS outputs.

Intended Audience

Before you start working with RWS, you must have:

- a good understanding of XML and methods for posting messages to specific URLs and
- an expert understanding of Rave study design and clinical data.

RWS Architecture

RWS uses the Representational State Transfer (REST) architecture. Data is posted to or retrieved from Rave using HTTP protocol messages posted to specific URLs. Each message receives an immediate success or failure response. If a failure occurs, any pending changes are rolled back. RWS uses RESTful web services and does not require the use of either Simple Object Access Protocol (SOAP) or Web Services Description Language (WSDL).



Note: The RWS version is the same as the Rave URL version that RWS is used within. The RWS version matches the Rave version exactly.

Features

RWS is a component of Rave, and supports the following features:

- Subject data (Clinical data) import
- Subject data (Clinical data) export
- Study designs (Metadata) import
- Study designs (Metadata) export
- Custom dataset export
- Transaction logging
- Error reporting.

For more information on RWS features, see the [Medidata Rave Releases History](#).

RWS uses vendor extensions to overcome some of the limitations of the ODM standard. See [ODM Schema](#) for more details.

The RWS Request Services

[Working with Subject Data](#)

[Working with Study and Library Metadata](#)

[Retrieving Clinical Study Lists](#)

[Retrieving Study Subject Lists](#)

[Retrieving Clinical View Datasets as ODM](#)

[Retrieving Clinical View Form Datasets as CSV](#)

[Retrieving Clinical Datasets Metadata as ODM](#)

[Retrieving Study and Library Metadata Lists](#)

[Retrieving Study and Library Metadata](#)

[Retrieving Custom Datasets](#)

Error Reporting

For every transaction, RWS returns a response. If there is an error, the error description and error location are provided in XPath notation. See [Error responses - the complete list](#).

RWS checks the request XML for validity and against the CDISC [ODM 1.3 schema](#) (with backward compatibility support to ODM 1.2 for clinical data, only). For more information on how to correctly construct the request message in ODM 1.3 XML, see [URIs](#).

 **Important:** RWS patient initialization requests may fail without a proper error message if an external system is called from a Rave Custom Function. When this issue occurs, it results in a corrupt subject.

Encoding

You must ensure that all documents used with RWS are encoded as [UTF-8](#).

Security

By default, RWS accepts transactions over HTTPS. RWS supports:

- [basic HTTP authentication](#) using local Rave accounts and
- application-to-application authentication with the [mAuth service](#).



Note: You must ensure that your network is configured to allow communication on port 443, the standard HTTPS port.



Important:

As of December 02 2017, Medidata is disabling older, vulnerable encryption algorithms and updating load balancers for Rave and Rave Web Services. As a result, any custom client application integrated with Rave Web Services must use newer encryption algorithms. If you are using up-to-date operating systems and have removed older encryption algorithms, you are not required to act.

If you use a third party, it is critical to let them know, as their environment probably supports multiple customers. You may need to adjust the configuration of any network equipment, like load balancers, in the environment which brokers the transmissions to Medidata. See below details for Microsoft products:

- [Protocol Testing Tool - Notes on Use](#)
- [How to restrict the use of certain cryptographic algorithms and protocols in Schannel.dll](#)

If you have any questions, comments, or concerns, contact the Medidata Customer Support team at: helpdesk@mdsol.com.



Note:

Medidata has updated the security protocols known as Transport Layer Security (TLS) and SSL Certificate Authority, which protect the privacy of information communicated over the Internet. This ensures that our software operates with current security standards.

Medidata provides two test Rave instances to verify compatibility with both TLS 1.2 and Entrust root certificates. These test instances do not contain any test data and are not capable of responding to RWS commands. You can use them to establish an HTTPS (TLS) connection to the web service. That constitutes a successful test.

- Credentials are provided to test API connectivity only.
- You can also use these URLs to test browser compatibility.

You can contact your Medidata representative for further details and also refer to [RWS Security Protocols FAQs](#).

Rave Role Permissions

RWS uses the Rave security model to determine what actions are permitted and what data can be accessed. The Rave user account used by an external system that calls RWS must have an associated role with relevant permissions. For clinical data entry, an associated role must include at least the Entry action. The role must also be set up with form and field restrictions for the relevant studies.

To perform all clinical data operations, the Rave user account used with RWS must have all the following permissions:

- CanInactivateForm,
- CanInactivateLogLine,
- CanReactivateForm,
- CreateSubject,
- ModifyPrimaryRecord,
- SubjectAdmin,
- UsePrimarySubjectName, and
- ViewSubjectsWithAnyStatus.

See [here](#) a full description of the Rave roles and permissions.

User Accounts and Permissions

- To make RWS requests, create a dedicated user account with the User Administration module in Rave and correct role permissions.
- To allow multiple logins, set the Rave Core Config file Allow Multiple Logins property. Otherwise, when Rave receives an RWS request from a user who is already logged in to the Rave EDC, the EDC session closes to permit Rave to process the RWS request. You can avoid this problem by using a dedicated RWS user account.

A dedicated user account created solely for RWS access does not require specific training. However, you must complete all the required training and fulfill all the prerequisites. This allows you to use the same user account both for RWS access and for logging in to Rave.

For details on how to create a Rave local account to access RWS, see [Creating and Modifying Users in Rave User Administration](#).

Caching

RWS uses the same caching mechanism as Rave, but maintains a separate cache. By default, the RWS cache is flushed every 24 hours.



Note:

- Some Rave configuration changes result in the Rave cache being flushed automatically. However, in these cases the RWS cache is not refreshed.
- Certain Rave configuration changes, such as those relating to role permissions, are not available immediately in RWS.

Use the following URL to flush the cache to immediately implement configuration changes in RWS.

```
GET https://{{host}}/RaveWebServices/WebService.aspx?CacheFlush
```

For example:

```
https://innovate.mdsol.com/RaveWebServices/WebService.aspx?CacheFlush
```

You can install larger studies on more than one server. In this case, you must repeat the CacheFlush request for each node. Consult your Medidata representative to confirm the number of nodes used in your study.

Subjects

Clinical data relates to a specified subject. You can only see studies and subjects for which you have been granted security access through your role associations in Rave.



Note: When you use the clinical data import service to create a subject (including adding the default matrix), you must ensure the Rave user account has the CreateSubject permission assigned to at least one of its associated roles.

Folders

To create folders, you must assign the user account SubjectAdmin permission to at least one of its associated roles.

Forms and Data Pages

To create forms or data pages, you must assign the user account SubjectAdmin permission to at least one of its associated roles.

See [Error responses - the complete list](#) for more information on the error response codes associated with subjects, folders, forms, and data pages.

URIs

The Uniform Resource Identifier (URI) can be broken down into three smaller components:

- host and scheme, for example, <https://innovate.mdsol.com/RaveWebServices>
- path, for example, </datasets/LogMessages.csv>
- query string, for example, [?start=2013-05-01](#)

RWS supports most characters in the path. Below is a list of characters that are not supported:

- %25 (the encoded form for %)
- & and %26
- *
- : and %3A



Note:

- When you are designing your study in conjunction with RWS, you must ensure that the project name, site number, folder, form, and field OIDs do not contain any of these unsupported characters.
- See additional information on [URL Special Character Escaping](#).

RWS supports almost all characters in the query string except for '&', for example, [?item=abc&def](#). This is because '&' is the URL parameter separator.

Using the TransactionType Attribute

RWS uses the optional ODM attribute TransactionType in Clinical Data POST requests. This indicates if a data item exists in a study or not, and the appropriate action to take. For example:

- the TransactionType Insert indicates that the data item does not exist and should be added;
- the TransactionType Remove indicates that the data item does exist and should be deleted from the study; and
- if the TransactionType does not match the status of the data item in the study (for example, if Insert is used with a data item that exists), RWS returns an error.

The value of the TransactionType is inherited by all the child elements, unless the child element has its own TransactionType attribute.

The TransactionType values that you can use with RWS are as follows:

Transaction Type	Description
Insert	Used to add a new data entity. If the data entity exists, RWS returns an error.
Update	Used to modify an existing data entity. If the data entity does not exist, RWS returns an error.
Upsert	Used to modify an existing data entity or to add a new data entity. The TransactionType Upsert is supported in a limited number of circumstances.
Remove	Used to make a data entity inactive. If the data entity does not exist, RWS returns an error. Inactive elements show in Rave EDC as strikethrough text. To reactivate an inactive data entity, use the TransactionType Insert.
Context	Used to assert whether the data entity exists in a certain state. This is used for updates that are only relevant when the specified state exists. Generally, Context is used with ItemData elements or with its child elements.

Potential Issues with Web Proxies

System configurations that use of a web proxy between your client machine and the Rave installation may return stale data, in particular in response to 'GET' requests. This may be due to data caching by the proxy.

You can force the proxy to return fresh data by using the HTTP Cache-Control header with the value no-cache for every request. The way this setting is implemented depends on the programming language you are using.

Using RWS with Rave Edit Checks

Edit checks on datapoint actions are triggered once per form in Rave EDC. In contrast, every time a clinical data import action is completed with RWS and a response is returned, Rave triggers the appropriate edit check action. RWS has no influence or effect on the content of the edit checks themselves.

1.1.2 RWS Authentication Methods

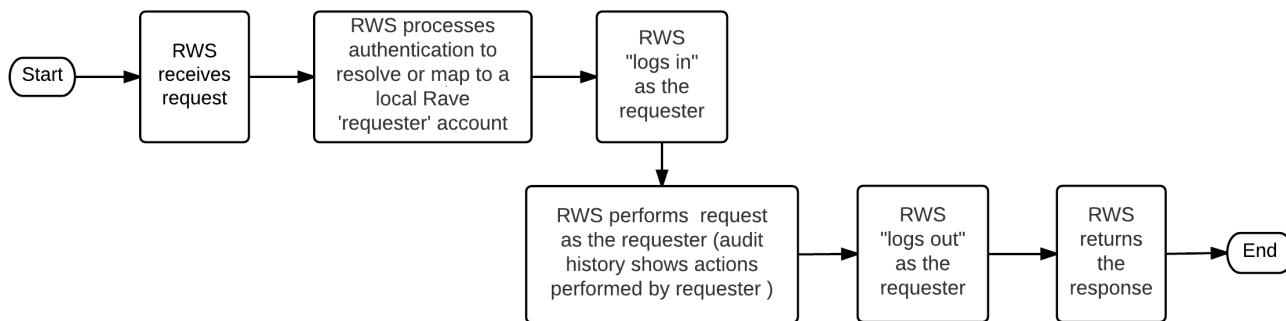
RWS supports the following authentication methods:

- Application-to-application authentication using the Medidata Authentication protocol.
- Basic HTTP authentication using local Rave accounts.

RWS requires that you authenticate every request. The Rave EDC user interface (UI) uses web sessions. Therefore, users can perform numerous tasks after successfully logging in, subject to the session-length restrictions.

RWS Request Authentication Process Flow

The following graphic displays the process that RWS uses to authenticate requests for data.



Dependencies

- The authentication process is independent of Rave role-based access control (RBAC). Authorization for access to protected resources requires that you have the appropriate role and access assigned through the [User Administration Module](#) in Rave.
- RWS records the audit history for each authentication action. The local Rave account performs this audit.

Medidata Authentication Protocol

Use the more secure and preferred [Medidata authentication protocol](#) on top of [HTTPS authentication](#). The easiest way to authenticate API requests is to use an [authentication client](#). See the [Medidata Authentication](#) video for more information about using these clients.



Important: If you use mAuth to authenticate Rave Web Services requests, you must use mAuth v1. We discuss this protocol below. See the [Medidata Authentication Protocol](#) documentation for more information.

Note that for v1, signing the body is not hashed before inclusion in the *string_to_sign* component. Instead, directly include the body contents in the *string_to_sign* that is eventually hashed.

In the case of an empty body, the trailing linefeed character should still be added to the *string_to_sign*.

The Medidata authentication protocol adds one layer of protection and eliminates the need to store your credentials.



Note: The Medidata Authentication Protocol:

- Takes precedence over HTTP basic authentication where you supply both.
- Does not support the slash ('/') character in the study name for data sets.

Set Up Medidata Authentication for RWS

1. If you want a new Rave study authenticated using the Medidata authentication protocol with RWS, tell your Medidata Rave project manager.
2. The Rave project manager requests support from Medidata Technical Services and instructs you on what to do to authenticate using RWS.



Note: As noted above:

- You cannot use an iMedidata account with RWS.
- To use RWS with the Medidata authentication protocol, you need a local Rave account to map the application to.
- A local account is required, since the application is mapped to a user on the URL.

Prerequisites

To succeed:

- I have a valid Rave account to map the application to.
- The local Rave account is not locked out.
- The local Rave account has the required permissions to perform my tasks.
- I have [registered my app and public key with the Medidata Authentication Protocol](#) and have an Application UUID issued by the protocol.
- My Rave URL is configured to support the Medidata Authentication Protocol.

Setting Up the Client

Send authenticated HTTP requests to request Medidata API resources.

1. Download and install the [authentication client](#).



Important: You can use your own client. See [the authentication protocol description](#). For more information about the Medidata authentication protocol, public/private key pairs, and mAuth clients, see the [Service Authentication](#) video.

2. Configure the client:

- **Private key file** – Where the private key is. See Generate Private-Public Key Pair section on the [Get Started with iMedidata Restful APIs](#) page.
- **app UUID** – Obtained when you register your API app.

3. Send an authenticated API request.

Use Case

I am authenticating my request using the Medidata authentication protocol.

C#

 **Note:** If this tab does not update when you click it, refresh your browser window.

C# Use Case Example

```
using System;
using System.Net.Http;
using System.Threading.Tasks;
using Medidata.MAuth.Core;

namespace DotnetApiCallExample
{
    class Program
    {
        static async Task Main(string[] args)
        {
            var mauthSigningHandler = new MAuthSigningHandler(
                new MAuthSigningOptions()
                {
                    ApplicationUuid = new Guid("Your application uuid goes here"),
                    PrivateKey = "Your private key contents or local path to the key
file goes here"
                }
            );
        }
    }
}
```

```
) ;

var client = new HttpClient(mauthSigningHandler);

var request = new HttpRequestMessage(HttpMethod.Get, "https://
api.mdsol.com/countries");

request.Headers.Add("Accept", "application/json");
request.Headers.Add("Mcc-version", "v2019-03-22");

var response = await client.SendAsync(request);

response.EnsureSuccessStatusCode();

Console.WriteLine("The response body is the following:");
Console.WriteLine($"{await response.Content.ReadAsStringAsync()}");

Console.WriteLine("\nPress ANY key to exit.");
Console.ReadKey();
}

}

}
```

Python

Python code using the Medidata authentication protocol

```
from requests_mauth import MAuth
from rwslib import RWSConnection
from rwslib.rws_requests import StudySubjectsRequest, ClinicalStudiesRequest
import json
import os

def load_config(filename):
    """
    Load the MAuth configuration
    Note: The configuration file is of the format
    ```

 {"APP_UUID": "d03b5130-5d25-4276-aaed-888888888888",
 "KEY_FILE": "hackathon1_next_nyc2018_mauth_PrivateKey.pem"}
    ```

    """
    print(f"Loading {filename}")
    if os.path.exists(filename):
        with open(filename, 'r') as fh:
            cfg = json.load(fh)
        return cfg
    else:
        raise ValueError("No such file")

def get_subjects(project_name, project_environment, urlname="hackathon1"):
```

```
"""
Get a list of the subjects for a given study environment
"""

# Load in the configuration
cfg = load_config(f"{urlname}.json")
# Create the MAuth Signer
auth = MAuth(
    app_uuid=cfg["APP_UUID"], private_key_data=open(cfg["KEY_FILE"]).read()
)
# Note: Using the MAuth object from the requests_mauth module make this very easy
client = RWSConnection(urlname, auth=auth)
# This is the same call as you would use with Basic Authentication
subjects = client.send_request(
    StudySubjectsRequest(
        project_name=project_name,
        environment_name=project_environment,
        subject_key_type="SubjectUUID",
    )
)
return subjects

def get_studies(urlname="hackathon1"):
    """
    Get a list of the Studies for the URL (this is predicated on the application
    being authorised in Rave)
    """

    cfg = load_config(f"{urlname}.json")
    # Create the MAuth Signer
    auth = MAuth(
        app_uuid=cfg["APP_UUID"], private_key_data=open(cfg["KEY_FILE"]).read()
    )
    # Note: Using the MAuth object from the requests_mauth module make this very easy
    client = RWSConnection(urlname, auth=auth)
    # This is the same call as you would use with Basic Authentication
    studies = client.send_request(ClinicalStudiesRequest())
    return studies

if __name__ == "__main__":
    studies = get_studies()
    for study in studies:
        print(f"Study: {study.oid}")
```

Basic HTTP Authentication

Basic HTTP authentication involves using a Rave account to send a user name and password in the HTTP request headers. All Rave services are available for use with valid Rave credentials that are local to the URL.

Use Case

I am authenticating my request using HTTP basic authentication.



Note: This is the most common method to authenticate a request.

Example: C# code using basic HTTP authentication.

```
using System.IO;
using System.Net;

namespace WriteOdmWebServiceDatasetToFile
{
    class Program
    {
        static void Main()
        {
            // Create request
            var myURLPath = "/studies/MyStudy/datasets/raw/MyDataset";
            var myRequestBody = String.Empty;
            var myHTTPVerb = "GET";
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create("https://
MyURL.mdsol.com/RaveWebServices" + myURLPath);

            // Set HTTP Basic authentication
            request.Credentials = new NetworkCredential("MyUserName",
"MyPassword");

            // Perform request
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();

            // Handle response
            Stream stream = response.GetResponseStream();
            StreamReader reader = new StreamReader(stream);
            StreamWriter writer = new StreamWriter("odm.xml");
            writer.Write(reader.ReadToEnd());
            writer.Close();
            reader.Close();
        }
    }
}
```

Prerequisites

To succeed:

- I have a valid Rave account.
- My Rave account is local and not managed by an external system such as iMedidata.
- My password has not expired and my account is not locked.
- My account has the required permission to perform the task.

Header

My HTTP Basic authentication request will set the standard HTTP Authorization header as follows:

```
Authorization: Basic <credentials>
```

where <credentials> is the base64 encoded concatenation of the Rave username and password using a ':' to separate the fields, that is:

```
{Base64.encode64 (my-userid:my-password) }
```

AttributeDescriptionNotes

{my-userid}	My Rave account Login
{my-password}	My Rave account Password

Scenarios

Background:

I am a valid user

Scenario Overview: Unauthorized access to services is denied

I am using <invalid credentials>

When I make a request with my username and password authenticated using HTTP Basic Authentication

I should expect to receive an "Unauthorized" response code

Examples:

invalid credentials	
a blank username	
a username that doesn't exist	
a blank password	

```
| an incorrect password          |
| a username that doesn't exist and is too long |
| an incorrect password that is too long          |
```

Scenario: Response when user account is locked

I am a valid user, however, my Rave account is locked

When I make a request with my username and password authenticated using HTTP Basic Authentication

I should expect to receive an HTTP response with a status code of "Unauthorized"
And reason code "RWS00005"

The expected response should typically read; "User is locked out"

1.1.3 RESTClient Browser Add-On

You can use the RESTClient browser add-on to send HTTP requests, using all the available HTTP methods (GET, POST, PUT, DELETE, OPTIONS, HEAD, TRACE, and CONNECT).

- RESTClient home page: <http://restclient.net/>.
- The Firefox add-on (version 2.0.3) supports basic authentication (suitable for RWS), oAuth signing, custom headers, and various other features. The HTTP response is separated into tabs for the header and the content.
- The Firefox add-on is available at: <https://addons.mozilla.org/en-US/firefox/addon/restclient/>
- RESTClient has become the tool of choice for the RWS development team to test functionality.

Features

- Make a web service request to any URI such as RWS.
- Supports HTTP methods: GET, POST, PUT, DELETE, OPTIONS, HEAD, TRACE, CONNECT.
- Add custom headers to the request such as Content-Type = text/xml.
- View the HTTP headers from the response.
- View the HTTP response, if any, from the URI request.
- None, basic and oAuth authentication methods.

Accessibility

You can access RestClient as a Firefox add-on, Chrome extension, or Safari extension. The source is available on GitHub.

The following graphic illustrates how the RestClient presents in Firefox:

The screenshot shows the RESTClient interface. At the top, there are tabs for File, Authentication, Headers, and View. On the right, there are buttons for Favorite Requests, Setting, and a large red button labeled "RESTClient". Below the tabs, under the "Request" section, the method is set to POST, the URL is https://innovate.mdsol.com/RaveWebServices/webservice.aspx?PostODMClinicalData, and there is a "SEND" button. Under "Headers", there are two entries: Authorization: Basic ZGV2OnBhc3N... and Content-Type: text/xml. Under the "Body" section, the XML payload is shown:

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3" FileType="Transactional" FileOID="Example-1" CreationDateTime="2008-01-01T00:00:00">
<ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">
<ODM>
```

1.1.4 Sample Code for RWS

The examples on this page illustrate the use of Rave Web Services (RWS) requests in different coding languages.

- Every request to RWS generates an HTTP response code, for example "200", or "404".
- The "200" HTTP response code signifies a successful outcome. All other HTTP response codes signify a failure.
- Where the service supports response body, this will contain details of the request outcome. See [Error Responses - the complete list](#).
- A useful testing tool is the [RESTClient browser add-on](#).

[Sample Code for Working with Clinical Data](#)

C# Code for Updating Clinical Data

The XML included in this code snippet performs a simple update.

```
using System;
using System.Net;
using System.IO;

namespace Medidata.Integrations.RaveWebServices
{
    class ExampleClinicalDataUpdate
    {
        [STAThread]
        static void Main()
        {
            // ODM dates must be in ISO 8601 format, Rave dates must be in the field
format.
            String now = DateTime.UtcNow.ToString("s");
            string subjectName = "123 ABC", siteNumber = "4567";
            string birthDate = "01 JAN 1980", gender = "MALE", race = "3";
            string server = "innovate.mdsol.com", username = "myuser", password =
"mypassword";
            string odmTemplate = @"<?xml version="1.0" encoding="utf-8" ?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"
FileType="Transactional"
FileOID="N/A" CreationDateTime="{0}">
    <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">
        <SubjectData SubjectKey="{1}" TransactionType="Update">
            <SiteRef LocationOID="{2}"></SiteRef>
            <StudyEventData StudyEventOID="SCREEN">
                <FormData FormOID="DM">
                    <ItemGroupData ItemGroupOID="DM">
                        <ItemData ItemOID="BIRTHDTC" Value="{3}"/>
                        <ItemData ItemOID="SEX" Value="{4}"/>
                        <ItemData ItemOID="RACE" Value="{5}"/>
                    </ItemGroupData>
                </FormData>
            </StudyEventData>
        </SubjectData>
    </ClinicalData>
</ODM>";
            // This will break if any values inserted into the template contain un-
escaped
            // special characters.
            String odm = String.Format(odmTemplate, now, subjectName, siteNumber,
birthDate, gender, race);
            try
            {
                SendRequestToRave(odm, server, username, password);
                Console.WriteLine("SUCCESS");
            }
        }
    }
}
```

```
        catch (Exception exception)
        {
            Console.WriteLine("FAILURE: {0}", exception.Message);
        }
        Console.ReadLine();
    }

    static private void SendRequestToRave(string odm, string server, string
username, string password)
    {
        string url = String.Format("https://{0}/RaveWebServices/WebService.aspx?
PostODMClinicalData",
            server);
        HttpWebRequest request = (HttpWebRequest) WebRequest.Create(url);
        // Pass the username and password to the web service using HTTP Basic
Authentication.
        Request.Credentials = new NetworkCredential(username, password);
        request.Method = "POST";
        request.ContentType = "text/xml";
        using (StreamWriter requestWriter = new
StreamWriter(request.GetRequestStream()))
        {
            requestWriter.Write(odm);
        }
        try
        {
            using (HttpWebResponse response = (HttpWebResponse)
request.GetResponse())
            {
                WriteResponseToConsole(response.GetResponseStream());
            }
        }
        catch (WebException exception)
        {
            WriteResponseToConsole(exception.Response.GetResponseStream());
            throw exception;
        }
    }

    static private void WriteResponseToConsole(Stream responseStream)
    {
        using (StreamReader responseStreamReader = new
StreamReader(responseStream))
        {
            Console.WriteLine(responseStreamReader.ReadToEnd());
            Console.WriteLine();
        }
    }
}
```

Sample C# Code for Retrieving CV Datasets as ODM

How to use this example:

- Replace MyURL, MyStudy and MyDataset with your URL, study name, and dataset name.
- Replace MyUserName and MyPassword with an appropriate user name and password.



Note: RWS writes the response to the file **odm.xml** in the working directory.

```
using System.IO;
using System.Net;

namespace WriteOdmWebServiceDatasetToFile
{
    class Program
    {
        static void Main()
        {
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create("http://
MyURL.mdsol.com/RaveWebServices/studies/MyStudy/datasets/raw/MyDataset");
            request.Credentials = new NetworkCredential("MyUserName",
"MyPassword");
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            Stream stream = response.GetResponseStream();
            StreamReader reader = new StreamReader(stream);
            StreamWriter writer = new StreamWriter("odm.xml");
            writer.Write(reader.ReadToEnd());
            writer.Close();
            reader.Close();
        }
    }
}
```

Erlang Code for Updating Clinical Data

The XML in this code snippet performs a simple update.

```
-module(rave_web_services).
-export([example1/0]).

Post_odm_clinical_data(Protocol,Domain,URL,Data,Username,Password) ->
{ ok, {{_A,HTTPResponseCode,_C}, _Headers, _Body }} = http:request(
```

```

post,
{
    Protocol ++ Domain ++ URL,
    [{"Authorization", "Basic " ++ base64:encode_to_string(Username ++ ":" ++
Password)},{ {"content-type", "text/xml"}],
    "text/xml",
    Data
},
[],
[]
),
[HTTPResponseCode].
Example1() ->
Protocol="https://",
Domain="innovate.mdsol.com",
URL="/Ravewebservices/WebService.aspx?PostODMClinicalData",
Username="myuser",
Password="mypassword",
SubjectKey="100 ABC",
LocationOID="MDSOL",
DEMO_BIRTHDT="16 Jul 1978",
DEMO_SEX="MALE",
DEMO_RAVE="3",
DEMO_COUNTRY="GBR",

ExampleODM="<?xml version="1.0" encoding="UTF-8" ?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"
FileType="Transactional" FileOID="NA"
    CreationDateTime="2008-06-06T12:33:35">
<ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">
    <SubjectData SubjectKey=''' ++ SubjectKey ++ ''' TransactionType="Update">
        <SiteRef LocationOID=''' ++ LocationOID ++ ''' />
        <StudyEventData StudyEventOID="SCREEN">
            <FormData FormOID="DM">
                <ItemGroupData ItemGroupOID="DM">
                    <ItemData ItemOID="BIRTHDT" Value=''' ++ DEMO_BIRTHDT ++ ''' />
                    <ItemData ItemOID="SEX" Value=''' ++ DEMO_SEX ++ ''' />
                    <ItemData ItemOID="RACE" Value=''' ++ DEMO_RAVE ++ ''' />
                    <ItemData ItemOID="COUNTRY" Value=''' ++ DEMO_COUNTRY ++ ''' />
                </ItemGroupData>
            </FormData>
        </StudyEventData>
    </SubjectData>
</ClinicalData>
</ODM>",

ssl:start(),
inets:start(),
HTTPResponseCode =

```

```
post_odm_clinical_data(Protocol,Domain,URL,ExampleODM,Username,Password),  
    io:fwrite(~w~n,[HTTPResponseCode]),  
    inets:stop(),  
    ssl:stop().
```

Java Code (Apache HttpClient) for Updating Clinical Data

This Java example uses the Apache HttpClient class library to connect as a client to RWS.

```
/*  
 * SimpleMethodClient.java  
  
 * Description:  
  
 * This simple Java class uses the Apache HttpClient class library to connect as a  
 * client to the  
 * Medidata Rave Web Service (Inbound) over SSL and enables an ODM 1.3.x data file to  
 * be loaded  
 * into Rave versions 5.6.3.  
  
 * Build Details:  
  
 * Requires the Apache class libraries,  
 * commons-codec-1.3.jar  
 * commons-httpclient-3.1.jar  
 * commons-logging-1.1.1.jar  
  
 * Compiled using Sun SDK EE5 update 5  
  
 * NOTE: Modify the user and host details accordingly before compiling.  
  
 * Usage:  
 * Simply run from the command line after successfully compiling.  
 * The program expects to find an ODM 1.3 data file named 'odm13data.xml' in the  
 * path.  
  
 * Copyright Medidata Solutions 2008  
 */  
package simpleexample.client;  
  
import java.io.File;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.BufferedReader;  
  
import org.apache.commons.httpclient.UsernamePasswordCredentials;  
import org.apache.commons.httpclient.Credentials;  
import org.apache.commons.httpclient.auth.AuthScope;
```

```
import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.HttpStatus;
import org.apache.commons.httpclient.HttpVersion;
import org.apache.commons.httpclient.methods.PostMethod;
import org.apache.commons.httpclient.methods.FileRequestEntity;
import org.apache.commons.httpclient.methods.RequestEntity;

/**
 * SimpleMethodClient class
 * Acts as a simple web service client to enable loading of ODM 1.3 data into Rave
 */
public class SimpleMethodClient
{
    /* Modify the username, password, hostname and odmXMLFileName accordingly */
    final static String username = "username";
    final static String password = "password";
    final static String hostname = "host.company.com";
    final static String odmXMLFileName = "odm13data.xml";

    /* Main method to connect, read the data file and upload via the web service */
    public static void main(String[] args) throws Exception
    {
        String request = "https://" + hostname + "/Ravewebservices/WebService.aspx?
PostODMClinicalData";

        HttpClient client = new HttpClient(); // Apache's Http client
        Credentials credentials = new UsernamePasswordCredentials(username, password);

        client.getState().setCredentials(AuthScope.ANY, credentials);
        client.getState().setProxyCredentials(AuthScope.ANY, credentials); // may not
be necessary

        client.getParams().setAuthenticationPreemptive(true); // send authentication
details in the header

        PostMethod post = new PostMethod(request); // uses HTTP POST
        post.setDoAuthentication( true ); // must authenticate
        post.getParams().setParameter("http.protocol.version", HttpVersion.HTTP_1_1);

        File f = new File(odmXMLFileName);
        RequestEntity entity = new FileRequestEntity(f, "text/xml");
        post.setRequestEntity(entity);

        try
        {
            int statusCode = client.executeMethod(post); // send POST request with data
            if (statusCode != HttpStatus.SC_OK)
            { // did it work ?
                System.err.println("Method failed: " + post.getStatusLine());
            }
        }
    }
}
```

```
// Process the response from Rave Web Services
InputStream rstream = post.getResponseBodyAsStream();
BufferedReader br = new BufferedReader(new InputStreamReader(rstream));
String line;
while ((line = br.readLine()) != null)
{
    System.out.println(line); // just display the response
}
br.close();

}
finally
{
    post.releaseConnection(); // remember to free up the connection
}
}
}
```

Java Code (Sun JAX-WS) for Updating Clinical Data

This Java example uses the Sun JAX-WS class library to connect as a client to RWS.

```
/*
```

SimpleMethodClient.java

Description:

This simple Java class uses the Sun JAX-WS class library to connect as a client to the

Medidata Rave Web Service (Inbound) and enables an ODM 1.3.x data file to be loaded into Rave versions 5.6.3.

Build Details:

Requires the Sun EE5 JAX-WS class libraries (in tools.jar),

Compiled using Sun SDK EE5 update 5

NOTE: Modify the user and host details accordingly before compiling.

Usage:

Simply run from the command line after successfully compiling.

The program expects to find an ODM 1.3 data file named 'odm13data.xml' in the path.

```
Copyright Medidata Solutions 2008
*/
package simpleexample.client;

import javax.xml.ws.*;
import javax.xml.ws.http.*;
import javax.xml.ws.handler.*;
import javax.xml.namespace.Qname;
import javax.xml.transform.*;
import javax.xml.transform.stream.*;
import java.io.File;
import java.util.*;

/**
 * SimpleMethodClient class
 * Acts as a simple web service client to enable loading of ODM 1.3 data into Rave
 */
public class SimpleMethodClient
{
    /* Modify the username, password, hostname and odmXMLFileName accordingly */
    final static String    username = "username";
    final static String    password = "password";
    final static String    hostname = "host.company.com";
    final static String    odmXMLFileName = "odm13data.xml";

    /* Main method to connect, read the data file and upload via the web service */
    public static void main  (String[] args)
    {
        QName qname    = new QName("",    "");
        String url = "http://"+hostname+"/Ravewebservices/WebService.aspx?
PostODMClinicalData";
        Service    service    = Service.create(qname);
        service.addPort(qname, HTTPBinding.HTTP_BINDING, url);
        Dispatch<Source> dispatcher    = service.createDispatch(qname, Source.class,
Service.Mode.MESSAGE);
        Map<String, Object>    requestContext = dispatcher.getRequestContext();
        requestContext.put(MessageContext.HTTP_REQUEST_METHOD, "POST");
        requestContext.put(BindingProvider.USERNAME_PROPERTY, username);
        requestContext.put(BindingProvider.PASSWORD_PROPERTY, password);

        Source result = null;
        File f = new File(odmXMLFileName);
        try
        {
            result = dispatcher.invoke(new StreamSource(f)); // post the xml
            printSource(result); // display the result
        }
        catch (Exception x)
        {
```

```
        System.out.printf ("Caught Exception: %s\n", x.toString());
        if (result != null)    printSource(result);
    }
}
/*
PrintSource
Helper method to output the results of the post

@param Source s the result source to read from

*/
public static void printSource(Source s)
{
    try
    {
        System.out.println("\n===== Response Received
=====");
        TransformerFactory factory = TransformerFactory.newInstance();
        Transformer transformer = factory.newTransformer();
        transformer.transform(s, new StreamResult(System.out));

        System.out.println("\n=====");
    }
    catch (Exception e)
    {
        System.out.println(e);
    }
}
}
```

Python Code for Updating Clinical Data

The XML in this code snippet performs a simple update.

```
import urllib2

import datetime

domain = 'innovate.mdsol.com'
webservice = '/Ravewebservices/WebService.aspx?PostODMClinicalData'
protocol = 'https://'
username = 'myuser'
password = 'mypassword'

xml='''<?xml version="1.0" encoding="utf-8" ?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"
```

```

FileType="Transactional" FileOID="1" CreationDateTime="%{creationdatetime}s">
<ClinicalData StudyOID="Mediflex (Dev)" MetaDataVersionOID="1">
  <SubjectData SubjectKey="%{subjectkey}s" >
    <SiteRef LocationOID="%{locationoid}s"></SiteRef>
    <StudyEventData StudyEventOID="SCREEN">
      <FormData FormOID="DM">
        <ItemGroupData ItemGroupOID="DM">
          <ItemData ItemOID="BRTHDTC" Value="%{birthdt}s"/>
          <ItemData ItemOID="SEX" Value="%{gender}s"/>
          <ItemData ItemOID="RACE" Value="%{race}s"/>
          <ItemData ItemOID="COUNTRY" Value="%{country}s"/>
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
</ODM>'''
```

```

data = xml % \
  {'creationdatetime' : datetime.datetime.now().isoformat() ,
   'subjectkey'       : '100 ABC' ,
   'locationoid'      : 'MDSOL' ,
   'birthdt'          : '16 Jul 1978' ,
   'gender'           : 'MALE' ,
   'race'             : '3' ,
   'country'          : 'GBR' }
```

```

# Create a password manager and add authentication details for the Medidata Rave realm
passman = urllib2.HTTPPasswordMgr()
passman.add_password('Medidata Rave', domain, username, password)
# Create an authentication handler using the password manager
authhandler = urllib2.HTTPBasicAuthHandler(passman)
# Create an opener using the authentication handler and install it
opener = urllib2.build_opener(authhandler)
urllib2.install_opener(opener)
# Create request object
req = urllib2.Request(protocol + domain + webservice,data)
# Add content type header
req.add_header("Content-type", "text/xml")

# Make the request and print any errors
try:
  f = urllib2.urlopen(req)
  print f.read()
except IOError, e:
  if hasattr(e, 'code'):
```

```
print e.code
print e.read()
```

Ruby Code for Updating Clinical Data <sample-code-ruby-clinical-post>

The XML included in this code snippet performs a simple update.

```
require 'net/http'

require 'net/https'
require 'time'

def send_request_to_rave(odm, server, username, password)
  Net::HTTP.start(server, Net::HTTP.https_default_port) do |http|
    path = "/RaveWebServices/WebService.aspx?PostODMClinicalData"
    request = Net::HTTP::Post.new(path)
    request.basic_auth(username, password)
    request.content_type = "text/xml"
    request.body = odm

    response = http.request(request)
    puts response.body
    response.value
  end
end

odm_template = <<<-XML
<?xml version="1.0" encoding="utf-8" ?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"
FileType="Transactional" FileOID="N/A" CreationDateTime="%s">
  <ClinicalData StudyOID="Beproxen(Dev)" MetaDataVersionOID="N/A">
    <SubjectData SubjectKey="%s" TransactionType="Update">
      <SiteRef LocationOID="%s"></SiteRef>
      <StudyEventData StudyEventOID="SCREEN">
        <FormData FormOID="DEMOG">
          <ItemGroupData ItemGroupOID="N/A">
            <ItemData ItemOID="DEMO_BIRTHDT" Value="%s"/>
            <ItemData ItemOID="DEMO_GENDER" Value="%s"/>
            <ItemData ItemOID="DEMO_RACE" Value="%s"/>
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
</ODM>
XML
```

```
odm = odm_template % [
  Time.now.iso8601,
  "AK005",
  "RwsTestSite1",
  "16 Jul 1978",
  "Male",
  "Asian"
]

begin
  send_request_to_rave(odm, "innovate.mdsol.com", "myusername", "mypassword")
  puts "SUCCESS"
rescue
  puts "FAILURE: #{$!}"
end
```

Sample Code for Retrieving Clinical View (CV) Datasets as ODM

C# Code for Retrieving CV Datasets as ODM

How to use this example:

- Replace MyURL, MyStudy and MyDataset with your URL, study name, and dataset name.
- Replace MyUserName and MyPassword with an appropriate user name and password.



Note: RWS writes the response to the file **odm.xml** in the working directory.

```
using System.IO;
using System.Net;

namespace WriteOdmWebServiceDatasetToFile
{
    class Program
    {
        static void Main()
        {
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create("http://
MyURL.mdsol.com/RaveWebServices/studies/MyStudy/datasets/raw/MyDataset");
            request.Credentials = new NetworkCredential("MyUserName",
"MyPassword");
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            Stream stream = response.GetResponseStream();
            StreamReader reader = new StreamReader(stream);
```

```
        StreamWriter writer = new StreamWriter("odm.xml");
        writer.Write(reader.ReadToEnd());
        writer.Close();
        reader.Close();
    }
}
}
```

Command Line Example Using cURL Tool to Retrieve CV Datasets as ODM

cURL is a free, open source tool available from: <http://curl.haxx.se/>

How to use this example:

- Replace MyURL, MyStudy and MyDataset with your URL, study name, and dataset name.
- Replace MyUserName with an appropriate user name.

The program prompts for a password, then writes the response to the “**odm.xml**” file in the working directory.

```
> curl --user MyUserName --output odm.xml http://MyURL.mdsol.com/RaveWebServices/
studies/MyStudy/datasets/raw/MyDataset
```

Sample Android RWS Post Code

- [Android app sample overview](#)
- [Android POST RWS code](#)

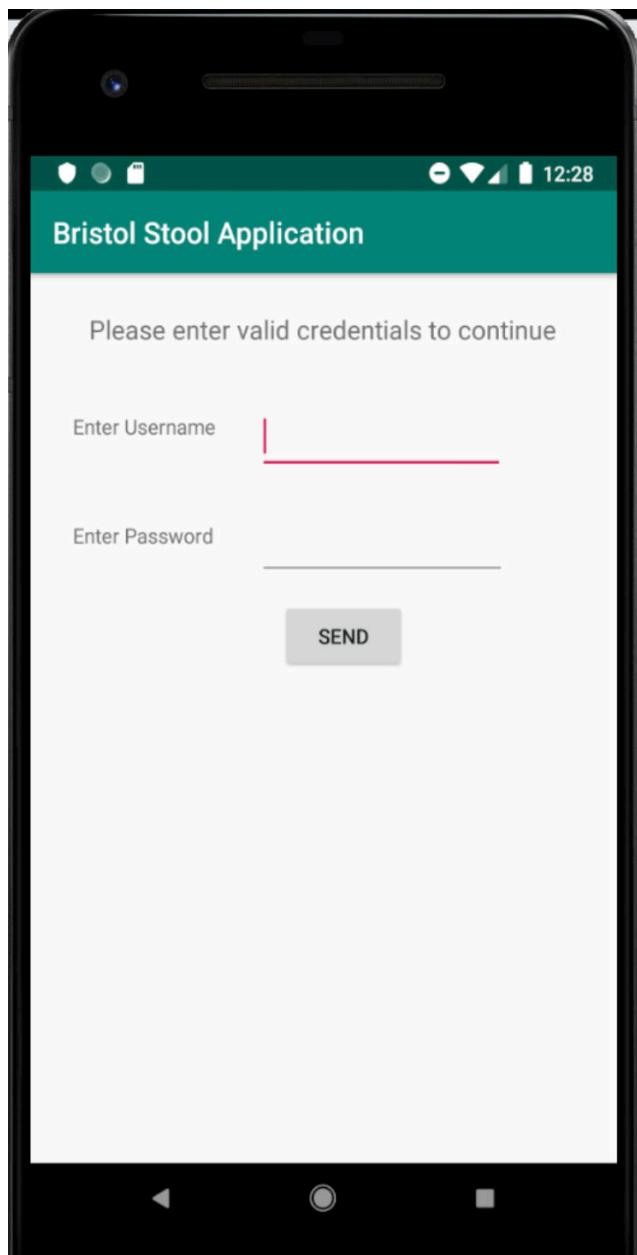
1.1.4.1 Sample Android App and RWS PostMan Code

Android App Overview

This example shows how RWS code used to POST data into a Rave EDC study can be adapted to fit into an Android app in order to perform the same purpose. The sample Rave EDC study AnsteyBSC has been specially built to contain a form called Bristol Stool Chart that contains the user selections for an mHealth Android app called the Bristol Stool Chart Application. This app is a proof of concept (PoC) to show how Android could be used to send data from a portable device into Rave using a standard medical scale for assessing bowel health and receive feedback from Rave related to the success or failure of the transaction.

The [Open Source PostMan tool](#) is used to show how the initial RWS POST code works before it is applied to the Android app. Afterward, you can [see how the extra Android code wraps around this standard RWS code to make the same call within the app](#). This first basic explanation of the sequence of events helps determine what happens in the Android app with respect to data transfer to Rave.

The Android app has a sign-in screen that allows the Rave user to apply their Rave EDC account name and password which is subsequently used to use the Rave basic authentication mechanism for RWS in order to push the data into Rave.



This Android app is only a PoC and includes certain hard coded form name, subject number, and so on. Below is an RWS POST example that inserts data points into the AnsteyBSC Rave study on a URL called `Hackathon2.mdsol.com` using PostMan. You can see that the HTTP Post action was successful by reading the lower section of the POST example:

POST https://hackathon2.mdsol.com/RaveWebServices/webservice.aspx?PostODMClinicalData

Send Save

Params Authorization Headers (11) Body Pre-request Script Tests Cookies Code Comments

none form-data x-www-form-urlencoded raw binary GraphQL BETA XML (text/xml) Beautify

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <ODM CreationDateTime="2019-02-18T00:00:00" FileOID="hackathon2_post_xml" FileType="Transactional" ODMVersion="1.3" xmlns="http://www
   .cdisc.org/ns/odm/v1.3">
3  <ClinicalData MetaDataVersionOID="55" StudyOID="AnsteyBSC(Prod)">
4  <SubjectData TransactionType="Update" SubjectKey="1003">
5  <SiteRef LocationOID="186742"/>
6  <StudyEventData TransactionType="Update" StudyEventRepeatKey="1" StudyEventOID="SCREEN">
7  <FormData TransactionType="Update" FormOID="BSC">
8  <ItemGroupData TransactionType="Update" ItemGroupOID="BSC">
9  <ItemData Value="Severe Constipation" ItemOID="BSC.TYPE1"/>
10 <ItemData Value="Mild Constipation" ItemOID="BSC.TYPE2"/>
11 <ItemData Value="Normal" ItemOID="BSC.TYPE3"/>
12 <ItemData Value="Normal" ItemOID="BSC.TYPE4"/>
13 <ItemData Value="Lacking Fibre" ItemOID="BSC.TYPE5"/>
14 <ItemData Value="Mild Diarrhea" ItemOID="BSC.TYPE6"/>
15 <ItemData Value="Severe Diarrhea" ItemOID="BSC.TYPE7"/>
16 </ItemGroupData>
17 </FormData>
18 </StudyEventData>
19 </SubjectData>
20 </ClinicalData>
21 </ODM>

```

Status: 200 OK Time: 1579ms Size: 597 B | Save Response

Body Cookies (1) Headers (13) Test Results

Pretty Raw Preview XML

Response

```

1 ReferenceNumber="6955b972-dd21-4df9-bb0f-21b6ca1163ad"
2 InboundODMFileOID="hackathon2_post_xml"
3 IsTransactionSuccessful = "1"
4 SuccessStatistics="Rave objects touched: Subjects=0; Folders=0; Forms=0; Fields=7; LogLines=0"
5 NewRecords=""</Response>

```

Below is a screenshot of the Rave Bristol Stool Chart form into which the data from the action above is inserted:

The screenshot shows the Rave web interface for the Bristol Stool Chart. The subject is 1003. The chart lists seven stool types with their corresponding constipation levels:

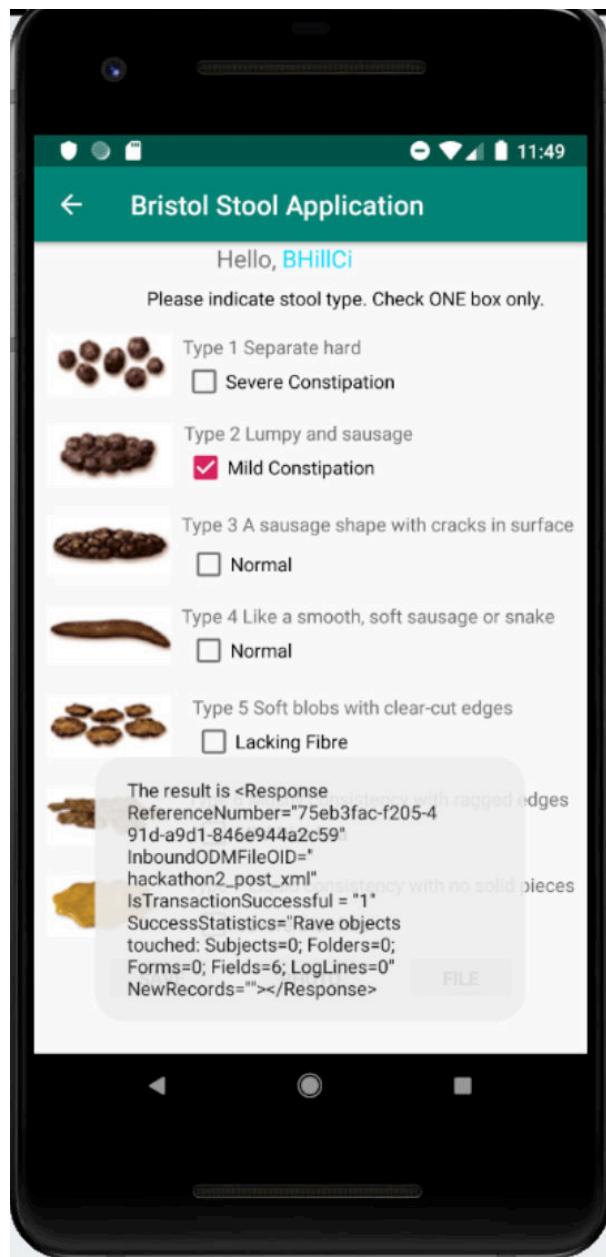
- 1 Stool Type 1: SEVERE CONSTIPATION
- 2 Stool Type 2: MILD CONSTIPATION
- 3 Stool Type 3: NORMAL
- 4 Stool Type 4: NORMAL
- 5 Stool Type 5: LACKING FIBRE
- 6 Stool Type 6: MILD DIARRHEA
- 7 Stool Type 7: SEVERE DIARRHEA

Each row has a 'Verify' button. The left sidebar shows the subject's visit history and other forms like Medical History and Vital Signs Screening.

The image below shows the Android emulator displaying a Bristol Stool Chart mHealth app screen. This Android app helps you enter a selection in this screen and have it transmitted to the AnsteyBSC Rave EDC study Bristol Stool Chart form and populate the corresponding record.



Note: All the entries previously populated in the PostMan RWS Post action are overwritten by this Android selection with just the Mild Constipation selection having an entry. Also note the success message informing you that the data has been sent from Android to Rave.

[Back to top](#)

RWS PostMan Code

Below is the code for the standard RWS transaction as displayed in PostMan in order to send data to Rave. The RWS call is:

```
https://hackathon2.mdsol.com/RaveWebServices/webservice.aspx?PostODMClinicalData
```

RWS PostMan Body

```
<?xml version="1.0" encoding="UTF-8"?>
<ODM CreationDateTime="2019-02-18T00:00:00" FileOID="hackathon2_post_xml"
FileType="Transactional" ODMVersion="1.3" xmlns="http://www.cdisc.org/ns/odm/v1.3">
<ClinicalData MetaDataVersionOID="55" StudyOID="AnsteyBSC(Prod)">
<SubjectData TransactionType="Update" SubjectKey="1003">
<SiteRef LocationOID="186742"/>
<StudyEventData TransactionType="Update" StudyEventRepeatKey="1"
StudyEventOID="SCREEN">
<FormData TransactionType="Update" FormOID="BSC">
<ItemGroupData TransactionType="Update" ItemGroupOID="BSC">
<ItemData Value="Severe Constipation" ItemOID="BSC.TYPE1"/>
<ItemData Value="Mild Constipation" ItemOID="BSC.TYPE2"/>
<ItemData Value="Normal" ItemOID="BSC.TYPE3"/>
<ItemData Value="Normal" ItemOID="BSC.TYPE4"/>
<ItemData Value="Lacking Fibre" ItemOID="BSC.TYPE5"/>
<ItemData Value="Mild Diarrhea" ItemOID="BSC.TYPE6"/>
<ItemData Value="Severe Diarrhea" ItemOID="BSC.TYPE7"/>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>
```

The authorization is the user name and password entered manually for user BHillCI on the Medsym Hackathon2 URL.

[Back to top](#)

1.1.4.2 Android POST RWS Code

The Android RWS code detailed below is equivalent to the [RWS PostMan code](#).

Perform the following step:

1. [Pass in the user credentials](#) for the [RWS basic authentication](#) from the login screen.
2. [Perform an Android HTTP asynchronous class object](#). You must run this code in a separate thread, hence the asynchronous call.
3. [Create the URL](#) as shown in the performPostCall method including the URL call itself, parameters, basic authorization, the body and an error message indicating success or failure.
4. [Call all of the methods 1-3 in a single method](#) that runs in the onCreate method of the Android Activity.

Get Rave User Credentials

1. Capture the data from two text objects for name and password on an opening Android screen.
2. Save the data to an Intent using a message.
3. Save the name and password entered to a new text variable called raveUserNameAnd Password, as shown below:

Get the Intent that started this activity and extract the string

```
// Get the Intent that started this activity and extract the string
Intent intent = getIntent();
String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

//Use the credentials entered for Rave authentication and display username
raveUserNameAndPassword = message;
```

[Back to top](#)

Call an Asynchronous Thread

As of Android version 3.3.4, you cannot call a web service from the current user thread. Hence, you need to launch a thread in the background in order to run the RWS HTTP call. This is what the **HTTPAsync** class does by running the **performPostCall** URL in a **doInBackground** method, as shown below:

HTTPAsync class

```
private class HTTPAsync extends AsyncTask<String, Void, String>{

    @Override
    protected void onPreExecute() {}

    @Override
    protected String doInBackground(String... params) {

        //Send checkBox info to an example PHP webservice
        String result = performPostCall("https://hackathon2.mdsol.com/RaveWebServices/
webservice.aspx?PostODMClinicalData");

        return result;
    }

    @Override
    protected void onProgressUpdate(Void... values) {}

    @Override
```

```
protected void onPostExecute(String result) {  
  
    Toast.makeText(DisplayMessageActivity.this, "The result is "+ result,  
    Toast.LENGTH_LONG).show();  
  
    if (result != null && !result.equals("")){  
        try {  
            JSONObject resObject = new JSONObject(result);  
        } catch (Exception e){  
            e.printStackTrace();  
        }  
    }  
}  
}
```

[Back to top](#)

Perform POST Call Method

The performPostCall method is where the URL call, parameters, authorization, body and the resultant error state itself is combined and passed back as a complete response object to the calling Asynch object. The body contains the answers that the user of the app entered and that becomes the input to the corresponding fields in the Rave EDC.

performPostCall method

```
public String performPostCall(String requestURL) {  
  
    URL url;  
    String response = "";  
    try {  
        url = new URL(requestURL);  
  
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
        conn.setReadTimeout(15000);  
        conn.setConnectTimeout(15000);  
        conn.setRequestMethod("POST");  
        conn.setDoInput(true);  
        conn.setDoOutput(true);  
        conn.setRequestProperty("Content-Type", "text/xml; charset=UTF-8");  
        final String basicAuth = "Basic " +  
        Base64.encodeToString(raveUserNameAndPassword.getBytes(), Base64.NO_WRAP);  
        conn.setRequestProperty("Authorization", basicAuth);  
  
        //Make a manual call to the RWS Subject Service in in order to get the  
        correct LocationId  
        String postBody = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +
```

```
"<ODM CreationDateTime=\"2019-03-12T00:00:00\"  
FileOID=\"hackathon2_post_xml\" FileType=\"Transactional\" ODMVersion=\"1.3\"  
xmlns=\"http://www.cdisc.org/ns/odm/v1.3\">\n" +  
    "<ClinicalData MetaDataVersionOID=\"55\" StudyOID=\"AnsteyBSC\">\n" +  
        "<SubjectData TransactionType=\"Update\" SubjectKey=\"1003\">\n" +  
            "<SiteRef LocationOID=\"186742\"/>\n" +  
            "<StudyEventData TransactionType=\"Update\" StudyEventRepeatKey=\"1\"  
StudyEventOID=\"SCREEN\">\n" +  
                "<FormData TransactionType=\"Update\" FormOID=\"BSC\">\n" +  
                    "<ItemGroupData TransactionType=\"Update\" ItemGroupOID=\"BSC\">\n" +  
                        "<ItemData Value=\""+ checkBoxStr1 +"\" ItemOID=\"BSC.TYPE1\"/>\n" +  
                        "<ItemData Value=\""+ checkBoxStr2 +"\" ItemOID=\"BSC.TYPE2\"/>\n" +  
                        "<ItemData Value=\""+ checkBoxStr3 +"\" ItemOID=\"BSC.TYPE3\"/>\n" +  
                        "<ItemData Value=\""+ checkBoxStr4 +"\" ItemOID=\"BSC.TYPE4\"/>\n" +  
                        "<ItemData Value=\""+ checkBoxStr5 +"\" ItemOID=\"BSC.TYPE5\"/>\n" +  
                        "<ItemData Value=\""+ checkBoxStr6 +"\" ItemOID=\"BSC.TYPE6\"/>\n" +  
                        "<ItemData Value=\""+ checkBoxStr7 +"\" ItemOID=\"BSC.TYPE7\"/>\n" +  
                    "</ItemGroupData>\n" +  
                "</FormData>\n" +  
            "</StudyEventData>\n" +  
        "</SubjectData>\n" +  
    "</ClinicalData>\n" +  
"</ODM>;  
  
OutputStream os = conn.getOutputStream();  
BufferedWriter writer = new BufferedWriter(  
    new OutputStreamWriter(os, "UTF-8"));  
writer.write(postBody);  
  
writer.flush();  
writer.close();  
os.close();  
int responseCode=conn.getResponseCode();  
  
if (responseCode ==HttpsURLConnection.HTTP_OK) {  
    String line;  
    BufferedReader br=new BufferedReader(new  
InputStreamReader(conn.getInputStream()));  
    while ((line=br.readLine()) != null) {  
        response+=line;  
    }  
}  
else {  
    response= "";  
}  
}  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

```
    return response;  
}
```

[Back to top](#)

Make an Asynchronous Call

Finally, make an asynchronous call using the following code in the onCreate method of the Android Activity. This means that it is executed each time the Activity is activated, which in this app is done with a button click.

Asynchronous call

```
//Run the web service connect logic to send checkBox values to a Rave Study Form  
through RWS  
  
new HTTPAsync().execute("");
```

[Back to top](#)

1.2 Quick Reference

Click the following tabs to get the entire set of RWS request services or separate lists of request services by [solution](#).

All RWS Services

Request Service	Description	URI
Retrieve Clinical Data with the Clinical Audit Records Dataset	Retrieves the clinical data transactions using an auto-paginating, preinstalled custom dataset.	Up to Rave 2019.2.x: GET https://{host}/RaveWebServices/datasets/ClinicalAuditRecords.odm? studyoid={study-oid}&startid={start-id}&per_page={per-page} Up to Rave 2022.2.x: GET https://{host}/RaveWebServices/datasets/ClinicalAuditRecords.odm? studyoid={study-oid}&startid={start-

Request Service	Description	URI
		id}&per_page={per-page}&unicode={unicode} From Rave 2022.3.0 onwards: GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}&unicode={{unicode}}&mode={{mode}}
Retrieve Subject Calendar Visit Dates	Retrieves the Rave Subject Calendar data, that is, the Visit Actual Date and Visit Expected Date.	GET https://{{host}}/RaveWebServices/datasets/SubjectsCalendar.odm?studyoid={{study-oid}}&StudySiteNumber={{studysitenumber}}&SubjectName={{subjectname}}
Retrieve CSV Format Clinical Views	Retrieves a CSV-formatted version of a Clinical View.	
Retrieve Clinical View Metadata	Retrieves the CSV-formatted version of the Clinical View metadata dataset, the CSV-formatted version of the metadata for a specific Clinical View, or the CSV-formatted version of the metadata for a specific project's Clinical Views.	GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv?ViewName={{VIEW NAME}} GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv?ProjectName={{PROJECT NAME}}

Request Service	Description	URI
Retrieve the Comments Dataset	Retrieves a CSV-formatted dataset of investigator comments for a given project.	GET https://{{host}}/RaveWebServices/datasets/SDTMComments.csv?studyid={{ProjectName}}({{EnvironmentName}})
Retrieve the Protocol Deviations Dataset	Retrieves a CSV-formatted dataset of protocol deviations comments for a project.	GET https://{{host}}/RaveWebServices/datasets/SDTMDProtocolDeviations?studyid={{ProjectName}}({{EnvironmentName}})
Retrieve the Data Dictionaries Dataset	Retrieves a CSV-formatted dataset of the latest CRF version's data dictionaries for a specified project and environment.	GET https://{{host}}/RaveWebServices/datasets/SDTMDDataDictionaries.csv?studyid={{ProjectName}}({{EnvironmentName}})
Retrieve Metadata with the Version Folders with Forms Dataset	Retrieves metadata that helps construct a complete hierarchy of Rave objects from the Study level, through matrices and study events to their corresponding forms. The Study Metadata service can then be used in conjunction with this output to trace the structure of each individual form down to its components such as ItemGroups, Items, dictionaries, and so on, thus providing a full ontology.	GET https://{{host}}/ravewebservices/datasets/VersionFoldersWithForms.odm?studyoid={{study-oid}}&metadataversionoid={{versionID}}
Retrieve Admin Data with the Sites Dataset	Retrieves Admin data for all the sites in a specified study.	GET https://{{host}}/RaveWebServices/datasets/Sites.odm?studyoid={{study-oid}}

Request Service	Description	URI
		GET https://{{host}}/RaveWebServices/datasets/Sites.odm
Retrieve Admin Data with the Version Folders Dataset	Retrieves a list of all folders across all matrices for each CRF version in use for a specified study.	GET https://{{host}}/RaveWebServices/datasets/VersionFolders.odm?studyoid={{study-oid}}
Retrieve Lab Data with the Lab Analyte Ranges CSV Dataset	Retrieves the local lab analyte ranges for a study across all sites.	GET https://{{host}}/RaveWebServices/datasets/LabAnalyteRanges.csv?studyoid={{study_oid}}
Retrieve Admin Data with the Signatures Dataset	Retrieves Admin data for all the signatures for a specified study.	GET https://{{host}}/RaveWebServices/datasets/Signatures.odm?studyid={{study-id}}
Retrieve Admin Data with the Users Dataset	Retrieves Admin data for all the users for a specified study.	GET https://{{host}}/RaveWebServices/datasets/Users.odm?studyoid={{study-oid}}&locationoid={{location-oid}}&locationoidtype={{location-oid-type}}
Retrieve Metadata for CV ODM Extensions	Retrieves metadata for CV ODM extensions.	GET {{Clinical Datasets Metadata as ODM URI}}?versionitem={{version-suffix}}[&{{other-query-string-parameters}}]
Include Code and Decode Values for Data Dictionaries	Retrieves Clinical View Datasets as ODM and includes code and decode values for Data Dictionaries.	GET {{clinical view datasets as ODM URI}}?codelistsuffix={{codelists-suffix}}&decodesuffix={{decode-suffix}}[&{{other-query-string-parameters}}]

Request Service	Description	URI
Include Standard Coded and Decode Values for Data Dictionaries	Retrieves Clinical View Datasets as ODM and includes standard coded and decode values for Data Dictionaries.	GET {clinical view datasets as ODM URI}/regular?stdsuffix={std-suffix}&decodesuffix={decode-suffix}[&{other-query-string-parameters}]
Include Standard and Unit Values in a Regular Service	Retrieves Clinical View Datasets as ODM and includes standard and unit values.	GET {clinical view datasets as ODM URI}/regular?stdsuffix={std-suffix}[&{other-query-string-parameters}]
Include Raw Values in a Regular Service	Retrieves Clinical View Datasets as ODM and includes raw data values.	GET {clinical view datasets as ODM URI}/regular?rawsuffix={raw-suffix}[&{other-query-string-parameters}]
Include CRF Version	Retrieves Clinical View Datasets as ODM and includes the CRF version.	GET {clinical view datasets as ODM URI}?versionitem={version-suffix}[&{other-query-string-parameters}]
Retrieve Clinical Datasets Metadata as ODM	Retrieves Clinical View Datasets metadata as ODM.	GET https://{host}/RaveWebServices/studies/{study-oid}/datasets/metadata/regular[?{query-string-parameters}]
Retrieve Clinical View Datasets as ODM	Retrieves Clinical View Datasets as ODM.	GET https://{host}/RaveWebServices/studies/{study-oid}/datasets/{regular-or-raw} GET https://{host}/RaveWebServices/studies/{study-oid}/datasets/{regular-or-raw}/{form-oid}[?{query-string-parameters}] GET https://{host}/RaveWebServices/studies/{study-oid}/versions/{version id}/datasets/{regular-or-raw}

Request Service	Description	URI
		GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/versions/{{version id}}/datasets/{{regular-or-raw}}[/{{form-oid}}][?{{query-string-parameters}}] GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/subjects/{{subject name}}/datasets/{{regular-or-raw}}[/{{form-oid}}][?{{query-string-parameters}}] GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/subjects/{{subject name}}/datasets/{{regular-or-raw}}
Retrieve Study and Library Metadata	Retrieves study and library metadata.	GET https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/versions/{{version-id}} GET https://{{host}}/RaveWebServices/metadata/libraries/{{library-name}}/versions/{{version-id}}
Retrieve Custom Datasets	Retrieves custom datasets.	GET https://{{host}}/RaveWebServices/datasets/{{dataset-name}}[.{format}][[?{{parameter-list}}]] [[&{{pagination-query}}]]
Retrieve Custom Dataset Log Messages	Retrieves custom Dataset Log Messages.	GET https://{{host}}/RaveWebServices/datasets/LogMessages.xml?{{query-string-filters}}

Request Service	Description	URI
Retrieve Incremental ODM Datasets	Retrieves incremental ODM datasets.	GET {clinical view datasets as ODM URI}?start={start-datetime}[&{other-query-string-parameters}]
Retrieve Clinical View Form Datasets as CSV	Retrieves Clinical View Form Datasets as CSV.	GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/datasets/{{regular-or-raw}}/{{form-oid}}.csv[?start={{start-time}}]]
Work With Study and Library Metadata	Posts study or library metadata to a draft.	GET https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/{{project-uuid}}/drafts POST https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/drafts POST https://{{host}}/RaveWebServices/metadata/libraries/{{library-name}}/drafts
Retrieve the List of Subjects in a Study	Retrieves the list of subjects in a study.	GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/Subjects[?status=all&include={{inactive} inactiveAndDeleted}}]
Work With Subject Data	Updates subject data.	POST https://{{host}}/RaveWebServices/webservice.aspx?PostODMClinicalData POST https://{{host}}/RaveWebServices/webservice.aspx?PostODMClinicalData[&RequestIds={{True}}]

Request Service	Description	URI
Retrieve Study and Library Metadata Lists	Retrieves study and library metadata lists.	GET https://{{host}}/RaveWebServices/metadata/studies GET https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/drafts GET https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/versions GET https://{{host}}/RaveWebServices/metadata/libraries GET https://{{host}}/RaveWebServices/metadata/libraries/{{library-name}}/drafts GET https://{{host}}/RaveWebServices/metadata/libraries/{{library-name}}/versions
Retrieve Incremental CSV Datasets	Retrieves incremental CSV datasets.	GET {{clinical view form datasets as CSV URI}}?start={{start-datetime}} [&{{other-query-string-parameters}}]
Retrieve Clinical Study Lists	Retrieves clinical study lists.	GET https://{{host}}/RaveWebServices/studies

ODM Adapter Solution

ODM Adapter	Description	URI
Retrieve Clinical Data with the Clinical Audit Records Dataset	Retrieves the clinical data transactions using an auto-paginating, preinstalled custom dataset.	Up to Rave 2019.2.x: GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm? studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}} Up to Rave 2022.2.x:

ODM Adapter	Description	URI
		GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}&unicode={{unicode}} From Rave 2022.3.0 onwards: GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}&unicode={{unicode}}&mode={{mode}}
Retrieve Admin Data with the Sites Dataset	Retrieves Admin data for all the sites in a specified study.	GET https://{{host}}/RaveWebServices/datasets/Sites.odm?studyoid={{study-oid}} GET https://{{host}}/RaveWebServices/datasets/Sites.odm
Retrieve Admin Data with the Version Folders Dataset	Retrieves a list of all folders across all matrices for each CRF version in use for a specified study.	GET https://{{host}}/RaveWebServices/datasets/VersionFolders.odm?studyoid={{study-oid}}
Retrieve Lab Data with the Lab Analyte Ranges CSV Dataset	Retrieves the local lab analyte ranges for a study across all sites.	GET https://{{host}}/RaveWebServices/datasets/LabAnalyteRanges.csv?studyoid={{study_oid}}
Retrieve Admin Data with the Signatures Dataset	Retrieves Admin data for all the signatures for a specified study.	GET https://{{host}}/RaveWebServices/datasets/Signatures.odm?studyid={{study-id}}

ODM Adapter	Description	URI
Retrieve Admin Data with the Users Dataset	Retrieves Admin data for all the users for a specified study.	GET https://{{host}}/RaveWebServices/datasets/Users.odm?studyoid={{study-oid}}&locationoid={{location-oid}}&locationoidtype={{location-oid-type}}
Retrieve Study and Library Metadata	Retrieves study and library metadata.	GET https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/versions/{{version-id}} GET https://{{host}}/RaveWebServices/metadata/libraries/{{library-name}}/versions/{{version-id}}
Retrieve the List of Subjects in a Study	Retrieves the list of subjects in a study.	GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/Subjects[?status=all&include={{inactive} inactiveAndDeleted}]

Biostat Adapter Solution

Biostat Adapter	Description	URI
Retrieve CSV Format Clinical Views	Retrieves a CSV-formatted version of a Clinical View.	
Retrieve Clinical View Metadata	Retrieves the CSV-formatted version of the Clinical View metadata dataset, the CSV-formatted version of the metadata for a specific Clinical View, or the CSV-formatted version of the metadata for a specific project's Clinical Views.	GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv?ViewName={{VIEW NAME}}

Biostat Adapter	Description	URI
		GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv?ProjectName={{PROJECT NAME}}
Retrieve the Comments Dataset	Retrieves a CSV-formatted dataset of investigator comments for a given project.	GET https://{{host}}/RaveWebServices/datasets/SDTMComments.csv?studyid={{ProjectName}}({{EnvironmentName}})
Retrieve the Protocol Deviations Dataset	Retrieves a CSV-formatted dataset of protocol deviations comments for a project.	GET https://{{host}}/RaveWebServices/datasets/SDTMDProtocolDeviations?studyid={{ProjectName}}({{EnvironmentName}})
Retrieve the Data Dictionaries Dataset	Retrieves a CSV-formatted dataset of the latest CRF version's data dictionaries for a specified project and environment.	GET https://{{host}}/RaveWebServices/datasets/SDTMDDataDictionaries.csv?studyid={{ProjectName}}({{EnvironmentName}})
Retrieve Clinical View Form Datasets as CSV	Retrieves Clinical View Form Datasets as CSV.	GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/datasets/{{regular-or-raw}}/{{form-oid}}.csv[?start={{start-time}}]]

1.3 Cookbook

Rave Web Services (RWS) integrates Medidata Rave with third-party systems to exchange CDISC ODM standard clinical data or metadata synchronously and with immediate confirmation.

In the RWS cookbook, you find API recipes that identify steps to perform tasks as simple as updating a form and as complex as retrieving CSV extracts of Clinical Views with the Biostat Adapter - all using API HTTP requests.

Click your desired recipe below to learn more.

[Retrieving Clinical View Data for Reporting with the Biostat Adapter](#)

Using the Biostat Adapter to View Comments and Protocol Deviations

1. [Authenticate](#)
2. [Retrieve CSV Format Clinical Views](#)
3. [Retrieve Clinical View Metadata](#)
4. [Retrieve the Comments Dataset](#)

5. [Retrieve the Protocol Deviations Dataset](#)
6. [Retrieve the Data Dictionaries Dataset](#)

Using the Biostat Adapter to View Options in Data Dictionaries

1. [Authenticate](#)
2. [Retrieve CSV Format Clinical Views](#)
3. [Retrieve Clinical View Metadata](#)

4. [Retrieve the Data Dictionaries Dataset](#)

[Using the ODM Adapter for Data Warehouse Integration](#)

Medidata's Operational Data Model (ODM) Adapter is a Rave Web Services capability that provides users with transactional ODM extracts of clinical transaction records for fields, field workflow status, field comments and queries for a given study. This solution enables partners and sponsors to load Rave EDC data into their enterprise warehousing systems.

Retrieving CSV Format Clinical Views

1. [Authenticate](#)
2. [Retrieve CSV Format Clinical Views](#)
3. [Retrieve Clinical View Metadata](#)

Retrieving ODM Format Clinical Views

1. [Authenticate](#)

2. [Retrieve ODM Format Clinical View Dataset](#)
3. [Retrieve Study and Library Metadata Lists](#)
4. [Retrieve Admin Data with the Sites Dataset](#)
5. [Retrieve Admin Data with the Users Dataset](#)
6. [Retrieve Admin Data with the Signatures Dataset](#)

Retrieving the List of Subjects in a Study

1. [Authenticate](#)
2. [Retrieve the List of Subjects in a Study](#)

Retrieving Rave Audit Data Incrementally

You can use this workflow to pull Rave audit data incrementally, for example, from a certain date and time onwards.

1. [Authenticate](#)
2. [Retrieve Clinical Data with the Clinical Audit Records Dataset](#)
3. [Retrieve Admin Data with the Version Folders Dataset](#)
4. [Retrieve Study and Library Metadata](#)
5. [Retrieve Admin Data with the Sites Dataset](#)
6. [Retrieve Admin Data with the Users Dataset](#)
7. [Retrieve Admin Data with the Signatures Dataset](#)

[Migrating Metadata from Another EDC System and Publishing Data Into Rave](#)

Creating a Subject

1. [Authenticate](#)
2. [Create a Subject in a Study and Site](#)

Working with Field External Audits

1. [Authenticate](#)
2. [Work With Field External Audits](#)

Updating a Field

1. [Authenticate](#)
2. [Update a Field](#)

Updating a Whole Item Group

1. [Authenticate](#)
2. [Update a Whole Item Group](#)

Working with Field Workflow Properties

1. [Authenticate](#)
2. [Work With Field Workflow Properties](#)

[Using RWS with Interactive Voice Recognition \(IVR/IVRS\)](#)

Creating a Subject

1. [Authenticate](#)
2. [Retrieve CSV Format Clinical Views](#)

Upserting a Form

1. [Authenticate](#)
2. [Upsert a Form](#)

Working with Field External Audits

1. [Authenticate](#)
2. [Work With Field External Audits](#)

1.3.1 Create a Subject

To create a subject in a study and site, follow these steps:

1. [Authenticate](#)
2. [Create a Subject in a Study and Site](#)

1.3.2 Retrieve a Clinical View CSV-formatted Version

To retrieve CSV format Clinical Views, follow these steps:

1. [Authenticate](#)
2. [Retrieve CSV Format Clinical Views](#)
3. [Retrieve Clinical View Metadata](#)

You can retrieve a CSV-formatted version of the Clinical View metadata dataset for:

- all the Clinical Views,
- a specific Clinical View, or
- a specific project's Clinical Views.

1.3.3 Retrieve ODM Format Clinical Views

1. [Authenticate](#)
2. [Retrieve ODM Format Clinical View Dataset](#)
3. [Retrieve Study and Library Metadata Lists](#)
4. [Retrieve Admin Data with the Sites Dataset](#)
5. [Retrieve Admin Data with the Users Dataset](#)
6. [Retrieve Admin Data with the Signatures Dataset](#)

To retrieve ODM format Clinical Views, follow the steps below:

This also involves retrieving the following:

- related study and library metadata lists,
- clinical data with the sites standard dataset,
- clinical data with the users dataset, and
- clinical data with the signatures dataset.

1.3.4 Retrieve Rave Audit Data Incrementally

To retrieve Rave audit data incrementally, follow these steps:

1. [Authenticate](#)
2. [Retrieve Clinical Data with the Clinical Audit Records Dataset](#)
3. [Retrieve Admin Data with the Version Folders Dataset](#)
4. [Retrieve Study and Library Metadata](#)
5. [Retrieve Admin Data with the Sites Dataset](#)
6. [Retrieve Admin Data with the Users Dataset](#)
7. [Retrieve Admin Data with the Signatures Dataset](#)

1.3.5 Retrieve List of Subjects in a Study

To retrieve the list of all accessible subjects in a given study, follow the steps below:

1. [Authenticate](#)
2. [Retrieve the List of Subjects in a Study](#)

1.3.6 Update Fields

Follow the steps below to update these fields:

- data dictionary dropdown list field,
- data dictionary search list field,
- date field,
- dynamic search list field,
- text field, or
- unit dictionary field.

1. [Authenticate](#)
2. [Update a Field](#)

1.3.7 Update Item Groups

Follow the steps below to update the whole standard form or logline to reflect the submitted ODM ItemGroupData:

1. [Authenticate](#)
2. [Update a Whole Item Group](#)

Pre-existing fields get updated if they are included in the ItemGroupData, or cleared if they are not included in the ItemGroupData.

1.3.8 Upsert Forms

Follow the steps below to upsert a repeating form that is located by its OID and a context value in one of its fields:

1. [Authenticate](#)
2. [Upsert a Form](#)

1.3.9 Use Biostat Adapter to View Comments and Protocol Deviations

You can use the Biostat Adapter to view comments and protocol deviations by performing the steps below:

1. [Authenticate](#)
2. [Retrieve CSV Format Clinical Views](#)
3. [Retrieve Clinical View Metadata](#)
4. [Retrieve the Comments Dataset](#)

5. [Retrieve the Protocol Deviations Dataset](#)
6. [Retrieve the Data Dictionaries Dataset](#)

This involves retrieving first the Clinical Views and related metadata and then the following datasets:

- comments,
- protocol, and
- data dictionaries.

1.3.10 Use Biostat Adapter to View Options in Data Dictionaries

You can use the Biostat Adapter to view options in data dictionaries by performing the steps below:

1. [Authenticate](#)
2. [Retrieve CSV Format Clinical Views](#)
3. [Retrieve Clinical View Metadata](#)

4. [Retrieve the Data Dictionaries Dataset](#)

This involves retrieving first the Clinical Views and related metadata, and then the dictionary dataset.

1.3.11 Add External Audits to a Field

You can add external audits to a field in RWS by following the steps below:

1. [Authenticate](#)
2. [Work With Field External Audits](#)

1.3.12 Update Field Statuses

Follow the steps below to update one of the following statuses on a field in RWS:

- Lock,
 - Freeze, or
 - Verify.
1. [Authenticate](#)
 2. [Work With Field Workflow Properties](#)

1.4 Solutions

The following are the main out of the box solutions available for Rave Web Services.

[Retrieve Clinical View Data with the Biostat Adapter](#)

The Biostat Adapter is a Rave Web Services capability, which provides users with CSV extracts of Clinical Views in order to enable biostatisticians, statistical programmers, data managers, and others to create data sets for themselves. The Biostat Adapter was designed in particular to support CDISC's Study Data Tabulation Model (SDTM) format, which is the FDA's preferred standard for submitting data in a New Drug Application (NDA).

[Retrieve Clinical Data with the ODM Adapter](#)

The ODM Adapter is a Rave Web Services capability, which provides users with transactional ODM extracts of clinical transaction records for fields, field workflow status, field comments, and queries for a given study. This solution enables partners and sponsors to load Rave EDC data into their enterprise warehousing systems.

- [Retrieving clinical data with the Clinical Audit Records dataset](#)
 - [ODM Adapter Included Audit Subcategories](#)
- [Retrieving admin data with the Version Folders dataset](#)
- [Retrieving Study and Library metadata](#)
- [Retrieving admin data with the Sites dataset](#)
- [Retrieving admin data with the Users dataset](#)
- [Retrieving admin data with the Signatures dataset](#)

- Retrieving the list of subjects in a Study in RWS

[Back to top](#)

1.4.1 Biostat Adapter

Medidata's Biostat Adapter is a Rave Web Services capability which provides users with CSV extracts of Clinical Views in order to enable biostatisticians, statistical programmers, data managers, and others to create datasets for themselves. The Biostat Adapter was designed in particular to support datasets defined in the CDISC Study Data Tabulation Model (SDTM) format, which is the FDA's preferred standard for submitting data in a New Drug Application (NDA). To ensure SDTM format, the study design has to be SDTM compliant.



Note: The Biostat Adapter does not perform any transformations to SDTM format. It compiles datasets defined in any format supported by the study design.

Use Case

As an integrating system

I want to retrieve CSV datasets from Rave so that I can convert them to the Study Data Tabulation Model (SDTM) format

So that I can submit data for my New Drug Application

Medidata customers typically use SAS to prepare SDTM datasets for their FDA submissions. As Base SAS does not currently have a programmatic way to parse ODM data, Rave now provides the automatic generation of metadata and clinical data in CSV format which can be processed by SAS. You can use RWS to retrieve metadata and clinical data from Rave for further processing, and also to retrieve three additional datasets required by the SDTM standard: the Comments dataset, the Protocol Deviations dataset, and the Data Dictionaries dataset.

Access Restrictions

You can only use RWS to retrieve datasets described on this page and to which you have the necessary access permissions. This includes the Clinical View datasets, the metadata dataset and the additional datasets for Comments, Protocol Deviations, and Data Dictionaries.

Clinical Views

Clinical Views are created or rebuilt when you click the Rebuild Clinical Views button in the Rave Configuration module's Clinical Views page or when you publish a CRF version.

Automatic Generation of CSV Clinical Views and Metadata

For Rave 5.6.4 Patch 04 onwards; whenever you create or rebuild a study's Clinical Views, Rave automatically creates or updates a CSV template for each Clinical View and registers the full name of each Clinical View, (For example V_MEDIFLEX_AE), in the RWS catalog of configurable datasets. Clinical Views from the production (Prod) schema, if any exist, are registered in the catalog with the prefix *prod.* in front of the Clinical View name. (For example *prod.V_MEDIFLEX_AE*).

For Rave Release 2012 onwards, the auto-generated CSV Clinical view configurable datasets are superceded by the more performant [Retrieve Clinical View Form Datasets as CSV](#) service.

Rave also creates a dataset of Clinical View metadata for active projects (as defined in Rave).

The Datasets

[Retrieving CSV Format Clinical Views](#)

[Retrieving Clinical View Metadata](#)

[Retrieving the Comments Dataset](#)

[Retrieving the Protocol Deviations Dataset](#)

[Retrieving the Data Dictionaries Dataset](#)

1.4.2 ODM Adapter

Medidata's Operational Data Model (ODM) Adapter is a Rave Web Services capability that provides users with transactional ODM extracts of clinical transaction records for fields, field workflow status, field comments and queries for a given study. This solution enables partners and sponsors to load Rave EDC data into their enterprise warehousing systems.

- The solution is comprised of a set of pre-installed *custom datasets*.
- The data is returned in [CDISC ODM v1.3.1 format with Medidata vendor extensions](#).
- For more information on the ODM Adapter, see:

- [Operational Data Model \(ODM\) Schema](#)
- [Medidata ODM \(MODM\) Extensions](#)
- [ODM Adapter Guidelines](#)
- [ODM Adapter Included Audit Subcategories](#)

ODM Adapter Version 1 Compared to Version 2



Important:

- There is about a 30 minute latency, on average, for ODM Adapter Version 2 (V2) data.
- Rave 2025.1.0 introduces the following enhancements:
 - ODM Adapter V2 returns X-Last-AuditStudiesBatch-Time in the response header to indicate when the latest refresh completed.
 - ODM Adapter V2 includes the InstanceName attribute in extracted audit XML files. The InstanceName attribute appears after the InstanceID attribute in the ODM Adapter V2 XML file, for example:

```
mdsol:InstanceID=123 mdsol:InstanceName="Cycle 2 Day 1"
```



Note: When translation is available in the user locale, InstanceName shows the translated folder name.

- From Rave 2023.1.0 onwards, the ODM Adapter V2 endpoint is enabled by default for all new URLs.
- From Rave EDC 2022.3.0 onwards, the ODM Adapter Version 2 provides Clinical Audit Record (CAR) extracts for all the Audit Subcategories (ASCs) available for Rave EDC clinical data.
- From Rave EDC 2022.3.0 onwards, the ODM Adapter V2 [Clinical Audit Records \(CAR\)](#) service introduces an optional query string parameter, `mode`, to allow you to extract additional ASCs.
- Prior to Classic Rave 2019.2.0, entering Unicode characters in Rave EDC was supported but retrieving the same Unicode characters in ODM extracts through RWS was not supported. The Unicode characters supported in RWS were equivalent to Latin ASCII characters and special characters like '?', '!', '/', and so on. Many Unicode characters were converted to question mark ?, as specified in the [ODM Unicode character conversion list](#).

- From Classic Rave 2019.2.0 onwards, the ODM Adapter V2 [Clinical Audit Records \(CAR\) service](#) has introduced an optional Boolean query string parameter, `unicode`, to support Unicode characters.
 - Add `unicode=true` to the query string to return Unicode data in RWS, for example:

```
{url}/datasets/ClinicalAuditRecords.odm?studyoid=<Study  
OID>&startid=<Start Audit ID>&per_page=<Per Page>&unicode=true
```
 - If you do not require Unicode data, either omit specifying the `unicode` query parameter or add `unicode=false` to the query string, for example:

```
{url}/datasets/ClinicalAuditRecords.odm?studyoid=<Study  
OID>&startid=<Start Audit ID>&per_page=<Per Page>&unicode=false
```



```
{url}/datasets/ClinicalAuditRecords.odm?studyoid=<Study  
OID>&startid=<Start Audit ID>&per_page=<Per Page>
```

ODM Adapter V1 has the following limitations:

- Duplicate marking
- Poor speed and performance
- [Limited set of Included Audit Subcategories \(ASCs\)](#)
- You have to rely on configurable datasets for additional ASCs and metadata
- ODM Adapter V1 is no longer available for new URLs
- All ODM V1 enabled URLs will be upgraded to ODM Adapter V2 in the near future
- Once upgraded, ODM Adapter V1 will be deprecated.

ODM Adapter V2 provides:

- Faster data extraction, up to 3 million records per hour
- Uniquely identifies all the markings in your data warehouse for data entered after upgrading to Rave 2016.5.0 or later
- Includes all the [ASCs available for EDC Clinical Data entry](#)
- Additional metadata associated with clinical data
- Does not require additional configurable datasets
- All Rave EDC URLs are on ODM Adapter V2

- Represents the default endpoint for all new RWS URLs.

Use Case

As an integrating system

I want to retrieve the clinical transaction data for a Rave study

So that I can report from this data in my own data warehouse

Retrieve Clinical Data Audits

Retrieve clinical data with the Clinical Audit Records dataset

Up to Rave 2019.2.x:

```
GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}
```

From Rave 2019.2.x onwards:

```
GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}&unicode={{unicode}}
```

From Rave 2022.3.0 onwards:

```
GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}&unicode={{unicode}}&mode={{mode}}
```

The main “ClinicalAuditRecords” pre-installed custom dataset provides access to the Rave EDC audit trail to feed every change that was applied to every field through to a client’s data warehouse. It uses [pagination](#) to page the large amount of data over multiple requests in manageable sized chunks, and includes the following:

- Audits at the Subject, Folder, Form, Record and Field levels
- IFreeze, Verify and Lock status changes at field level
- Queries and Comments

Example

ODM Adapter Included Audit Subcategories

Rave Audit Categories and Subcategories

Retrieve Metadata and Admin Data

To interpret the clinical data, the study metadata, user, site and signature admin data can be retrieved first. This can be achieved using the following services.

Retrieve admin data with the Version Folders dataset

```
GET https://host/RaveWebServices/datasets/VersionFolders.odm?studyoid={study-oid}
```

The Version Folders pre-installed custom dataset identifies all study versions in-use.

Example

Retrieve Study and Library metadata

```
GET https://host/RaveWebServices/metadata/studies/{study-oid}/versions/{version-id}
```

Metadata definitions for each study version in-use can be retrieved using the Study Metadata service.

Examples

Retrieve admin data with the Sites dataset

```
GET https://host/RaveWebServices/datasets/Sites.odm?studyoid={study-oid}
```

Admin data for all sites in the study can be retrieved from the Sites pre-installed custom dataset.

Example

Retrieve admin data with the Users dataset

```
GET https://host/RaveWebServices/datasets/Users.odm?studyoid={study-oid}&locationoid={location-oid}
```

Users admin data can be retrieved for each site using the Users pre-installed custom dataset.

Example

Retrieve admin data with the Signatures dataset

```
GET https://{{host}}/RaveWebServices/datasets/Signatures.odm?studyid={{study-id}}
```

Signatures admin data for the study can be retrieved using the Signatures pre-installed custom dataset.

Example

Retrieve the list of subjects in a Study in RWS

```
GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/Subjects[?  
include=inactiveAndDeleted]
```

Listing of all subjects in the study, including those currently inactive or deleted.

1.5 Request Services

The following request services provide you with a means of integrating with Medidata [Classic Rave](#) and [Rave EDC](#) to exchange [CDISC ODM standard](#) clinical data or metadata synchronously and with immediate confirmation. For additional information on [Solutions](#)-related request services, see [Biostat Adapter](#) and [Operational Data Model \(ODM\) Adapter](#).

Biostat Adapter Solution

- Retrieve CSV Format Clinical Views
- Retrieve Clinical View Metadata
- Retrieve the Protocol Deviations Dataset
- Retrieve the Comments Dataset
- Retrieve the Data Dictionaries Dataset
- Retrieve Clinical View Form Datasets as CSV

ODM Adapter Solution

- Retrieve Clinical Data with the Clinical Audit Records Dataset
- Retrieve Admin Data with the Sites Dataset
- Retrieve Admin Data with the Version Folders Dataset
- Retrieve Lab Data with the Lab Analyte Ranges CSV Dataset
- Retrieve Study and Library Metadata
- Retrieve Admin Data with the Signatures Dataset
- Retrieve Admin Data with the Users Dataset

- Retrieve the List of Subjects in a Study

Post Subject Data

- Work With Subject Data

Post Study or Library Metadata

- Work With Study and Library Metadata

Retrieve Clinical View Datasets

- Retrieve Raw or Regular Data
- Retrieve Clinical View Datasets as ODM
- Retrieve Incremental ODM Datasets
- Retrieve Incremental CSV Datasets

Retrieve Clinical View Datasets and Include a Relevant Value

- Include Code and Decode Values for Data Dictionaries
- Include Standard Coded and Decode Values for Data Dictionaries
- Include Standard and Unit Values in a Regular Service
- Include Raw Values in a Regular Service
- Include CRF Version

Retrieve Clinical View Datasets Metadata

- Retrieve Metadata for CV ODM Extensions
- Retrieve Clinical Datasets Metadata as ODM

Retrieve Custom Datasets

- Retrieve Paged Results
- Retrieve Results in Default XML Format
- Configurable Datasets
- Retrieve Custom Datasets
- Retrieve Custom Dataset Log Messages

Retrieve Clinical Study Lists

- Retrieve Clinical Study Lists

Retrieve Clinical Study and Library Metadata Lists

- Retrieve Study and Library Metadata Lists

Retrieve Metadata for Complete Hierarchy of Rave Objects

- Retrieve Metadata with the Version Folders with Forms Dataset

Retrieve Project Information

Content by label

There is no content with the specified labels

[Back to top](#)

1.5.1 Clinical View Extracts

Clinical Views provide you with a method to browse up to date subject data entered into Medidata Rave.

1.5.1.1 Retrieve Clinical Datasets Metadata as ODM

Service	Retrieve Clinical View Datasets Metadata as ODM
Category	Request Services
URI	GET <code>https://{{host}}/RaveWebServices/studies/{{study-oid}}/datasets/metadata/regular[?{{query-string-parameters}}]</code>
Description	Retrieves Clinical View Datasets metadata as ODM.

Use Case

As an integrating system

I want to retrieve my clinical datasets metadata in ODM

So that I can understand my clinical datasets in ODM



Note:

- Clinical views frequently span multiple CRF versions. Use this service to retrieve a combined metadata definition for clinical data in the clinical view.
- See also [retrieving clinical view datasets as ODM](#).

Assumptions

To succeed, I have satisfied these prerequisites:

1. I am a valid Rave User
2. I can view clinical data in my study
3. There is a Clinical View for my study

[Back to top](#)

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/datasets/metadata/regular[?{{query-string-parameters}}]
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{my-org}.mdsol.com"
{study-oid}	The study name	Yes	a URL-escaped string See specifying a study oid
?{query-string-parameters}	See "Optional query string parameters" below	No	

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/datasets/  
metadata/regular
```

Optional query string parameters:

Parameter	Description	Mandatory?	Notes
versionitem={version-suffix}	Adds the subject's CRF version to the dataset, and identifies it with {version-suffix}.	No	See including CRF version
decodesuffix={decode-suffix}	Add decoded values of items that have an associated code list to the dataset, and identifies these values with {decode-suffix}.	No	See including code and decode values for data dictionaries
codelistsuffix={codelists-suffix}	Adds code list OIDS for fields that use a code list to the dataset, and identifies these values with {codelists-suffix}.	No	See including code and decode values for data dictionaries
rawsuffix={raw-suffix}	Adds raw data values to a full or incremental dataset, and identifies these values with {raw-suffix}.	No	See including raw values in the regular service

**Note:**

- The list of parameters in {query-string-parameters} must begin with a question mark (?) and parameters must be separated by &.
- Any combination of these parameters is permitted, in any order.
- See also [retrieving metadata for CV ODM extensions](#).

[Back to top](#)

The Response

Header	Reason	Body	Notes
200	Dataset retrieved	ODM representation of the dataset	See Response Note for more information on the returned ODM.
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See error responses - the complete list

Header

The header contains the following elements:

- HTTP Response code
- Content-Type : "text/xml"

Body

The response Body is a [valid ODM 1.3 snapshot document](#)

My ODM document is constructed as follows:

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<?xml version="1.0" ?>
<ODM>
  <Study OID="{study-oid}">
    <GlobalVariables>
      <StudyName>{study-oid}</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>{study-oid}</ProtocolName>
    </GlobalVariables>

    <!-- "Measurement unit definitions" -->

    <MetaDataVersion
      OID="{metadata-oid}"
      Name="{draft-name}"
      mdsol:PrimaryFormOID="{primary-form}"
      mdsol:DefaultMatrixOID="{default-matrix-oid}"
      mdsol:SignaturePrompt="{signature-prompt}">
      <!-- "Folder definitions" -->
      <!-- "Form definitions" -->
      <!-- "Log line definitions " -->
      <!-- "Field definitions" -->
    </MetaDataVersion>
```

```
</Study>
</ODM>
```

i Note: The MetaDataVersion OID attribute shows the date and time of the last update of any clinical view in the study.

Example success response

Header:

```
HTTP Response code : 200 OK
Content-Type : "text/xml"
```

Body:

```
<ODM FileType="Snapshot" Granularity="Metadata"
CreationDateTime="2013-01-13T09:54:53.839-00:00" FileOID="edb011d4-d4d7-43c7-bb39-
cdaf508b76d8" ODMVersion="1.3" xmlns="http://www.cdisc.org/ns/odm/v1.3">
    <Study OID="1Test_20130108_003731 (Prod)">
        <GlobalVariables>
            <StudyName>1Test_20130108_003731</StudyName>
            <StudyDescription>
            </StudyDescription>
            <ProtocolName>1Test_20130108_003731</ProtocolName>
        </GlobalVariables>
        <MetaDataVersion OID="2013-01-08T00:37:42.000-00:00"
Name="2013-01-08T00:37:42.000-00:00">
            <FormDef OID="DM" Name="V_1TEST_20130108_003731_DM" Repeating="Yes">
                <ItemGroupRef ItemGroupOID="DM_LOG_LINE" Mandatory="No" />
            </FormDef>
            <FormDef OID="PRIMARY" Name="V_1TEST_20130108_003731_PRIMARY"
Repeating="Yes">
                <ItemGroupRef ItemGroupOID="PRIMARY_LOG_LINE" Mandatory="No" />
            </FormDef>
            <ItemGroupDef OID="DM_LOG_LINE" Name="DM_LOG_LINE" Repeating="Yes">
                <ItemRef ItemOID="DM.DOB" Mandatory="No" />
            </ItemGroupDef>
            <ItemGroupDef OID="PRIMARY_LOG_LINE" Name="PRIMARY_LOG_LINE"
Repeating="Yes">
                <ItemRef ItemOID="PRIMARY.FIELD1" Mandatory="No" />
            </ItemGroupDef>
            <ItemDef OID="DM.DOB" Name="DOB" DataType="text" Length="200" />
```

```
<ItemDef OID="PRIMARY.FIELD1" Name="FIELD1" DataType="text"  
Length="200" />  
</MetaDataVersion>  
</Study>  
</ODM>
```

[Back to top](#)

Scenarios

Background:

Given I am a user

Scenario Outline: Success

Given there is a "<production_or_auxiliary>" study that I can see
And clinical views have been generated for the project
When I make an HTTP GET request to the URL "/studies/{<study oid>}/datasets/metadata/regular"

Then I should receive a successful response
And the response should be a valid ODM 1.3 document
And there should be one MetaDataVersion element
And the MetaDataVersionOID should be the latest updated datetime across the views
And there should be one FormDef for each clinical view ordered by OID
And there should be one ItemDef for each data column in the view ordered by clinical view column ordinal

Examples:

production_or_auxiliary study oid	
production project_name	
production project_name (production_environment)	
production project_name(production_environment)	
auxiliary project_name (auxiliary_environment)	
auxiliary project_name(auxiliary_environment)	

Scenario: Clinical views not present

Given there is a study with no clinical views
When I request the clinical view dataset metadata for the study
Then I should receive a successful response
And the response should be a valid ODM 1.3 document
But there should be no MetaDataVersion elements

Scenario: Include code list OID

Given there is a study with clinical views
When I request the clinical view dataset metadata for the study with parameter "codelistsuffix=_CL"
Then I should receive the clinical view dataset metadata
And there should be one ItemDef suffixed "_CL" for each field with an associated dictionary

Scenario: Study does not exist

When I request the clinical view dataset metadata for a study that does not exist
Then I should receive an HTTP response with status code "Not Found"

Scenario: Study not authorised

Given there is a study with clinical views
And I do not have permission to view the study data
When I request the clinical view dataset metadata for the study
Then I should receive an HTTP response with status code "Not Found"



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

1.5.1.2 Retrieve Clinical View Form Datasets as CSV

Service	Retrieve Clinical View Form Datasets as CSV
Category	Request Services
URI	GET https://[{host}]/RaveWebServices/studies/{study-oid}/datasets/{regular-or-raw}/{form-oid}.csv[?start={start-time}]
Description	Retrieves Clinical View Form Datasets as CSV.

Use Case

As an integrating system

I want to retrieve my clinical view form datasets in CSV

So that I can manage my study data

New in version 2012.1.0.

Assumptions

To succeed, I have satisfied these prerequisites:

1. I am a valid Rave User
2. I can view clinical data in my study
3. There is a Clinical View for my study

[Back to top](#)

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/datasets/{{regular-or-raw}}/  
{{form-oid}}.csv[?start={{start-time}}]
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name, e.g. my-organisation.mdsol.com	Yes	usually "{my-org}.mdsol.com"
studies/{study-oid}	The study name. e.g. Mediflex (Prod)	Yes	a URL-escaped string See specifying a study oid
{regular-or-raw}	The dataset type. Values: 'raw' or 'regular'.	Yes	See retrieving raw or regular data
{form-oid}	The form identifier	Yes	a URL-escaped string
?start={start-time}	Incremental inserts, updates or removes after start-time or includes all current values if this is not present	No	See retrieving incremental CSV datasets

**Note:**

- dataset type of raw or regular
- cumulative or incremental extracts

Examples:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod) /  
datasets/raw/DEMOG.csv  
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod) /datasets/  
regular/DEMOG.csv?start=2011-11-23T10:00:00
```

[Back to top](#)

The Response

Header	Reason	Body	Notes
200	Dataset retrieved	CSV representation of the dataset	
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See error responses - the complete list

Header

The header contains the following elements:

- HTTP Response code
- Content-Type : "text/csv"
- X-MWS-CV-Last-Updated : {ISO 8601 format date time}



Note:

- A timestamp of when the clinical view was last updated is returned in the response header.
- See also [clinical view service limitations](#).

New in version 2012.1.0: X-MWS-CV-Last-Updated

Body

The response body is a CSV format document:

```
userid,projectid,project,studyid,environmentName,subjectId,StudySiteId,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNumber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,{field1},{field2}...  
"{userid}","{project id}","{project}","{study id}","{environment name}","{subject id}","{study site id}","{subject}","{site id}","{site}","{site number}","{site group}","{instance id}","{instance name}","{instance repeat number}","{folder id}","{folder}","{folder name}","{folder seq}","{target days}","{data page id}","{data page name}","{page repeat number}","{record date}","{record id}","{record position}","{record active}","{save ts}","{min created}","{max updated}","{field1 value}","{field2 value}"...
```

Example success response

Header:

```
HTTP Response code : 200 OK  
Content-Type : "text/csv"  
X-MWS-CV-Last-Updated : "2010-04-22T12:46:02"
```

Body:

```

userid,projectid,project,studyid,environmentName,subjectId,StudySiteId,Subject,siteid,
Site,SiteNumber,SiteGroup,
instanceId,InstanceName,InstanceRepeatNumber,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,
PageRepeatNumber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,BRTHDTC,AGE,SEX,SEX_STD,
CHILDBEAR,CHILDBEAR_STD,RACE,RACE_STD,COUNTRY,COUNTRY_STD,DSSTDT
EOF

```

[Back to top](#)

Scenarios

Background:

```

Given I am a user
And there is a clinical view with date type data that I can see
And there is a repeating form in the clinical view

```

Scenario Outline: Full dataset for a form as csv

```

When I make an HTTP GET request to the URL "/studies/{study oid}/datasets/<regular or raw>/{{the form name}}.csv"

```

```

Then I should receive a successful response
And the response header should have a X-MWS-CV-Last-Updated value of when the CV was last updated
And the response header should have a content type of "text/csv"
And the response data should stream
And the response body should be:
"""

```

```

userid,projectid,project,studyid,environmentName,subjectId,StudySiteId,Subject,siteid,
Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNumber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD1>
  "{user id}","{project id}","{project name}","{study id}","{environment name}","{subject id}","{study site id}","ABC001","{siteid}","{site}","{site number}","{site group}","","","","","","","{folder seq}","","","{datapage id}","Basic","{page repeat number}","","","{record id}","{record position}","1","{save Ts}","{min created}","{max created}","<FIELD1_value1>
  "{user id}","{project id}","{project name}","{study id}","{environment name}","{subject id}","{study site id}","ABC002","{siteid}","{site}","{site number}","{site group}","","","","","","","{folder seq}","","","{datapage id}","Basic","{page repeat number}","","","{record id}","{record position}","1","{save Ts}","{min created}","{max created}","<FIELD1_value2>
EOF
"""

```

Examples:

```

| regular or raw | FIELD1
FIELD1_value1
FIELD1_value2
| raw           | FIELD1
"11 JAN 2010"
2010"
| regular      | FIELD1,FIELD1_RAW,FIELD1_INT,FIELD1_YYYY,FIELD1_MM,FIELD1_DD |
"2010-01-11T00:00:00","11 JAN 2010","2010-01-11T00:00:00","2010","1","11" |
"2010-01-11T00:00:00","11 JAN 2010","2010-01-11T00:00:00","2010","1","11" |

```

Scenario Outline: Incremental dataset for Inserted data

```

When I observe the current time
And I add a new form with data
And I have refreshed the clinical views for the study
And I make an HTTP GET request to the URL "/studies/{study oid}/datasets/<regular
or raw>/{{the form name}}.csv?start={observed time}"
Then I should receive a successful response
And the response header should have a X-MWS-CV-Last-Updated value of when the CV
was last updated
And the response header should have a content type of "text/csv"
And the response data should stream
And the response body should have a TransactionType of "I"
And the response body should be:
"""

```

```

TransactionType,userid,projectid,project,studyid,environmentName,subjectId,StudySiteId
,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber
,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNum
ber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD1>

```

```

    "I","{user id}","{project id}","{project name}","{study id}","{environment
name}","{subject id}","{study site id}","ABC003","{siteid}","{site}","{site
number}","{site group}","","","","","","","{folder seq}","","","{datapage
id}","Basic","{page repeat number}","","","{record id}","{record position}","1","{save
Ts}","{min created}","{max created}<FIELD1_value1>
    EOF
"""

```

Examples:

```

| regular or raw | FIELD1
FIELD1_value1
| raw           | FIELD1
"05 JAN 2012"
| regular      | FIELD1,FIELD1_RAW,FIELD1_INT,FIELD1_YYYY,FIELD1_MM,FIELD1_DD |
"2012-01-05T00:00:00","05 JAN 2012","2012-01-05T00:00:00","2012","1","5" |

```

Scenario Outline: Incremental dataset for updated data

```

When I observe the current time

```

```

And I update the form
And I have refreshed the clinical views for the study
And I make an HTTP GET request to the URL "/studies/{study oid}/datasets/<regular
or raw>/{{the form name}}.csv?start={{observed time}}"
Then I should receive a successful response
And the response header should have a X-MWS-CV-Last-Updated value of when the CV
was last updated
And the response header should have a content type of "text/csv"
And the response data should stream
And the response body should have a TransactionType of "U"
And the response body should be:
"""

```

```

TransactionType,userid,projectid,project,studyid,environmentName,subjectId,StudySiteId
,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber
,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNum
ber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD1>
    "U","{user id}","{project id}","{project name}","{study id}","{environment
name}","{subject id}","{study site id}","ABC002","{siteid}","{site}","{site
number}","{site group}","","","","","","","{folder seq}","","","{datapage
id}","Basic","{page repeat number}","","","{record id}","{record position}","1","{save
Ts}","{min created}","{max created}<FIELD1_value1>
    EOF
"""

```

Examples:

regular or raw FIELD1	
FIELD1_value1	
raw FIELD1	
"05 JAN 2012"	
regular FIELD1,FIELD1_RAW,FIELD1_INT,FIELD1_YYYY,FIELD1_MM,FIELD1_DD	
"2012-01-05T00:00:00","05 JAN 2012","2012-01-05T00:00:00","2012","1","5"	

Scenario Outline: Incremental dataset for deactivated data

```

When I observe the current time
And I deactivate the form
And I have refreshed the clinical views for the study
And I make an HTTP GET request to the URL "/studies/{study oid}/datasets/<regular
or raw>/{{the form name}}.csv?start={{observed time}}"
Then I should receive a successful response
And the response header should have a X-MWS-CV-Last-Updated value of when the CV
was last updated
And the response header should have a content type of "text/csv"
And the response data should stream
And the response body should have a TransactionType of "D"
And the response body should be:
"""

```

```

TransactionType,userid,projectid,project,studyid,environmentName,subjectId,StudySiteId

```

```
,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber
,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNumber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD1>
"D","{user id}","{project id}","{project name}","{study id}","{environment name}","{subject id}","{study site id}","ABC002","{siteid}","{site}","{site number}","{site group}","","","","","","","{folder seq}","","","{datapage id}","Basic","{page repeat number}","","","{record id}","{record position}","0","{save Ts}","{min created}","{max created}<FIELD1_value1>
EOF
"""

```

Examples:

```
| regular or raw | FIELD1
FIELD1_value1
| raw           | FIELD1 | "11 JAN 2010" |
| regular       | FIELD1,FIELD1_RAW,FIELD1_INT,FIELD1_YYYY,FIELD1_MM,FIELD1_DD |
"2010-01-11T00:00:00","11 JAN 2010","2010-01-11T00:00:00","2010","1","11" |
```

Scenario Outline: Incremental dataset for no changes to data

```
When I observe the current time
And I make no change to the form
And I have refreshed the clinical views for the study
And I make an HTTP GET request to the URL "/studies/{study oid}/datasets/<regular or raw>/{{the form name}}.csv?start={{observed time}}"
Then I should receive a successful response
And the response header should have a X-MWS-CV-Last-Updated value of when the CV was last updated
And the response header should have a content type of "text/csv"
And the response data should stream
And the response body should be:
"""

```

```
TransactionType,userid,projectid,project,studyid,environmentName,subjectId,StudySiteId
,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber
,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNumber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD1>
EOF
"""

```

[Back to top](#)

1.5.1.3 Retrieve CSV Format Clinical Views

Service	Retrieve CSV Format Clinical Views
Category	Biostat Adapter
URIs	GET https://{{host}}/RaveWebServices/studies/{{study-name}}/datasets/regular/{{form-oid}}.csv GET https://{{host}}/RaveWebServices/datasets/{{clinical view name}}.csv GET https://{{host}}/RaveWebServices/datasets/prod.{{clinical view name}}.csv
Description	Retrieves a CSV-formatted version of a Clinical View.

You can retrieve a CSV-formatted version of a Clinical View by making an RWS request to the following URLs:

```
GET https://{{host}}/RaveWebServices/studies/{{study-name}}/datasets/regular/{{form-oid}}.csv
GET https://{{host}}/RaveWebServices/datasets/{{clinical view name}}.csv
GET https://{{host}}/RaveWebServices/datasets/prod.{{clinical view name}}.csv
```

See [Retrieve Clinical View Form Datasets as CSV](#) for more details of this service.

1.5.1.4 Retrieve Incremental CSV Datasets

Service	Retrieve Incremental CSV Datasets
Category	Request Services
URI	GET {{clinical view form datasets as CSV URI}}?start={{start-datetime}}[&{{other-query-string-parameters}}]
Description	Retrieves incremental CSV datasets.

Use Case

I am [retrieving my clinical view form datasets as CSV](#) and...

I am retrieving updates since my start-datetime

i Note: You can request recent updates by including a start-datetime query-string parameter. The start-datetime must be in ISO 8601 format. The response will add a TransactionType as the first column that identifies net inserts or updates since your start-datetime, in order for you to keep your snapshot up-to-date.

My GET Request

URI:

```
GET {clinical view form datasets as CSV URI}?start={start-datetime} [&{other-query-string-parameters}]
```

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod) / datasets/raw/DM.csv?start="2012-01-05T00:00:00"
```

The Response

Header

The header contains the following elements:

- HTTP Response code : 200 OK
- Content-Type : "text/csv"
- X-MWS-CV-Last-Updated={ISO 8601 format date time}

i Note:

- A timestamp of when the clinical view was last updated is returned in the response header.
- See also [clinical view service limitations](#).

Body

- Data items created after my start-datetime have TransactionType value "I".
- Data items updated after my start-datetime have TransactionType value "U".
- Data items removed after my start-datetime have TransactionType value "R".
- Data items added and then removed between my start-datetime and the time at which the request was received appear twice, with TransactionType of "I" and "R".
- Data items removed and then reactivated between my start-datetime and the time at which the request was received have TransactionType "U".

Example Response CSV document

```
TransactionType,userid,projectid,project,studyid,environmentName,subjectId,StudySiteId
,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber
,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNum
ber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD
D1>
"I","{user id}","{project id}","{project name}","{study id}","{environment
name}","{subject id}","{study site id}","ABC001","{siteid}","{site}","{site
number}","{site group}","","","","","","","{folder seq}","","","{datapage
id}","Basic","{page repeat number}","","","{record id}","{record position}","1","{save
Ts}","{min created}","{max created}<FIELD1_value1>
"U","{user id}","{project id}","{project name}","{study id}","{environment
name}","{subject id}","{study site id}","ABC002","{siteid}","{site}","{site
number}","{site group}","","","","","","","{folder seq}","","","{datapage
id}","Basic","{page repeat number}","","","{record id}","{record position}","1","{save
Ts}","{min created}","{max created}<FIELD1_value2>
EOF
```

Scenarios

Scenario Outline: Incremental dataset for Inserted data

When I observe the current time
 And I add a new form with data
 And I have refreshed the clinical views for the study
 And I make an HTTP GET request to the URL "/studies/{study oid}/datasets/<regular
 or raw>/{{the form name}}.csv?start={observed time}"
 Then I should receive a successful response
 And the response header should have a X-MWS-CV-Last-Updated value of when the CV
 was last updated
 And the response header should have a content type of "text/csv"
 And the response data should stream
 And the response body should have a TransactionType of "I"
 And the response body should be:
 """

```
TransactionType,userid,projectid,project,studyid,environmentName,subjectId,StudySiteId
,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber
,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNum
ber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD
D1>
```

```
    "I","{user id}","{project id}","{project name}","{study id}","{environment
name}","{subject id}","{study site id}","ABC003","{siteid}","{site}","{site
number}","{site group}","","","","","","","{folder seq}","","","{datapage
id}","Basic","{page repeat number}","","","{record id}","{record position}","1","{save
Ts}","{min created}","{max created}<FIELD1_value1>
    EOF
"""
```

Examples:

```
| regular or raw | FIELD1
FIELD1_value1
| raw           | FIELD1
"05 JAN 2012"
| regular       | FIELD1,FIELD1_RAW,FIELD1_INT,FIELD1_YYYY,FIELD1_MM,FIELD1_DD |
"2012-01-05T00:00:00","05 JAN 2012","2012-01-05T00:00:00","2012","1","5" |
```

Scenario Outline: Incremental dataset for updated data

```
When I observe the current time
And I update the form
And I have refreshed the clinical views for the study
And I make an HTTP GET request to the URL "/studies/{study oid}/datasets/<regular
or raw>/{{the form name}}.csv?start={observed time}"
Then I should receive a successful response
And the response header should have a X-MWS-CV-Last-Updated value of when the CV
was last updated
And the response header should have a content type of "text/csv"
And the response data should stream
And the response body should have a TransactionType of "U"
And the response body should be:
"""
```

```
TransactionType,userid,projectid,project,studyid,environmentName,subjectId,StudySiteId
,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber
,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNum
ber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD
D1>
```

```
    "U","{user id}","{project id}","{project name}","{study id}","{environment
name}","{subject id}","{study site id}","ABC002","{siteid}","{site}","{site
number}","{site group}","","","","","","","{folder seq}","","","{datapage
id}","Basic","{page repeat number}","","","{record id}","{record position}","1","{save
Ts}","{min created}","{max created}<FIELD1_value1>
    EOF
"""
```

Examples:

```
| regular or raw | FIELD1
FIELD1_value1
| raw           | FIELD1
"05 JAN 2012"
| regular       | FIELD1,FIELD1_RAW,FIELD1_INT,FIELD1_YYYY,FIELD1_MM,FIELD1_DD |
"2012-01-05T00:00:00","05 JAN 2012","2012-01-05T00:00:00","2012","1","5" |
```

Scenario Outline: Incremental dataset for deactivated data

```
When I observe the current time
And I deactivate the form
And I have refreshed the clinical views for the study
And I make an HTTP GET request to the URL "/studies/{study oid}/datasets/<regular
or raw>/<the form name>.csv?start={observed time}"
Then I should receive a successful response
And the response header should have a X-MWS-CV-Last-Updated value of when the CV
was last updated
And the response header should have a content type of "text/csv"
And the response data should stream
And the response body should have a TransactionType of "D"
And the response body should be:
"""

```

```
TransactionType,userid,projectid,project,studyid,environmentName,subjectId,StudySiteId
,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber
,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNum
ber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD1>
"D","{user id}","{project id}","{project name}","{study id}","{environment
name}","{subject id}","{study site id}","ABC002","{siteid}","{site}","{site
number}","{site group}","","","","","","","{folder seq}","","","{datapage
id}","Basic","{page repeat number}","","","{record id}","{record position}","0","{save
Ts}","{min created}","{max created}","<FIELD1_value1>
EOF
"""

```

Examples:

```
| regular or raw | FIELD1
FIELD1_value1
| raw           | FIELD1 | "11 JAN 2010" |
| regular       | FIELD1,FIELD1_RAW,FIELD1_INT,FIELD1_YYYY,FIELD1_MM,FIELD1_DD |
"2010-01-11T00:00:00","11 JAN 2010","2010-01-11T00:00:00","2010","1","11" |
```

Scenario Outline: Incremental dataset for no changes to data

```
When I observe the current time
And I make no change to the form
And I have refreshed the clinical views for the study
And I make an HTTP GET request to the URL "/studies/{study oid}/datasets/<regular
or raw>/<the form name>.csv?start={observed time}"
Then I should receive a successful response
```

And the response header should have a X-MWS-CV-Last-Updated value of when the CV was last updated

And the response header should have a content type of "text/csv"

And the response data should stream

And the response body should be:

"""

```
TransactionType,userid,projectid,project,studyid,environmentName,subjectId,StudySiteId
,Subject,siteid,Site,SiteNumber,SiteGroup,instanceId,InstanceName,InstanceRepeatNumber
,folderid,Folder,FolderName,FolderSeq,TargetDays,DataPageId,DataPageName,PageRepeatNum
ber,RecordDate,RecordId,recordposition,RecordActive,SaveTs,MinCreated,MaxUpdated,<FIELD1>
```

EOF

"""

Examples:

regular or raw FIELD1	
FIELD1_value1	
raw	FIELD1
regular	FIELD1,FIELD1_RAW,FIELD1_INT,FIELD1_YYYY,FIELD1_MM,FIELD1_DD

1.5.1.5 Retrieve Incremental ODM Datasets

Service	Retrieve Incremental ODM Datasets
Category	Request Services
URI	GET {clinical view datasets as ODM URI}?start={start-datetime}[&{other-query-string-parameters}]
Description	Retrieves incremental ODM datasets.

Use Case

I am *retrieving my clinical view datasets as ODM* and...

I am retrieving updates since my start-datetime



Note: You can request recent updates by including a start-datetime query-string parameter. The start-datetime must be in ISO 8601 format. The response will be a transactional ODM document that

identifies net inserts or updates since your start-datetime, in order for you to keep your snapshot up-to-date. The transactions will not however contain audit-trail style individual changes, nor who made them.

My GET Request

URI:

```
GET {clinical view datasets as ODM URI}?start={start-datetime} [&{other-query-string-parameters}]
```

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/datasets/raw?start="2012-01-05T00:00:00"
```

The Response

Header

The header contains the following elements:

- HTTP Response code : 200 OK
- Content-Type : "text/xml"
- X-MWS-CV-Last-Updated={ISO 8601 format date time}



Note:

- A timestamp of when the clinical view was last updated is returned in the response header.
- See also [clinical view service limitations](#).

Body

- The body is a valid ODM 1.3 transactional document
- Data items created after my start-datetime have TransactionType value "Insert".
- Data items updated after my start-datetime have TransactionType value "Update".

- Data items removed after my start-datetime have TransactionType value "Remove".
- Data items added and then removed between my start-datetime and the time at which the request was received appear twice, with TransactionType of "Insert" and "Remove".
- Data items removed and then reactivated between my start-datetime and the time at which the request was received have TransactionType "Upsert".

Example Response ODM document

```

<ODM FileType="Transactional" FileOID="FILE569"
CreationDateTime="2008-02-12T09:30:00" ODMVersion="1.3" xmlns:mdsol="http://
www.mdsol.com/ns/odm/metadata" xmlns="http://www.cdisc.org/ns/odm/v1.3">
    <ClinicalData StudyOID="TestClinicalViews" MetaDataVersionOID="30">
        <SubjectData SubjectKey="CVTest 0001">
            <SiteRef LocationOID="12345" />
            <StudyEventData StudyEventOID="SUBJECT">
                <FormData FormOID="AE" FormRepeatKey="1">
                    <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1"
                        TransactionType="Update">
                        <ItemData ItemOID="AE.AEYN" Value="Y" />
                        <ItemData ItemOID="AE.AETERM" Value="HEART PAIN" />
                        <ItemData ItemOID="AE.AESTDTC" Value="12 DEC 2009" />
                        <ItemData ItemOID="AE.AEONG" Value="N" />
                        <ItemData ItemOID="AE.AEENDTC" Value="14 DEC 2009" />
                        <ItemData ItemOID="AE.AELOC" Value="HEART" />
                        <ItemData ItemOID="AE.AESEV" Value="Mild" />
                    </ItemGroupData>
                </FormData>
            </StudyEventData>
        </SubjectData>
    </ClinicalData>
</ODM>

```

Scenarios

Scenario Outline: Incremental dataset for a form inside a folder

When I observe the current time

And update the subject with the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	
	form_repeat	logline_repeat	logline_transaction	field	value	
	field_transaction					
1	SCREEN	Update		1	DM	Update
Update					BIRTHDT	30 JAN 1978
1					AE	Update
					AEDESC	BACK PAIN
1			Insert			

```

Insert      |
And I make an HTTP GET request to the URL "<url>"  

Then I should receive a successful response  

And the response should be a valid ODM 1.3 transactional document  

And the response data should stream  

And the response header should have a X-MWS-CV-Last-Updated value of when the CV  

was last updated  

And ODM element in the response should have the following child elements:  

"""
<ClinicalData StudyOID="{study oid} (Prod)" MetaDataVersionOID="{version id}">  

  <SubjectData SubjectKey="{subject name}">  

    <SiteRef LocationOID="{site number}" />  

    <StudyEventData StudyEventOID="SUBJECT">  

      <FormData FormOID="AE" FormRepeatKey="1">  

        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2"  

        TransactionType="Insert">  

          <ItemData ItemOID="AE.AEYN" Value="YES"/>  

          <ItemData ItemOID="AE.AEDESC" Value="BACK PAIN" />  

        </ItemGroupData>  

      </FormData>  

    </StudyEventData>  

  </SubjectData>  

</ClinicalData>  

<ClinicalData StudyOID="{study oid} (Prod)" MetaDataVersionOID="{version id}">  

  <SubjectData SubjectKey="{subject name}">  

    <SiteRef LocationOID="{site number}" />  

    <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1">  

      <FormData FormOID="DM" FormRepeatKey="1">  

        <ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Update">  

          <ItemData ItemOID="DM.BRTHDT" Value="1978-01-30" />  

        </ItemGroupData>  

      </FormData>  

    </StudyEventData>  

  </SubjectData>  

</ClinicalData>
"""

```

Examples:

```

|  

url | /studies/{study oid}/datasets/regular?start={observed  

time} |  

  | /studies/{study oid}/versions/{version id}/datasets/regular?start={observed  

time} |  

  | /studies/{study oid}/subjects/{subject name}/datasets/regular?start={observed  

time} |

```

Scenario Outline: Full dataset for a subject level form

When I observe the current time

And update the subject with the following data:

```

| form      | form_transaction | form_repeat | logline_repeat |

```

```

logline_transaction | field      | value      | field_transaction |
| AE             | Update          | 1           | 2
Insert           | AEDESC   | BACK PAIN | Insert
And I make an HTTP GET request to the URL "<url>"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And ODM element in the response should have the following child elements:
"""
<ClinicalData StudyOID="{study oid}(Prod)" MetaDataVersionOID="{version id}">
  <SubjectData SubjectKey="{subject name}">
    <SiteRef LocationOID="{site number}" />
    <StudyEventData StudyEventOID="SUBJECT">
      <FormData FormOID="AE" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2">
          TransactionType="Insert"
            <ItemData ItemOID="AE.AEYN" Value="YES"/>
            <ItemData ItemOID="AE.AEDESC" Value="BACK PAIN" />
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
"""

Examples:
|
url
|
  | /studies/{study oid}/datasets/regular/AE?start={observed
time}           |
  | /studies/{study oid}/versions/{version id}/datasets/regular/AE?
start={observed time}   |
  | /studies/{study oid}/subjects/{subject name}/datasets/regular/AE?
start={observed time} |

```

1.5.1.6 Retrieve Clinical View Datasets as ODM

Service	Retrieve Clinical View Datasets as ODM
Category	Request Services
URI	<p>GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/datasets/{{regular-or-raw}}</p> <p>GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/datasets/{{regular-or-raw}}[/{form-oid}][?{{query-string-parameters}}]</p>

Service	Retrieve Clinical View Datasets as ODM
	GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/versions/{{version id}}/datasets/{{regular-or-raw}} GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/versions/{{version id}}/datasets/{{regular-or-raw}}[[{{form-oid}}][?{{query-string-parameters}}]] GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/subjects/{{subject name}}/datasets/{{regular-or-raw}}[[{{form-oid}}][?{{query-string-parameters}}]] GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/subjects/{{subject name}}/datasets/{{regular-or-raw}}
Description	Retrieves Clinical View Datasets as ODM.

Use Case

As an integrating system

I want to retrieve my clinical view datasets as ODM

So that I can manage my clinical data



Note:

- Use this service to retrieve clinical data from Rave clinical views. A clinical view is typically used to review data relating to all subjects rather than to an individual subject. Each clinical view corresponds to one Rave form definition, such as a Demographics form; each row in the view contains data for one instance of the form or one record on a log form. For more information on types of clinical views, see Rave documentation on Clinical Views.
- See also [retrieving clinical datasets metadata as ODM](#).

Assumptions

To succeed, I have satisfied these prerequisites:

1. I am a valid Rave User

2. I can view clinical data in my study
3. There is a Clinical View for my study

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://host/RaveWebServices/studies/{study-oid}/datasets/{regular-or-raw}  
GET https://host/RaveWebServices/studies/{study-oid}/datasets/{regular-or-raw} [/ {form-oid}] [?{query-string-parameters}]  
GET https://host/RaveWebServices/studies/{study-oid}/versions/{version id}/datasets/{regular-or-raw}  
GET https://host/RaveWebServices/studies/{study-oid}/versions/{version id}/datasets/{regular-or-raw} [/ {form-oid}] [?{query-string-parameters}]  
GET https://host/RaveWebServices/studies/{study-oid}/subjects/{subject name}/datasets/{regular-or-raw} [/ {form-oid}] [?{query-string-parameters}]  
GET https://host/RaveWebServices/studies/{study-oid}/subjects/{subject name}/datasets/{regular-or-raw}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	Usually "{my-org}.mdsol.com"
{study-oid}	The study name	Yes	URL-escaped string See specifying a study oid
versions/{version-number}	Returns data for this CRF version	No	URL-escaped string
subjects/{subject-name}	Returns data for this subject name	No	URL-escaped string
{regular-or-raw}	The dataset type. Values - 'raw' or 'regular'.	Yes	See retrieving raw or regular
{form-oid}	Returns data for this form	No	URL-escaped string
?{query-string-parameters}	See "Optional query string parameters" below	No	

**Note:**

- Versions/{version-number} and subjects/{subject-name} cannot be combined in the same URI.
(A clinical dataset can be subject or version-specific, but not both.)
- URI parameters must be URL-escaped strings.

Examples:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod) /  
datasets/raw  
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod) /  
datasets/raw/DEMOG  
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod) /versions/  
1234/datasets/raw  
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod) /versions/  
1234/datasets/raw/DEMOG  
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod) /subjects/  
ABAN8/datasets/raw  
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod) /subjects/  
ABAN8/datasets/raw/DEMOG
```

Optional query string parameters:

Parameter	Description	Mandatory?	Notes
start={start-datetime}	Returns an incremental dataset from this start datetime.	No	<p>See retrieving incremental ODM datasets.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> Note: The {start-datetime} value should be greater than 1753-01-01T00:00:00. </div>
versionitem={version-suffix}	Adds the subject's CRF version to the dataset, and identifies it with {version-suffix}.	No	See including CRF version
decodesuffix={decode-suffix}	Add decoded values of items that have an associated code list to the dataset, and identifies these values with {decode-suffix}.	No	See including code and decode values for data dictionaries
codelistsuffix={codelists-suffix}	Adds code list OIDS for fields that use a code list to the dataset, and identifies these values with {codelists-suffix}.	No	See including code and decode values for data dictionaries

Parameter	Description	Mandatory?	Notes
rawsuffix={raw-suffix}	Adds raw data values to a full or incremental dataset, and identifies these values with {raw-suffix}.	No	See including raw values in the regular service
stdsuffix={std-suffix}	Adds standard and unit data values to a full or incremental dataset, and identifies these values with {std-suffix}.	No	See including standard and unit values in the regular service

**Note:**

- The list of parameters in {query-string-parameters} must begin with ? and parameters must be separated by &.
- Any combination of these parameters is permitted, in any order.

The Response

Header	Reason	Body	Notes
200	Dataset retrieved	ODM representation of the dataset	See Response Note for more information on the returned ODM.
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See error responses - the complete list

Header

The header contains the following elements:

- HTTP Response code
- Content-Type : "text/xml"
- X-MWS-CV-Last-Updated : {ISO 8601 format date time}



Note:

- A timestamp of when the clinical view was last updated is returned in the response header.
- See also [clinical view service limitations](#).

Body

- For cumulative datasets the body is [a valid ODM 1.3 snapshot document](#)
- For incremental datasets the body is [a valid ODM 1.3 transactional document](#) - see [retrieving incremental ODM datasets](#)
- When a form contains only standard fields the ItemGroupRepeatKey attribute of the ItemGroupData element is not used.
- When a form contains only log-line fields the ItemGroupRepeatKey attribute of the ItemGroupData element is included in the dataset.
- When a form contains both standard and log-line fields the non-log-line ItemData elements are repeated with the same values for every ItemData element within the ItemGroupData element.
- RWS can export partial dates as long as Rave is configured to allow partial dates and the missing part of the date is indicated in Rave with UN or UNK. If a partial date does not contain UN or UNK, RWS exports it as null.

As an example partial date, the dates "13 UNK 2010" and "13 JAN" will be represented as follows:

```
...
<ItemData ItemOID="Date2" Value="2010--" />
<ItemData ItemOID="Date3" IsNull ="Yes" />
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Example success response

Header:

```
HTTP Response code : 200 OK
Content-Type : "text/xml"
X-MWS-CV-Last-Updated : "2010-04-22T12:46:02"
```

Body:

```
...
<ClinicalData StudyOID="TestClinicalViews" MetaDataVersionOID="2012-01-05T00:00:00">
    <SubjectData SubjectKey="CVTest 0001">
        <SiteRef LocationOID="12345" />
        <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1">
            <FormData FormOID="DM" FormRepeatKey="1">
                <ItemGroupData ItemGroupOID="DM">
                    <ItemData ItemOID="DM.BRTHDTC" Value="06 JUL 1978" />
                    <ItemData ItemOID="DM.AGE" Value="31" />
                    <ItemData ItemOID="DM.SEX" Value="MALE" />
                    <ItemData ItemOID="DM.CHILDBEAR" IsNull="Yes" />
                    <ItemData ItemOID="DM.RACE" Value="1" />
                    <ItemData ItemOID="DM.COUNTRY" Value="PAK" />
                </ItemGroupData>
            </FormData>
        </StudyEventData>
    </SubjectData>
</ClinicalData>
...
<!-- further ClinicalData elements -->
...
```

 **Note:** The MetaDataVersionOID attribute of the ClinicalData element shows the date and time of the last update of any clinical view in the study.

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Example error response

Header:

```
HTTP Response code : 404 Not Found
Content-Type : "text/xml; charset=utf-8"
X-MWS-CV-Last-Updated : "2010-04-22T12:46:02"
```

Body:

```
<ODM FileType="Snapshot" FileOID="bd5f3d59-18cf-4f88-913d-316d94539e9a"
CreationDateTime="2013-01-13T09:38:54.851-00:00" ODMVersion="1.3"
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata"
xmlns:xlink="http://www.w3.org/1999/xlink"
mdsol:ErrorDescription="Dataset does not exist [RWS00134]"
xmlns="http://www.cdisc.org/ns/odm/v1.3" />
```

Scenarios

```
Scenario Outline: Cumulative Regular dataset for the whole study
  When I make an HTTP GET request to the URL "<url>"
  Then I should receive a successful response
  And the response should be a valid ODM 1.3 snapshot document
  And the response data should stream
  And the response header should have a X-MWS-CV-Last-Updated value of when the CV
was last updated
  And ODM element in the response should have the following child elements:
    """
<ClinicalData StudyOID="{study oid}(Prod)" MetaDataVersionOID="{version id}">
  <SubjectData SubjectKey="{subject name}">
    <SiteRef LocationOID="{site number}" />
  <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1">
    <FormData FormOID="DM" FormRepeatKey="1">
      <ItemGroupData ItemGroupOID="DM_LOG_LINE">
        <ItemData ItemOID="DM.BRTHDTC" Value="1978-01-29" />
        <ItemData ItemOID="DM.COUNTRY" Value="ENGLAND" />
        <ItemData ItemOID="DM.HEIGHT" Value="5.0" />
      </ItemGroupData>
    </FormData>
  </StudyEventData>
</ClinicalData>
```

```

    </SubjectData>
</ClinicalData>
<ClinicalData StudyOID="{study oid} (Prod)" MetaDataVersionOID="{version id}">
    <SubjectData SubjectKey="{subject name}">
        <SiteRef LocationOID="{site number}" />
        <StudyEventData StudyEventOID="SUBJECT">
            <FormData FormOID="PRIMARY" FormRepeatKey="1">
                <ItemGroupData ItemGroupOID="PRIMARY_LOG_LINE">
                    <ItemData ItemOID="PRIMARY.FIELD1" Value="ANYVALUE" />
                </ItemGroupData>
            </FormData>
        </StudyEventData>
    </SubjectData>
</ClinicalData>
"""

```

Examples:

url	
/studies/{study oid}/datasets/regular	
/studies/{study oid}/versions/{version id}/datasets/regular	
/studies/{study oid}/subjects/{subject name}/datasets/regular	

Scenario Outline: Cumulative Regular dataset for a form inside a folder

When I make an HTTP GET request to the URL "<url>"
 Then I should receive a successful response
 And the response should be a valid ODM 1.3 snapshot document
 And the response data should stream
 And the response header should have a X-MWS-CV-Last-Updated value of when the CV
 was last updated
 And ODM element in the response should have the following child elements:

```

"""
<ClinicalData StudyOID="{study oid} (Prod)" MetaDataVersionOID="{version id}">
    <SubjectData SubjectKey="{subject name}">
        <SiteRef LocationOID="{site number}" />
        <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1">
            <FormData FormOID="DM" FormRepeatKey="1">
                <ItemGroupData ItemGroupOID="DM_LOG_LINE">
                    <ItemData ItemOID="DM.BIRTHDATE" Value="1978-01-29" />
                    <ItemData ItemOID="DM.AGE" Value="32" />
                    <ItemData ItemOID="DM.SEX" Value="M" />
                    <ItemData ItemOID="DM.COUNTRY" Value="ENGLAND" />
                    <ItemData ItemOID="DM.HEIGHT" Value="5.0" />
                    <ItemData ItemOID="DM.DATEOFCONSENT" IsNull="Yes" />
                </ItemGroupData>
            </FormData>
        </StudyEventData>
    </SubjectData>
</ClinicalData>
"""

```

Examples:

url	
-----	--

```
| /studies/{study oid}/datasets/regular/DM
| /studies/{study oid}/versions/{version id}/datasets/regular/DM |
| /studies/{study oid}/subjects/{subject name}/datasets/regular/DM |
```

Scenario Outline: Incremental dataset for a form inside a folder

When I observe the current time

And update the subject with the following data:

folder	folder_transaction	folder_repeat	form	form_transaction	
form_repeat	logline_repeat	logline_transaction	field	value	
field_transaction					
1	Update	1	DM	Update	
Update			BRTHDT	30 JAN 1978	
1	2	Insert	AE	Update	
Insert			AEDESC	BACK PAIN	

And I make an HTTP GET request to the URL "<url>"

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And the response data should stream

And the response header should have a X-MWS-CV-Last-Updated value of when the CV was last updated

And ODM element in the response should have the following child elements:

"""

```
<ClinicalData StudyOID="{study oid} (Prod)" MetaDataVersionOID="{version id}">
  <SubjectData SubjectKey="{subject name}">
    <SiteRef LocationOID="{site number}" />
    <StudyEventData StudyEventOID="SUBJECT">
      <FormData FormOID="AE" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2"
          TransactionType="Insert">
          <ItemData ItemOID="AE.AEYN" Value="YES"/>
          <ItemData ItemOID="AE.AEDESC" Value="BACK PAIN" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
<ClinicalData StudyOID="{study oid} (Prod)" MetaDataVersionOID="{version id}">
  <SubjectData SubjectKey="{subject name}">
    <SiteRef LocationOID="{site number}" />
    <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1">
      <FormData FormOID="DM" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Update">
          <ItemData ItemOID="DM.BRTHDT" Value="1978-01-30" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
```

```
"""
Examples:
|
url
| /studies/{study oid}/datasets/regular?start={observed
time} |
| /studies/{study oid}/versions/{version id}/datasets/regular?start={observed
time} |
| /studies/{study oid}/subjects/{subject name}/datasets/regular?start={observed
time} |

Scenario Outline: Incremental dataset for a form inside a folder
  When I observe the current time
  And update the subject with the following data:
    | folder | folder_transaction | folder_repeat | form      | form_transaction |
form_repeat | field   | value       | field_transaction | 
  | SCREEN | Update      | 1             | DM        | Update      |
1           | BRTHDTCT | 30 JAN 1978 | Update      | 
  | SCREEN | Update      | 1             | DM        | Update      |
1           | SEX      | F            | Update      | 
  And I make an HTTP GET request to the URL "<url>"
  Then I should receive a successful response
  And the response should be a valid ODM 1.3 transactional document
  And the response data should stream
  And the response header should have a X-MWS-CV-Last-Updated value of when the CV
was last updated
  And ODM element in the response should have the following child elements:
  """
<ClinicalData StudyOID="{study oid}(Prod)" MetaDataVersionOID="{version id}">
  <SubjectData SubjectKey="{subject name}">
    <SiteRef LocationOID="{site number}" />
    <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1">
      <FormData FormOID="DM" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Update">
          <ItemData ItemOID="DM.BRTHDTCT" Value="1978-01-30" />
          <ItemData ItemOID="DM.SEX" Value="F" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
"""

Examples:
|
url
|
| /studies/{study oid}/datasets/regular/DM?start={observed
time} |
| /studies/{study oid}/versions/{version id}/datasets/regular/DM?start={observed
time} |
|
```

```

| /studies/{study oid}/subjects/{subject name}/datasets/regular/DM?
start={observed time} |

Scenario Outline: Full dataset for a subject level form
When I observe the current time
And update the subject with the following data:
| form      | form_transaction | form_repeat | logline_repeat | logline_transaction
| field     | value           | field_transaction |           |           |
| AE        | Update          | 1             | 2             | Insert
| AEDESC   | BACK PAIN       | Insert        |               |
And I make an HTTP GET request to the URL "<url>"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And ODM element in the response should have the following child elements:
"""
<ClinicalData StudyOID="{study oid} (Prod)" MetaDataVersionOID="{version id}">
  <SubjectData SubjectKey="{subject name}">
    <SiteRef LocationOID="{site number}" />
    <StudyEventData StudyEventOID="SUBJECT">
      <FormData FormOID="AE" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2"
TransactionType="Insert">
          <ItemData ItemOID="AE.AEYN" Value="YES"/>
          <ItemData ItemOID="AE.AEDESC" Value="BACK PAIN" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
"""

Examples:
|
url
|
| /studies/{study oid}/datasets/regular/AE?start={observed
time}           |
| /studies/{study oid}/versions/{version id}/datasets/regular/AE?start={observed
time}           |
| /studies/{study oid}/subjects/{subject name}/datasets/regular/AE?
start={observed time} |

```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.1.7 Include CRF Version

Service	Retrieve Clinical View Datasets as ODM and Include CRF Version
Category	Request Services
URI	GET {clinical view datasets as ODM URI}?versionitem={version-suffix}[&{other-query-string-parameters}]
Description	Retrieves Clinical View Datasets as ODM and includes the CRF version.

Use Case

I am retrieving my clinical view datasets as ODM and...

I am augmenting my results with the CRF version the data belongs to.



Note:

- Clinical views frequently span multiple CRF versions. You can track which version the data belongs to by adding the CRF version to the results.
- See also [retrieving metadata for CV ODM extensions](#)

My GET Request

URI:

```
GET {clinical view datasets as ODM URI}?versionitem={version-suffix}[&{other-query-string-parameters}]
```

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/datasets/raw?versionitem=VERSION
```



Note: Do not enclose the {versionitem-suffix} value in quotation marks.

[Back to top](#)

The Response

Adding versionitem={version-suffix} to the request query string adds a ItemData element to each form in the response.



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<ItemData ItemDataOID="{field-oid}.{version-suffix}" Value="{crf-version}" />
...
```

Example ODM snippet:

In this example, the data entered into CRF version is 1234 and my request query string includes versionitem=VERSION.



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<ItemGroupData ItemGroupDataOID="DM">
  <ItemData ItemDataOID="DM.VERSION" Value="1234"/>
  <ItemData ItemDataOID="DM.SEX" Value="1"/>
  <ItemData ItemDataOID="DM.DOB" Value="01 Jan 1980"/>
```

```
</ItemGroupData>  
...
```

Note: The MetaDataVersionOID attribute of the ClinicalData element shows the date and time of the last update of any clinical view in the study.

Scenarios

Scenario Outline: Add CRF versions to dataset

When I request the <example dataset> for the form with parameter "versionitem=VERSION"

Then I should receive the <example dataset>

But there should be an ItemData FORM.VERSION for each ItemGroupData that contains a number for the CRF version

Examples:

example dataset	
full raw dataset	
full regular dataset	
incremental raw dataset	
incremental regular dataset	

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

1.5.2 Retrieve Raw or Regular Data

Use Case

I am retrieving my clinical view datasets [as ODM](#) or [as CSV](#) and...

I am retrieving raw, verbatim data values

Or

I am receiving regular, conformant data values

i Note: You can retrieve either raw data (as entered) or regular data (converted to conformant values). The distinction between raw and regular datatypes is particularly important for date and time data, where user data entry errors can occur.

The Response

Body

- Raw datasets show the data values exactly as entered in EDC.
- Regular datasets show data values converted to appropriate datatypes and exclude non-conformant values.
- In both raw and regular datasets field values for data dictionaries are mapped to the coded values, in accordance to the ODM specification.
- In raw datasets, unit dictionary values (for measurement units) are supported so that ItemData elements include the MeasurementUnitRef child element where appropriate.

Example Raw Response ODM document

In this example, the "DM.BRTHDTC" date field value is as entered by the user.

```
<ODM FileType="Snapshot" FileOID="FILE567" CreationDateTime="2008-02-12T07:30:00" ODMVersion="1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" xmlns="http://www.cdisc.org/ns/odm/v1.3">
    <ClinicalData StudyOID="TestClinicalViews" MetaDataVersionOID="30">
        <SubjectData SubjectKey="CVTest 0001">
            <SiteRef LocationOID="12345" />
            <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1">
                <FormData FormOID="DM" FormRepeatKey="1">
                    <ItemGroupData ItemGroupOID="DM">
                        <ItemData ItemOID="DM.BRTHDTC" Value="06 JUL 1978" />
                        <ItemData ItemOID="DM.AGE" Value="31" />
                        <ItemData ItemOID="DM.SEX" Value="MALE" />
                        <ItemData ItemOID="DM.CHILDBEAR" IsNull="Yes" />
                        <ItemData ItemOID="DM.RACE" Value="1" />
                        <ItemData ItemOID="DM.COUNTRY" Value="PAK" />
                    </ItemGroupData>
                </FormData>
            </StudyEventData>
        </SubjectData>
    </ClinicalData>
</ODM>
```

Example Regular Response ODM document

In this example, the "DM.BRTHDTC" date field value is converted into the standard format.

```

<ODM FileType="Snapshot" FileOID="FILE568" CreationDateTime="2008-02-12T08:30:00"
ODMVersion="1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" xmlns="http://
www.cdisc.org/ns/odm/v1.3">
  <ClinicalData StudyOID="TestClinicalViews" MetaDataVersionOID="30">
    <SubjectData SubjectKey="CVTest 0001">
      <SiteRef LocationOID="12345" />
      <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1">
        <FormData FormOID="DM" FormRepeatKey="1">
          <ItemGroupData ItemGroupOID="DM">
            <ItemData ItemOID="DM.BRTHDTC" Value="1978-07-06" />
            <ItemData ItemOID="DM.AGE" IsNull="Yes" />
            <ItemData ItemOID="DM.SEX" Value="MALE" />
            <ItemData ItemOID="DM.CHILDBEAR" IsNull="Yes" />
            <ItemData ItemOID="DM.RACE" Value="1" />
            <ItemData ItemOID="DM.COUNTRY" Value="PAK" />
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
</ODM>
```

1.5.2.1 Include Raw Values in a Regular Service

Service	Retrieve Clinical View Datasets as ODM and Include Raw Values in the Regular Service
Category	Request Services
URI	GET {clinical view datasets as ODM URI}/regular? rawsuffix={raw-suffix}[&{other-query-string-parameters}]
Description	Retrieves Clinical View Datasets as ODM and includes raw data values.

Use Case

I am retrieving my clinical view datasets as ODM and...

I am augmenting my regular service results to include the raw data values as well.

**Note:**

- With this extension, you can retrieve your raw data (as entered) with your regular data (converted to conformant values).
- See also [retrieving metadata for CV ODM extensions](#).

My GET Request

URI:

```
GET {clinical view datasets as ODM URI}/regular?rawsuffix={raw-suffix} [&{other-query-string-parameters}]
```

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/datasets/regular?rawsuffix=.RAW
```

**Note:** Do not enclose the {raw-suffix} value in quotation marks.

The Response

Adding rawsuffix={raw-suffix} to the request query string adds an additional ItemData element to each form in the response.

```
...
<ItemData ItemDataOID="{field-oid}{raw-suffix}" Value="{raw-value}" />
...
```

Example ODM snippet:

In this example my request query string included: rawsuffix=.RAW

```

...
<ItemGroupData ItemGroupDataOID="VS_LOG_LINE">
  ...
    <ItemData ItemOID="VS.VSDTC" Value="2010-01-12" />
    <ItemData ItemOID="VS.VSDTC.RAW" Value="12 JAN 2010" />
  ...
<ItemGroupData ItemGroupDataOID="DM">
  ...

```

 Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Scenario Outline: Cumulative regular dataset for a form including raw values
 When I make an HTTP GET request to the URL "<base dataset url>?rawsuffix=.RAW"
 Then I should receive a successful response
 And the response should be a valid ODM 1.3 snapshot document
 And the response data should stream
 And the response header should have a X-MWS-CV-Last-Updated value of when the CV was last updated

And should have one occurrence of each of the following "ItemData" elements:

ItemOID	Value	IsNull
DM.BRTHDTC	1978-01-29	
DM.BRTHDTC.RAW	29 JAN 1978	
DM.AGE	32	
DM.AGE.RAW	32	
DM.SEX	M	
DM.COUNTRY	ENGLAND	
DM.HEIGHT	5.0	
DM.HEIGHT.RAW	5	
DM.DATEOFCONSENT		Yes
DM.DATEOFCONSENT.RAW		Yes

Examples:

base dataset url	
/studies/{study oid}/datasets/regular/DM	
/studies/{study oid}/versions/{version id}/datasets/regular/DM	
/studies/{study oid}/subjects/{subject name}/datasets/regular/DM	



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.2.2 Include Standard and Unit Values in a Regular Service

Service	Retrieve Clinical View Datasets as ODM and Include Standard and Unit Values in the Regular Service
Category	Request Services
URI	GET {clinical view datasets as ODM URI}/regular?stdsuffix={std-suffix}[&{other-query-string-parameters}]
Description	Retrieves Clinical View Datasets as ODM and includes standard and unit values.

Use Case

I am [retrieving my clinical view datasets as ODM](#) and...

I am augmenting my regular service results to include the standard and unit data values as well



Note:

- With this extension, you can retrieve entered unit, converted standard unit value, and standard unit associated with the unit dictionary field with your regular data.
- See also [retrieving metadata for CV ODM extensions](#).

My GET Request

URI:

```
GET {clinical view datasets as ODM URI}/regular?stdsuffix={std-suffix} [&{other-query-string-parameters}]
```

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/datasets/regular?stdsuffix=_STD
```



Important: Do **not** enclose the {std-suffix} value in quotation marks.

The Response

Adding stdsuffix={std-suffix} to the request query string adds an additional three ItemData elements to each form in the response.

```
...
<ItemData ItemDataOID="{field-oid}_UN" Value="{entered-unit}" />
<ItemData ItemDataOID="{field-oid}{std-suffix}" Value="{std-value}" />
<ItemData ItemDataOID="{field-oid}{std-suffix}_UN" Value="{std-unit}" />
...
```

Example ODM snippet:

In this example, my request query string included: stdsuffix=_STD.

```
...
<ItemGroupData ItemGroupDataOID="DM_LOG_LINE">
  ...
    <ItemData ItemOID="DM.WEIGHT" Value="120" />
    <ItemData ItemOID="DM.WEIGHT_UN" Value="LB" />
    <ItemData ItemOID="DM.WEIGHT_STD" Value="50" />
    <ItemData ItemOID="DM.WEIGHT_STD_UN" Value="KG" />
  ...
<ItemGroupData ItemGroupDataOID="DM">
  ...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Scenario Outline: Full dataset for a form and the whole study

Given the study has the following structure:

unit_dictionary	field	field_data_type	field_length	field_control_type	
SCREEN	DM	BRTHDTC	date		DateTime
WEIGHT	WEIGHT	float	6.2	Text	

And there is a subject in the study with the following data:

form_repeat	field	value	measurement_unit	field_transaction	
1	SCREEN	Update	1	DM	Update
1		BRTHDTC	29 JAN 1978		Insert
1	SCREEN	Update	1	DM	Update
	WEIGHT	50	KG		Insert

When I make an HTTP GET request to the URL "/studies/{study oid}/<url>"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And the response data should stream

And ODM ItemGroupData element in the the clinical view response should have the following child elements for form "DM":

```
"""
<ItemGroupData ItemGroupOID="DM_LOG_LINE">
  <ItemData ItemOID="DM.VERSION" Value="{version id}" />
  <ItemData ItemOID="DM.BRTHDTC" Value="1978-01-29" />
  <ItemData ItemOID="DM.BRTHDTC.RAW" Value="29 JAN 1978" />
  <ItemData ItemOID="DM.WEIGHT" Value="50.00" />
  <ItemData ItemOID="DM.WEIGHT.RAW" Value="50" />
  <ItemData ItemOID="DM.WEIGHT_UN" Value="KG" />
  <ItemData ItemOID="DM.WEIGHT_STD" Value="50.00" />
  <ItemData ItemOID="DM.WEIGHT_STD_UN" Value="KG" />
</ItemGroupData>
"""
```

Examples:

```

| url
|   |
| datasets/regular/DM?
rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD
| versions/{version id}/datasets/regular/DM?
versionitem=VERSION&rawsuffix=.RAW&stdsuffix=_STD
| subjects/{subject name}/datasets/regular/DM?
&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD
| datasets/regular?
rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD
| versions/{version id}/datasets/regular?
```

```
&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD           |
 | subjects/{subject name}/datasets/regular?
rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD           |

Scenario Outline: Incremental dataset for a form and the whole study
  Given the study has the following structure:
    | folder | form | field   | field_data_type | field_length | field_control_type |
unit_dictionary |
    | SCREEN | DM    | BRTHDTC | date          |             | DateTime
|
    | SCREEN | DM    | WEIGHT  | float         | 6.2          | Text
WEIGHT           |
  And there is a subject in the study with the following data:
    | folder | folder_transaction | folder_repeat | form | form_transaction |
form_repeat | field   | value        | measurement_unit | field_transaction |
    | SCREEN | Update       | 1             | DM   | Update
1           | BRTHDTC | 29 JAN 1978 |             | Insert
    | SCREEN | Update       | 1             | DM   | Update
1           | WEIGHT  | 50            | KG   | Insert
And I observe the current time
And I update the subject with the following data:
    | folder | folder_transaction | folder_repeat | form | form_transaction |
form_repeat | field   | value        | measurement_unit | field_transaction |
    | SCREEN | Update       | 1             | DM   | Update
1           | BRTHDTC | 30 JAN 1978 |             | Update
    | SCREEN | Update       | 1             | DM   | Update
1           | WEIGHT  | 110           | LB   | Update
When I make an HTTP GET request to the URL "/studies/{study oid}/{<url>}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And ODM ItemGroupData element in the the clinical view response should have the
following child elements for form "DM":
"""
<ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Update">
  <ItemData ItemOID="DM.VERSION" Value="{version id}" />
  <ItemData ItemOID="DM.BRTHDTC" Value="1978-01-30" />
  <ItemData ItemOID="DM.BRTHDTC.RAW" Value="30 JAN 1978" />
  <ItemData ItemOID="DM.WEIGHT" Value="110.00" />
  <ItemData ItemOID="DM.WEIGHT.RAW" Value="110" />
  <ItemData ItemOID="DM.WEIGHT_UN" Value="LB" />
  <ItemData ItemOID="DM.WEIGHT_STD" Value="50.00" />
  <ItemData ItemOID="DM.WEIGHT_STD_UN" Value="KG" />
</ItemGroupData>
"""

Examples:
  |
url
  |
  | datasets/regular/DM?start={observed}
```

```

time}&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD
| versions/{version id}/datasets/regular/DM?start={observed
time}&versionitem=VERSION&rawsuffix=.RAW&stdsuffix=_STD    |
| subjects/{subject name}/datasets/regular/DM?start={observed
time}&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD    |
| datasets/regular?start={observed
time}&versionitem=VERSION&rawsuffix=.RAW&stdsuffix=_STD
| versions/{version id}/datasets/regular?start={observed
time}&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD    |
| subjects/{subject name}/datasets/regular?start={observed
time}&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD    |

```

Scenario Outline: Full dataset for a form with two fields and the whole study

Given the study has the following structure:

	folder	form	field	field_data_type	field_length	field_control_type	
	unit_dictionary						
	SCREEN	DM	BRTHDTC	date			DateTime
WEIGHT	SCREEN	DM	WEIGHT	float	6.2		Text
TEMP	SCREEN	DM	TEMP	float	6.2		Text

And there is a subject in the study with the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	
	form_repeat	field	value	measurement_unit	field_transaction	
1	SCREEN	Update	1	DM	Update	
		BRTHDTC	29 JAN 1978		Insert	
1	SCREEN	Update	1	DM	Update	
		WEIGHT	50	KG	Insert	
1	SCREEN	Update	1	DM	Update	
		TEMP	100	C	Insert	

When I make an HTTP GET request to the URL "/studies/{study oid}/{url}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And the response data should stream

And ODM ItemGroupData element in the the clinical view response should have the following child elements for form "DM":

```

"""
<ItemGroupData ItemGroupOID="DM_LOG_LINE">
  <ItemData ItemOID="DM.VERSION" Value="{version id}" />
  <ItemData ItemOID="DM.BRTHDTC" Value="1978-01-29" />
  <ItemData ItemOID="DM.BRTHDTC.RAW" Value="29 JAN 1978" />
  <ItemData ItemOID="DM.WEIGHT" Value="50.00" />
  <ItemData ItemOID="DM.WEIGHT.RAW" Value="50" />
  <ItemData ItemOID="DM.WEIGHT_UN" Value="KG" />
  <ItemData ItemOID="DM.WEIGHT_STD" Value="50.00" />
  <ItemData ItemOID="DM.WEIGHT_STD_UN" Value="KG" />
  <ItemData ItemOID="DM.TEMP" Value="100.00" />
  <ItemData ItemOID="DM.TEMP.RAW" Value="100" />
  <ItemData ItemOID="DM.TEMP_UN" Value="C" />

```

```

<ItemData ItemOID="DM.TEMP_STD" Value="100.00" />
<ItemData ItemOID="DM.TEMP_STD_UN" Value="C" />
</ItemGroupData>
"""

```

Examples:

```

| url
|
| datasets/regular/DM?
rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD
| versions/{version id}/datasets/regular/DM?
versionitem=VERSION&rawsuffix=.RAW&stdsuffix=_STD | 
| subjects/{subject name}/datasets/regular/DM?
&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD |
| datasets/regular?
rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD
| versions/{version id}/datasets/regular?
&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD | 
| subjects/{subject name}/datasets/regular?
rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD |

```

Scenario Outline: Incremental dataset for a form with two fields and the whole study
Given the study has the following structure:

	folder	form	field	field_data_type	field_length	field_control_type	unit_dictionary
	SCREEN	DM	BRTHDT	date			DateTime
	SCREEN	DM	WEIGHT	float	6.2		Text
WEIGHT							
	SCREEN	DM	TEMP	float	6.2		Text
TEMP							

And there is a subject in the study with the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	form_repeat	field	value	measurement_unit	field_transaction
1	SCREEN	Update			DM			1		Update
	SCREEN	Update			DM			1		Insert
1										
	SCREEN	Update			KG			50		Update
1										
	SCREEN	Update			DM			1		Insert
1										
	TEMP	Update			C			100		Update
1										

And I observe the current time

And I update the subject with the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	form_repeat	field	value	measurement_unit	field_transaction
1	SCREEN	Update			DM			1		Update
	SCREEN	Update			DM			1		Update
1										
	SCREEN	Update			LB			110		Update
1										
	SCREEN	Update			DM			1		Update
1										
	TEMP	Update			FH			220		Update
1										

When I make an HTTP GET request to the URL "/studies/{study oid}/<url>"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And ODM ItemGroupData element in the the clinical view response should have the following child elements for form "DM":

```
"""
<ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Update">
  <ItemData ItemOID="DM.VERSION" Value="{version id}" />
  <ItemData ItemOID="DM.BIRTHDT" Value="1978-01-30" />
  <ItemData ItemOID="DM.BIRTHDT.RAW" Value="30 JAN 1978" />
  <ItemData ItemOID="DM.WEIGHT" Value="110.00" />
  <ItemData ItemOID="DM.WEIGHT.RAW" Value="110" />
  <ItemData ItemOID="DM.WEIGHT_UN" Value="LB" />
  <ItemData ItemOID="DM.WEIGHT_STD" Value="50.00" />
  <ItemData ItemOID="DM.WEIGHT_STD_UN" Value="KG" />
  <ItemData ItemOID="DM.TEMP" Value="220.00" />
  <ItemData ItemOID="DM.TEMP.RAW" Value="220" />
  <ItemData ItemOID="DM.TEMP_UN" Value="FH" />
  <ItemData ItemOID="DM.TEMP_STD" Value="100.00" />
  <ItemData ItemOID="DM.TEMP_STD_UN" Value="C" />
</ItemGroupData>
"""
```

Examples:

```
|  
url  
|  
| datasets/regular/DM?start={observed  
time}&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD  
| versions/{version id}/datasets/regular/DM?start={observed  
time}&versionitem=VERSION&rawsuffix=.RAW&stdsuffix=_STD |  
| subjects/{subject name}/datasets/regular/DM?start={observed  
time}&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD |  
| datasets/regular?start={observed  
time}&versionitem=VERSION&rawsuffix=.RAW&stdsuffix=_STD  
| versions/{version id}/datasets/regular?start={observed  
time}&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD |  
| subjects/{subject name}/datasets/regular?start={observed  
time}&rawsuffix=.RAW&versionitem=VERSION&stdsuffix=_STD |
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.2.3 Include Standard Coded and Decode Values for Data Dictionaries

Service	Retrieve Clinical View Datasets as ODM and Include Standard Coded and Decode Values for Data Dictionaries
Category	Request Services
URI	GET {clinical view datasets as ODM URI}/regular?stdsuffix={std-suffix}&decodesuffix={decode-suffix} [&{other-query-string-parameters}]
Description	Retrieves Clinical View Datasets as ODM and includes standard coded and decode values for Data Dictionaries.

Use Case

I am *retrieving my clinical view datasets as ODM* and...

I am augmenting my regular service results to include the standard coded and decode values as well for data dictionaries.



Note:

- With this extension, you can retrieve standard value and decode value associated with the data dictionary field with your regular data.
- See also [retrieving metadata for CV ODM extensions](#).

My GET Request

URI

```
GET {clinical view datasets as ODM URI}/regular?stdsuffix={std-suffix}&decodesuffix={decode-suffix} [&{other-query-string-parameters}]
```

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/datasets/regular?stdsuffix=_STD&decodesuffix=_DCD
```



Note: Do not enclose the {std-suffix} and {decode-suffix} value in quotation marks.

The Response

Adding stdsuffix={std-suffix} and decodesuffix={decode-suffix} to the request query string adds an additional three ItemData elements to each form in the response:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<ItemData ItemDataOID="{field-oid}" Value="{entered-decode-value}" />
<ItemData ItemDataOID="{field-oid}{std-suffix}" Value="{coded-std-value}" />
<ItemData ItemDataOID="{field-oid}{decode-suffix}" Value="{entered-decode-value}" />
...
```

Example ODM snippet

In this example, my request query string includes stdsuffix=_STD and decodesuffix=_DCD.



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<ItemGroupData ItemGroupDataOID="DM_LOG_LINE">
  ...
    <ItemData ItemOID="DM.GENDER" Value="FEMALE" />
    <ItemData ItemOID="DM.GENDER_STD" Value="F" />
```

```

<ItemData ItemOID="DM.GENDER_DCD" Value="FEMALE" />
...
<ItemGroupData ItemGroupDataOID="DM">
...

```

Scenarios

Scenario Outline: Regular dataset with stdsuffix and rawsuffix parameters for a field with a data dictionary

Given the study has the following structure:

folder	form	field	field_visible	field_type	field_data_type	field_length	field_control_type	data_dictionary
		SCREEN	DM	GENDER	Yes		Standard	text
7							DropDownList	
								GENDER

And there is a subject in the study with the following data:

folder	folder_transaction	folder_repeat	form	form_transaction	form_repeat	field	value	field_transaction
						SCREEN	Update	
1							1	
								DM
								Update
						GENDER	F	
								Insert

When I make an HTTP GET request to the URL "/studies/{study oid}/datasets/regular/DM?<url>"

Then I should receive a successful response

And the response should be a valid ODM 1.3 <odm_type> document

And the response data should stream

And ODM ItemGroupData element in the the clinical view response should have the following child elements for form "DM":

```

"""
<item_group_data_element>
    <ItemData ItemOID="DM.GENDER" Value="FEMALE" />
    <ItemData ItemOID="DM.GENDER_STD" Value="F" />
    <ItemData ItemOID="DM.GENDER_DCD" Value="FEMALE" />
</ItemGroupData>
"""

```

Examples:

odm_type
url
item_group_data_element
snapshot
decodesuffix=_DCD&rawsuffix=_RAW&stdsuffix=_STD
<ItemGroupData ItemGroupOID="DM_LOG_LINE">
transactional
Start=2013-02-01T12:00:00&decodesuffix=_DCD&rawsuffix=_RAW&stdsuffix=_STD
<ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Insert">

Scenario Outline: Regular dataset with stdsuffix and rawsuffix parameters for a SearchList field with other specify value

Given the study has the following structure:

```

    | folder | form | field | field_visible | field_type | field_data_type |
field_length | field_control_type | data_dictionary |
    | SCREEN | DM | SEX | Yes | Standard | text |
7       | SearchList | GENDER |

```

And there is a subject in the study with the following data:

```

    | folder | folder_transaction | folder_repeat | form | form_transaction |
form_repeat | field | value | field_transaction |
    | SCREEN | Update | 1 | DM | Update |
1       | SEX | OTHER | Insert |

```

When I make an HTTP GET request to the URL "/studies/{study oid}/datasets/regular/DM?<url>"

Then I should receive a successful response

And the response should be a valid ODM 1.3 <odm_type> document

And the response data should stream

And ODM ItemGroupData element in the the clinical view response should have the following child elements for form "DM":

```

"""
<item_group_data_element>
    <ItemData ItemOID="DM.SEX" Value="OTHER" />
    <ItemData ItemOID="DM.SEX_STD" Value="OTHER" />
    <ItemData ItemOID="DM.SEX_DCD" Value="OTHER" />
</ItemGroupData>
"""

```

Examples:

```

| odm_type |
url |
item_group_data_element |
| snapshot |
decodesuffix=_DCD&rawsuffix=_RAW&stdsuffix=_STD |
<ItemGroupData ItemGroupOID="DM_LOG_LINE">
    | transactional |
Start=2013-02-01T12:00:00&decodesuffix=_DCD&rawsuffix=_RAW&stdsuffix=_STD |
<ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Insert"> |

```

Scenario Outline: Regular dataset with only stdsuffix and without rawsuffix parameter

Given the study has the following structure:

```

    | folder | form | field | field_visible | field_type | field_data_type |
field_length | field_control_type | data_dictionary |
    | SCREEN | DM | GENDER | Yes | Standard | text |
7       | DropDownList | GENDER |

```

And there is a subject in the study with the following data:

```

    | folder | folder_transaction | folder_repeat | form | form_transaction |
form_repeat | field | value | field_transaction |
    | SCREEN | Update | 1 | DM | Update |
1       | GENDER | F | Insert |

```

When I make an HTTP GET request to the URL "/studies/{study oid}/datasets/regular/DM?<url>"

Then I should receive a successful response

And the response should be a valid ODM 1.3 <odm_type> document

And the response data should stream
 And ODM ItemGroupData element in the the clinical view response should have the following child elements for form "DM":

```
"""
<item_group_data_element>
  <ItemData ItemOID="DM.GENDER" Value="FEMALE"/>
  <ItemData ItemOID="DM.GENDER_STD" Value="F"/>
  <ItemData ItemOID="DM.GENDER_DCD" Value="FEMALE"/>
</ItemGroupData>
"""
```

Examples:

odm_type url	
item_group_data_element	
snapshot decodesuffix=_DCD&stdsuffix=_STD	
<ItemGroupData ItemGroupOID="DM_LOG_LINE">	
transactional Start=2013-02-01T12:00:00&decodesuffix=_DCD&stdsuffix=_STD	
<ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Insert">	

Scenario Outline: Regular dataset without stdsuffix parameter

Given the study has the following structure:

folder form field field_visible field_type field_data_type	
field_length field_control_type data_dictionary	
SCREEN DM GENDER Yes Standard text	
7 DropDownList GENDER	

And there is a subject in the study with the following data:

folder folder_transaction folder_repeat form form_transaction	
form_repeat field value field_transaction	
SCREEN Update 1 DM Update	
1 GENDER F Insert	

When I make an HTTP GET request to the URL "/studies/{study oid}/datasets/regular/DM?<url>"

Then I should receive a successful response

And the response should be a valid ODM 1.3 <odm_type> document

And the response data should stream

And ODM ItemGroupData element in the the clinical view response should have the following child elements for form "DM":

```
"""
<item_group_data_element>
  <ItemData ItemOID="DM.GENDER" Value="F" />
  <ItemData ItemOID="DM.GENDER_DCD" Value="FEMALE" />
</ItemGroupData>
"""
```

Examples:

odm_type url	
item_group_data_element	
snapshot decodesuffix=_DCD&rawsuffix=_RAW	
<ItemGroupData ItemGroupOID="DM_LOG_LINE">	
snapshot decodesuffix=_DCD	

```
<ItemGroupData ItemGroupOID="DM_LOG_LINE">
    | transactional | Start=2013-02-01T12:00:00&decodesuffix=_DCD&rawsuffix=_RAW |
<ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Insert"> |
    | transactional | Start=2013-02-01T12:00:00&decodesuffix=_DCD |
<ItemGroupData ItemGroupOID="DM_LOG_LINE" TransactionType="Insert"> |
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.2.4 Include Code and Decode Values for Data Dictionaries

Service	Retrieve Clinical View Datasets as ODM and Include Code and Decode Values for Data Dictionaries
Category	Request Services
URI	GET {clinical view datasets as ODM URI}? codelistsuffix={codelists-suffix}&decodesuffix={decode-suffix}[&{other-query-string-parameters}]
Description	Retrieves Clinical View Datasets as ODM and includes code and decode values for Data Dictionaries.

Use Case

I am [retrieving my clinical view datasets as ODM](#) and...

I am augmenting data dictionary selection with the corresponding code OID and decode value.



Note:

- To interpret data dictionary selections, the corresponding code OID and decode value can be added to the results.
- See also [retrieving metadata for CV ODM extensions](#).

My GET Request

URI:

```
GET {clinical view datasets as ODM URI}?codelistsuffix={codelists-suffix}&decodesuffix={decode-suffix} [&{other-query-string-parameters}]
```

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/datasets/raw?codelistsuffix=.CL&decodesuffix=.DECODE
```



Note: Do not enclose the {codelists-suffix} or {decode-suffix} values in quotation marks.

The Response

Adding codelistsuffix={codelists-suffix} to the request query string adds an ItemData element to each form in the response.

```
...
<ItemData ItemDataOID="{field-oid}{codelists-suffix}" Value="{codelist-oid}" />
...
```

Adding decodesuffix={decode-suffix} to the request query string adds an ItemData element to each form in the response.

```
...
<ItemData ItemDataOID="{field-oid}{decode-suffix}" Value="{decode-value}" />
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Example ODM snippets:

In this example, the data dictionary OID is "GENDERS", the selected item decodes to "Male", and my request query string includes decodesuffix=.DECODED

```
...
<ItemGroupData ItemGroupDataOID="DM">
  <ItemData ItemDataOID="DM.SEX" Value="1"/>
  <ItemData ItemDataOID="DM.SEX.DECODED" Value="Male"/>
  <ItemData ItemDataOID="DM.DOB" Value="01 Jan 1980"/>
</ItemGroupData>
...
```

In this example, the data dictionary OID is "GENDERS", the selected item decodes to "Male" and my request query string includes codelistssuffix=.CL

```
...
<ItemGroupData ItemGroupDataOID="DM">
  <ItemData ItemDataOID="DM.SEX" Value="1"/>
  <ItemData ItemDataOID="DM.SEX.CL" Value="GENDERS"/>
  <ItemData ItemDataOID="DM.DOB" Value="01 Jan 1980"/>
</ItemGroupData>
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Scenario Outline: Add decoded values to dataset
When I request the <example dataset> for the form with parameter
"decodesuffix=_DECODE"
Then I should receive the <example dataset>
But there should be an ItemData with suffix "_DECODE" containing the decoded value
for each field with a data dictionary

Examples:

example dataset
full raw dataset
full regular dataset
incremental raw dataset
incremental regular dataset

Scenario Outline: Add code list OIDs to dataset

When I request the <example dataset> for the form with parameter "codelistsuffix=_CL"

Then I should receive the <example dataset>

But there should be an ItemData with suffix "_CL" containing the data dictionary OID for each field with a data dictionary

Examples:

example dataset	
full raw dataset	
full regular dataset	
incremental raw dataset	
incremental regular dataset	



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.2.5 Retrieve Metadata for CV ODM Extensions

Service	Retrieve Metadata for CV ODM Extensions
Category	Request Services
URI	GET {Clinical Datasets Metadata as ODM URI}?versionitem={version-suffix}[&{other-query-string-parameters}]
Description	Retrieves metadata for CV ODM extensions.

Use Case

I am [retrieving my clinical view datasets metadata as ODM](#) and...

I am augmenting my results with ODM extensions.



Note:

- Use this to extend the CV Metadata service to include version, raw value, and data dictionary code OID and decode value.
- Do not enclose the {...-suffix} values in quotation marks.
- See also [including code and decode values for data dictionaries](#), [including CRF version](#), and [including raw values in the regular service](#).

CRF Version Extension

Request URI:

```
GET {Clinical Datasets Metadata as ODM URI}?versionitem={version-suffix} [&{other-query-string-parameters}]
```

ODM snippet:

```
...
<ItemRef ItemOID="{form-oid}.{version-suffix}" Mandatory="No" />
...
<ItemDef OID="{form-oid}.{version-suffix}" Name="{version-suffix}" DataType="integer" Length="8" />
...
```

Example:

```
...
<ItemRef ItemOID="AE.VERSION" Mandatory="No" />
...
<ItemDef OID="AE.VERSION" Name="VERSION" DataType="integer" Length="8" />
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Decoded Values Extension

Request URI:

```
GET {Clinical Datasets Metadata as ODM URI}?decodesuffix={decode-suffix} [&{other-query-string-parameters}]
```

ODM snippet:

```
...
<ItemRef ItemOID="{field-oid}{decode-suffix}" Mandatory="No" />
...
<ItemDef OID="{field-oid}{decode-suffix}" Name="{field-oid}{decode-suffix}"
DataType="text" Length="4000" />
...
```

Example:

```
...
<ItemRef ItemOID="AE.AESYN.DECODED" Mandatory="No" />
...
<ItemDef OID="AE.AESYN.DECODED" Name="AESYN.DECODED" DataType="text" Length="4000" />
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Code List OID Extension

Request URI:

```
GET {Clinical Datasets Metadata as ODM URI}?codelistsuffix={codelists-suffix} [&{other-query-string-parameters}]
```

ODM snippet:

```
...
<ItemRef ItemOID="{field-oid}{codelists-suffix}" Mandatory="No" />
...
<ItemDef OID="{field-oid}{codelists-suffix}" Name="{field-oid}{codelists-suffix}"
DataType="text" Length="50" />
...
```

Example:

```
...  
<ItemRef ItemOID="AE.AESYN.CL" Mandatory="No" />  
...  
<ItemDef OID="AE.AESYN.CL" Name="AESYN.CL" DataType="text" Length="50" />  
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Raw Value Extension

Request URI:

```
GET {Clinical Datasets Metadata as ODM URI}?rawsuffix={raw-suffix} [&{other-query-string-parameters}]
```

ODM snippet:

```
...  
<ItemRef ItemOID="{field-oid}{raw-suffix}" Mandatory="No" />  
...  
<ItemDef OID="{field-oid}{raw-suffix}" Name="{field-oid}{raw-suffix}" DataType="text" Length="11" />  
...
```

Example:

```
...  
<ItemRef ItemOID="AE.AEENDTC.RAW" Mandatory="No" />  
...  
<ItemDef OID="AE.AEENDTC.RAW" Name="AEENDTC.RAW" DataType="text" Length="11" />  
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Scenario: Metadata export with all options

When I make an HTTP GET request to the URL "/studies/{study oid}/datasets/metadata/regular?rawsuffix=.RAW&versionitem=VERSION&decodesuffix=.DECODE&codelistsuffix=.CL"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have one occurrence of each of the following "ItemRef" elements:

ItemOID	
DM.VERSION	
DM.BRTHDTC	
DM.BRTHDTC.RAW	
DM.AGE	
DM.AGE.RAW	
DM.SEX	
DM.SEX.DECODE	
DM.SEX.CL	

And one occurrence of each of the following "ItemDef" elements:

OID	Name	DataType	
DM.VERSION	VERSION	integer	
DM.BRTHDTC	BRTHDTC	datetime	
DM.BRTHDTC.RAW	BRTHDTC.RAW	text	
DM.AGE	AGE	integer	
DM.AGE.RAW	AGE.RAW	text	
DM.SEX	SEX	text	
DM.SEX.DECODE	SEX.DECODE	text	
DM.SEX.CL	SEX.CL	text	

Scenario: Blank rawsuffix parameter

When I make an HTTP GET request to the URL "/studies/{study oid}/datasets/metadata/regular?rawsuffix=&versionitem=VERSION&decodesuffix=.DECODE&codelistsuffix=.CL"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have one occurrence of each of the following "ItemRef" elements:

ItemOID	
DM.VERSION	
DM.BRTHDTC	
DM.AGE	
DM.SEX	
DM.SEX.DECODE	
DM.SEX.CL	

But no occurrence of each of the following "ItemRef" elements:

ItemOID	

DM.BRTHDT.C.RAW
DM.AGE.RAW

And one occurrence of each of the following "ItemDef" elements:

OID Name DataType
DM.VERSION VERSION integer
DM.BRTHDT.C BRTHDT.C datetime
DM.AGE AGE integer
DM.SEX SEX text
DM.SEX.DECODE SEX.DECODE text
DM.SEX.CL SEX.CL text

But no occurrence of each of the following "ItemDef" elements:

OID Name DataType
DM.BRTHDT.C.RAW BRTHDT.C.RAW text
DM.AGE.RAW AGE.RAW text



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.3 Configurable Datasets

Configurable Datasets provide the facility to extend custom datasets in Rave Web Services (RWS). Configurable Datasets allow Rave EDC information to be extracted in CSV, TXT, XML, or JSON format through a custom RWS endpoint.

1.5.3.1 Retrieve Results in Default XML Format

Use Case

I am *retrieving my custom dataset* and...

I am retrieving my results in the default XML format.



Note: When you do not specify the format, data is returned in XML format. One "record" element is returned per data row. Column names are returned as attribute names in the record elements.

Example

```
GET request URL: "/datasets/Coding_dictionaries"
```

Response Body

```
<?xml version="1.0" encoding="utf-8"?>
<datasets>
  <dataset>
    <record DictionaryName="Meddra" DictionaryVersion="11.1"/>
    <record DictionaryName="WhoDrug" DictionaryVersion="5.1"/>
  </dataset>
</datasets>
```

Scenarios

Background:

```
Given I am a user
And there is a configurable dataset with name "testdataset"
```

Scenario Outline: Get data

```
Given the configurable dataset has the recordset "RecordSet1" with the type
"<objecttype>"
```

And the recordset "RecordSet1" has the following data:

DictionaryName	DictionaryVersion
Meddra	11.1
WhoDrug	5.1

When I make a time tracked HTTP GET request to the URL "/datasets/testdataset"

Then I should receive a successful response

And the response should have a header of content type "application/xml"

And the body of the response should be following XML:

"""

```
<?xml version="1.0" encoding="utf-8"?>
<datasets>
  <dataset>
    <record DictionaryName="Meddra" DictionaryVersion="11.1"/>
    <record DictionaryName="WhoDrug" DictionaryVersion="5.1"/>
  </dataset>
</datasets>
```

"""

And the response data should stream

Examples:

objecttype	
view	
stored procedure	

Scenario Outline: No data

Given the configurable dataset has the recordset "RecordSet1" with the type "<objecttype>"

And the recordset "RecordSet1" has no data

When I make an HTTP GET request to the URL "/datasets/testdataset"

Then I should receive a successful response

And the body of the response should be following XML:

```
"""
<?xml version="1.0" encoding="utf-8"?>
<datasets>
    <dataset />
</datasets>
"""
```

And the response should have a header of content type "application/xml"

And the response data should stream

Examples:

objecttype
view
stored procedure

Scenario: Dataset does not exist

When I make an HTTP GET request to the URL "/datasets/dataset_that_does_not_exist"

Then I should receive an HTTP response with status code "Not Found"

Scenario Outline: Database view or store procedure does not exist

Given the configurable dataset has the recordset "RecordSet1" with the type "<objecttype>"

And the recordset "RecordSet1" is pointing to a non-existent "<objecttype>"

When I make an HTTP GET request to the URL "/datasets/testdataset"

Then I should receive an HTTP response with status code "InternalServerError"

Examples:

objecttype
view
stored procedure

1.5.3.2 Retrieve Paged Results

Use Case

I am *retrieving my custom dataset* and...

I am retrieving my data paged over multiple requests.



Note: Datasets configured to support paging provide paging links in the response header. The integrating system can check the header of the response for paging links, and follow the "next" link if it is present.

Examples

Request URL examples:

```
GET  "/datasets/GetData?study=Mediflex"
GET  "/datasets/GetData?study=Mediflex&page=29"
GET  "/datasets/GetData?study=Mediflex&page=29&per_page=100"
```

Response Headers:

```
Link:      <https://api.mdsol.com/users?page=28&per_page=100&filters >;
rel="previous",
          <https://api.mdsol.com/users?page=30&per_page=100&filters >; rel="next"
```

or

```
Link:      <https://api.mdsol.com/users?page=28&per_page=100&filters >;
rel="previous",
          <https://api.mdsol.com/users?page=30&per_page=100&filters >; rel="next",
          <https://api.mdsol.com/users?page=1&per_page=100&filters >; rel="first",
          <https://api.mdsol.com/users?page=100&per_page=100&filters >; rel="last"
```



Note:

- Datasets configured to support paging serve up paged results even when no query string pagination parameters are supplied in the request URL.
- You can configure datasets to make the @page parameter optional, so long as the stored procedure implements its own paging mechanism, for example, an audit identifier. The stored procedure must add this custom parameter to the pagination metadata table.

Scenarios

Background:

Given I am a user
 And there is a pageable configurable dataset with name "GetData"
 And there is a stored procedure which supports pagination that returns two recordsets
 And the second recordset has 20 records

Scenario Outline: Expected behaviour when paging is enabled or disabled

Given the first recordset has the following data:

PaginationKey	PaginationValue
page	1
per_page	5
more_pages	true

And there is a non-pageable configurable dataset with name "GetNonPagedData"
 And there is a stored procedure which does not support pagination that returns one recordset
 And paging is <enabled_or_disabled>
 When I make an HTTP GET request to the URL "<dataset_query>"
 Then I should receive data with <count> records
 And the response data should stream after the headers have been sent

Examples:

dataset_query	enabled_or_disabled	count
/datasets/GetNonPagedData?study=Mediflex	not enabled	20
/datasets/GetData?study=Mediflex	enabled	5

Scenario Outline: Appropriate links appear in the HTTP header

Given the first recordset has the following data:

PaginationKey	PaginationValue
page	<page>
per_page	5
more_pages	<more_pages>
total_count	<total_count>

And paging is enabled
 When I make an HTTP GET request to the URL "/datasets/GetData?
 study=Mediflex&page=<page>"
 Then the body of the response should contain data
 And the response data should stream after the headers have been sent
 And the header should contain <link_key> link key
 And the header should not contain <invalid_link_keys> link keys

Examples:

page	more_pages	total_count	link_key	
invalid_link_keys				
1	true		next	first, previous,
last				
2	true		next, previous	first,
last				
4	false		previous	first, next,

last					
1	true	20	first, next, last		
previous					
2	true	20	first, previous, next, last		
4	false	20	first, previous		next,
last					

Scenario Outline: Appropriate URIs in the HTTP header for each pagination link

Given the first recordset has the following data:

PaginationKey	PaginationValue	
page	<page>	
per_page	5	
more_pages	<more_pages>	
total_count	<total_count>	

And paging is enabled

And the default page size for the configurable dataset is set to 5

When I make an HTTP GET request to the URL "/datasets/GetData?

study=Mediflex<pagination_query>"

Then I should receive one page containing 5 records

And the response data should stream after the headers have been sent

And the header should contain <link_key> with <link_value>

Examples:

link_value	more_pages	total_count	pagination_query	link_key	
&page=1&per_page=5	true	20		next	
&page=2&per_page=5	true	20	&page=1	next	
&page=3&per_page=5	true	20	&page=1&per_page=5	next	
&page=4&per_page=5	false	20	&page=2	next	
&page=1&per_page=5	true	20	&page=4	previous	
&page=2&per_page=5	true	20		first	
&page=3&per_page=5	true	20		next	
&page=4&per_page=5	true	20		last	
&page=1&per_page=5	true	20	&page=1	first	
&page=2&per_page=5	true	20	&page=1	next	
&page=3&per_page=5	true	20	&page=1	last	
&page=1&per_page=5	true	20	&page=1&per_page=5	first	

1 true 20 &page=1&per_page=5 next
&page=2&per_page=5
1 true 20 &page=1&per_page=5 last
&page=4&per_page=5
2 true 20 &page=2 first
&page=1&per_page=5
2 true 20 &page=2 previous
&page=1&per_page=5
2 true 20 &page=2 next
&page=3&per_page=5
2 true 20 &page=2 last
&page=4&per_page=5
4 false 20 &page=4 first
&page=1&per_page=5
4 false 20 &page=4 previous
&page=3&per_page=5

Scenario Outline: Expected page size

Given the first recordset has the following data:

PaginationKey	PaginationValue
page	1
per_page	<per_page_rs>
more_pages	true

And paging is enabled

And the default page size for the configurable dataset is set to 5

And the maximum page size for the configurable dataset is set to 15

And the minimum page size for the configurable dataset is set to 3

When I make an HTTP GET request to the URL "/datasets/GetData?

study=Mediflex<pagination_query>"

Then I should receive one page containing <actual_size> records

And the response data should stream after the headers have been sent

Examples:

pagination_query	per_page_rs	actual_size
	5	5
&per_page=8	8	8
&per_page=20	15	15
&per_page=2	3	3
&per_page=10	10	10

Scenario Outline: Failure when paging parameters are not expected

Given paging is not enabled

And there is a non-pageable configurable dataset with name "GetNonPagedData"

And there is a stored procedure which does not support pagination that returns one recordset

When I make an HTTP GET request to the URL "/datasets/GetNonPagedData?

study=Mediflex<pagination_query>"

Then I should receive an HTTP response with status code "<error>"

Examples:

pagination_query	error	
&page=2	BadRequest	
&per_page=15	BadRequest	
&page=2&per_page=15	BadRequest	

Scenario Outline: Failure when paging parameters are not integers

Given paging is enabled

When I make an HTTP GET request to the URL "/datasets/GetData?

study=Mediflex<pagination_query>"

Then I should receive an HTTP response with status code "<error>"

Examples:

pagination_query	error	
&per_page=?#!	BadRequest	
&per_page=abc	BadRequest	

Scenario: Minimum default per page size when negative @per_page is passed in the url query string

Given the first recordset has the following data:

PaginationKey	PaginationValue	
page	1	
per_page	5	
more_pages	true	
total_count		

And paging is enabled

And the minimum page size for the configurable dataset is set to 3

When I make an HTTP GET request to the URL "/datasets/GetData?

study=Mediflex&per_page=-88"

Then I should receive a successful response

And I should receive one page containing 3 records

And the response data should stream after the headers have been sent

And the header should contain next with &per_page=5&page=2

Scenario Outline: PageParameterRequired set to true for a CDS SP which requires page param

Given the background

But the page parameter of the configurable dataset "GetData" is required

And the first recordset has the following data:

PaginationKey	PaginationValue	
page	1	
per_page	5	
more_pages	true	

And paging is enabled

When I make an HTTP GET request to the URL "/datasets/GetData?

study=Mediflex&per_page=10<page_param>"

Then I should receive a successful response

And I should receive one page containing 10 records

And the response data should stream after the headers have been sent

And the header should contain next with &per_page=5&page=2

Examples:

page_param	
&page=1	

Scenario: Without page param in url query when PageParameterRequired set to false for a CDS SP which does not requires page param

Given the background

But there is a pageable configurable dataset with name "GetDataWithoutPageParam" which does not require page param

And the page parameter of the configurable dataset "GetDataWithoutPageParam" is not required

And the first recordset has the following data:

PaginationKey	PaginationValue
page	1
per_page	5
more_pages	true

And paging is enabled

When I make an HTTP GET request to the URL "/datasets/GetDataWithoutPageParam?study=Mediflex&per_page=10"

Then I should receive a successful response

And I should receive one page containing 10 records

And the response data should stream after the headers have been sent

And the header should contain next with &per_page=5

Scenario: With page in url query when PageParameterRequired is set to false for a CDS SP which does not requires page param

Given the background

But there is a pageable configurable dataset with name "GetDataWithoutPageParam" which does not require page param

And the page parameter of the configurable dataset "GetDataWithoutPageParam" is not required

And the first recordset has the following data:

PaginationKey	PaginationValue
page	1
per_page	5
more_pages	true

And paging is enabled

When I make an HTTP GET request to the URL "/datasets/GetDataWithoutPageParam?study=Mediflex&per_page=10&page=1"

Then I should receive a "BadRequest" response code

Scenario Outline: PageParameterRequired set to true for a CDS SP which does not require page param

Given the background

But there is a pageable configurable dataset with name "GetDataWithoutPageParam" which does not require page param

And the page parameter of the configurable dataset "GetDataWithoutPageParam" is required

And the first recordset has the following data:

```

| PaginationKey | PaginationValue |
| page          | 1                  |
| per_page       | 5                  |
| more_pages     | true               |
And paging is enabled
When I make an HTTP GET request to the URL "/datasets/GetDataWithoutPageParam?
study=Mediflex&per_page=10<page_param>"  

Then I should receive a "BadRequest" response code

```

Examples:

```

| page_param |
|             |
| &page=1    |

```

Scenario: Request with page param where PageParameterRequired is false for a CDS SP which requires page param

Given the background

But the page parameter of the configurable dataset "GetData" is not required

And the first recordset has the following data:

```

| PaginationKey | PaginationValue |
| page          | 1                  |
| per_page       | 5                  |
| more_pages     | true               |

```

And paging is enabled

```

When I make an HTTP GET request to the URL "/datasets/GetData?
study=Mediflex&per_page=10&page=1"

```

Then I should receive a successful response

And I should receive one page containing 10 records

And the response data should stream after the headers have been sent

And the header should contain next with &per_page=5

Scenario: Request without page param where PageParameterRequired is false for a CDS SP which requires page param

Given the background

But the page parameter of the configurable dataset "GetData" is not required

And the first recordset has the following data:

```

| PaginationKey | PaginationValue |
| page          | 1                  |
| per_page       | 5                  |
| more_pages     | true               |

```

And paging is enabled

```

When I make an HTTP GET request to the URL "/datasets/GetData?
study=Mediflex&per_page=10"

```

Then I should receive an "InternalServerError" response code

[Back to top](#)

1.5.3.3 Retrieve Lab Data with the Lab Analyte Ranges CSV Dataset

Service	Retrieve Lab Data with the Lab Analyte Ranges CSV Dataset
Category	ODM Adapter
URI	GET https://{{host}}/RaveWebServices/datasets/LabAnalyteRanges.csv?studyoid={{study_oid}}
Description	Retrieves the local lab analyte ranges for a study across all sites.

Use Case

I am retrieving lab data with the lab analyte ranges CSV dataset and ...

As a consumer of lab data

I want to get the local lab analyte ranges for a study across all sites.



Note: This service only outputs local lab data, and lab analyte unit ranges only.

Assumptions

To succeed, I have satisfied this pre-requisite:

I am a valid Rave User.

My GET Request

Header

My header contains the following element:

1. [My authentication](#)

URI

```
GET https://host/RaveWebServices/datasets/LabAnalyteRanges.csv?studyoid={study_oid}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
-----------	-------------	------------	-------

{host}	The host name	Yes	usually "{client}.mdsol.com"
studies/{study-oid}	The study OID, example: "Mediflex(Prod)"	Yes	a URL-escaped string

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/LabAnalyteRanges.csv?  
studyoid=Mediflex(Prod)
```

The Response

Header	Reason	Body	Notes
200	Dataset retrieved	CSV representation of the dataset	
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See The error Responses Listing

Header

The response header contains the following elements:

- Content-type: "text/csv"
- HTTP response code

Body

The response Body is a CSV format document.

```
SiteNumber,LabName,AnalyteName,FromDate,ToDate,LowRange,HighRange,Units,Dictionary,Comments  
{site number},{lab name},{analyte name},{from date},{to date},{low range},{high range}, {units},{dictionary},{comments}
```

Example success response

Header:

```
HTTP Response code : 200 OK  
Content-Type : "text/csv"
```

Body:

```
SiteNumber,LabName,AnalyteName,FromDate,ToDate,LowRange,HighRange,Units,Dictionary,Comments  
12345,my_local_lab,test_analyte_rws_101,,,0,10,my_unit_rws_101,,  
12345,my_local_lab_2,test_analyte_rws_101,,,0,10,my_unit_rws_101,,  
EOF
```

Scenarios

```
Background:  
    Given I am a Rave user  
    And there is a valid lab configuration  
    And there is a study with the following structure:  
        | form           | field             | field_data_type   | field_length |  
field_control_type | clinical_significance |  
        | LAB_DATA      | MY_VARIABLE_RWS_101 | integer          | 3            |  
Text              | Yes               |  
    And there is a subject in the study
```

And there is a dataset called "LabAnalyteRanges"

Scenario: Local Lab is setup for one site

Given the local labs have been setup for the site

And Analyte Ranges have been configured for one site

When I make an HTTP GET request to the URL "/datasets/LabAnalyteRanges.csv?studyoid={study_oid}"

Then I should receive a successful response

And the response header should have a content type of "text/csv"

And the response body should contain the lab analyte ranges

1.5.3.4 Retrieve Clinical Data with the Clinical Audit Records Dataset

The main `ClinicalAuditRecords` preinstalled custom dataset provides access to the Rave EDC audit trail to feed every change that was applied to every field through to a client's data warehouse. It uses [pagination](#) to page the large amount of data over multiple requests in manageable sized chunks, and includes the following:

- Audits at the Subject, Folder, Form, Record, and Field levels
- IFreeze, Verify, and Lock status changes at field level
- Queries and Comments



Note:

- From Rave EDC 2022.3.0 onwards, the ODM Adapter V2 Retrieve Clinical Data with the Clinical Audit Records Dataset has introduced an optional query string parameter, `mode`, to allow you to extract more ASCs.
- The historical audits of all the additional ASCs get populated in ODM Adapter Version 2 with an included backfill SQL job.
- The current enhancement becomes enabled only after the backfill job completes.
- The Retrieve Clinical Data with the Clinical Audit Records Dataset service returns an error, if you make a call for more ASCs before the backfill job completes.
- After the backfill job completes, you can use `mode=enhanced` to extract all the additional ASCs in your CAR extract, and verify that there are no gaps in your downstream systems.
- After the historical audits are populated, you can start using `mode=all` to extract all the ASCs available for Rave clinical data.

- For more details see the **URI Parameters** section, and contact your Medidata representative.

Service	Retrieve Clinical Data with the Clinical Audit Records Dataset
Category	ODM Adapter
URI	<p>Up to Rave 2019.2.x: GET <code>https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}</code></p> <p>Up to Rave 2022.2.x: GET <code>https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}&unicode={{unicode}}</code></p> <p>From Rave 2022.3.0 onwards: GET <code>https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}&unicode={{unicode}}&mode={{mode}}</code></p>
Description	Retrieves the clinical data transactions using an auto-paginating, preinstalled custom dataset.

Use Case

I am retrieving clinical transaction data with the ODM Adapter and ...

I am retrieving the clinical data transactions using this auto-paginating, preinstalled custom dataset.

Example

Assumptions

To succeed, I have satisfied these prerequisites:

1. I am a valid Rave User

2. I am a power user or similar with full view-access to my study
3. I can follow the "next" link in the response header to get the next page.

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

Up to Rave 2019.2.x:

```
GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}
```

From Rave 2019.2.x onwards:

```
GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}&unicode={{unicode}}
```

From Rave 2022.3.0 onwards:

```
GET https://{{host}}/RaveWebServices/datasets/ClinicalAuditRecords.odm?studyoid={{study-oid}}&startid={{start-id}}&per_page={{per-page}}&unicode={{unicode}}&mode={{mode}}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
{study-oid}	The study name and environment, for example: "Mediflex(Prod)"	Yes	a URL-escaped string
{start-id}	The audit ID to start from	No	see below
{per-page}	The amount of audits to consider for export	No	see below
{unicode}	Boolean parameter you can use to specify if Unicode characters are required in the response.	No	Available from Classic Rave 2019.2.0 ODM Adapter V2 onwards.
{mode}	Allows you to extract more ASCs.	No	<p>Available from Rave EDC 2022.3.0 ODM Adapter V2 onwards.</p> <p>Available values:</p> <ul style="list-style-type: none"> • mode=default – Extracts the existing Included Audit Subcategories supported by ODM Adapter in Rave version 2022.2.0 and lower. • mode=enhanced – Extracts only the additional

Parameter	Description	Mandatory?	Notes
			<p>ASCs that were not originally supported in Rave version 2022.2.0 and earlier.</p> <ul style="list-style-type: none">• mode=all – Extracts all the ASCs currently available in Rave for EDC clinical data, including both default and enhanced.

Examples:

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/  
ClinicalAuditRecords.odm?studyoid=Mediflex(Prod)  
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/  
ClinicalAuditRecords.odm?  
studyoid=Mediflex(Prod) &startid=4736745&per_page=10000&unicode=true
```

Making requests to the Clinical Audit Records standard dataset

The Clinical Audit Records standard dataset has been designed and optimized to work through a URL's audit trail to retrieve the audit data for a single study. All the audit entries in one Rave URL are unique and sequential.

Clinical Audit Record extraction for a study may have nonsequential SourceID (AuditID) because different users could modify clinical data across many studies, at any given time. Each audit ID corresponds to the AuditRecord element in ODM. AuditRecord has several child elements, and describes the audit data.

- SourceID is the AuditID in Rave.
- The UserRef element has the UserID attribute, whose value represents the user who entered the audit.
- The LocationRef element has the LocationOID attribute, which has the value of Site number.

**Important:**

- The LocationOID attribute is derived from the Site Number.
- Because the SiteNumber can change, the LocationRef may change as well.
- Hence, the contents of the audit record can vary between different transactions.

- The DateTimeStamp element represents the database audit timestamp.
- The ReasonForChange element represents the reason for the audit.

Audit ODM:

```
<ClinicalData StudyOID="RWS_study_with_folders" MetaDataVersionOID="37"  
mdsol:AuditSubCategoryName="Entered">
```

```

<SubjectData SubjectKey="007522fb-a22b-425d-b3b9-da5df80203bd"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="Sub1">
    <SiteRef LocationOID="12345"/>
    <StudyEventData StudyEventOID="SCREEN"
StudyEventRepeatKey="SCREEN[1]" mdsol:InstanceId="275" mdsol:InstanceName="Screening">
        <FormData FormOID="EN" FormRepeatKey="1"
mdsol:DataPageId="1383">
            <ItemGroupData ItemGroupOID="EN">
                <ItemData ItemOID="EN.SUBJECTNAME"
TransactionType="Upsert" Value="SOME NAME">
                    <AuditRecord>
                        <UserRef UserID="defuser"/>
                        <LocationRef LocationOID="12345"/>
                        <DateTimeStamp>2025-02-20T17:06:20</
DateTimeStamp>
                        <ReasonForChange/>
                        <SourceID>207336</SourceID>
                    </AuditRecord>
                </ItemData>
                </ItemGroupData>
            </FormData>
        </StudyEventData>
    </SubjectData>
</ClinicalData>

```

Dynamic features:

1) Getting up to date and staying up to date with Paginated requests

Pagination is employed to break up the large amount of data over multiple requests into manageable sized chunks. The "next" link in the response header should be followed after the initial request, and for all subsequent calls until a request is received with no "next" link in the response header.

An accumulative request followed by incremental requests

Typically, a synchronization will start with a series of paged requests to the dataset until all data is fetched. Then on some schedule - perhaps daily - the client will make a subsequent series of paged requests, starting at the next point in the audit trail.

- The first-ever request should not include pagination items in the query string.
- Receiving a response that contains no "next" link in the header indicates that all data has been received.
- The last response contains an empty ODM element, with no child elements.

- After each request the Maximum received AuditID must be captured, to determine where to start from for the next scheduled synchronization.

2) Controlling the request size with per_page

Request size is controlled by means of the 'per_page' value in the request query string.

The 'per-page' parameter is the size of the search range. 'per_page' controls the number of audit records *considered* for extraction. For performance reasons, ClinicalAuditRecords will search for all the audits within the 'per-page' range.

After optimization, the number of results returned is not equal to the 'per_page' value.



Note: The acceptable 'per_page' range is between 1000 to 1000000. If the ODM Adaptor is taking longer than you expect, lower the 'per_page' parameter value to 1000.

Example

The Response

The whole request will result in either:

1. SUCCESS: Return an HTTP Response code of 200 OK - with all the ODM data (the ODM element is closed).
2. FAIL: Return an HTTP Response code of 4xx or 5xx. Further details are logged in the database.

HTTP response code	Reason	Body	Notes
200	The transaction was successful	The retrieved data in ODM format " ... " "	(see below)
4xx	Request Failure	n/a (Details will be logged in the database)	(see below)
5xx	Service Failure	n/a (Details will be logged in the database)	(see below)

Handling Errors

We strongly advise that clients parse the ODM received to check that the ODM element is closed. An unclosed ODM element indicates that not all the streamed data was received. Some retry logic should be in place for this possibility.

Error reporting for *custom datasets* is limited to just the HTTP response code such as 400 or 500. A single HTTP response code can have any number of causes. Upon receiving such errors, further investigation is advised.

Header

The response header contains the following elements:

- Content-type: "text/xml"
- HTTP response code
- Pagination links

Body

The response Body is *a valid ODM 1.3 transactional document*.

Note: If your `StudyEnvironemntSiteNumber` is defined in Rave, then RWS returns it and provides it in ODM next to `LocationRef` and `SiteRef` in the `mdsol:StudyEnvSiteNumber` attribute.

The response ODM document is constructed as follows:

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM ODMVersion="1.3" FileType="Transactional" FileOID="8f8ec541-6694-4609-93b5-c13e83f730c9" CreationDateTime="2019-11-19T21:47:33" xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">
<ClinicalData StudyOID="RWS_study_with_folders" MetaDataVersionOID="376"
mdsol:AuditSubCategoryName="SubjectCreated">
<SubjectData SubjectKey="e0b9a535-0cff-4f88-b40e-ecdb66a01648"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="ASUBJECT"
TransactionType="Upsert" >
<AuditRecord>
<UserRef UserOID="defuser"/>
<LocationRef LocationOID="12345" mdsol:StudyEnvSiteNumber="9000" />
<DateTimeStamp>2015-09-18T20:01:18</DateTimeStamp>
<ReasonForChange></ReasonForChange>
<SourceID>846036</SourceID>
</AuditRecord>
<SiteRef LocationOID="12345" mdsol:StudyEnvSiteNumber="9000" />
</SubjectData>
</ClinicalData>
</ODM>
```

Attributes for this feature:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema for more details.](#)

Attribute or element	Description	Mandatory?	Example
UserOID="{user-oid}"	The User OID is the User's Rave Login. This user performed the action, which created the audit record.		"JBlogs"
LocationOID="{location-oid}"	The Location OID. Maps to site number.  Note: The LocationOID is case-sensitive.		"1234"
{time-stamp}	The timestamp of the audit record.		2012-12-06T09:45:39
{reason-for-change}	The reason for the audit record.		"Enrolled"
{source-id}	The database key for the Audit record.		"1234"
{record-id}	The database key or ID for the record.		"1570"
{instance-id}	The database key or ID for the instance.		"154"
{datapage-id}	The database key or ID for the datapage.		"930"
{mdsol:Freeze="{Yes No}"}	This attribute is included if the audit record changes the Freeze status.		mdsol:Freeze="Yes"

Attribute or element	Description	Mandatory?	Example
{mdsol:Verify="{Yes No}"}	This attribute is included if the audit record changes the Verify status.		mdsol:Verify="Yes"
{mdsol:Lock="{Yes No}"}	This attribute is included if the audit record changes the Lock status.		mdsol:Lock="Yes"
mdsol:AuditSubCategoryName	The Audit SubCategory Name.		"Verify"
mdsol: Query	The query in Rave.		
QueryRepeatKey="{query id}"	The query ID.		QueryRepeatKey="1"
Response="{response}"	The response from the query.		Response="Please review"
Recipient="{recipient}"	The marking group.		Recipient="Site from System"
Value="{value}"	The query or comment value.		Value="Please reconcile"
Status="{status}"	The query status.		Status="Answered"
mdsol: Comment	The comment in Rave.		
mdsol: Status	The status of the subject.		mdsol:Status="Enrolled"
CommentRepeatKey="{comment id}"	The comment ID in Rave.		CommentRepeatKey="1"
TransactionType= "Insert Update Remove Upsert"	The comment transaction type.		TransactionType= "Insert"

Attribute or element	Description	Mandatory?	Example
Signature	The signature in Rave.		
mdsol:StudyEnvSiteNumber	Indicator for the location of the subject in Rave EDC.		mdsol:StudyEnvSiteNumber="9000"

- Audit records are exported at each object level: for Subjects, Folders, Forms, Records, and Fields.
- Each audit record is represented as a separate ODM ClinicalData element.
- Audits for the same subject are represented as separate elements, and not grouped into a single ClinicalData.
- The TransactionTypes for ItemData elements are 'Upsert' or 'Remove'. It is impossible to reliably distinguish between 'Insert' and 'Update' transactions when sequentially reading through the Rave audit trail.
- The database key for Audit records, AuditID, is exported as SourceID in the ODM AuditRecord.
- Field level Status audits are exported for 'Verify', 'Freeze', 'Lock'.
- Audit records include Protocol Deviations, Queries, Comments, and Signatures.
- For Subject Migrations, the metadata version attribute value will change.
- Records are ordered sequentially by SourceID.

Scenarios

Background:

```

Given I am a Rave user
And there is a study with the following structure:
| folder | form | field | field_data_type | field_control_type |
| VISIT | AE | AEDESC | text | Text |
| VISIT | AE | AESDT | date | DateTime |
And there is a subject in the study with the following data:
| folder | form | field | value |
| VISIT | AE | AEDESC | HEADACHE |
| VISIT | AE | AESDT | 29 JAN 2009 |
And there is a dataset called "ClinicalAuditRecords"
And there is clinical data for the "VISIT" folder
Scenario: Get ClinicalAuditRecords service returns UUID as Subject key
Given the background
And the subject has a UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" not visible in
Rave
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1&per_page=10"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And should have the following ODM elements:
"""
<ClinicalData mdsol:AuditSubCategoryName="SubjectCreated" ...>
<SubjectData SubjectKey="c45f3bda-efd5-400f-aa72-b174dc0c93dd"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="ASUBJECT"
TransactionType="Upsert" >
  <AuditRecord>
    <UserRef .../>
    <LocationRef LocationOID="12345"/>
    <DateTimeStamp>...</DateTimeStamp>
```

```
<ReasonForChange></ReasonForChange>
<SourceID>...</SourceID>
</AuditRecord>
<SiteRef LocationOID="12345" />
</SubjectData>
</ClinicalData>
"""

```

Scenario: Including ReasonForChange

```
Given I update the field "AESDT" in folder "VISIT" with value "26 Feb 2012"
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1&per_page=100"
```

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And should have the following ODM elements:

```
"""

```

```
<ClinicalData mdsol:AuditSubCategoryName="EnteredWithChangeCode" ...>
  <ItemData ItemOID="AE.AESDT" TransactionType="Upsert" Value="26 Feb 2012">
    <ReasonForChange>Data Entry Error</ReasonForChange>
  </ItemData>
</ClinicalData>
"""

```

Scenario: Clinical audit request without environment in the studyoid parameter

```
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study name}&startid=1&per_page=100"
```

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And should have the following ODM elements:

```
"""

```

```
<ClinicalData mdsol:AuditSubCategoryName="SubjectCreated" ...>
  ...
</ClinicalData>
"""

```

Scenario Outline: Clinical audit data for Queries where value has more than one pipe [and special characters and comments and protocol deviations]

Given I query the field "AEDESC" in folder "VISIT" with the value "<query_value>"

```
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1&per_page=1000"
```

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And should have the following ODM elements:

```
"""

```

```
  <mdsol:Query Value="<query_output_value>" Status="Open" Recipient="{marking
group}" .../>
"""

```

Examples:

```
| query_value |
```

```
query_output_value |  
      | this is the value\\| | this is the  
value |  
      | this\\| and this\\| and this | this\\| and  
this |  
      | this\\| and this\\| this\\| and this | this\\| and this\\|  
this |  
      | this\\| and this\\| this\\| and this\\| and this | this\\| and this\\| this\\|  
and this |  
      | this \\| and this\\| and this\\| and this\\| and this | this \\| and this\\| and  
this\\| and this |
```

Scenario: Clinical audit record extract must include comments

Given there is a comment "This is a new approximate value." against the field "AEDESC" located in the form "AE" inside folder "VISIT" for the subject

When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?studyoid={study_oid}&startid=1&per_page=100"

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And should have the following ODM elements:

三

```
<ClinicalData mdsol:auditSubCategoryName="CommentAdd" ...>
    <ItemData ItemOID="AE.AEDESC" ...>
        ...
        <mdsol:Comment TransactionType="Insert" Value="This is a new approximate
value." .../>
    </ItemData>
</ClinicalData>
"""
```

Scenario: Clinical Audit data export for inactivated Subject

Given the background

But the subject has been inactivated

And the subject has a UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" not visible in Rave

When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?studyoid={study oid}&startid=1&per_page=10"

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And should have the following ODM elements:

11

```
<SubjectData SubjectKey="c45f3bda-efd5-400f-aa72-b174dc0c93dd"  
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="ASUBJECT"  
TransactionType="Remove">  
    ...  
</SubjectData>  
....
```

Scenario: Inactivated Subject clinical Audit data should be exported

Given the background

But the subject has been inactivated
And the subject has a UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" not visible in Rave
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?studyoid={study_oid}&startid=1&per_page=10"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And should have the following ODM elements:
"""

```
<ClinicalData mdsol:AuditSubCategoryName="SubjectCreated" ...>
<SubjectData mdsol:SubjectName="ASUBJECT" TransactionType="Upsert" ... >
    <AuditRecord>
        ...
    </AuditRecord>
    <SiteRef LocationOID="12345" />
</SubjectData>
</ClinicalData>
```

""

Scenario: Clinical Audit data export for duplicate subject
Given the background
But there is another subject with the same name
And the subject has a UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" not visible in Rave
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?studyoid={study_oid}&startid=1&per_page=10"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And should have the following ODM ClinicalData elements:
"""

```
<ClinicalData mdsol:AuditSubCategoryName="SubjectCreated" ...>
<SubjectData mdsol:SubjectName="ASUBJECT" TransactionType="Upsert" ... >
    ...
<ClinicalData mdsol:AuditSubCategoryName="SubjectCreated" ...>
    <SubjectData mdsol:SubjectName="ASUBJECT" TransactionType="Upsert" ... >
    ...
"
```


Scenario Outline: Clinical Audit data export for object name changed
Given the background
But the <object> has been renamed
And the subject has a UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" not visible in Rave
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?studyoid={study_oid}&startid=1&per_page=10"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And should have the following ODM elements:
"""

```
<ClinicalData mdsol:AuditSubCategoryName="" ...>
```

```

<SubjectData ...>
...
</SubjectData>
</ClinicalData>
"""

```

Examples:

object	audit_subcategory_name	
subject	SubjectNameChanged	
instance	ObjectNameChanged	

Scenario Outline: Clinical audit request without studyoid parameter

When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
<studyoid_param>startid=1&per_page=100"
Then I should receive an HTTP response with status code "InternalServerError"

Examples:

studyoid_param	
studyoid=somethinginvalid&	
studyoid=	

Scenario: Clinical audit request where the study is associated with inactive site

Given the background

But the site associated with the study is inactive

When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1&per_page=100"

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And should have the following ODM elements:

```
"""

```

```
<ClinicalData mdsol:AuditSubCategoryName="SubjectCreated" ...>
...
</ClinicalData>
"""

```

Scenario: Clinical audit request by an inactive the user

Given the background

But the user is inactive

When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1&per_page=100"

Then I should receive an HTTP response with status code "Unauthorized"

Scenario: No audit data for new study

Given the background

But there is another study with the following structure:

folder	form	field	field_data_type	field_control_type	
VISIT	AE	AEDESC	text	Text	
VISIT	AE	AESDT	date	DateTime	

And the study is pushed to site "12345"

And there is no clinical data for the study
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And the response body should be an empty ODM document

Scenario: Clinical audit record extract must include signature

Given the background
And I sign and save the data page -/pending
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1&per_page=10"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And should have the following ODM elements:
"""

```
<ClinicalData StudyOID="{study_oid}" mdsol:AuditSubCategoryName="Signed">
    <ItemGroupData ItemGroupOID="AE"  >
        <ItemData ItemOID="AE.AEDST" TransactionType="Upsert"  >
            <Signature>
                <UserRef UserOID="{user}" />
                <LocationRef LocationOID="{}" />
                <SignatureRef SignatureOID="{oid}" />
                <DateTimeStamp>{date time}</DateTimeStamp>
            </Signature>
        </ItemData>
    </ItemGroupData>
"""
```

Scenario Outline: Error when service is requested with invalid start ID
Given the background
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=<startID>&per_page=<perpage>"
Then I should receive an HTTP response with status code "<response_code>"

Examples:

startID	perpage	response_code
-1	10	InternalServerError
a	10	BadRequest
1	abc	BadRequest

Scenario: Calling user not assigned to study
Given the background
But I am not associated with the study
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1"
Then I should receive an HTTP response with status code "InternalServerError"

Scenario: Calling user not assigned to any site
Given the background

```
But I am not assigned to any site
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And the response body should be an empty ODM document
```

```
Scenario: Calling user assigned to a specific site only
Given the background
And the study is pushed to another site "99999"
And there is a subject "BSUBJECT" in site "99999"
But I am not assigned to site "99999"
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And should not have the following ODM elements:
"""
...
<LocationRef LocationOID="99999"/>
...
"""

"""


```

```
Scenario: View all sites permission
Given the study is pushed to another site "99999"
And the study is pushed to another site "99999"
And there is a subject "BSUBJECT" in site "99999"
And I have ViewAllSites permission for the study
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And the response should contain audit data for every site in the study
```

```
Scenario: Get GetAuditData Service returns UUID as Subject key
Given there is a subject in a study
And the subject has a UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" not visible in
Rave
When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1"
Then I should receive a successful response
And the response should be a valid ODM 1.3 transactional document
And should have the following ODM elements:
"""
<ClinicalData mdsol:AuditSubCategoryName="SubjectCreated" ...>
  <SubjectData SubjectKey="c45f3bda-efd5-400f-aa72-b174dc0c93dd"
    mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="ASUBJECT"
    TransactionType="Upsert">
    ...
  </SubjectData>

```

```
</ClinicalData>
"""

```

Scenario Outline: Get ClinicalAuditRecords service must include SubCategories
Given the background

But the field "AEDESC" has been <state>

When I make an HTTP GET request to the URL "/datasets/ClinicalAuditRecords.odm?
studyoid={study_oid}&startid=1&per_page=10"

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And should have the following ODM elements:

```
"""

```

```
<ClinicalData mdsol:AuditSubCategoryName="" ...>
  <ItemData ItemOID="AE.AEDESC" .../>
</ClinicalData>
"""

```

Examples:

state SubCategoryName		
frozen Freeze		
locked Lock		

Scenario: Include InstanceId for every Folder in the audit trail

Given the background

When I make an HTTP GET request to the URL "/datasets/
ClinicalAuditRecords.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And the response body should include the InstanceId for every Folder

Scenario: Include DataPageId for every Form in the audit trail

Given the background

When I make an HTTP GET request to the URL "/datasets/
ClinicalAuditRecords.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And the response body should include the DataPageId for every Form

Scenario: Include RecordId for every Record in the audit trail

Given the background

When I make an HTTP GET request to the URL "/datasets/
ClinicalAuditRecords.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 transactional document

And the response body should include the RecordId for every Record



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.3.4.1 Example - Retrieve Clinical Data with the Clinical Audit Records Dataset

GET Request



Note: The first event for a given study may not be at the beginning of the audit trail.

In this example, the first event is at SourceID = 670. However we are requesting AuditID = 1 and a page size of 1000.

It is not necessary to retrieve multiple empty pages before the first record is reached. The service fast-forwards to the first relevant event and serves up audit records from that point.

Header

```
Authorization : Basic ZGV2OnBhc3N
```

URI

```
http://client.mdsol.com/RaveWebservices/datasets/ClinicalAuditRecords.odm?  
studyoid=Mediflex%28Dev%29&startid=1&per_page=1000
```

The Response

Header

```
HTTP Response code : 200 OK  
Content-Type : "text/xml"  
Link: <https://client.mdsol.com/RaveWebservices/datasets/ClinicalAuditRecords.odm?  
startid=1&per_page=1000&studyoid=Mediflex(Prod)>; rel="previous",  
      <https://client.mdsol.com/RaveWebservices/datasets/ClinicalAuditRecords.odm?  
startid=101&per_page=1000&studyoid=Mediflex(Proc)>; rel="next"
```

Body

ODM Adapter/CAR without SES number

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" ODMVersion="1.3" FileType="Transactional" FileOID="52805c53-34b6-4690-8f0f-825c9410fa16" CreationDateTime="2013-03-27T09:37:30">
  <ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18" mdsol:AuditSubCategoryName="SubjectCreated">
    <SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e" mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="123 AB" TransactionType="Upsert">
      <AuditRecord>
        <UserRef UserID="dev"/>
        <LocationRef LocationOID="12345"/>
        <TimeStamp>2009-12-01T15:27:46</TimeStamp>
        <ReasonForChange/>
        <SourceID>670</SourceID>
      </AuditRecord>
      <SiteRef LocationOID="12345"/>
    </SubjectData>
  </ClinicalData>
  <ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18" mdsol:AuditSubCategoryName="SubjectNameChanged">
    <SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e" mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="123 AB" TransactionType="Upsert">
      <AuditRecord>
        <UserRef UserID="dev"/>
        <LocationRef LocationOID="12345"/>
        <TimeStamp>2009-12-01T15:27:46</TimeStamp>
        <ReasonForChange/>
        <SourceID>671</SourceID>
      </AuditRecord>
      <SiteRef LocationOID="12345"/>
    </SubjectData>
  </ClinicalData>
  <ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18" mdsol:AuditSubCategoryName="SubjectStatusChanged">
    <SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e" mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="123 AB" TransactionType="Upsert" mdsol>Status="Enrolled">
      <AuditRecord>
        <UserRef UserID="dev"/>
        <LocationRef LocationOID="12345"/>
        <TimeStamp>2009-12-01T15:27:46</TimeStamp>
        <ReasonForChange>Enrolled</ReasonForChange>
        <SourceID>672</SourceID>
      </AuditRecord>
```

```
<SiteRef LocationOID="12345"/>
</SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="Entered">
<SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="123 AB">
<SiteRef LocationOID="12345"/>
<StudyEventData StudyEventOID="SUBJECT">
<FormData FormOID="EN" FormRepeatKey="1">
<ItemGroupData ItemGroupOID="EN">
<ItemData ItemOID="EN.SUBJID" TransactionType="Upsert" Value="123">
<AuditRecord>
<UserRef UserOID="dev"/>
<LocationRef LocationOID="12345"/>
<DateTimeStamp>2009-12-01T15:27:46</DateTimeStamp>
<ReasonForChange/>
<SourceID>816</SourceID>
</AuditRecord>
</ItemData>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="Entered">
<SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="123 AB">
<SiteRef LocationOID="12345"/>
<StudyEventData StudyEventOID="SUBJECT">
<FormData FormOID="EN" FormRepeatKey="1">
<ItemGroupData ItemGroupOID="EN">
<ItemData ItemOID="EN.SUBJINIT" TransactionType="Upsert" Value="AB">
<AuditRecord>
<UserRef UserOID="dev"/>
<LocationRef LocationOID="12345"/>
<DateTimeStamp>2009-12-01T15:27:46</DateTimeStamp>
<ReasonForChange/>
<SourceID>817</SourceID>
</AuditRecord>
</ItemData>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="Entered">
<SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e">
```

```
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="123 AB">
    <SiteRef LocationOID="12345"/>
    <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="SCREEN[1]">
        <FormData FormOID="DM" FormRepeatKey="1">
            <ItemGroupData ItemGroupOID="DM">
                <ItemData ItemOID="DM.BIRTHDTC" TransactionType="Upsert" Value="05 May
1983">
                    <AuditRecord>
                        <UserRef UserOID="dev"/>
                        <LocationRef LocationOID="12345"/>
                        <DateTimeStamp>2009-12-01T15:27:46</DateTimeStamp>
                        <ReasonForChange/>
                        <SourceID>818</SourceID>
                    </AuditRecord>
                </ItemData>
            </ItemGroupData>
        </FormData>
    </StudyEventData>
</SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="Entered">
    <SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e">
        <SiteRef LocationOID="12345"/>
        <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="SCREEN[1]">
            <FormData FormOID="DM" FormRepeatKey="1">
                <ItemGroupData ItemGroupOID="DM">
                    <ItemData ItemOID="DM.RACE" TransactionType="Upsert" Value="4">
                        <AuditRecord>
                            <UserRef UserOID="dev"/>
                            <LocationRef LocationOID="12345"/>
                            <DateTimeStamp>2009-12-01T15:27:46</DateTimeStamp>
                            <ReasonForChange/>
                            <SourceID>819</SourceID>
                        </AuditRecord>
                    </ItemData>
                </ItemGroupData>
            </FormData>
        </StudyEventData>
    </SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="Entered">
    <SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e">
        <SiteRef LocationOID="12345"/>
        <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="SCREEN[1]">
            <FormData FormOID="DM" FormRepeatKey="1">
                <ItemGroupData ItemGroupOID="DM">
```

```
<ItemData ItemOID="DM.SEX" TransactionType="Upsert" Value="FEMALE">
  <AuditRecord>
    <UserRef UserID="dev"/>
    <LocationRef LocationOID="12345"/>
    <DateTimeStamp>2009-12-01T15:27:46</DateTimeStamp>
    <ReasonForChange/>
    <SourceID>820</SourceID>
  </AuditRecord>
</ItemData>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="Entered">
  <SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="123 AB">
    <SiteRef LocationOID="12345"/>
    <StudyEventData StudyEventOID="VISIT01" StudyEventRepeatKey="VISIT01[1]">
      <FormData FormOID="VISIT" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="VISIT">
          <ItemData ItemOID="VISIT.DTC" TransactionType="Upsert" Value="03 Feb
2009">
            <AuditRecord>
              <UserRef UserID="dev"/>
              <LocationRef LocationOID="12345"/>
              <DateTimeStamp>2009-12-01T15:27:46</DateTimeStamp>
              <ReasonForChange/>
              <SourceID>821</SourceID>
            </AuditRecord>
          </ItemData>
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="Entered">
  <SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="123 AB">
    <SiteRef LocationOID="12345"/>
    <StudyEventData StudyEventOID="SUBJECT">
      <FormData FormOID="EN" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="EN">
          <ItemData ItemOID="EN.USUBJID" TransactionType="Upsert" Value="123 AB">
            <AuditRecord>
              <UserRef UserID="systemuser"/>
              <LocationRef LocationOID="12345"/>
              <DateTimeStamp>2009-12-01T15:27:46</DateTimeStamp>
```

```
        <ReasonForChange/>
        <SourceID>823</SourceID>
    </AuditRecord>
</ItemData>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="SetVisible">
    <SubjectData SubjectKey="3bddb79c-0023-47bb-8861-5345f8987c7e"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="123 AB">
        <SiteRef LocationOID="12345"/>
    <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="SCREEN[1]">
        <FormData FormOID="DM" FormRepeatKey="1">
            <ItemGroupData ItemGroupOID="DM">
                <ItemData ItemOID="DM.CHILDBEAR" TransactionType="Upsert">
                    <AuditRecord>
                        <UserRef UserOID="systemuser"/>
                        <LocationRef LocationOID="12345"/>
                        <DateTimeStamp>2009-12-01T15:27:46</DateTimeStamp>
                        <ReasonForChange/>
                        <SourceID>824</SourceID>
                    </AuditRecord>
                </ItemData>
            </ItemGroupData>
        </FormData>
    </StudyEventData>
</SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="SubjectCreated">
    <SubjectData SubjectKey="7e8892ba-1e20-4ee2-b23c-63658782fe95"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="456 CD"
TransactionType="Upsert">
        <AuditRecord>
            <UserRef UserOID="dev"/>
            <LocationRef LocationOID="12345"/>
            <DateTimeStamp>2009-12-01T15:28:24</DateTimeStamp>
            <ReasonForChange/>
            <SourceID>825</SourceID>
        </AuditRecord>
        <SiteRef LocationOID="12345"/>
    </SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18"
mdsol:AuditSubCategoryName="SubjectNameChanged">
    <SubjectData SubjectKey="7e8892ba-1e20-4ee2-b23c-63658782fe95"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="456 CD"
```

```


  <AuditRecord>
    <UserRef UserOID="dev"/>
    <LocationRef LocationOID="12345"/>
    <DateTimeStamp>2009-12-01T15:28:24</DateTimeStamp>
    <ReasonForChange/>
    <SourceID>826</SourceID>
  </AuditRecord>
  <SiteRef LocationOID="12345"/>
</SubjectData>
</ClinicalData>
<ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="18">
  mdsol:AuditSubCategoryName="SubjectStatusChanged"
    <SubjectData SubjectKey="7e8892ba-1e20-4ee2-b23c-63658782fe95">
      mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="456 CD"
      TransactionType="Upsert" mdsol:Status="Enrolled">
        <AuditRecord>
          <UserRef UserOID="dev"/>
          <LocationRef LocationOID="12345"/>
          <DateTimeStamp>2009-12-01T15:28:24</DateTimeStamp>
          <ReasonForChange>Enrolled</ReasonForChange>
          <SourceID>827</SourceID>
        </AuditRecord>
        <SiteRef LocationOID="12345"/>
      </SubjectData>
    </ClinicalData>
  </ODM>

```

ODM Adapter/CAR with SES number

```

<ClinicalData StudyOID="RWS_study_with_folders" MetaDataVersionOID="376">
  mdsol:AuditSubCategoryName="Entered"
    <SubjectData SubjectKey="e0b9a535-0cff-4f88-b40e-ecdb66a01648">
      mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="ASUBJECT" >
        <SiteRef LocationOID="12345" mdsol:StudyEnvSiteNumber="9000" />
        <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="SCREEN[1]">
          mdsol:InstanceId="5496" >
            <FormData FormOID="EN" FormRepeatKey="1" mdsol:DataPageId="24906" >
              <ItemGroupData ItemGroupOID="EN" mdsol:RecordId="33404" >
                <ItemData ItemOID="EN.SUBJECTNAME" TransactionType="Upsert" Value="SOME NAME" >
                  <AuditRecord>
                    <UserRef UserOID="defuser"/>
                    <LocationRef LocationOID="12345" mdsol:StudyEnvSiteNumber="9000" />
                    <DateTimeStamp>2015-09-18T20:01:18</DateTimeStamp>
                    <ReasonForChange></ReasonForChange>
                    <SourceID>846047</SourceID>
                  </AuditRecord>
                </ItemData>
              </ItemGroupData>
            </StudyEventData>
          </SiteRef>
        </ClinicalData>
      </ODM>

```

```
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
```

1.5.3.5 Retrieve Admin Data with the Version Folders Dataset

The Version Folders pre-installed custom dataset identifies all study versions in-use.

Service	Retrieve Admin Data with the Version Folders Dataset
Category	ODM Adapter
URI	GET https://{host}/RaveWebServices/datasets/VersionFolders.odm?studyoid={study-oid}
Description	Retrieves a list of all folders across all matrices for each CRF version in use for a specified study.

Use Case

I am retrieving clinical transaction data with the ODM Adapter and ...

As a consumer of clinical transaction data

I want to get a list of all folders across all matrices for each CRF version in use

So that I can understand clinical audit data.

The minimum ODM version required to support StudyEventDefs with no child elements is 1.3.1.



Note: A CRF version which is “in use” must satisfy the following criteria:

- It is the latest version
- It is pushed to at least one site
- It has at least one subject

Example

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User.
2. I am a power user or similar with full view-access to my study.

[Back to top](#)

My GET Request

Header

My header contains the following element:

- *My authentication*

URI

```
GET https://{{host}}/RaveWebServices/datasets/VersionFolders.odm?studyoid={{study-oid}}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
{study-oid}	The study name and environment, for example: "Mediflex(Prod)"	Yes	a URL-escaped string



Note: The study OID that includes the environment is passed in to determine metadata versions "in use" for this study.

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/VersionFolders.odm?  
studyoid=Mediflex(Prod)
```

Example

[Back to top](#)

The Response

The whole request results in either:

1. SUCCESS: Return an HTTP Response code of 200 OK - with all ODM data (the ODM element is closed).
2. FAIL: Return an HTTP Response code of 200 OK - but not with all the ODM data (the ODM element is not closed).
3. FAIL: Return an HTTP Response code of 4xx or 5xx. Further details will be logged in the database.

Header	Reason	Body	Notes
200	The transaction was successful	The retrieved data in ODM format " ... "	(see below)
200	The transaction was successful	Retrieved data in ODM format, chopped. " ... "	(see below)
4xx	Request Failure	n/a (Details will be logged in the database)	(see below)
5xx	Service Failure	n/a (Details will be logged in the database)	(see below)

Handling Errors

We strongly advise that clients parse the ODM received to check that the ODM element is closed. An unclosed ODM element indicates that not all the streamed data was received. Some re-try logic should be in place for this possibility.

Error reporting for custom datasets includes HTTP response codes such as 400 or 500. A single HTTP response code can have any number of causes. Upon receiving such errors, further investigation is advised.

Header

The response header contains the following elements:

- Content-type : "text/xml"
- HTTP response code

Body

The response Body is *a valid ODM 1.3 snapshot document*.

The response ODM document is constructed as follows:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema for more details](#).

```
<?xml version="1.0" encoding="utf-8"?>
<ODM>
  <Study OID="{study-oid}">
    <GlobalVariables>
      <StudyName>{study-oid}</StudyName>
      <StudyDescription />
      <ProtocolName>{study-id}</ProtocolName>
    </GlobalVariables>

    <MetaDataVersion OID="{metadata-oid}" Name="{draft-name}" mdsol:PrimaryFormOID="{primary-form}">
      <Protocol>

        <!-- StudyEventRef elements - for each Folder in and outside of the base matrix -->
```

```

    </Protocol>
</MetaDataTable>

<!-- More MetaDataTable elements - for each Version "in-use" -->

</Study>
</ODM>

```

Example

[Back to top](#)

Scenarios

Background:

Given I am a Rave user
 And there is a dataset called "VersionFolders"

Scenario: Export folders

Given there is a study with the following structure:

folder	folder_in_default_matrix	form	field
SCREEN	Yes	DM	BIRTHDATE
VISIT	Yes	VISIT	VISIT_DATE
SCREEN	Yes	AE	AE_DESC

And there is a subject in the study

When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?
 studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM elements:

"""

```

...
<MetaDataTable Name="Version1" mdsol:DefaultMatrixOID="BASE" ...>
  <Protocol>
    <StudyEventRef StudyEventOID="SCREEN"
      mdsol:StudyEventDefName="SCREEN" .../>
      <StudyEventRef StudyEventOID="VISIT"   mdsol:StudyEventDefName="VISIT" .../>
    </Protocol>
  </MetaDataTable>
...
"""

```

Scenario: Folder not in any matrix should also be exported

Given there is a study with the following structure:

folder	folder_in_default_matrix	form	field
SCREEN	Yes	DM	BIRTHDATE

SCREEN Yes	AE	AE_DESC	
VISIT No	VISIT	VISIT_DATE	

And there is a subject in the study

When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM MetadataVersion elements:

"""

```
<MetaDataVersion ...>
  <Protocol>
    <StudyEventRef StudyEventOID="VISIT" mdsol:StudyEventDefName="VISIT" .../>
  </Protocol>
</MetaDataVersion>
<MetaDataVersion mdsol:DefaultMatrixOID="BASE" ...>
  <Protocol>
    <StudyEventRef StudyEventOID="SCREEN"
mdsol:StudyEventDefName="SCREEN" .../>
  </Protocol>
</MetaDataVersion>
"""
```

Scenario: Folder from any CRF version which has at least a subject should be exported

Given there is a study design with the following structure:

folder form field		
SCREEN DM BIRTHDATE		
SCREEN AE AE_DESC		
VISIT VISIT VISIT_DATE		

And there is a subject in one of the version "Version1"

And the study has few more recent CRF versions

When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM MetadataVersion elements:

"""

```
<MetaDataVersion Name="Version1" mdsol:DefaultMatrixOID="BASE"
mdsol:PrimaryFormOID="PRIMARY" ...>
  <Protocol>
    <StudyEventRef StudyEventOID="SCREEN" mdsol:StudyEventDefName="SCREEN" .../>
    <StudyEventRef StudyEventOID="VISIT" mdsol:StudyEventDefName="VISIT" .../>
  </Protocol>
</MetaDataVersion>
"""
```

Scenario: Folders from the latest CRF version which is pushed to at least one site should be exported

Given there is a study design with the following structure:

folder form field		
SCREEN DM BIRTHDATE		

SCREEN	DM	BRTHDTC
SCREEN	AE	AE_DESC
VISIT	VISIT	VISIT_DATE

And there is a version "Version1" pushed to a site

And the study has few more recent CRF versions

And CRF versions "Version3" is the latest CRF version pushed to a site

When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM MetadataVersion elements:

"""

```
<MetaDataVersion Name="Version3" mdsol:DefaultMatrixOID="BASE"
mdsol:PrimaryFormOID="PRIMARY" ...>
<Protocol>
    <StudyEventRef StudyEventOID="SCREEN" mdsol:StudyEventDefName="SCREEN" .../>
    <StudyEventRef StudyEventOID="VISIT" mdsol:StudyEventDefName="VISIT" .../>
</Protocol>
</MetaDataVersion>
"""
```

Scenario: Folders from the latest CRF version not pushed to a site should be exported

Given there is a study design with the following structure:

folder	form	field
SCREEN	DM	BRTHDTC
SCREEN	AE	AE_DESC
VISIT	VISIT	VISIT_DATE

And there is a subject in one of the version "Version1"

And the study has a new recent crf version "Version2" not pushed to a site

When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?

studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM MetadataVersion elements:

"""

```
<MetaDataVersion Name="Version2" mdsol:DefaultMatrixOID="BASE"
mdsol:PrimaryFormOID="PRIMARY" ...>
<Protocol>
    <StudyEventRef StudyEventOID="SCREEN" mdsol:StudyEventDefName="SCREEN" .../>
    <StudyEventRef StudyEventOID="VISIT" mdsol:StudyEventDefName="VISIT" .../>
</Protocol>
</MetaDataVersion>
"""
```

Scenario: Folders in non default matrix should also be exported

Given there is a study design with the following structure:

folder	form	field
SCREEN	DM	BRTHDTC
SCREEN	AE	AE_DESC

```

| VISIT | VISIT | VISIT_DATE |
And there is a non default matrix "TEST" with folder "VISIT" mapped to form
"VISIT"
When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM MetadataVersion elements:
"""

<MetaDataVersion Name="Version1" mdsol:DefaultMatrixOID="BASE"
mdsol:PrimaryFormOID="PRIMARY">
<Protocol>
<StudyEventRef StudyEventOID="SCREEN" OrderNumber="1" Mandatory="No"
mdsol:StudyEventDefName="SCREEN" .../>
<StudyEventRef StudyEventOID="VISIT" OrderNumber="2" Mandatory="No"
mdsol:StudyEventDefName="VISIT" .../>
</Protocol>
</MetaDataVersion>
<MetaDataVersion Name="Version1" mdsol:PrimaryFormOID="PRIMARY" >
<Protocol>
<StudyEventRef StudyEventOID="VISIT" mdsol:StudyEventDefName="VISIT" .../>
</Protocol>
</MetaDataVersion>
"""

```

Scenario: Calling the versions service by user who does not have access to the Architect module

Given there is a study with the following structure:

folder	form	field
SCREEN	DM	BRTHDTC
VISIT	VISIT	VISIT_DATE
SCREEN	AE	AE_DESC

And the study is pushed to site "12345"

But I do not have Projects Architect Module Read permission

When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?
studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And the response body should not contain any "StudyEventRef" elements

Scenario: Calling the versions service by user who does not have access to study

Given there is a study with the following structure:

folder	form	field
SCREEN	DM	BRTHDTC
VISIT	VISIT	VISIT_DATE
SCREEN	AE	AE_DESC

And the study is pushed to site "12345"

But I am not associated with the study

When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?
studyoid={study_oid}"

Then I should receive a successful response
 And the response should be a valid ODM 1.3 snapshot document
 And the response body should not contain any "StudyEventRef" elements

Scenario: Multiple CRF version for the same study

Given there is a study design with the following structure:

folder	form	field
SCREEN	DM	BRTHDTC
VISIT	VISIT	VISIT_DATE

And there is a subject in the study

And the study has a new recent crf version "Version2" not pushed to a site

When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM MetadataVersion elements:

"""

```
<MetaDataVersion Name="Version1" mdsol:DefaultMatrixOID="BASE"
mdsol:PrimaryFormOID="PRIMARY" ...>
<Protocol>
  <StudyEventRef StudyEventOID="SCREEN" mdsol:StudyEventDefName="SCREEN" .../>
  <StudyEventRef StudyEventOID="VISIT" mdsol:StudyEventDefName="VISIT" .../>
</Protocol>
</MetaDataVersion>
<MetaDataVersion Name="Version2" mdsol:DefaultMatrixOID="BASE"
mdsol:PrimaryFormOID="PRIMARY" ...>
<Protocol>
  <StudyEventRef StudyEventOID="SCREEN" mdsol:StudyEventDefName="SCREEN" .../>
  <StudyEventRef StudyEventOID="VISIT" mdsol:StudyEventDefName="VISIT" .../>
</Protocol>
</MetaDataVersion>
"""
```

Scenario: CRF version if not the latest and not pushed to a site must not be exported

Given there is a study design with the following structure:

folder	form	field
SCREEN	DM	BRTHDTC

And the study has a CRF version "Version0" which is not pushed to a site

And there is a subject in another version "Version1"

When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And the response body should have 1 occurrence of the "MetaDataVersion" element

Scenario: Unpublished draft in a study should not be exported

Given there is a study design with the following structure:

folder	form	field
SCREEN	DM	BRTHDTC

```
And there is a subject in the study
And there is another unpublished draft "draft2" in the study
When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And the response body should have 1 occurrence of the "MetaDataVersion" element

Scenario: Empty MetadataVersion when invalid parameters are passed
Given there is a study design with the following structure:
| folder | form | field |
| SCREEN | DM   | BRTHDTC |
And there is a subject in the study
When I make an HTTP GET request to the URL "/datasets/VersionFolders.odm?
studyoid=Se;ect * from users(Prod)"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And the response body should have 0 occurrence of the "MetaDataVersion" element
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

1.5.3.5.1 Example - Retrieve Admin Data with the Version Folders Dataset

I am retrieving clinical data audits and ...

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
```

URI

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/VersionFolders.odm?
studyoid=Mediflex(Prod)
```

The Response

Header

```
HTTP Response code : 200 OK
Content-Type : "text/xml"
```

Body

```
<?xml version="1.0" encoding="utf-8"?>
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot" FileOID="8bc9193d-
acb0-4e92-9186-72be5b985b60" CreationDateTime="2013-01-09T11:41:13.000-00:00"
xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/
metadata">
  <Study OID="Mediflex(Prod)">
    <GlobalVariables>
      <StudyName>Mediflex(Prod)</StudyName>
      <StudyDescription />
      <ProtocolName>Mediflex</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="18" Name="1" mdsol:PrimaryFormOID="EN">
      <Protocol>
        <StudyEventRef StudyEventOID="YEAR1" OrderNumber="7" Mandatory="No"
mdsol:StudyEventDefName="Year 01" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
      </Protocol>
    </MetaDataVersion>
    <MetaDataVersion OID="18" Name="1" mdsol:PrimaryFormOID="EN">
      <Protocol>
        <StudyEventRef StudyEventOID="ADDCYCLE" OrderNumber="11" Mandatory="No"
mdsol:StudyEventDefName="Additional Cycle" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
      </Protocol>
    </MetaDataVersion>
    <MetaDataVersion OID="18" Name="1" mdsol:DefaultMatrixOID="BASE"
mdsol:PrimaryFormOID="EN">
      <Protocol>
        <StudyEventRef StudyEventOID="SCREEN" OrderNumber="1" Mandatory="No"
mdsol:StudyEventDefName="Screening" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="VISIT01" OrderNumber="2" Mandatory="No"
mdsol:StudyEventDefName="Visit 01" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="VISIT02" OrderNumber="3" Mandatory="No"
mdsol:StudyEventDefName="Visit 02" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="VISIT03" OrderNumber="4" Mandatory="No"
```

```
mdsol:StudyEventDefName="Visit 03" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
    <StudyEventRef StudyEventOID="VISIT04" OrderNumber="5" Mandatory="No"
mdsol:StudyEventDefName="Visit 04" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
    <StudyEventRef StudyEventOID="CYCLE1" OrderNumber="8" Mandatory="No"
mdsol:StudyEventDefName="Cycle 01" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
    <StudyEventRef StudyEventOID="CYCLE2" OrderNumber="9" Mandatory="No"
mdsol:StudyEventDefName="Cycle 02" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
    </Protocol>
</MetaDataVersion>
<MetaDataVersion OID="18" Name="1" mdsol:PrimaryFormOID="EN">
    <Protocol>
        <StudyEventRef StudyEventOID="EXTCYCLE" OrderNumber="10" Mandatory="No"
mdsol:StudyEventDefName="Extended Cycle" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
        </Protocol>
    </MetaDataVersion>
    <MetaDataVersion OID="18" Name="1" mdsol:PrimaryFormOID="EN">
        <Protocol>
            <StudyEventRef StudyEventOID="UNSCHEDULED" OrderNumber="6" Mandatory="No"
mdsol:StudyEventDefName="Visit" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
            </Protocol>
        </MetaDataVersion>
        <MetaDataVersion OID="18" Name="1" mdsol:PrimaryFormOID="EN">
            <Protocol>
                <StudyEventRef StudyEventOID="VISIT01" OrderNumber="2" Mandatory="No"
mdsol:StudyEventDefName="Visit 01" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
                <StudyEventRef StudyEventOID="VISIT02" OrderNumber="3" Mandatory="No"
mdsol:StudyEventDefName="Visit 02" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
                <StudyEventRef StudyEventOID="VISIT03" OrderNumber="4" Mandatory="No"
mdsol:StudyEventDefName="Visit 03" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
                </Protocol>
            </MetaDataVersion>
            <MetaDataVersion OID="28" Name="decimalRange" mdsol:PrimaryFormOID="EN">
                <Protocol>
                    <StudyEventRef StudyEventOID="YEAR1" OrderNumber="7" Mandatory="No"
mdsol:StudyEventDefName="Year 01" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
                </Protocol>
            </MetaDataVersion>
            <MetaDataVersion OID="28" Name="decimalRange" mdsol:PrimaryFormOID="EN">
                <Protocol>
                    <StudyEventRef StudyEventOID="ADDCYCLE" OrderNumber="11" Mandatory="No"
```

```
mdsol:StudyEventDefName="Additional Cycle" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
    </Protocol>
</MetaDataVersion>
<MetaDataVersion OID="28" Name="decimalRange" mdsol:DefaultMatrixOID="BASE">
    <Protocol>
        <StudyEventRef StudyEventOID="SCREEN" OrderNumber="1" Mandatory="No"
    mdsol:StudyEventDefName="Screening" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="VISIT01" OrderNumber="2" Mandatory="No"
    mdsol:StudyEventDefName="Visit 01" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="VISIT02" OrderNumber="3" Mandatory="No"
    mdsol:StudyEventDefName="Visit 02" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="VISIT03" OrderNumber="4" Mandatory="No"
    mdsol:StudyEventDefName="Visit 03" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="VISIT04" OrderNumber="5" Mandatory="No"
    mdsol:StudyEventDefName="Visit 04" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="CYCLE1" OrderNumber="8" Mandatory="No"
    mdsol:StudyEventDefName="Cycle 01" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="CYCLE2" OrderNumber="9" Mandatory="No"
    mdsol:StudyEventDefName="Cycle 02" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
    </Protocol>
</MetaDataVersion>
<MetaDataVersion OID="28" Name="decimalRange" mdsol:PrimaryFormOID="EN">
    <Protocol>
        <StudyEventRef StudyEventOID="EXTCYCLE" OrderNumber="10" Mandatory="No"
    mdsol:StudyEventDefName="Extended Cycle" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
    </Protocol>
</MetaDataVersion>
<MetaDataVersion OID="28" Name="decimalRange" mdsol:PrimaryFormOID="EN">
    <Protocol>
        <StudyEventRef StudyEventOID="UNSCHEDULED" OrderNumber="6" Mandatory="No"
    mdsol:StudyEventDefName="Visit" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
    </Protocol>
</MetaDataVersion>
<MetaDataVersion OID="28" Name="decimalRange" mdsol:PrimaryFormOID="EN">
    <Protocol>
        <StudyEventRef StudyEventOID="VISIT01" OrderNumber="2" Mandatory="No"
    mdsol:StudyEventDefName="Visit 01" mdsol:StudyEventDefType="Common"
    mdsol:StudyEventDefRepeating="No"/>
        <StudyEventRef StudyEventOID="VISIT02" OrderNumber="3" Mandatory="No"
```

```

mdsol:StudyEventDefName="Visit 02" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
    <StudyEventRef StudyEventOID="VISIT03" OrderNumber="4" Mandatory="No"
mdsol:StudyEventDefName="Visit 03" mdsol:StudyEventDefType="Common"
mdsol:StudyEventDefRepeating="No"/>
    </Protocol>
</MetaDataVersion>
</Study>
</ODM>

```

1.5.3.6 Retrieve Admin Data with the Users Dataset

Users admin data can be retrieved for each site using the Users pre-installed custom dataset.

Service	Retrieve Admin Data with the Users Dataset
Category	ODM Adapter
URI	GET https://{{host}}/RaveWebServices/datasets/Users.odm?studyoid={{study-oid}}&locationoid={{location-oid}}&locationoidtype={{location-oid-type}}
Description	Retrieves Admin data for all the users for a specified study.

Use Case

I am retrieving clinical transaction data with the ODM Adapter and ...

As a consumer of clinical transaction data

I want to get Admin data for all users for the study

So that I can understand clinical audit data

Example

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User

2. I have access to AllModules in UserAdmin

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://{{host}}/RaveWebServices/datasets/Users.odm?studyoid={{study-oid}}&locationoid={{location-oid}}&locationoidtype={{location-oid-type}}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
{study-oid}	The study name and environment, for example: "Mediflex(Prod)"	Yes	a URL-escaped string
{location-oid}	The site number, example: "1234567"	No	a URL-escaped string
{location-oid-type}	Allows you to input either the Site Number or the Study Environment Site (SES) Number from Rave.	No	a URL-escaped string

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/Users.odm?  
studyoid=Mediflex(Prod) &locationoid=2886161&locationoidtype=StudyEnvSiteNumber
```

[Example](#)

The Response

The whole request will result in either:

1. SUCCESS: Return an HTTP Response code of 200 OK - with all of the ODM data (the ODM element is closed).
2. FAIL: Return an HTTP Response code of 200 OK - but not with all of the ODM data (the ODM element is not closed).
3. FAIL: Return an HTTP Response code of 4xx or 5xx. Further details will be logged in the database.

Header	Reason	Body	Notes
200	The transaction was successful	The retrieved data in ODM format " ... "	(see below)
200	The transaction was successful	Retrieved data in ODM format, chopped. " ... "	(see below)
4xx	Request Failure	n/a (Details will be logged in the database)	(see below)
5xx	Service Failure	n/a (Details will be logged in the database)	(see below)

Handling Errors

We strongly advise that clients parse the ODM received to check that the ODM element is closed. An unclosed ODM element indicates that not all of the streamed data was received. Some re-trial logic should be in place for this possibility.

Error reporting for custom datasets are limited to just the HTTP response code such as 400 or 500. A single HTTP response code can have any number of causes. Upon receiving such errors, further investigation is advised.

Header

The response header contains the following elements:

- Content-type : "text/xml"
- HTTP response code

Body

The response Body is *a valid ODM 1.3 snapshot document*.



Note: If your `StudyEnvironmentSiteNumber` is defined in Rave, then RWS returns it and provides it in ODM next to `SiteRef` in the `mdsol:StudyEnvSiteNumber` attribute.

The response ODM document is constructed as follows:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Body Sample - Classic Rave

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ODM ODMVersion="1.3" FileType="Snapshot" FileOID="33b0414f-0afb-4306-b71f-8657c16986ae" CreationDateTime="2013-01-09T11:08:41" xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">
  <AdminData>
    <User OID="datamanager" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All Modules" mdsol:SiteGroup="World" >
      <LoginName>datamanager</LoginName>
      <DisplayName>Data Manager</DisplayName>
      <FullName>Data Manager</FullName>
      <FirstName>Data</FirstName>
      <LastName>Manager</LastName>
      <Address> </Address>
      <Email></Email>
      <Fax></Fax>
      <Phone></Phone>
      <LocationRef LocationOID="12345" mdsol:Active="Yes" mdsol:StudyEnvSiteNumber="9000" />
        <LocationRef LocationOID="99999" />
      </User>
      <User OID="dev" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All Modules" mdsol:SiteGroup="World" >
        <LoginName>dev</LoginName>
        <DisplayName>A Developer</DisplayName>
        <FullName>A Developer</FullName>
        <FirstName>A</FirstName>
        <LastName>Developer</LastName>
        <Address> </Address>
        <Email></Email>
        <Fax></Fax>
        <Phone></Phone>
        <LocationRef LocationOID="12345" mdsol:Active="Yes" mdsol:StudyEnvSiteNumber="9000" />
          <LocationRef LocationOID="12345123456789012345678901234567890123456789012345" />
          <LocationRef LocationOID="55555" />
          <LocationRef LocationOID="66666" />
          <LocationRef LocationOID="99999" />
      </User>
      <User OID="investigator" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All Modules" mdsol:SiteGroup="World" >
        <LoginName>investigator</LoginName>
        <DisplayName>site investigator</DisplayName>
        <FullName>site investigator</FullName>
        <FirstName>site</FirstName>
        <LastName>investigator</LastName>
        <Address> </Address>
        <Email></Email>
        <Fax></Fax>
        <Phone></Phone>
        <LocationRef LocationOID="12345" mdsol:Active="Yes" mdsol:StudyEnvSiteNumber="9000" />
```

```
<LocationRef LocationOID="99999" />
</User>
<User OID="jpn" UserType="Other" mdsol:Active="Yes"
mdsol:UserGroup="Administrator" mdsol:SiteGroup="World" >
<LoginName>jpn</LoginName>
<DisplayName>Japanese User</DisplayName>
<FullName>Japanese User</FullName>
<FirstName>Japanese</FirstName>
<LastName>User</LastName>
<Address> </Address>
<Email></Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
<LocationRef LocationOID="99999" />
</User>
<User OID="monitoringdatamanager" UserType="Other" mdsol:Active="Yes"
mdsol:UserGroup="Administrator" mdsol:SiteGroup="World" >
<LoginName>monitoringdatamanager</LoginName>
<DisplayName>Monitering Data Manager</DisplayName>
<FullName>Monitering Data Manager</FullName>
<FirstName>Monitering Data</FirstName>
<LastName>Manager</LastName>
<Address> </Address>
<Email></Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
<LocationRef LocationOID="99999" />
</User>
<User OID="noadminauth" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All
Modules" mdsol:SiteGroup="World" >
<LoginName>noadminauth</LoginName>
<DisplayName>no adminauth</DisplayName>
<FullName>no adminauth</FullName>
<FirstName>no</FirstName>
<LastName>adminauth</LastName>
<Address> </Address>
<Email></Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
<LocationRef LocationOID="99999" />
</User>
<User OID="NonArchitectUser" UserType="Other" mdsol:Active="Yes"
mdsol:UserGroup="RS - DEV" mdsol:SiteGroup="World" >
<LoginName>NonArchitectUser</LoginName>
```

```

<DisplayName>Non Architect</DisplayName>
<FullName>Non Architect</FullName>
<FirstName>Non</FirstName>
<LastName>Architect</LastName>
<Address> </Address>
<Email></Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
<LocationRef LocationOID="99999" />
</User>
<User OID="systemuser" UserType="Investigator" mdsol:Active="Yes"
mdsol:UserGroup="SystemUserGroup" mdsol:SiteGroup="World" >
<LoginName>systemuser</LoginName>
<DisplayName>System User</DisplayName>
<FullName>System User</FullName>
<FirstName>System</FirstName>
<LastName>User</LastName>
<Address> </Address>
<Email></Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
<LocationRef LocationOID="12345123456789012345678901234567890123456789012345" />
<LocationRef LocationOID="66666" />
<LocationRef LocationOID="99999" />
</User>
</AdminData>
</ODM>

```

Body Sample - Rave EDC

```

<?xml version="1.0" encoding="utf-8"?>

<ODM ODMVersion="1.3" FileType="Snapshot" FileOID="33b0414f-0afb-4306-
b71f-8657c16986ae" CreationDateTime="2013-01-09T11:08:41" xmlns="http://
www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">
<AdminData>
<User OID="datamanager" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All
Modules" mdsol:SiteGroup="World" >
<LoginName>datamanager</LoginName>
<DisplayName>Data Manager</DisplayName>
<FullName>Data Manager</FullName>
<FirstName>Data</FirstName>
<LastName>Manager</LastName>
<Address> </Address>

```

```
<Email></Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
<LocationRef LocationOID="99999" />
<mdsol:UserGroups>
    <mdsol:UserGroupName>All Modules</mdsol:UserGroupName>
</mdsol:UserGroups>
</User>
<User OID="dev" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All Modules"
mdsol:SiteGroup="World" >
    <LoginName>dev</LoginName>
    <DisplayName>A Developer</DisplayName>
    <FullName>A Developer</FullName>
    <FirstName>A</FirstName>
    <LastName>Developer</LastName>
    <Address> </Address>
    <Email></Email>
    <Fax></Fax>
    <Phone></Phone>
    <LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
    <LocationRef LocationOID="12345123456789012345678901234567890123456789012345" />
    <LocationRef LocationOID="55555" />
    <LocationRef LocationOID="66666" />
    <LocationRef LocationOID="99999" />
    <mdsol:UserGroups>
        <mdsol:UserGroupName>All Modules</mdsol:UserGroupName>
    </mdsol:UserGroups>
</User>
<User OID="investigator" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All
Modules" mdsol:SiteGroup="World" >
    <LoginName>investigator</LoginName>
    <DisplayName>site investigator</DisplayName>
    <FullName>site investigator</FullName>
    <FirstName>site</FirstName>
    <LastName>investigator</LastName>
    <Address> </Address>
    <Email></Email>
    <Fax></Fax>
    <Phone></Phone>
    <LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
    <LocationRef LocationOID="99999" />
    <mdsol:UserGroups>
        <mdsol:UserGroupName>All Modules</mdsol:UserGroupName>
    </mdsol:UserGroups>
</User>
<User OID="jpn" UserType="Other" mdsol:Active="Yes"
```

```
mdsol:UserGroup="Administrator" mdsol:SiteGroup="World" >
    <LoginName>jpn</LoginName>
    <DisplayName>Japanese User</DisplayName>
    <FullName>Japanese User</FullName>
    <FirstName>Japanese</FirstName>
    <LastName>User</LastName>
    <Address> </Address>
    <Email></Email>
    <Fax></Fax>
    <Phone></Phone>
    <LocationRef LocationOID="12345" mdsol:Active="Yes">
        <LocationRef LocationOID="9000" />
        <LocationRef LocationOID="99999" />
        <mdsol:UserGroups>
            <mdsol:UserGroupName>Administrator</mdsol:UserGroupName>
        </mdsol:UserGroups>
    </User>
    <User OID="monitoringdatamanager" UserType="Other" mdsol:Active="Yes">
        <mdsol:UserGroup="Administrator" mdsol:SiteGroup="World" >
            <LoginName>monitoringdatamanager</LoginName>
            <DisplayName>Monitering Data Manager</DisplayName>
            <FullName>Monitering Data Manager</FullName>
            <FirstName>Monitering Data</FirstName>
            <LastName>Manager</LastName>
            <Address> </Address>
            <Email></Email>
            <Fax></Fax>
            <Phone></Phone>
            <LocationRef LocationOID="12345" mdsol:Active="Yes">
                <LocationRef LocationOID="9000" />
                <LocationRef LocationOID="99999" />
                <mdsol:UserGroups>
                    <mdsol:UserGroupName>Administrator</mdsol:UserGroupName>
                </mdsol:UserGroups>
            </User>
            <User OID="noadminauth" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All
Modules" mdsol:SiteGroup="World" >
                <LoginName>noadminauth</LoginName>
                <DisplayName>no adminauth</DisplayName>
                <FullName>no adminauth</FullName>
                <FirstName>no</FirstName>
                <LastName>adminauth</LastName>
                <Address> </Address>
                <Email></Email>
                <Fax></Fax>
                <Phone></Phone>
                <LocationRef LocationOID="12345" mdsol:Active="Yes">
                    <LocationRef LocationOID="9000" />
                    <LocationRef LocationOID="99999" />
                    <mdsol:UserGroups>
```

```
<mdsol:UserGroupName>All Modules</mdsol:UserGroupName>
</mdsol:UserGroups>
</User>
<User OID="NonArchitectUser" UserType="Other" mdsol:Active="Yes"
mdsol:UserGroup="RS - DEV" mdsol:SiteGroup="World" >
    <LoginName>NonArchitectUser</LoginName>
    <DisplayName>Non Architect</DisplayName>
    <FullName>Non Architect</FullName>
    <FirstName>Non</FirstName>
    <LastName>Architect</LastName>
    <Address> </Address>
    <Email></Email>
    <Fax></Fax>
    <Phone></Phone>
    <LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
    <LocationRef LocationOID="99999" />
    <mdsol:UserGroups>
        <mdsol:UserGroupName>RS - DEV</mdsol:UserGroupName>
    </mdsol:UserGroups>
</User>
<User OID="systemuser" UserType="Investigator" mdsol:Active="Yes"
mdsol:UserGroup="SystemUserGroup" mdsol:SiteGroup="World" >
    <LoginName>systemuser</LoginName>
    <DisplayName>System User</DisplayName>
    <FullName>System User</FullName>
    <FirstName>System</FirstName>
    <LastName>User</LastName>
    <Address> </Address>
    <Email></Email>
    <Fax></Fax>
    <Phone></Phone>
    <LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
    <LocationRef LocationOID="12345123456789012345678901234567890123456789012345" />
    <LocationRef LocationOID="66666" />
    <LocationRef LocationOID="99999" />
    <mdsol:UserGroups>
        <mdsol:UserGroupName>SystemUserGroup</mdsol:UserGroupName>
    </mdsol:UserGroups>
</User>
</AdminData>
</ODM>
```

Example

Scenarios

Scenario - Classic Rave

Background:

Given I am a user with following details:

login	name	user group	site group
dev	A Developer	All Modules	World

And there is a study with the following structure:

form	field	field_data_type	field_control_type
AE	AEDESC	text	Text

And there is a subject in the study with the following data:

form	field	value
AE	AEDESC	HEADACHE

And there is a dataset called "Users"

Scenario Outline: Getting user admin data with or without studyoid parameter

When I make an HTTP GET request to the URL "/datasets/Users.odm?<studyoid_param>"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM elements:

"""

```
<User OID="dev" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All
Modules"
```

```
    mdsol:SiteGroup="World" >
    <LoginName>dev</LoginName>
    <DisplayName>A Developer</DisplayName>
    <FullName>A Developer</FullName>
    <FirstName>A</FirstName>
    <LastName>Developer</LastName>
    <Address/>
    <Email/>
    <Fax/>
    <Phone/>
    <LocationRef LocationOID="12345" />
    ....
</User>
<User OID="systemuser" .../>
```

"""

Examples:

studyoid_param	
studyoid={study_oid}	

Scenario: More than one user in a study

Given there is another user "cra" assigned to the study

When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"

Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:

```
"""
...
<User OID="dev" ... />
<User OID="cra" ... />
...
"""
```

Scenario: User has not received training on Rave should fail
Given the background
But I have not had any training to use Rave
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
Then I should receive an HTTP response with status code "InternalServerError"

Scenario: User without User Admin Access should fail
Given the background
But I do not have "User Admin" access
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
Then I should receive an HTTP response with status code "InternalServerError"

Scenario: Another user with ViewAllSites permission and not assigned to a Site
should be exported
Given the background
But there is another user "adminuser" with ViewAllSites permission not assigned
to the StudySite
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And the response body should have User element for "adminuser"

Scenario: Another user without ViewAllSites permission and not assigned to a Site
should not be exported
Given the background
But there is another user "adminuser" without ViewAllSites permission not
assigned to the StudySite
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And the response body should not have User element for "adminuser"

Scenario: All sites assigned to a study should all be exported
Given the study is assigned to site "99999"
When I make an HTTP GET request to the URL "/datasets/Users.odm?studyoid={study
name}"

Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:

```
"""
<User OID="dev" UserType="Other" ... >
<LoginName>dev</LoginName>
<DisplayName>A Developer</DisplayName>
<FullName>A Developer</FullName>
<FirstName>A</FirstName>
<LastName>Developer</LastName>
<Address> </Address>
<Email></Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" />
<LocationRef LocationOID="99999" />
</User>
"""
```

Scenario: All sites assigned to a study should be exported
Given the background
But the user does not have access to the study any more
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"

Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:

```
"""
<User OID="dev" UserType="Other" ... >
<LoginName>dev</LoginName>
<DisplayName>A Developer</DisplayName>
<FullName>A Developer</FullName>
<FirstName>A</FirstName>
<LastName>Developer</LastName>
<Address> </Address>
<Email></Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" />
</User>
"""
```

Scenario: Study parameter passed without environment
When I make an HTTP GET request to the URL "/datasets/Users.odm?studyoid={study
name}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:

```
"""
<User OID="dev" ... />
```

```
"""

```

Scenario: Invalid studyoid parameter should return empty ODM
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid=StudyDoesNotExist"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And the response body should have 0 occurrence of the "User" element

Scenario: Study with no site assigned
Given the background
But there is a study
And the study is not pushed to a site
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And the response body should have 0 occurrence of the "User" element

Scenario - Rave EDC

Background:

Given I am a user with following details:
| login | name | user group | site group |
| dev | A Developer | All Modules | World |
And there is a study with the following structure:
| form | field | field_data_type | field_control_type |
| AE | AEDESC | text | Text |
And there is a subject in the study with the following data:
| form | field | value |
| AE | AEDESC | HEADACHE |
And there is a dataset called "Users"
And the site "12345" is assigned study site number "45678"

Scenario: Getting user admin data with studyoid parameter
Given the background
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:
"""

```
<User OID="dev" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All  

Modules"  

    mdsol:SiteGroup="World" >  

<LoginName>dev</LoginName>  

<DisplayName>A Developer</DisplayName>  

<FullName>A Developer</FullName>
```

```

<FirstName>A</FirstName>
<LastName>Developer</LastName>
<Address/>
<Email/>
<Fax/>
<Phone/>
<LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="45678"/>
    ...
    <mdsol:UserGroups>
        <mdsol:UserGroupName>All Modules</mdsol:UserGroupName>
    ...
</mdsol:UserGroups>
</User>
<User OID="systemuser" .../>
"""

```

Scenario: Getting user admin data without studyoid parameter
Given the background
When I make an HTTP GET request to the URL "/datasets/Users.odm"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And the response body should have 0 occurrence of the "User" element

Scenario: More than one user in a study
Given there is another user "cra" assigned to the study
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:
"""
 ...
 <User OID="dev" ... />
 <User OID="cra" ... />
...
"""

@SECURITY
@Undo_User_Training_Signed
Scenario: User has not received training on Rave should fail
Given the background
But I have not had any training to use Rave
When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
Then I should receive an HTTP response with status code "InternalServerError"

@SECURITY
@Undo_User_Admin_Access

```
Scenario: User without User Admin Access should fail
  Given the background
  But I do not have "User Admin" access
  When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
    Then I should receive an HTTP response with status code "InternalServerError"

@SECURITY
Scenario: Another user with ViewAllSites permission and not assigned to a Site
should be exported
  Given the background
  But there is another user "adminuser" with ViewAllSites permission not assigned
to the StudySite
  When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
    Then I should receive a successful response
    And the response should be a valid ODM 1.3 snapshot document
    And the response body should have User element for "adminuser"

@SECURITY
Scenario: Another user without ViewAllSites permission and not assigned to a Site
should not be exported
  Given the background
  But there is another user "adminuser" without ViewAllSites permission not
assigned to the StudySite
  When I make an HTTP GET request to the URL "/datasets/Users.odm?
studyoid={study_oid}"
    Then I should receive a successful response
    And the response should be a valid ODM 1.3 snapshot document
    And the response body should not have User element for "adminuser"

Scenario: All sites assigned to a study should all be exported
  Given the study is assigned to site "99999"
  And the site "99999" is assigned study site number "88888"
  When I make an HTTP GET request to the URL "/datasets/Users.odm?studyoid={study
name}"
    Then I should receive a successful response
    And the response should be a valid ODM 1.3 snapshot document
    And should have the following ODM elements:
    """
<User OID="dev" UserType="Other" ... >
<LoginName>dev</LoginName>
<DisplayName>A Developer</DisplayName>
<FullName>A Developer</FullName>
<FirstName>A</FirstName>
<LastName>Developer</LastName>
<Address>  </Address>
<Email></Email>
<Fax></Fax>
<Phone></Phone>
```

```
<LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="45678"/>
<LocationRef LocationOID="99999" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="88888"/>
<mdsol:UserGroups>
  <mdsol:UserGroupName>...</mdsol:UserGroupName>
  ...
</mdsol:UserGroups>
</User>
"""

```

Scenario: Unassigned user should not be exported

Given the background

But the user does not have access to the study any more

When I make an HTTP GET request to the URL "/datasets/Users.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And the response body should not have User element for "dev"

And the response body should have User element for "systemuser"

Scenario: Study parameter passed without environment

When I make an HTTP GET request to the URL "/datasets/Users.odm?studyoid={study_name}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM elements:

"""

```
<User OID="dev" ... />
```

"""

Scenario: Invalid studyoid parameter should return empty ODM

When I make an HTTP GET request to the URL "/datasets/Users.odm?studyoid=StudyDoesNotExist"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And the response body should have 0 occurrence of the "User" element

Scenario: Study with no site assigned

Given the background

But there is a study

And the study is not pushed to a site

When I make an HTTP GET request to the URL "/datasets/Users.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And the response body should have 0 occurrence of the "User" element

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.3.6.1 Example - Retrieve Admin Data with the Users Dataset

Important: The user making the request must have **All Modules** permission.

I am retrieving *clinical data audits* and ...

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
```

URI

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/Users.odm?  
studyoid=Mediflex(Prod) & locationoidtype={location-oid-type}
```

The Response

Header

```
HTTP Response code : 200 OK  
Content-Type : "text/xml"
```

Body

Note: If your `StudyEnvironmentSiteNumber` is defined in Rave, RWS returns it and provides it in ODM next to `SiteRef`.

Users dataset with SES number example - Classic Rave

```
<ODM ODMVersion="1.3" FileType="Snapshot"  
FileOID="21e5c353-7889-4d77-9b1b-0b66ec94a289" CreationDateTime="2019-11-19T21:17:54"
```

```
xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/
metadata">
<AdminData>
<User OID="defuser" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All Modules"
mdsol:SiteGroup="World" mdsol:UserRole="Power User" >
<LoginName>defuser</LoginName>
<DisplayName>Default User</DisplayName>
<FullName>Default User</FullName>
<FirstName>Default</FirstName>
<LastName>User</LastName>
<Address> </Address>
<Email>bgross@mdsol.com, spatel@mdsol.com</Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" mdsol:Active="Yes" mdsol:StudyEnvSiteNumber="9000" />
</User>
</AdminData>
</ODM>
```

Users dataset with SES number example - Rave EDC

```
<ODM ODMVersion="1.3" FileType="Snapshot"
FileOID="21e5c353-7889-4d77-9b1b-0b66ec94a289" CreationDateTime="2019-11-19T21:17:54"
xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/
metadata">
<AdminData>
<User OID="defuser" UserType="Other" mdsol:Active="Yes" mdsol:UserGroup="All
Modules" mdsol:SiteGroup="World" mdsol:UserRole="Power User" >
<LoginName>defuser</LoginName>
<DisplayName>Default User</DisplayName>
<FullName>Default User</FullName>
<FirstName>Default</FirstName>
<LastName>User</LastName>
<Address> </Address>
<Email>bgross@mdsol.com, spatel@mdsol.com</Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" mdsol:Active="Yes"
mdsol:StudyEnvSiteNumber="9000" />
<mdsol:UserGroups>
<mdsol:UserGroupName>All Modules</mdsol:UserGroupName>
</mdsol:UserGroups>
</User>
<User OID="datamanager" UserType="Other" mdsol:Active="Yes"
mdsol:UserGroup="ReportAdmin" mdsol:SiteGroup="World" mdsol:UserRole="Power User" >
<LoginName>defuser</LoginName>
<DisplayName>Default User</DisplayName>
<FullName>Default User</FullName>
```

```

<FirstName>Default</FirstName>
<LastName>User</LastName>
<Address> </Address>
<Email>bgross@mdsol.com, spatel@mdsol.com</Email>
<Fax></Fax>
<Phone></Phone>
<LocationRef LocationOID="12345" mdsol:Active="Yes">
  mdsol:StudyEnvSiteNumber="9000" />
  <mdsol:UserGroups>
    <mdsol:UserGroupName>ReportAdmin</mdsol:UserGroupName>
    <mdsol:UserGroupName>UserAdmin</mdsol:UserGroupName>
    <mdsol:UserGroupName>Architect</mdsol:UserGroupName>
  </mdsol:UserGroups>
</User>
</AdminData>
</ODM>

```

[Back to top](#)

1.5.3.7 Retrieve Admin Data with the Sites Dataset

You can retrieve Admin data for all sites in the study from the Sites pre-installed custom dataset.

Service	Retrieve Admin Data with the Sites Dataset
Category	ODM Adapter
URI	GET https://{{host}}/RaveWebServices/datasets/Sites.odm?studyoid={{study-oid}} GET https://{{host}}/RaveWebServices/datasets/Sites.odm
Description	Retrieves Admin data for all the sites in a specified study.

Use Case

I am retrieving clinical transaction data with the ODM Adapter and ...

As a consumer of clinical transaction data

I want to get Admin data for all sites in the study

So that I can understand clinical audit data.

Example

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User
2. I am a power user or similar with full view-access to my study

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://host/RaveWebServices/datasets/Sites.odm?studyoid={study-oid}
GET https://host/RaveWebServices/datasets/Sites.odm
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
{study-oid}	The study name and environment, for example: "Mediflex(Prod)"	No	a URL-escaped string

Examples:

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/Sites.odm?  
studyoid=Mediflex(Prod)  
GET https://{{host}}/RaveWebServices/datasets/Sites.odm
```

Example

The Response

The whole request results in either:

1. SUCCESS: Return an HTTP Response code of 200 OK - with all the ODM data (the ODM element is closed).
2. FAIL: Return an HTTP Response code of 200 OK - but not with all the ODM data (the ODM element is not closed).
3. FAIL: Return an HTTP Response code of 4xx or 5xx. Further details are logged in the database.

Header	Reason	Body	Notes
200	The transaction was successful	The retrieved data in ODM format "<ODM> ... </ODM>"	(see below)
200	The transaction was successful	Retrieved data in ODM format, chopped. "<ODM> ... "	(see below)
4xx	Request Failure	n/a (Details are logged in the database)	(see below)
5xx	Service Failure	n/a (Details are logged in the database)	(see below)

Handling Errors

We strongly advise that clients parse the ODM received to check that the ODM element is closed. An unclosed ODM element indicates that not all the streamed data was received. Some re-try logic should be in place for this possibility.

Error reporting for custom datasets are limited to just the HTTP response code such as 400 or 500. A single HTTP response code can have any number of causes. Upon receiving such errors, you should do further investigations.

Header

The response header contains the following elements:

- Content-type : "text/xml"
- HTTP response code

Body

The response Body is *a valid ODM 1.3 snapshot document*.

The response ODM document is constructed as follows:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<?xml version="1.0" ?>
<ODM>
  <AdminData>

    <Location OID="{location-oid}" Name="{name}" LocationType="Site"
      mdsol:Active="{Yes|No}" >

      <!-- MetaDataVersionRef elements - for each study + version combination -->

    </Location>

    <!-- More Location elements - for each Site in the study -->
```

```
</AdminData>
</ODM>
```

Example

Scenarios

Background:

```
Given I am a Rave user
And there is a study with the following structure:
| form | field | field_data_type | field_control_type |
| AE   | AESDT | date           | DateTime          |
And the study is pushed to site "12345"
And there is a dataset called "Sites"
```

Scenario: Getting site admin data

```
When I make an HTTP GET request to the URL "/datasets/Sites.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 document
And should have the following ODM elements:
"""
```

```
<AdminData>
<Location OID="12345" Name="ActiveSite" LocationType="Site" mdsol:Active="Yes" >
  <MetaDataVersionRef ... />
</Location>
</AdminData>
"""
```

Scenario: Study assigned to more than one site

```
Given the study has a version pushed to site "99999"
When I make an HTTP GET request to the URL "/datasets/Sites.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 document
And should have the following ODM elements:
"""
```

```
<AdminData>
<Location OID="12345" Name="ActiveSite" LocationType="Site" mdsol:Active="Yes" >
  <MetaDataVersionRef ... />
</Location>
<Location OID="99999" Name="ActiveSite2" LocationType="Site" mdsol:Active="Yes" >
  <MetaDataVersionRef ... />
</Location>
</AdminData>
"""
```

Scenario: CRF version not pushed to a site must not be exported
Given there is another CRF version for the study not pushed to a site
When I make an HTTP GET request to the URL "/datasets/Sites.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 document
Then the response body should have 1 occurrence of the "Location" element

Scenario: User does not have view permission for the study data should return an empty ODM
Given the background
But I am not associated with the study
When I make an HTTP GET request to the URL "/datasets/Sites.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 document
And the response body should be an empty ODM document

Scenario: User not assigned to any site should return an empty ODM
Given the background
But I am not assigned to any site
When I make an HTTP GET request to the URL "/datasets/Sites.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 document
And the response body should be an empty ODM document

Scenario: User assigned to a specific site only
Given the background
And the study is pushed to another site "99999"
But I am not assigned to site "99999"
When I make an HTTP GET request to the URL "/datasets/Sites.odm?
studyoid={study_oid}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 document
And should not have the following ODM elements:
"""
...
<Location OID="99999" ... />
...
"""

Scenario: User has ViewAllSites permission
Given the background
And the study is pushed to another site "99999"
And I have ViewAllSites permission for the study
When I make an HTTP GET request to the URL "/datasets/Sites.odm?
studyoid={study_oid}"
Then I should receive a successful response

And the response should be a valid ODM 1.3 document
And should have the following ODM elements:

```
"""
    ...
<Location OID="12345" ... />
    ...
<Location OID="99999" ... />
    ...
"""
```

Scenario: Study parameter passed without environment

When I make an HTTP GET request to the URL "/datasets/Sites.odm?studyoid={study name}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 document

And should have the following ODM elements:

```
"""
<AdminData>
    <Location OID="12345" Name="ActiveSite" LocationType="Site" mdsol:Active="Yes" >
        <MetaDataVersionRef ... />
    </Location>
</AdminData>
"""
```

Scenario: CRF versions which is not the latest with at least a subject should be exported

Given there is a subject in an existing CRF version

And the study has a new recent crf version "Version2" not pushed to a site

When I make an HTTP GET request to the URL "/datasets/Sites.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 document

And should have the following ODM elements:

```
"""
<AdminData>
    <Location OID="12345" Name="ActiveSite" LocationType="Site" mdsol:Active="Yes" >
        <MetaDataVersionRef ... />
    </Location>
</AdminData>
"""
```

Scenario: Study Site Number

Given the background

And the site "12345" is assigned study site number "45678"

When I make an HTTP GET request to the URL "/datasets/Sites.odm?studyoid={study_oid}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 document

And should have the following ODM elements:

```
"""
```

```
<AdminData>
    <Location OID="12345" Name="ActiveSite" LocationType="Site"
mdsol:Active="Yes" >
    <MetaDataVersionRef mdsol:StudySiteNumber="45678" ... />
    </Location>
</AdminData>
"""

```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.3.7.1 Example - Retrieve Admin Data with the Sites Dataset

I am retrieving clinical data audits and ...

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
```

URI

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/Sites.odm?
studyoid=Mediflex(Prod)
```

The Response

Header

```
HTTP Response code : 200 OK
Content-Type : "text/xml"
```

Body

```
<?xml version="1.0" encoding="utf-8"?>
<ODM ODMVersion="1.3" FileType="Snapshot" FileOID="ced147de-ba45-4d5d-9539-
ca31b5887329" CreationDateTime="2013-01-09T11:08:37" xmlns="http://
```


1.5.3.8 Retrieve Admin Data with the Signatures Dataset

Service	Retrieve Admin Data with the Signatures Dataset
Category	ODM Adapter
URI	GET https://{host}/RaveWebServices/datasets/Signatures.odm?studyid={study-id}
Description	Retrieves Admin data for all the signatures for a specified study.

Use Case

I am retrieving clinical transaction data with the ODM Adapter and ...

*As a consumer of clinical transaction data
I want to get Admin data for all signatures for the study
So that I can understand clinical audit data*

Example

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User
2. I am a power user or similar with full view-access to my study

My GET Request

Header

My header contains the following element:

1. [My authentication](#)

URI

```
GET https://host/RaveWebServices/datasets/Signatures.odm?studyid={study-id}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
{study-id}	The study name, example: "Mediflex"	Yes	a URL-escaped string

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/Signatures.odm?  
studyid=Mediflex
```

[Example](#)

The Response

The whole request will result in either:

1. SUCCESS: Return an HTTP Response code of 200 OK - with all of the ODM data (the ODM element is closed).
2. FAIL: Return an HTTP Response code of 200 OK - but not with all of the ODM data (the ODM element is not closed).
3. FAIL: Return an HTTP Response code of 4xx or 5xx. Further details will be logged in the database.

Header	Reason	Body	Notes
200	The transaction was successful	The retrieved data in ODM format " ... "	(see below)
200	The transaction was successful	Retrieved data in ODM format, chopped. " ... "	(see below)
4xx	Request Failure	n/a (Details will be logged in the database)	(see below)
5xx	Service Failure	n/a (Details will be logged in the database)	(see below)

Handling Errors

We strongly advise that clients parse the ODM received to check that the ODM element is closed. An unclosed ODM element indicates that not all of the streamed data was received. Some re-trial logic should be in place for this possibility.

Error reporting for custom datasets are limited to just the HTTP response code such as 400 or 500. A single HTTP response code can have any number of causes. Upon receiving such errors, further investigation is advised.

Header

The response header contains the following elements:

- Content-type : "text/xml"
- HTTP response code

Body

The response Body is *a valid ODM 1.3 snapshot document*.

The response ODM document is constructed as follows:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<?xml version="1.0" ?>
<ODM>
  <AdminData>
    <SignatureDef Methodology="Electronic" OID="18" mdsol:Study="Mediflex">
      <Meaning>Approval</Meaning>
      <LegalReason>
        <small>I understand that I am responsible for data entered into the
        Medidata Clinical Research System under my account name and password. I understand
        that sharing of passwords is illegal. I hereby confirm that all data is accurate to
        the best of my knowledge.</small>
      </LegalReason>
    </SignatureDef>
    <SignatureDef Methodology="Electronic" OID="19" mdsol:Study="Mediflex">
```

```

<Meaning>Approval</Meaning>
<LegalReason>
    <small>I understand that I am responsible for data entered into the
Medidata Clinical Research System under my account name and password. I understand
that sharing of passwords is illegal. I hereby confirm that all data is accurate to
the best of my knowledge.</small>
</LegalReason>
</SignatureDef>
<SignatureDef Methodology="Electronic" OID="28" mdsol:Study="Mediflex">
    <Meaning>Approval</Meaning>
    <LegalReason>
        <small>I understand that I am responsible for data entered into the
Medidata Clinical Research System under my account name and password. I understand
that sharing of passwords is illegal. I hereby confirm that all data is accurate to
the best of my knowledge.</small>
    </LegalReason>
    </SignatureDef>
</AdminData>
</ODM>

```

Example

Scenarios

Background:

Given I am a Rave user
And there is a study draft with the following structure:
| form | field | field_data_type |
| AE | AEDES | text |
And there is a dataset called "Signatures"

Scenario: Get Signatures for a study

Given a CRF version "version1" is pushed with signature prompt set as "Please sign the data value"

When I make an HTTP GET request to the URL "/datasets/Signatures.odm?studyid={project_name}"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:

"""

```

<SignatureDef ...>
    <Meaning>Approval</Meaning>
    <LegalReason>Please sign the data value</LegalReason>
</SignatureDef>
"""

```

Scenario: Study parameter passed without environment

Given a CRF version "version1" is pushed with signature prompt set as "Please

```
sign the data value"
When I make an HTTP GET request to the URL "/datasets/Signatures.odm?
studyid={project_name}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:
"""
<SignatureDef ...>
  <Meaning>Approval</Meaning>
  <LegalReason>Please sign the data value</LegalReason>
</SignatureDef>
"""
```

Scenario: Multiple CRF version for a study with different signatures
And a CRF version "version1" is pushed with signature prompt set as "Please sign the data value"
And another CRF version "version2" is pushed with signature prompt set as "Please sign the data page"
When I make an HTTP GET request to the URL "/datasets/Signatures.odm?
studyid={project_name}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And the response body should have 2 occurrences of the "SignatureDef" element

Scenario: Empty document when user does not have view permission on the study
Given the signature prompt is set as "Please sign the data value"
And the study has a version pushed to a site
But I am not associated with the study
When I make an HTTP GET request to the URL "/datasets/Signatures.odm?
studyid={project_name}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And the response body should not contain any "SignatureDef" elements

Scenario: Calling user assigned to a site
Given the signature prompt is set as "Please sign the data value"
And the study has a version pushed to site "12345"
And I am assigned to site "12345"
When I make an HTTP GET request to the URL "/datasets/Signatures.odm?
studyid={project_name}"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:

```
"""
<SignatureDef ...>
  <Meaning>Approval</Meaning>
  <LegalReason>Please sign the data value</LegalReason>
</SignatureDef>
"""
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.3.8.1 Example - Retrieve Admin Data with the Signatures Dataset

I am retrieving clinical data audits and ...

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
```

URI

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/Signatures.odm?  
studyid=Mediflex
```

The Response

Header

```
HTTP Response code : 200 OK  
Content-Type : "text/xml"
```

Body

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://  
www.mdsol.com/ns/odm/metadata" ODMVersion="1.3" FileType="Snapshot" FileOID="3ad87471-  
ca6c-4320-ac93-0a3428596642" CreationDateTime="2013-02-19T14:23:59">  
  <AdminData>  
    <SignatureDef Methodology="Electronic" OID="18" mdsol:Study="Mediflex">  
      <Meaning>Approval</Meaning>  
      <LegalReason>  
        <small>I understand that I am responsible for data entered into the  
        Medidata Clinical Research System under my account name and password. I understand  
        that sharing of passwords is illegal. I hereby confirm that all data is accurate to  
        the best of my knowledge.</small>  
      </LegalReason>
```

```

</SignatureDef>
<SignatureDef Methodology="Electronic" OID="19" mdsol:Study="Mediflex">
    <Meaning>Approval</Meaning>
    <LegalReason>
        <small>I understand that I am responsible for data entered into the
        Medidata Clinical Research System under my account name and password. I understand
        that sharing of passwords is illegal. I hereby confirm that all data is accurate to
        the best of my knowledge.</small>
    </LegalReason>
</SignatureDef>
<SignatureDef Methodology="Electronic" OID="28" mdsol:Study="Mediflex">
    <Meaning>Approval</Meaning>
    <LegalReason>
        <small>I understand that I am responsible for data entered into the
        Medidata Clinical Research System under my account name and password. I understand
        that sharing of passwords is illegal. I hereby confirm that all data is accurate to
        the best of my knowledge.</small>
    </LegalReason>
</SignatureDef>
</AdminData>
</ODM>

```

1.5.3.9 Retrieve the Protocol Deviations Dataset

Service	Retrieve the Protocol Deviations Dataset
Category	Biostat Adapter
URI	GET https://{host}/RaveWebServices/datasets/SDTMProtocolDeviations?studyid={ProjectName}({EnvironmentName})
Description	Retrieves a CSV-formatted dataset of protocol deviations comments for a project.

The Protocol Deviations domain (PD) is a requirement of the SDTM standard. You can retrieve a CSV-formatted dataset of protocol deviations comments for a project by making an RWS request to the following URL:

```
GET https://{host}/RaveWebServices/datasets/SDTMProtocolDeviations?
studyid={ProjectName}({EnvironmentName})
```

The following table lists the columns included in the protocol deviations dataset:

Column	Description
ProjectName	The project's name, as localized for the user's locale.
EnvironmentName	The environment's name, as localized for the user's locale.
SiteNumber	Taken from Sites.SiteNumber.
SubjectName	Taken from Subjects.SubjectName.
SubjectID	Taken from Subjects.SubjectID.
InstanceName	The Instance's name, as localized for the user's locale
InstanceID	Taken from Instances.InstanceID.
InstanceRepeatNumber	Instances.InstanceRepeatNumber,
DataPageName	The DataPage's name, as localized for the user's locale.
DataPageID	Taken from DataPages.DatapageID.
PageRepeatNumber	Taken from DataPages.PageRepeatNumber.
DatapointID	Taken from Datapoints.DatapointID.
FormOID	Taken from Forms.OID.
RecordID	Taken from Records.RecordID.
RecordPosition	Taken from Records.RecordPosition.
FieldOID	Taken from Fields.OID.
Text	Taken from Markings.Text.
PDClass	The protocol deviation class's name, as localized for the user's locale.

Column	Description
PDCode	The protocol deviation code's name, as localized for the user's locale.
DataActive	Indicates whether the data point associated with the protocol deviation is active.
RecordActive	Indicates whether the data point associated with the protocol deviation is active. Matches the RecordActive status as given in the Clinical Views by the ReportingRecords table.
Updated	When the Protocol Deviation was last updated.

1.5.3.10 Retrieve the Comments Dataset

Service	Retrieve the Comments Dataset
Category	Biostat Adapter
URI	GET https://{{host}}/RaveWebServices/datasets/SDTMComments.csv?studyid={{ProjectName}}({{EnvironmentName}})
Description	Retrieves a CSV-formatted dataset of investigator comments for a given project.

The Comments domain (CO) is a requirement of the SDTM standard. You can retrieve a CSV-formatted dataset of investigator comments for a project by making an RWS request to the following URL:

```
GET https://{{host}}/RaveWebServices/datasets/SDTMComments.csv?studyid={{ProjectName}}({{EnvironmentName}})
```

The dataset only returns active comments, as defined in Rave.

The following table lists the columns included in the comments dataset:

Column	Description
ProjectName	The project's name, as localized for the user's locale.
EnvironmentName	The environment's name, as localized for the user's locale.
SiteNumber	Taken from Sites.SiteNumber.
SubjectName	Taken from Subjects.SubjectName.
SubjectID	Taken from Subjects.SubjectID.
InstanceName	The instance's name, as localized for the user's locale.
InstanceID	Taken from Instances.InstanceID.
InstanceRepeatNumber	Instances.InstanceRepeatNumber,
DataPageName	The data page's name, as localized for the user's locale.
DataPageID	Taken from DataPages.DatapageID.
PageRepeatNumber	Taken from DataPages.PageRepeatNumber.
DatapointID	Taken from Datapoints.DatapointID.
FormOID	Taken from Forms.OID.
RecordID	Taken from Records.RecordID.
RecordPosition	Taken from Records.RecordPosition.
FieldOID	Taken from Fields.OID.
Text	Taken from Markings.Text.
Updated	Taken from Markings.Updated.

Column	Description
DataActive	Indicates whether the data point associated with the comment is active.
RecordActive	Indicates whether the data point associated with the comment is active. Matches the RecordActive status as given in the Clinical Views by the ReportingRecords table.

1.5.3.11 Retrieve Custom Dataset Log Messages

Service	Retrieve Custom Dataset Log Messages
Category	Request Services
URI	GET https://{{host}}/RaveWebServices/datasets/LogMessages.xml?{{query-string-filters}}
Description	Retrieves custom Dataset Log Messages.

Use Case

As a sponsor

I want to see what errors my integration is generating

So that I can diagnose and fix my integration



Note:

- Use this pre-installed Custom Dataset to diagnose your request failures.
- See also [error responses - the complete list](#) and the [troubleshooting guide](#).

Assumptions

To succeed, I have satisfied these pre-requisite:

- I am a valid Rave User

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https:// {host} /RaveWebServices/datasets/LogMessages.xml?{query-string-filters}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{my-org}.mdsol.com"
{query-string-filters}	Filters	Yes	refer to below

The results can be filtered by:

- counting back from the latest log messages using the "last" parameter
- per url using the "url" parameter
- the earliest date using the "after" parameter
- the latest date using the "before" parameter

The Timestamp formats supported for the "before" & "after" parameters are:

- yyyy-MM-ddTHH:mm:ss
- yyyy-MM-dd

Examples:

```
GET http://abc.mdsol.com/ravewebservices/datasets/LogMessages.xml?last=100
GET http://abc.mdsol.com/ravewebservices/datasets/LogMessages.xml?
after=2011-11-23T10:00:00
GET http://abc.mdsol.com/ravewebservices/datasets/LogMessages.xml?
before=2011-11-25T10:00:00&after=2011-11-23T10:00:00
GET http://abc.mdsol.com/ravewebservices/datasets/LogMessages.xml?
after=2011-11-20T10:00:00&before=2011-11-23T10:00:00
GET http://abc.mdsol.com/ravewebservices/datasets/LogMessages.xml?
before=2011-11-25T10:00:00
GET http://abc.mdsol.com/ravewebservices/datasets/LogMessages.xml?
before=2011-11-25T10:00:00&url=datasets/LogMessages.xml
GET http://abc.mdsol.com/ravewebservices/datasets/LogMessages.xml?
after=2011-11-23T10:00:00&page=2&per_page=25
```

The Response

- The data is returned in XML format
- The results will be returned in [pages](#)
- The order of the results will be descending through time starting from the after parameter or the current date.
- The earliest log message date returned by RWS will be the Rave Release 2012.1.0 install date.
- The body of the Xml contains RunLog entries with errors only.

Example output of one page:

```
<RunLogs Next="" >
    <RunLog Guid="" Date="" Url="" Outcome="" >
```

```
<Log Date="" Level="" >
    <Exception>Object reference not set to an instance of an object</
Exception>
</Log>
...
</RunLog>
...
</RunLogs>
```

Scenarios

Scenario Outline: Get LogMessages configurable dataset

Given the background

And I have made a RWS <service> request

And there is a service-related error that gets logged

When I call the LogMessages configurable dataset

Then I should receive a successful response

And the body of the response should be an XML document

And I should only see my own errors

And I should see the error message for my request that matches the same entry in the database

And I should only see errors from RWS

And the error messages should traverse backwards through time

Examples:

service	
POST Clinical Data	
POST Metadata Study Data	
POST Metadata Library Data	
POST Metadata Study Attribute	
POST Metadata Library Attribute	
POST Coder Configuration	
GET Metadata Study Attribute	
GET Metadata Library Attribute	
GET Cache Flush	
GET Study Metadata Version	
GET Library Metadata Version	
GET Study Crf Versions List	
GET Library Crf Versions List	
GET Configurable Dataset	
GET Configurable Dataset by format	
POST Configurable Dataset	
POST Configurable Dataset by format	
PUT Configurable Dataset	
PUT Configurable Dataset by format	
DELETE Configurable Dataset	
DELETE Configurable Dataset by format	
GET Regular Clinical View Dataset	

```
| GET Raw Clinical View Dataset           |
| GET Raw Clinical View Dataset as csv   |
| GET Regular Clinical View Version Dataset filtered |
| GET Raw Clinical View Version Dataset filtered |
| GET Regular Clinical View Subject Dataset filtered |
| GET Raw Clinical View Subject Dataset filtered |
| GET Clinical Views Metadata           |
| GET Subjects in a Study               |
| GET Study Draft list                 |
| GET Studies                          |
```

Scenario Outline: Get LogMessages with valid query parameters

```
Given I have made many RWS service requests
And there are many service-related error that get logged
When I call the LogMessages configurable dataset with "<query_string>"
Then I should receive a successful response
And the body of the response should be an XML document
And I should only see my own errors
And I should see the error message for my request that matches the same entry in
the database
And I should only see errors from RWS
And the error messages should traverse backwards through time
```

Examples:

```

|
query_string
|
|
last=10
|
|
before=2012-12-31
|
|
before=2012-12-31T23:59
|
|
before=2012-12-31T23:59:59
|
|
after=2012-06-01
|
|
after=2012-06-01T00:01
|
|
after=2012-06-01T00:01:01
|
|
after=2012-06-01&before=2012-12-31
```

```
|  
|  
before=2012-12-31&after=2012-06-01  
|  
|  
before=2012-02-01&after=2012-02-01  
|  
|  
before=2012-12-31T23:59&after=2012-06-01T00:30  
|  
|  
before=2012-12-31T23:59&after=2012-06-01T00:30:00  
|  
|  
before=2012-12-31T23:59:59&after=2012-06-01T00:30  
|  
|  
before=2012-12-31T23:59:59&after=2012-06-01T00:30:00  
|  
| last=10&url=webservice.aspx?  
PostODMClinicalData |  
| before=2012-12-31&url=webservice.aspx?  
PostODMClinicalData |  
| before=2012-12-31T23:59:59&url=webservice.aspx?  
PostODMClinicalData |  
| after=2012-06-01T00:01:00&url=webservice.aspx?  
PostODMClinicalData |  
| before=2012-12-31&after=2012-06-01&url=webservice.aspx?  
PostODMClinicalData |  
| before=2012-12-31T23:50:00&after=2012-06-01T01:00:00&url=webservice.aspx?  
PostODMClinicalData |
```

Scenario Outline: Get the latest logs

```
Given I have made many RWS service requests  
And there are many service-related errors that get logged  
When I call the LogMessages configurable dataset with "<query_string>"  
Then I should receive a successful response  
And the body of the response should be an XML document  
And I should see pagination information in the HTTP header and the HTTP body  
And I should only see my own errors  
And I should see the error message for my request that matches the same entry in  
the database  
And I should only see errors from RWS  
And I should receive <number> of entries  
And the error messages should traverse backwards through time
```

Examples:

query_string	number	
last=12	12	
last=25	25	

```
| last=100      | 100    |
```

Scenario Outline: Failure when configurable dataset called incorrectly
 Given I have made many RWS service requests
 And there are many service-related errors that get logged
 When I call the LogMessages configurable dataset with "<query_string>"
 Then I should not receive an HTTP response with status code "OK"

Examples:

```
| query_string
|
| abc=
| before=2011-11-23T10:00:00&abc=not_supported
| before=
| last=
| after=
| after=
| before=1991-01-01&after=2001-01-01
| before=2011-11-23&after=2011-11-21&last=15
| before=not_a_supported_date_time_format
| after=not_a_supported_date_time_format
| last=not_a_positive_integer
```

1.5.3.12 Retrieve Metadata with the Version Folders with Forms Dataset

The Version Folders with Forms service is used to retrieve metadata that helps construct a complete hierarchy of Rave objects from the Study level, through matrices and study events to their corresponding forms. The Study Metadata service can then be used in conjunction with this output to trace the structure of each individual form down to its components such as ItemGroups, Items, dictionaries, and so on, thus providing a full ontology.

Service	Retrieve Metadata with the Version Folders with Forms Dataset
Category	MODM Dataset
URI	GET https://{{host}}/ravewebservices/datasets/VersionFoldersWithForms.odm?studyoid={{studyoid}}&metadataversionoid={{versionID}}
Description	Retrieves metadata that helps construct a complete hierarchy of Rave objects from the Study level, through matrices and study events to their corresponding forms. The Study Metadata service can then be used

Service	Retrieve Metadata with the Version Folders with Forms Dataset
	in conjunction with this output to trace the structure of each individual form down to its components such as ItemGroups, Items, dictionaries, and so on, thus providing a full ontology.

Use Case

I am retrieving metadata so that I have a hierarchy of Rave objects from the Study level, through matrices and study events to their corresponding forms.

New in version 2017.2.1

The minimum ODM version required to support StudyEventDefs with no child elements is 1.3.1.



Note: A CRF version which is “in use” needs to satisfy the following criteria:

- It is the latest version.
- It is pushed to at least one site.
- It has at least one subject.

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User.
2. I have permission to access Rave Architect.
3. I am a power user or similar with full view-access to my study.

My GET Request

Header

My header contains the following element:

- [*My authentication*](#)

URI

```
GET https://host/ravewebservices/datasets/VersionFoldersWithForms.odm?  
studyoid={study-oid}&metadataversionoid={versionID}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
{study-oid}	The study name and environment, for example: "Mediflex(Prod)"	Yes	a URL-escaped string
{versionID}	The CRF version ID.	Yes	none



Note: The study OID that includes the environment is passed in to determine metadata versions "in use" for this study.

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/VersionFolders.odm?  
studyoid=Mediflex(Prod)
```

The Response

The whole request will result in either:

1. SUCCESS: Return an HTTP Response code of 200 OK - with all of the ODM data (the ODM element is closed).
2. FAIL: Return an HTTP Response code of 200 OK - but not with all of the ODM data (the ODM element is not closed).
3. FAIL: Return an HTTP Response code of 4xx or 5xx. Further details will be logged in the database.

Header	Reason	Body	Notes
200	The transaction was successful	The retrieved data in ODM format " ... "	(see below)
200	The transaction was successful	Retrieved data in ODM format, chopped. " ... "	(see below)
4xx	Request Failure	n/a (Details will be logged in the database)	(see below)
5xx	Service Failure	n/a (Details will be logged in the database)	(see below)

Handling Errors

We strongly advise that clients parse the ODM received to check that the ODM element is closed. An unclosed ODM element indicates that not all of the streamed data was received. Some re-try logic should be in place for this possibility.

Error reporting for custom datasets are limited to just the HTTP response code such as 400 or 500. A single HTTP response code can have any number of causes. Upon receiving such errors, further investigation is advised.

Header

The response header contains the following elements:

- Content-type : "text/xml"
- HTTP response code

Body

The response Body is *a valid ODM 1.3 snapshot document*. The response ODM document is constructed as follows:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<?xml version="1.0" encoding="utf-8"?>
<ODM>
  <Study OID="{study-oid}">
    <GlobalVariables>
      <StudyName>{study-oid}</StudyName>
      <StudyDescription />
      <ProtocolName>{study-id}</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="{metadata-oid}" Name="{version-name}">
      <mdsol:PrimaryFormOID="{primary-form}">
        <Protocol>
          <!-- StudyEventRef elements - for each Folder in and outside of the base
matrix -->
        </Protocol>
      </mdsol:PrimaryFormOID>
    </MetaDataVersion>
  </Study>
</ODM>
```

```
<!-- StudyEventDef elements - for each Folder in and outside of the base matrix-->
<!-- FormRef elements - for each Form in the folder -->
</StudyEventDef>
</MetaDataVersion>
</Study>
<!-- More MetaDataVersion elements - for the specified Version "in-use" -->
</ODM>
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.3.13 Retrieve Clinical View Metadata

Service	Retrieve Clinical View Metadata
Category	Biostat Adapter
URI	<code>GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv</code> <code>GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv?ViewName={{VIEW NAME}}</code> <code>GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv?ProjectName={{PROJECT NAME}}</code>
Description	Retrieves the CSV-formatted version of the Clinical View metadata dataset, the CSV-formatted version of the metadata for a specific Clinical View, or the CSV-formatted version of the metadata for a specific project's Clinical Views.

You can retrieve a CSV-formatted version of the Clinical View metadata dataset by making an RWS request to the following URL:

```
GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv
```

You can retrieve a CSV-formatted version of the metadata for a specific Clinical View by making an RWS request to the following URL:

```
GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv?ViewName={{VIEW NAME}}
```

You can retrieve a CSV-formatted version of the metadata for a specific project's Clinical Views by making an RWS request to the following URL:

```
GET https://{{host}}/RaveWebServices/datasets/ClinicalViewMetadata.csv?ProjectName={{PROJECT NAME}}
```

The following table lists the columns included in the metadata dataset:

Column	Description
ProjectName	The name of the Project, as localized for the default locale.
ViewName	The name of the Clinical View (taken from ClinicalViews.ViewName). For production Clinical Views, the view name will include the prefix of "prod".
Ordinal	The column ordinal of the Clinical View column (taken from ClinicalViewColumns.ColumnOrdinal).
VarName	The name of the Clinical View column (taken from ClinicalViewColumns.ColumnName).
VarType	The datatype of the Clinical View column. The value is "char" for all types of character data and "num" for all types of numeric data.
VarLength	The length that SAS should assign to the Clinical View column, as follows: for columns with a character DataFormat, the same length as the original column; for columns with a time DataFormat, the number of characters required to express the time; for all other columns (integers, decimals, and datetimes), the VarLength is 8.

Column	Description
VarFormat	The SAS format for the Clinical View column (ClinicalViewColumns.SASFormat).
VarLabel	The SAS label for the Clinical View column (ClinicalViewColumns.SASLabel).

[Back to top](#)

1.5.3.14 Retrieve Subject Calendar Visit Dates

The SubjectsCalendar endpoint outputs the Rave Subject Calendar data, that is, the Visit Actual Date and Visit Expected Date.

- The dataset lists only the subject visits or folders in which the Target is set in the Folder Calendar, and the Visit Date is entered for Target 0 visit.
- The dataset does not list subjects where:
 - Subject Calendar is not triggered (Visit Date is not entered for Target Day 0 folder)
 - Folders (unscheduled visits) where the Target Date is not set, for example, AE and Commed folders.
- The dataset has the following input query string parameters:
 - Project Name - StudyOID, mandatory.
 - StudySiteNumber (StudEnvironmentSiteNumber (SES#)), optional.
 - Without SES# - <SiteRef LocationOID="123456" />
 - With SES# - <SiteRef LocationOID="123456" StudyEnvSiteNumber="5678"/>
 - SubjectName, optional.
- The dataset shows the following information in ODM format:
 - Study
 - Site
 - Subject
 - Instance/Folder Name
 - Instance Date – As entered by the user (Visit Actual Date)
 - Expected Date – Projected by the visit calendar (<date>-projected)
 - Instance Active – Status of Instance/Visit
 - Access
 - No Date Supplied – 'Not Set'
 - Otherwise specified date

- Start
 - No Date Supplied - 'Not Set'
 - Otherwise specified date
- End
 - No Date Supplied - 'Not Set'
 - Otherwise specified date
- Close
 - No Date Supplied - 'Not Set'
 - Otherwise specified date
- Overdue
 - No Date Supplied - 'Not Set'
 - Otherwise specified date
- All dates are shown in "YYYY-MM-DD" format.
- Unknown dates are interpolated based on the Rave configuration, and drive the visit calendar.
- The folders are sorted based on Folder Ordinal.

Service	Retrieve Subject Calendar Visit Dates
Category	Request Services
URI	GET https://{{host}}/RaveWebServices/datasets/SubjectsCalendar.odm?studyoid={{study-oid}}&StudySiteNumber={{studysitenumber}}&SubjectName={{subjectname}}
Description	Retrieves the Rave Subject Calendar data, that is, the Visit Actual Date and Visit Expected Date.

Use Case

I am retrieving clinical transaction data with the ODM Adapter and ...

As a consumer of clinical transaction data

I want to get Calendar data for all the subjects in a study site

So that I can understand clinical audit data

Assumptions

To succeed, I have satisfied these prerequisites:

1. I am a valid Rave User
2. I am a power user or similar with full view-access to my study

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://{{host}}/RaveWebServices/datasets/SubjectsCalendar.odm?studyoid={{study-oid}}&StudySiteNumber={{studysitenumber}}&SubjectName={{subjectname}}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
{study-oid}	The study name and environment, for example: "Mediflex(Prod)"	Yes	a URL-escaped string
{studysitenumber}	The study site number, for example: SSN003202	No	
{subjectname}	The subject name, for example: "DT_02"	No	

Example:

```
GET https://host/RaveWebServices/datasets/SubjectsCalendar.odm?  
studyoid=MCC-73443-001&StudySiteNumber=SSN003202&SubjectName=DT_02
```

The Response

The whole request results in either:

1. SUCCESS: Return an HTTP Response code of 200 OK - with all the ODM data (the ODM element is closed).
2. FAIL: Return an HTTP Response code of 200 OK - but not with all the ODM data (the ODM element is not closed).
3. FAIL: Return an HTTP Response code of 4xx or 5xx. Further details are logged in the database.

Header	Reason	Body	Notes
200	The transaction was successful	The retrieved data in ODM format "<ODM> ... </ODM>"	(see below)
200	The transaction was successful	Retrieved data in ODM format, chopped. "<ODM> ... "	(see below)
4xx	Request Failure	n/a (Details are logged in the database)	(see below)
5xx	Service Failure	n/a (Details are logged in the database)	(see below)

Handling Errors

We strongly advise that clients parse the ODM received to check that the ODM element is closed. An unclosed ODM element indicates that not all the streamed data was received. Some retry logic should be in place for this possibility.

Error reporting for custom datasets are limited to just the HTTP response code such as 400 or 500. A single HTTP response code can have any number of causes. Upon receiving such errors, further investigation is advised.

Header

The response header contains the following elements:

- Content-type: "text/xml"
- HTTP response code

Body

The response Body is *a valid ODM 1.3 snapshot document*.

The response ODM document is constructed as follows:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Example:

```
<ODM ODMVersion="1.3" FileType="Transactional" FileOID="72b9148e-792b-467a-b737-b7d51d9b66d2" CreationDateTime="2023-09--06T17:49:04" xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">
<ClinicalData StudyOID="MCC-73443-001" MetaDataVersionOID="3">
<SubjectData SubjectKey="DT_02" mdsol:SubjectKeyType="SubjectName">
<SiteRef LocationOID="site1_DT" mdsol:LocationOIDType="SiteNumber" StudyEnvSiteNumber="SSE001" />
<StudyEventData StudyEventOID="VISIT1" mdsol:InstanceName="Visit 1" mdsol:InstanceStatus="Inactive" mdsol:InstanceDate="2023-08-25" mdsol:ExpectedDate="2023-08-25" mdsol:InstanceOverdue="2023-08-28"
```

```
        mdsol:InstanceAccess="" mdsol:InstanceStartWindow=""
mdsol:InstanceEndWindow=""
    mdsol:InstanceClose="" />
<StudyEventData StudyEventOID="VISIT2" mdsol:InstanceName="Visit 2"
mdsol:InstanceStatus="Active"
    mdsol:InstanceDate="" mdsol:ExpectedDate="2023-09-01"
mdsol:InstanceOverdue="2023-09-04"
    mdsol:InstanceAccess="2023-08-30"
mdsol:InstanceStartWindow="2023-08-30"
    mdsol:InstanceEndWindow="2023-09-08"
mdsol:InstanceClose="2023-09-11" />
<StudyEventData StudyEventOID="VISIT3" mdsol:InstanceName="Visit 3"
mdsol:InstanceStatus="Active"
    mdsol:InstanceDate="" mdsol:ExpectedDate="2023-09-08"
mdsol:InstanceOverdue="2023-09-11"
    mdsol:InstanceAccess="2023-09-03"
mdsol:InstanceStartWindow="2023-09-06"
    mdsol:InstanceEndWindow="2023-09-22"
mdsol:InstanceClose="2023-09-28" />
<StudyEventData StudyEventOID="VISIT4" mdsol:InstanceName="Visit 4"
mdsol:InstanceStatus="Active"
    mdsol:InstanceDate="" mdsol:ExpectedDate="2023-09-15"
mdsol:InstanceOverdue="" mdsol:InstanceAccess=""
    mdsol:InstanceStartWindow="" mdsol:InstanceEndWindow=""
mdsol:InstanceClose="" />
<StudyEventData StudyEventOID="VISIT5" mdsol:InstanceName="Visit 5"
mdsol:InstanceStatus="Active"
    mdsol:InstanceDate="" mdsol:ExpectedDate="2023-09-22"
mdsol:InstanceOverdue=""
    mdsol:InstanceAccess="2023-09-17" mdsol:InstanceStartWindow=""
mdsol:InstanceEndWindow=""
    mdsol:InstanceClose="" />
<StudyEventData StudyEventOID="VISIT6" mdsol:InstanceName="Visit 6"
mdsol:InstanceStatus="Active"
    mdsol:InstanceDate="" mdsol:ExpectedDate="2023-09-29"
mdsol:InstanceOverdue="" mdsol:InstanceAccess=""
    mdsol:InstanceStartWindow="2023-09-25" mdsol:InstanceEndWindow=""
mdsol:InstanceClose="" />
<StudyEventData StudyEventOID="VISIT7" mdsol:InstanceName="Visit 7"
mdsol:InstanceStatus="Active"
    mdsol:InstanceDate="" mdsol:ExpectedDate="2023-10-06"
mdsol:InstanceOverdue="" mdsol:InstanceAccess=""
    mdsol:InstanceStartWindow="" mdsol:InstanceEndWindow="2023-11-20"
mdsol:InstanceClose="" />
<StudyEventData StudyEventOID="VISIT8" mdsol:InstanceName="Visit 8"
mdsol:InstanceStatus="Active"
    mdsol:InstanceDate="" mdsol:ExpectedDate="2023-10-13"
mdsol:InstanceOverdue="2023-10-16"
    mdsol:InstanceAccess="" mdsol:InstanceStartWindow=""
mdsol:InstanceEndWindow=""
```

```

        mdsol:InstanceClose="" />
    <StudyEventData StudyEventOID="VISIT9" mdsol:InstanceName="Visit 9"
mdsol:InstanceStatus="Active"
        mdsol:InstanceDate="" mdsol:ExpectedDate="2023-10-20"
mdsol:InstanceOverdue="" mdsol:InstanceAccess=""
        mdsol:InstanceStartWindow="" mdsol:InstanceEndWindow=""
mdsol:InstanceClose="2023-12-19" />
    <StudyEventData StudyEventOID="VISIT10" mdsol:InstanceName="Visit 10"
mdsol:InstanceStatus="Active"
        mdsol:InstanceDate="" mdsol:ExpectedDate="2023-10-27"
mdsol:InstanceOverdue="2023-10-30"
        mdsol:InstanceAccess="" mdsol:InstanceStartWindow=""
mdsol:InstanceEndWindow=""
        mdsol:InstanceClose="" />
    </SubjectData>
</ClinicalData>
</ODM>
```

Example With interpolated date:

```

<ODM ODMVersion="1.3" FileType="Transactional"
FileOID="e1691f77-6653-4f00-8851-227e29088d98" CreationDateTime="2023-09-11T01:17:10"
xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/
metadata">
    <ClinicalData StudyOID="MCC-73443-001" MetaDataVersionOID="14">
        <SubjectData SubjectKey="DT_10" mdsol:SubjectKeyType="SubjectName">
            <SiteRef LocationOID="site1_DT" mdsol:LocationOIDType="SiteNumber"
StudyEnvSiteNumber="SSE001" />
                <StudyEventData StudyEventOID="VISIT1" mdsol:InstanceName="Visit 1"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="1800-01-01"
mdsol:ExpectedDate="1800-01-01" mdsol:InstanceOverdue="1800-01-04"
mdsol:InstanceAccess="" mdsol:InstanceStartWindow="" mdsol:InstanceEndWindow=""
mdsol:InstanceClose="" />
                <StudyEventData StudyEventOID="VISIT2" mdsol:InstanceName="Visit 2"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="" mdsol:ExpectedDate="1800-01-08"
mdsol:InstanceOverdue="1800-01-11" mdsol:InstanceAccess="1800-01-06"
mdsol:InstanceStartWindow="1800-01-06" mdsol:InstanceEndWindow="1800-01-15"
mdsol:InstanceClose="1800-01-18" />
                <StudyEventData StudyEventOID="VISIT3" mdsol:InstanceName="Visit 3"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="" mdsol:ExpectedDate="1800-01-15"
mdsol:InstanceOverdue="1800-01-18" mdsol:InstanceAccess="1800-01-10"
mdsol:InstanceStartWindow="1800-01-13" mdsol:InstanceEndWindow="1800-01-29"
mdsol:InstanceClose="1800-02-04" />
                <StudyEventData StudyEventOID="VISIT4" mdsol:InstanceName="Visit 4"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="" mdsol:ExpectedDate="1800-01-22"
mdsol:InstanceOverdue="" mdsol:InstanceAccess="" mdsol:InstanceStartWindow=""
mdsol:InstanceEndWindow="" mdsol:InstanceClose="" />
                <StudyEventData StudyEventOID="VISIT5" mdsol:InstanceName="Visit 5"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="" mdsol:ExpectedDate="1800-01-29"
mdsol:InstanceOverdue="" mdsol:InstanceAccess="1800-01-24"
```

```

mdsol:InstanceStartWindow="" mdsol:InstanceEndWindow="" mdsol:InstanceClose="" />
    <StudyEventData StudyEventOID="VISIT6" mdsol:InstanceName="Visit 6"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="" mdsol:ExpectedDate="1800-02-05"
mdsol:InstanceOverdue="" mdsol:InstanceAccess=""
mdsol:InstanceStartWindow="1800-02-01" mdsol:InstanceEndWindow=""
mdsol:InstanceClose="" />
    <StudyEventData StudyEventOID="VISIT7" mdsol:InstanceName="Visit 7"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="" mdsol:ExpectedDate="1800-02-12"
mdsol:InstanceOverdue="" mdsol:InstanceAccess="" mdsol:InstanceStartWindow=""
mdsol:InstanceEndWindow="1800-03-29" mdsol:InstanceClose="" />
    <StudyEventData StudyEventOID="VISIT8" mdsol:InstanceName="Visit 8"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="" mdsol:ExpectedDate="1800-02-19"
mdsol:InstanceOverdue="1800-02-22" mdsol:InstanceAccess=""
mdsol:InstanceStartWindow="" mdsol:InstanceEndWindow="" mdsol:InstanceClose="" />
    <StudyEventData StudyEventOID="VISIT9" mdsol:InstanceName="Visit 9"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="" mdsol:ExpectedDate="1800-02-26"
mdsol:InstanceOverdue="" mdsol:InstanceAccess="" mdsol:InstanceStartWindow=""
mdsol:InstanceEndWindow="" mdsol:InstanceClose="1800-04-27" />
    <StudyEventData StudyEventOID="VISIT10" mdsol:InstanceName="Visit 10"
mdsol:InstanceStatus="Active" mdsol:InstanceDate="" mdsol:ExpectedDate="1800-03-05"
mdsol:InstanceOverdue="1800-03-08" mdsol:InstanceAccess=""
mdsol:InstanceStartWindow="" mdsol:InstanceEndWindow="" mdsol:InstanceClose="" />
</SubjectData>
</ClinicalData>
</ODM>

```

1.5.4 Retrieve the Data Dictionaries Dataset

Service	Retrieve the Data Dictionaries Dataset
Category	Biostat Adapter
URI	GET https://{{host}}/RaveWebServices/datasets/SDTMDictionaries.csv?studyid={{ProjectName}}({{EnvironmentName}})
Description	Retrieves a CSV-formatted dataset of the latest CRF version's data dictionaries for a specified project and environment.

Data Dictionaries are needed to produce the define.xml file required by the SDTM standard. You can retrieve a CSV-formatted dataset of the latest CRF version's data dictionaries for a project and environment by making an RWS request to the following URL:

```
GET https://{host}/RaveWebServices/datasets/SDTMDaDataDictionaries.csv?
studyid={ProjectName}({EnvironmentName})
```

The following table lists the columns included in the data dictionaries dataset:

Column	Description
ProjectName	The project's name, as localized for the user's locale.
EnvironmentName	The environment's name, as localized for the user's locale.
CRFVersionID	Taken from CRFVersions.CRFVersionID.
DictionaryName	The data dictionary's name, as localized for the user's locale.
CodedValue	Taken from DataDictionaryEntries.CodedData.
UserValue	The user value string, as localized for the user's locale.

1.5.5 Retrieve Clinical Study Lists

Service	Retrieve Clinical Study Lists
Category	Request Services
URI	GET https://{host}/RaveWebServices/studies
Description	Retrieves clinical study lists.

Use Case

*As an integrating system
I want to list my studies
So that I can manage my clinical data*



Note: This service will list your EDC studies. It will exclude studies you are not associated with.

Assumptions

To succeed, I have satisfied these prerequisites:

1. I am a valid Rave User
2. I am associated with one or more studies

[Back to top](#)

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://{{host}}/RaveWebServices/studies
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{my-org}.mdsol.com"

Example:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies
```

[Back to top](#)

The Response

Header	Reason	Body	Notes
200	Dataset retrieved	ODM representation of the dataset	See Response Note for more information on the returned ODM.
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See error responses - the complete list

Header

The response header contains the following elements:

- Content-type : "text/xml"
- HTTP response code

Body

The *successful* response Body is a [valid ODM 1.3 snapshot document](#).

The response Odm document is constructed as follows:

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>
  <Study OID="{study-oid}">
    <GlobalVariables>
      <StudyName>{study-oid}</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>{study-oid}</ProtocolName>
    </GlobalVariables>
  </Study>
  <!-- further study elements - for each study I can see -->
</ODM>
```

Attributes:

Attribute or element	Description	Mandatory?	Notes
Study OID	The study name, including the environment	Yes	example: "Mediflex(Dev)"
StudyName	The study name, including the environment	Yes	example: "Mediflex (Dev)"
ProtocolName	The study name	Yes	example: "Mediflex"

Example success response

Header:

```
HTTP Response code : 200 OK
Content-type : "text/xml"
```

Body:

```
<ODM FileType="Snapshot" FileOID="" CreationDateTime="" ODMVersion="1.3"
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" xmlns="http://www.cdisc.org/ns/odm/
v1.3">
<Study OID="Lab Test">
<GlobalVariables>
<StudyName>Lab Test</StudyName>
<StudyDescription />
<ProtocolName>Lab Test</ProtocolName>
</GlobalVariables>
</Study>
<Study OID="Mediflex">
<GlobalVariables>
<StudyName>Mediflex</StudyName>
<StudyDescription />
<ProtocolName>Mediflex</ProtocolName>
</GlobalVariables>
</Study>
</ODM>
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

1.5.6 Retrieve Custom Datasets

Service	Retrieve Custom Datasets
Category	Request Services
URI	GET https://{{host}}/RaveWebServices/datasets/{{dataset-name}}[.{format}][[?{{parameter-list}}]][&{{pagination-query}}]
Description	Retrieves custom datasets.

Use Case

As an integrating system

I want to retrieve my custom datasets

So that I can receive data that is not available through other services



Note: Custom datasets - also known as "Configurable Datasets" - use this generic URL route to provide customised datasets over HTTP. These are typically customer-specific datasets created by executing a SQL script against the URL database.

Assumptions

To succeed, I have satisfied these prerequisites:

1. I am a valid Rave User
2. I have satisfied any custom prerequisites that the custom dataset requires

[Back to top](#)

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://{{host}}/RaveWebServices/datasets/{{dataset-name}}[.{format}][[?{{parameter-list}}]] [[&{{pagination-query}}]]
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{my-org}.mdsol.com"
{dataset-name}	The dataset alias	No	See Parameter Notes.
{format}	The format name	No	See Parameter Notes.
{parameter-list}	The query string filters	No	
pagination-query	The query string pagination	No	

Examples:

```
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/LogMessages.xml  
GET https://my-organisation.mdsol.com/RaveWebServices/datasets/LogMessages.xml?  
last=100
```

**Note:**

- Default XML format
- Complex formats (odm, JSON, etc.)
- Query string filters
- Retrieving multiple recordsets
- Retrieving paged results
- Retrieving custom dataset log messages
- The list of dataset parameters in {parameter-list} must begin with ? and parameters must be separated by &. Any combination of these parameters is permitted.

[Back to top](#)

The Response

The whole response will either be:

1. SUCCESS: Return an HTTP Response code of 200 OK - with all of the ODM data (the ODM element is closed);
2. FAIL: Return an HTTP Response code of 4xx or 5xx. Further details will be logged in the database.

HTTP response code	Reason	Body	Notes
200	The transaction was successful	The retrieved data in ODM format " ... " "	(see below)
4xx	Request Failure	n/a (Details will be logged in the database)	(see below)
5xx	Service Failure	n/a (Details will be logged in the database)	(see below)

Handling Errors:

We strongly advise that clients parse the ODM received to check that the ODM element is closed. An unclosed ODM element indicates that not all of the streamed data was received. Some re-try logic should be in place for this possibility.

Error reporting for custom datasets is limited to just the HTTP response code such as 400 or 500. A single HTTP response code can have any number of causes. Upon receiving such errors, further investigation is advised.

Header

The header contains the following elements:

- HTTP Response code
- Content-Type : {custom-content-type}

Body

The response Body is a {custom-content-type}

Example success response

Header:

```
HTTP Response code : 200 OK
Content-Type : {custom-content-type}
```

Body:

```
SubjectID,OpenQueries
123,7
456,0
EOF
```



Note: The response content type is defined by the custom format.

Example error response

Header:

```
403 HTTP Response code
```

Body:

```
<empty body>
```

 **Note:** If an error occurs after the response has started streaming, it will be too late to change the HTTP Response code : 200 OK. Instead the stream will terminate. Therefore an EOF marker should be received to confirm that all data was received.

[Back to top](#)

1.5.7 Retrieve Study and Library Metadata

Service	Retrieve Study and Library Metadata
Category	Request Services
URI	GET https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/versions/{{version-id}} GET https://{{host}}/RaveWebServices/metadata/libraries/{{library-name}}/versions/{{version-id}}
Description	Retrieves study and library metadata.

Use Case

As an integrating system

I want to retrieve study or library metadata

So that I can manage my study or library metadata in Rave

Examples

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User
2. I can architect drafts in Rave

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://host/RaveWebServices/metadata/studies/{study-name}/versions/{version-id}  
GET https://host/RaveWebServices/metadata/libraries/{library-name}/versions/  
{version-id}
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
studies/{study-name}	The study name, example: "Mediflex"	No	a URL-escaped string
libraries/{library-name}	The global library name, example: "Mediflex"	No	a URL-escaped string
versions/{version-id}	The version identifier, example: "1234"	Yes	a URL-escaped string

Examples:

```
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/studies/Mediflex/  
versions/1234  
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/libraries/Mediflex/  
versions/1234
```

The Response

The whole transaction will either:

1. Succeed with an HTTP Response code of 200 OK.
2. Fail and roll back, with a 4xx or 5xx HTTP Response code and RWS error code in the response body.

Header	Reason	Body	Notes
200	The transaction was successful	The retrieved data in ODM format	(see below)
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See error responses - the complete list

Header

The response header contains the following elements:

- Content-type : "text/xml"
- HTTP response code

Body

The response Body is [a valid ODM 1.3 snapshot document](#)

The response Odm document is constructed as follows:

```
<?xml version="1.0" ?>
<ODM>
  <Study OID="{study-oid}">
    <GlobalVariables>
      <StudyName>{study-oid}</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>{study-name}</ProtocolName>
    </GlobalVariables>

    <!-- "Measurement unit definitions" -->

    <MetaDataVersion
      OID="{metadata-oid}"
      Name="{draft-name}"
      mdsol:PrimaryFormOID="{primary-form}"
      mdsol:DefaultMatrixOID="{default-matrix-oid}"
      mdsol:SignaturePrompt="{signature-prompt}">

      <!-- "Folder definitions" -->
      <!-- "Form definitions" -->
      <!-- "Log line definitions " -->
      <!-- "Field definitions" -->
      <!-- "Field Edit Check definitions" -->
      <!-- "Medidata Coder definitions" -->
      <!-- "Code list definitions" -->
      <!-- "Confirmation Message definitions" -->
      <!-- "Label definitions" -->
      <!-- "Edit Check definitions" -->
      <!-- "Derivation definitions" -->
      <!-- "Custom Function definitions" -->

    </MetaDataVersion>
```

```
</Study>  
</ODM>
```

Attributes:

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute or element	Description	Mandatory?	Notes
Study OID	The study name	Yes	example: "Mediflex"
StudyName	The study name	Yes	example: "Mediflex"
ProtocolName	The study name	Yes	example: "Mediflex"
MetaDataVersion OID	The draft version number	Yes	example: 123
MetaDataVersion Name	The draft name	Yes	example: "Draft Number 1"
mdsol:PrimaryFormOID	The form OID of the primary form	No	example: "ENROL"
mdsol:DefaultMatrixOID	The default matrix OID	No	example: "BASE"
mdsol:SignaturePrompt	The signature prompt for the draft version	No	example: "Please sign..."

I can see all of the following items in my draft:

- Measurement unit definitions
- Folder definitions
- Form definitions
- Log line definitions
- Field definitions
- Field edit check definitions
- Medidata Coder definitions
- Code list definitions
- Confirmation message definitions
- Label definitions

Example successful response

Header:

```
HTTP Response code : 200 OK
Content-Type : "text/xml; charset=utf-8"
```

Body:

```
<ODM FileType="Snapshot" Granularity="Metadata" CreationDateTime="" FileOID=""
ODMVersion="1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" xmlns="http://
www.cdisc.org/ns/odm/v1.3">
    <Study OID="TestLib" mdsol:ProjectType="GlobalLibraryVolume">
        <GlobalVariables>
            <StudyName>TestLib</StudyName>
            <StudyDescription></StudyDescription>
            <ProtocolName>TestLib</ProtocolName>
        </GlobalVariables>
        <BasicDefinitions/>
        <MetaDataVersion OID="1" Name="Draft 1" mdsol:PrimaryFormOID="EN" >
            <Protocol>
                <StudyEventRef StudyEventOID="SCREEN" OrderNumber="1"
Mandatory="No"/>
            </Protocol>
            <StudyEventDef OID="SCREEN" Name="Screening" Type="Common"
Repeating="No">
                <FormRef FormOID="DM" OrderNumber="1" Mandatory="No"/>
            </StudyEventDef>
            <FormDef OID="EN" Name="Enrolment" Repeating="Yes" >
                <ItemGroupRef ItemGroupOID="EN" Mandatory="Yes" />
```

```
</FormDef>
<FormDef OID="DM" Name="Demographics" Repeating="Yes" >
    <ItemGroupRef ItemGroupOID="DM" Mandatory="Yes" />
</FormDef>
<ItemGroupDef OID="DM" Name="DM" Repeating="No">
    <ItemRef ItemOID="DM.AGE" OrderNumber="1" Mandatory="No" />
</ItemGroupDef>
<ItemGroupDef OID="EN" Name="EN" Repeating="No">
    <ItemRef ItemOID="EN.SUBJECTNAME" OrderNumber="1" Mandatory="No" />
</ItemGroupDef>
<ItemDef OID="DM.AGE" Name="Age" DataType="text" Length="200" >
</ItemDef>
<ItemDef OID="EN.SUBJECTNAME" Name="Subject Name" DataType="text"
Length="200" >
</ItemDef>
</MetaDataVersion>
</Study>
</ODM>
```

Example error response

Header:

HTTP Response code : 403 Forbidden
Content-Type : "text/xml; charset=utf-8"

Body:

```
<Response ReferenceNumber=""
InboundODMFileOID="InputODM.xml"
IsTransactionSuccessful="0"
ReasonCode="RWS00102"
ErrorClientResponseMessage="Data posted to incorrect draft URL">
</Response>
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Examples

The following examples of ODM XML metadata export requests from Rave, using RWS, are based on the sample data available to registered users at: <https://innovate.mdsol.com>.

- Listing Studies
- Listing Global Library Studies
- Listing Study Drafts
- Listing Global Library Drafts
- Listing Study CRF Versions
- Listing Global Library CRF Versions
- Requesting a Complete Study CRF Version
- Requesting a Complete Global Library CRF Version
- Requesting a Complete CRF Version with Vendor Attributes
- Requesting a Complete Global Library CRF Version with Vendor Attributes
- Requesting Form Labels as Independent Items
- Requesting Form Labels as Independent Items together with Vendor Attributes

1.5.7.1 Example - List Global Library CRF Versions

This example captures a list of active global library CRF versions accessible to the user.

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/libraries/Library One/versions
```

Successful Response

Header

```
HTTP Response code : 200 OK  
Content-type : "text/xml"
```

Body

```
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot"
  FileOID="195c1546-cac5-411f-bfb8-13b2ff0cbfb6"
  CreationDateTime=" 2009-11-25T09:48:18 "
  xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">
  <Study OID="Library One" mdsol:ProjectType="GlobalLibraryVolume">
    <GlobalVariables>
      <StudyName>Library One</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>Library One</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="11" Name="2" />
    <MetaDataVersion OID="8" Name="1" />
  </Study>
</ODM>
```

1.5.7.2 Example - List Global Library Drafts

This example captures a list of global library drafts accessible to the user.

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/libraries/Library One/drafts
```

Successful Response

Header

```
HTTP Response code : 200 OK
Content-type : "text/xml"
```

Body

```
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot"
  FileOID="195c1546-cac5-411f-bfb8-13b2ff0cbfb6"
  CreationDateTime=" 2009-11-25T09:48:18 "
  xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">
  <Study OID="Library One" mdsol:ProjectType="GlobalLibraryVolume">
    <GlobalVariables>
      <StudyName>Library One</StudyName>
      <StudyDescription/>
      <ProtocolName>Library One</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="44" Name="Draft One"/>
    <MetaDataVersion OID="12" Name="Draft Two"/>
  </Study>
</ODM>
```

1.5.7.3 Example - List Global Library Studies

This example captures a list of global libraries accessible to the user.

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/libraries
```

Successful Response

Header

```
HTTP Response code : 200 OK
Content-type : "text/xml"
```

Body

```
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot"
  FileOID="eb8e4688-35c4-4508-ae7a-d3cc13de9ee0"
  CreationDateTime="2009-05-19T14:56:39.194-00:00"
  xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">
  <Study OID="Library One" mdsol:ProjectType="GlobalLibraryVolume">
    <GlobalVariables>
      <StudyName>Library One</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>Library One</ProtocolName>
    </GlobalVariables>
  </Study>
  <Study OID="Library Two" mdsol:ProjectType="GlobalLibraryVolume">
    <GlobalVariables>
      <StudyName>Library Two</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>Library Two</ProtocolName>
    </GlobalVariables>
  </Study>
</ODM>
</ODM>
```

1.5.7.4 Example - List Studies

You can send a request to retrieve a list of all the available studies in Rave. This is the equivalent to reviewing the list of studies in Rave Architect.

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
GET https://{{host}}/RaveWebServices/metadata/studies
```

Successful Response

Header

```
HTTP Response code : 200 OK
Content-type : "text/xml"
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM FileType="Snapshot" FileOID="" CreationDateTime="" ODMVersion="1.3"
  xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" xmlns="http://
www.cdisc.org/ns/odm/v1.3">
  <Study OID="Lab Test">
    <GlobalVariables>
      <StudyName>Lab Test</StudyName>
      <StudyDescription />
      <ProtocolName>Lab Test</ProtocolName>
    </GlobalVariables>
  </Study>
  <Study OID="Mediflex">
    <GlobalVariables>
      <StudyName>Mediflex</StudyName>
      <StudyDescription />
      <ProtocolName>Mediflex</ProtocolName>
    </GlobalVariables>
  </Study>
</ODM>
```

1.5.7.5 Example - List Study CRF Versions

This example captures a list of active study CRF versions accessible to the user.

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/Mediflex/versions
```

Successful Response

Header

```
HTTP Response code : 200 OK  
Content-type : "text/xml"
```

Body

```
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot"  
FileOID="195c1546-cac5-411f-bfb8-13b2ff0cbfb6"  
CreationDateTime=" 2009-11-25T09:48:18 "  
xmlns="http://www.cdisc.org/ns/odm/v1.3"  
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">  
<Study OID="Mediflex">  
  <GlobalVariables>  
    <StudyName>Mediflex</StudyName>  
    <StudyDescription></StudyDescription>  
    <ProtocolName>Mediflex</ProtocolName>  
  </GlobalVariables>  
  <MetaDataVersion OID="27" Name="V2.1" />  
  <MetaDataVersion OID="6" Name="V2.0" />  
  <MetaDataVersion OID="4" Name="V1.0" />  
</Study>  
</ODM>
```

1.5.7.6 Example - List Study Drafts

This example captures a list of study drafts accessible by the user.

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/Mediflex/drafts
```

Successful Response

Header

```
HTTP Response code : 200 OK  
Content-type : "text/xml"
```

Body

```
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot"  
FileOID="eb8e4688-35c4-4508-ae7a-d3cc13de9ee0"  
CreationDateTime="2009-11-25T09:35:23"  
xmlns="http://www.cdisc.org/ns/odm/v1.3"  
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">  
<Study OID="Beproxen">  
  <GlobalVariables>  
    <StudyName>Beproxen</StudyName>  
    <StudyDescription />  
    <ProtocolName>Beproxen</ProtocolName>  
  </GlobalVariables>  
  <MetaDataVersion OID="23" Name="3.0" />  
  <MetaDataVersion OID="6" Name="V4.0" />  
</Study>  
</ODM>
```

1.5.7.7 Example - Request a Complete CRF Version with Vendor Attributes

This example retrieves a complete CRF version with vendor attributes. The metadata elements returned by this request include:

- A study named Mediflex
- A CRF version named 1

- A subject-level form with the attributes OID="DM" and Name="Demographic"
- A field with the attribute OID="RACE"
- A custom mdsol:Attribute element with the attributes Namespace="MyIntegration", Name="Ignore" and Value="Yes"

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/Mediflex/versions/209/  
attributes?namespace=MyIntegration
```

Successful Response

Header

```
HTTP Response code : 200 OK  
Content-type : "text/xml"
```

Body

```
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot"  
FileOID="195c1546-cac5-411f-bfb8-13b2ff0cbfb6"  
CreationDateTime=" 2009-11-25T09:48:18 "  
xmlns="http://www.cdisc.org/ns/odm/v1.3"  
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">  
<Study OID="Mediflex">  
  <GlobalVariables>  
    <StudyName>Mediflex</StudyName>  
    <StudyDescription></StudyDescription>  
    <ProtocolName>Mediflex</ProtocolName>  
  </GlobalVariables>  
  <BasicDefinitions></BasicDefinitions>  
  <MetaDataVersion OID="209" Name="1">  
    <Protocol>  
      <StudyEventRef StudyEventOID="SUBJECT" Mandatory="Yes" />  
    </Protocol>  
    <StudyEventDef OID="SUBJECT" Name="Subject" Type="Common"  
      Repeating="No">
```

```
<FormRef FormOID="DM" OrderNumber="1" Mandatory="No" />
</StudyEventDef>
<FormDef OID="DM" Name="Demographic" Repeating="Yes">
  <ItemGroupRef ItemGroupOID="DM" Mandatory="Yes" />
</FormDef>
<ItemGroupDef OID="DM" Name="DM" Repeating="No">
  <ItemRef ItemOID="DM.RACE" OrderNumber="1" Mandatory="No">
    <mdsol:Attribute Namespace="MyIntegration" Name="Ignore"
      Value="Yes"/>
  </ItemGroupDef>
  <ItemDef OID="DM.RACE" Name="RACE" DataType="integer" Length="1">
    <mdsol:ControlType>DropDownList</mdsol:ControlType>
  </ItemDef>
</ItemGroupDef>
</MetaDataTableVersion>
</Study>
</ODM>
```

1.5.7.8 Example - Request a Complete Global Library CRF Version

This example retrieves a complete global library CRF version. The metadata elements returned by this request include:

- A global library titled ‘Library One’
- A CRF version titled ‘1’
- A subject-level form with the attributes OID=“DM” and Name=“Demographic”
- A field with the attribute OID=“RACE”
- A data dictionary with the attributes OID=“RACE” and Name=“Race”

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/libraries/Library One/versions/8
```

Successful Response

Header

```
HTTP Response code : 200 OK
Content-type : "text/xml"
```

Body

```
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot"
  FileOID="195c1546-cac5-411f-bfb8-13b2ff0cbfb6"
  CreationDateTime=" 2009-11-25T09:48:18 "
  xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">
  <Study OID="Library One" mdsol:ProjectType="GlobalLibraryVolume">
    <GlobalVariables>
      <StudyName>Library One</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>Library One</ProtocolName>
    </GlobalVariables>
    <BasicDefinitions></BasicDefinitions>
    <MetaDataVersion OID="8" Name="1">
      <Protocol>
        <StudyEventRef StudyEventOID="SUBJECT" Mandatory="Yes" />
      </Protocol>
      <StudyEventDef OID="SUBJECT" Name="Subject" Type="Common"
        Repeating="No">
        <FormRef FormOID="DM" OrderNumber="1" Mandatory="No" />
      </StudyEventDef>
      <FormDef OID="DM" Name="Demographic" Repeating="Yes">
        <ItemGroupRef ItemGroupOID="DM" Mandatory="Yes" />
      </FormDef>
      <ItemGroupDef OID="DM" Name="DM" Repeating="No">
        <ItemRef ItemOID="DM.RACE" OrderNumber="1" Mandatory="No" />
      </ItemGroupDef>
      <ItemDef OID="DM.RACE" Name="RACE" DataType="integer" Length="1"
        mdsol:ControlType="DropDownList">
        <ItemList CodedValue="1" mdsol:OrderNumber="1">
          <Decode>
            <TranslatedText xml:lang="en">Asian</TranslatedText>
          </Decode>
        </ItemList>
      </ItemDef>
    </MetaDataVersion>
```

```
</Study>  
</ODM>
```

1.5.7.9 Example - Request a Complete Global Library CRF Version with Vendor Attributes

This example retrieves a complete global library CRF version, whilst illustrating the ODM XML returned by RWS. It reads the MyIntegration attributes for CRF version 209 of the Library One study. The metadata elements returned by this request include:

- A library named Library One
- A CRF version named 1
- A subject-level form with the attributes OID="DM" and Name="Demographic"
- A field with the attribute OID="RACE" A custom mdsol:Attribute element with the attributes Namespace="MyIntegration", Name="Ignore" and Value="Yes"

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/libraries/Library One/  
versions/209/attributes?namespace=MyIntegration
```

Successful Response

Header

```
HTTP Response code : 200 OK  
Content-type : "text/xml"
```

Body

```
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot"  
FileOID="195c1546-cac5-411f-bfb8-13b2ff0cbfb6"  
CreationDateTime=" 2009-11-25T09:48:18 "  
xmlns="http://www.cdisc.org/ns/odm/v1.3"
```

```
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">
<Study OID="Library One" mdsol:ProjectType="GlobalLibraryVolume">
  <GlobalVariables>
    <StudyName>Library One</StudyName>
    <StudyDescription></StudyDescription>
    <ProtocolName>Library One</ProtocolName>
  </GlobalVariables>
  <BasicDefinitions></BasicDefinitions>
  <MetaDataVersion OID="209" Name="1">
    <Protocol>
      <StudyEventRef StudyEventOID="SUBJECT" Mandatory="Yes" />
    </Protocol>
    <StudyEventDef OID="SUBJECT" Name="Subject" Type="Common"
      Repeating="No">
      <FormRef FormOID="DM" OrderNumber="1" Mandatory="No" />
    </StudyEventDef>
    <FormDef OID="DM" Name="Demographic" Repeating="Yes">
      <ItemGroupRef ItemGroupOID="DM" Mandatory="Yes" />
    </FormDef>
    <ItemGroupDef OID="DM" Name="DM" Repeating="No">
      <ItemRef ItemOID="DM.RACE" OrderNumber="1" Mandatory="No">
        <mdsol:Attribute Namespace="MyIntegration" Name="Ignore"
          Value="Yes"/>
      </ItemRef>
    </ItemGroupDef>
    <ItemDef OID="DM.RACE" Name="RACE" DataType="integer" Length="1"
      mdsol:ControlType="DropDownList">
    </ItemDef>
  </MetaDataVersion>
</Study>
</ODM>
```

1.5.7.10 Example - Request a Complete Study CRF Version

This example retrieves a complete study CRF version. The metadata elements returned by this request include:

- A study named Mediflex
- A CRF version named 1
- A subject-level form with the attributes OID="DM" and Name="Demographic"
- A field with the attribute OID="RACE"
- A data dictionary with the attributes OID="RACE" and Name="Race"

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/Mediflex/versions/209
```

Successful Response

Header

```
HTTP Response code : 200 OK  
Content-type : "text/xml"
```

Body

```
<ODM ODMVersion="1.3" Granularity="Metadata" FileType="Snapshot"  
FileOID="195c1546-cac5-411f-bfb8-13b2ff0cbfb6"  
CreationDateTime=" 2009-11-25T09:48:18 "  
xmlns="http://www.cdisc.org/ns/odm/v1.3"  
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">  
<Study OID="Mediflex">  
  <GlobalVariables>  
    <StudyName>Mediflex</StudyName>  
    <StudyDescription></StudyDescription>  
    <ProtocolName>Mediflex</ProtocolName>  
  </GlobalVariables>  
  <BasicDefinitions></BasicDefinitions>  
  <MetaDataVersion OID="209" Name="1">  
    <Protocol>  
      <StudyEventRef StudyEventOID="SUBJECT" Mandatory="Yes" />  
    </Protocol>  
    <StudyEventDef OID="SUBJECT" Name="Subject" Type="Common"  
      Repeating="No">  
      <FormRef FormOID="DM" OrderNumber="1" Mandatory="No" />  
    </StudyEventDef>  
    <FormDef OID="DM" Name="Demographic" Repeating="Yes">  
      <ItemGroupRef ItemGroupOID="DM" Mandatory="Yes" />  
    </FormDef>  
    <ItemGroupDef OID="DM" Name="DM" Repeating="No">  
      <ItemRef ItemOID="DM.RACE" OrderNumber="1" Mandatory="No" />
```

```
</ItemGroupDef>
<ItemDef OID="DM.RACE" Name="RACE" DataType="integer" Length="1"
  mdsol:ControlType="DropDownList">
</ItemDef>
<CodeList OID="RACE" Name="Race" DataType="integer">
  <CodeListItem CodedValue="1" mdsol:OrderNumber="1">
    <Decode>
      <TranslatedText xml:lang="en">Asian</TranslatedText>
      <TranslatedText xml:lang="ja">Asian</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>
</MetaDataVersion>
</Study>
</ODM>
```

1.5.7.11 Example - Request Form Labels as Independent Items

RWS can be used to export label fields from Rave forms as independent items. Independent label fields can be created in Rave Architect.

Independent labels do not have a data type or field length and only have text which is displayed on the form. They may also be posted to Rave using RWS and the mdsol:LabelRef and mdsol:LabelDef elements (see Importing Label Fields).

This request returns all the labels in all the forms in the study. RWS returns the labels in the format illustrated below. The actual text of the label is displayed in the TranslatedText element.

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/{study-oid}/versions/
{version id}?labels=independent
```

Successful Response

Header

```
HTTP Response code : 200 OK
Content-type : "text/xml"
```

Body

```
<ODM>
...
<ItemGroupDef OID="VS">
    <ItemRef ItemOID="VS.HEART" OrderNumber="2" Mandatory="No" />
    <mdsol:LabelRef LabelOID="VS.LBL1" OrderNumber="1" />
</ItemGroupDef>
<ItemDef OID="VS.HEART">
    <mdsol:LabelDef OID="VS.LBL1" Name="HeartRateLabel">
        <TranslatedText>Please take resting heart rate</TranslatedText>
    </mdsol:LabelDef>
</ItemDef>
...
</ODM>
```

1.5.7.12 Example - Request Form Labels as Independent Items and Vendor Attributes

You can use RWS to export independent label fields together with vendor attributes. RWS returns the form labels and attributes in the format illustrated below.

GET Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/{study-oid}/versions/
{version id}/attributes?namespace={namespace id}&labels=independent
```

Successful Response

Header

```
HTTP Response code : 200 OK
Content-type : "text/xml"
```

Body

```
<ODM>
...
<ItemGroupDef OID="AE" Name="AE" Repeating="No">
    <ItemRef ItemOID="AE.AEYN" OrderNumber="1" Mandatory="No" >
        <mdsol:Attribute Namespace="namespace1" Name="attrib1" Value="value1" />
    </ItemRef>
    <mdsol:LabelRef LabelOID="AE.LBL1" OrderNumber="2" />
</ItemGroupDef>
<ItemGroupDef OID="AE_LOG_LINE" Name="AE_LOG_LINE" Repeating="Yes">
    <ItemRef ItemOID="AE.TERM" OrderNumber="4" Mandatory="No" >
        <mdsol:Attribute Namespace="namespace1" Name="attrib2" Value="value2" />
    </ItemRef>
    <mdsol:LabelRef LabelOID="AE.LBL2" OrderNumber="3" />
</ItemGroupDef>
<ItemDef OID="AE.AEYN" >...</ItemDef>
<ItemDef OID="AE.TERM" >...</ItemDef>
<mdsol:LabelDef OID="AE.LBL1" >...</mdsol:LabelDef>
<mdsol:LabelDef OID="AE.LBL2" >...</mdsol:LabelDef>
...
</ODM>
```

1.5.8 Retrieve Study and Library Metadata Lists

Service	Retrieve Study and Library Metadata Lists
Category	Request Services
URI	GET https://{{host}}/RaveWebServices/metadata/studies GET https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/drafts GET https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/versions

Service	Retrieve Study and Library Metadata Lists
	GET https://{{host}}/RaveWebServices/metadata/libraries GET https://{{host}}/RaveWebServices/metadata/libraries/{{library-name}}/drafts GET https://{{host}}/RaveWebServices/metadata/libraries/{{library-name}}/versions
Description	Retrieves study and library metadata lists.

Use Case

As an integrating system

I want to list my studies or libraries

or

I want to list my study or library drafts

or

I want to list my study or library versions

So that I can manage my study or library metadata data



Note: Use this to list items that you can see in the Architect module in Rave.

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User
2. I can architect drafts in Rave

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://host/RaveWebServices/metadata/studies  
GET https://host/RaveWebServices/metadata/studies/{study-name}/drafts  
GET https://host/RaveWebServices/metadata/studies/{study-name}/versions  
GET https://host/RaveWebServices/metadata/libraries  
GET https://host/RaveWebServices/metadata/libraries/{library-name}/drafts  
GET https://host/RaveWebServices/metadata/libraries/{library-name}/versions
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{my-org}.mdsol.com"
studies/{study-name}	The study name	Yes	Case-sensitive name of the project. E.g. "Mediflex"
libraries/{library-name}	The library name	No	Case-sensitive name of the library. E.g. "Mediflex"

 **Important:** Study or library names must **not** contain any characters that require URL-escaping, such as @ \$ £ &

Examples:

```
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/studies
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/studies/Mediflex/drafts
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/studies/Mediflex/
versions
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/libraries
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/libraries/Mediflex/
drafts
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/libraries/Mediflex/
versions
```

The Response

Header	Reason	Body	Notes
200	Dataset retrieved	ODM representation of the dataset	See Response Notes for more information on the returned ODM.
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See error responses - the complete list

Header

The response header contains the following elements:

- Content-type "text/xml; charset=utf-8"
- HTTP response code

Body

The response Body is a [valid ODM 1.3 snapshot document](#).

The response ODM document is constructed as follows:

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>
  <Study OID="{study-oid}">
    <GlobalVariables>
      <StudyName>{study-oid}</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>{study-oid}</ProtocolName>
    </GlobalVariables>
  </Study>
  <MetaDataVersion OID="{oid}" Name="{name}" />
  ...
</ODM>
```

Attributes:

Attribute or element	Description	Mandatory?	Notes
Study OID	The study name	Yes	example: "Mediflex"
StudyName	The study name	Yes	example: "Mediflex"
ProtocolName	The study name	Yes	example: "Mediflex"
MetaDataVersion OID	The draft or version OID	Yes	example: "D_ONE"
MetaDataVersion Name	The draft or version name	Yes	example: "draft-one"

Success examples

Example: I am listing my studies /metadata/studies

Header:

```
HTTP Response code : 200 OK
Content-type "text/xml; charset=utf-8"
```

Body:

```
<ODM>
  <Study OID="Beproxen">
    <GlobalVariables>
      <StudyName>Beproxen</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>Beproxen</ProtocolName>
    </GlobalVariables>
  </Study>
  <Study OID="Mediflex">
    <GlobalVariables>
      <StudyName>Mediflex</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>Mediflex</ProtocolName>
    </GlobalVariables>
  </Study>
</ODM>
```

Example: I am listing my study drafts /metadata/studies/Mediflex/drafts

Header:

```
HTTP Response code : 200 OK
Content-type "text/xml; charset=utf-8"
```

Body:

```
<ODM>
  <Study OID="Mediflex">
    <GlobalVariables>
      <StudyName>Mediflex</StudyName>
      <StudyDescription/>
      <ProtocolName>Mediflex</ProtocolName>
```

```
</GlobalVariables>
<MetaDataVersion OID="40" Name="Draft-name"/>
...
</Study>
</ODM>
```

Example: I am listing my study versions /metadata/studies/Mediflex/versions

Header:

```
HTTP Response code : 200 OK
Content-type "text/xml; charset=utf-8"
```

Body:

```
<ODM>
  <Study OID="Mediflex">
    <GlobalVariables>
      <StudyName>Mediflex</StudyName>
      <StudyDescription/>
      <ProtocolName>Mediflex</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="41" Name="Version-name1"/>
    <MetaDataVersion OID="42" Name="Version-name2"/>
    ...
  </Study>
</ODM>
```

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.5.9 Retrieve the List of Subjects in a Study

This service provides the facility of listing all the subjects in a study, including those currently inactive or deleted.

Service	Retrieve the List of Subjects in a Study
Category	Request Services
URI	GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/Subjects[?status=all&include={{inactive inactiveAndDeleted}}]
Description	Retrieves the list of subjects in a study.

Use Case

As an integrating system

I want to list all accessible subjects in my study

So that I can understand the subject statuses in the study



Note: This service can optionally include workflow status at the subject level.

Assumptions

To succeed, I have satisfied these prerequisites:

1. I am a valid Rave User
2. I can see subjects in my study

[Back to top](#)

My GET Request

Header

My header contains the following element:

- [My authentication](#)

URI

```
GET https://{{host}}/RaveWebServices/studies/{{study-oid}}/Subjects[?  
status=all&include={{inactive|inactiveAndDeleted}}]
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{my-org}.mdsol.com"
{study-oid}	The study name	Yes	a URL-escaped string. See specifying a study oid
?status=all	Optional query string parameter add subject level workflow status if present	No	
?include={inactive inactiveAndDeleted}	Optional query string parameter will include active, inactive and deleted subjects in the ODM returned	No	See notes below
?subjectKeyType= It {SubjectName SubjectUUID}	It is either the unique identifier or the subject name	No	See notes below

Optional query string parameters:

status

- all - additional workflow attributes (such as mdsol:NonConformant) are added to the SubjectData element.

include

- inactive - inactive subjects will be added to the list of subjects. Each SubjectData element will also include mdsol:SubjectActive attribute. Soft deleted subjects will be excluded
- deleted - soft-deleted subjects will be added to the list of subjects. Each SubjectData element will also include mdsol:Deleted attribute. Inactive subjects will be excluded
- inactiveAndDeleted - a combination of inactive and soft-deleted subjects will be added to the list of subjects. Each SubjectData element will also include mdsol:Deleted attribute

subjectKeyType

- SubjectName or empty - SubjectName will be returned in the subjectKey field
- SubjectUUID - a UUID (GUID) will be returned in the subjectKey field



Note: If invalid parameters are supplied, an empty ODM document is returned.

Examples:

```
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/subjects?  
status=all  
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/subjects?  
include=inactive  
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/subjects?  
status=all&include=inactiveAndDeleted  
GET https://my-organisation.mdsol.com/RaveWebServices/studies/Mediflex(Prod)/subjects?  
subjectKeyType=SubjectUUID
```

[Back to top](#)

The Response

Header	Reason	Body	Notes
200	Dataset retrieved	ODM representation of the dataset	See Response Notes for more information on the returned ODM.
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See error responses - the complete list

Header

The response header contains the following elements:

- Content-type : "text/xml"
- HTTP response code

Body

The *successful* response Body is a valid ODM 1.3 snapshot document.

 **Note:** If your `StudyEnvironmentSiteNumber` is defined in Rave, then RWS returns it and provides it in ODM next to `SiteRef` in the `mdsol:StudyEnvSiteNumber` attribute.

The response ODM document is constructed as follows:

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Without workflow status :

```
<?xml version="1.0" ?>
<ODM>

  <ClinicalData StudyOID="{study-oid}" MetaDataVersionOID="{version-oid}" >
    <SubjectData SubjectKey="{subject-key}" >
      <SiteRef LocationOID="{location-oid}" />
    </SubjectData>
  </ClinicalData>

  <!-- Further ClinicalData elements per subject -->

</ODM>
```

With workflow status :

```

<?xml version="1.0" ?>
<ODM>

    <ClinicalData StudyOID="{study-oid}" MetaDataVersionOID="{version-oid}">
        <SubjectData SubjectKey="{subject-key}">
            mdsol:Overdue="{Yes|No}"
            mdsol:Touched="{Yes|No}"
            mdsol:Empty="{Yes|No}"
            mdsol:Incomplete="{Yes|No}"
            mdsol:NonConformant="{Yes|No}"
            mdsol:RequiresSecondPass="{Yes|No}"
            mdsol:RequiresReconciliation="{Yes|No}"
            mdsol:RequiresVerification="{Yes|No}"
            mdsol:Verified="{Yes|No}"
            mdsol:Frozen="{Yes|No}"
            mdsol:Locked="{Yes|No}"
            mdsol:RequiresReview="{Yes|No}"
            mdsol:PendingReview="{Yes|No}"
            mdsol:Reviewed="{Yes|No}"
            mdsol:RequiresAnswerQuery="{Yes|No}"
            mdsol:RequiresPendingCloseQuery="{Yes|No}"
            mdsol:RequiresCloseQuery="{Yes|No}"
            mdsol:StickyPlaced="{Yes|No}"
            mdsol:Signed="{Yes|No}"
            mdsol:SignatureCurrent="{Yes|No}"
            mdsol:RequiresTranslation="{Yes|No}"
            mdsol:RequiresCoding="{Yes|No}"
            mdsol:RequiresPendingAnswerQuery="{Yes|No}"
            mdsol:RequiresSignature="{Yes|No}"
            mdsol:ReadyForFreeze="{Yes|No}"
            mdsol:ReadyForLock="{Yes|No}" >
            <SiteRef LocationOID="{location-oid}" />
        </SubjectData>
    </ClinicalData>

    <!-- Further ClinicalData elements per subject -->
</ODM>

```

With inactive :

```

<?xml version="1.0" ?>
<ODM>

    <ClinicalData StudyOID="{study-oid}" MetaDataVersionOID="{version-oid}">
        <SubjectData SubjectKey="{subject-key}">
            mdsol:SubjectActive="{Yes|No}" >
            <SiteRef LocationOID="{location-oid}" />

```

```

    </SubjectData>
</ClinicalData>

<!-- Further ClinicalData elements per subject -->

</ODM>

```

With inactive and deleted :

```

<?xml version="1.0" ?>
<ODM>

    <ClinicalData StudyOID="{study-oid}" MetaDataVersionOID="{version-oid}">
        <SubjectData SubjectKey="{subject-key}">
            mdsol:SubjectActive="{Yes|No}" mdsol:Deleted="{Yes|No}" >
                <SiteRef LocationOID="{location-oid}" />
            </SubjectData>
        </ClinicalData>

        <!-- Further ClinicalData elements per subject -->

    </ODM>

```

With SubjectUUID :

```

<?xml version="1.0" ?>
<ODM>

    <ClinicalData StudyOID="{study-oid}" MetaDataVersionOID="{version-oid}">
        <SubjectData SubjectKey="{subject-uuid}" mdsol:SubjectName="{subject-name}">
            mdsol:SubjectKeyType="SubjectUUID" >
                <SiteRef LocationOID="{location-oid}" />
            </SubjectData>
        </ClinicalData>

        <!-- Further ClinicalData elements per subject -->

    </ODM>

```

With mdsol:StudyEnvSiteNumber

```

<ODM FileType="Snapshot" FileOID="97c8cc1e-9ee4-4b48-a0c6-33d51be2690c"
CreationDateTime="2019-11-19T21:28:14.613-00:00" ODMVersion="1.3" xmlns:mdsol="http://
www.mdsol.com/ns/odm/metadata" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns="http://www.cdisc.org/ns/odm/v1.3">
    <ClinicalData StudyOID="RWS_study_with_folders(Prod)" MetaDataVersionOID="376">

```

```
<SubjectData SubjectKey="ASUBJECT" mdsol:SubjectKeyType="SubjectName"  
mdsol:Overdue="No" mdsol:Touched="Yes" mdsol:Empty="No" mdsol:Incomplete="Yes"  
mdsol:NonConformant="No" mdsol:RequiresSecondPass="No"  
mdsol:RequiresReconciliation="No" mdsol:RequiresVerification="No" mdsol:Verified="No"  
mdsol:Frozen="No" mdsol:Locked="No" mdsol:RequiresReview="No"  
mdsol:PendingReview="No" mdsol:Reviewed="No" mdsol:RequiresAnswerQuery="Yes"  
mdsol:RequiresPendingCloseQuery="No" mdsol:RequiresCloseQuery="No"  
mdsol:StickyPlaced="No" mdsol:Signed="No" mdsol:SignatureCurrent="No"  
mdsol:RequiresTranslation="No" mdsol:RequiresCoding="No"  
mdsol:RequiresPendingAnswerQuery="No" mdsol:RequiresSignature="No"  
mdsol:ReadyForFreeze="Yes" mdsol:ReadyForLock="Yes">  
<SiteRef LocationOID="12345" mdsol:StudyEnvSiteNumber="9000" />  
</SubjectData>  
</ClinicalData>  
  
<!-- Further ClinicalData elements per subject -->  
</ODM>
```



Note: Subjects are listed in the context of their ClinicalData and SiteRef elements. The StudyEventData elements are excluded for this service.

Attributes:

Attribute	Description	Mandatory?	Notes
StudyOID	The study identifier	Yes	
MetaDataVersionOID	The study identifier	Yes	
SubjectKey	The subject identifier	Yes	
LocationOID	The site identifier	Yes	The LocationOID is case sensitive.
mdsol:{workflow-status}="{Yes No}"	The subject's workflow status	No	All or none are returned
mdsol:SubjectName	The name of the subject	No	
mdsol:SubjectKeyType= "{SubjectName SubjectUUID}"	The identifier type of the key	No	Can either be the SubjectUUID or the SubjectName
mdsol:StudyEnvSiteNumber	Indicator for the location of the subject in Rave EDC.	No	Example: mdsol:StudyEnvSiteNumber="9000"

Example success response

Header:

```
HTTP Response code : 200 OK
Content-type : "text/xml"
```

Body:

```
<ODM FileType="Snapshot" FileOID="" CreationDateTime="" ODMVersion="1.3"
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" xmlns="http://www.cdisc.org/ns/odm/
v1.3" >
  <ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="30">
    <SubjectData SubjectKey="CVTest 0001" mdsol:Overdue="No" mdsol:Touched="Yes"
      mdsol:Empty="No"
      mdsol:Incomplete="Yes" mdsol:NonConformant="No" mdsol:RequiresSecondPass="No"
      mdsol:RequiresReconciliation="No" mdsol:RequiresVerification="Yes"
      mdsol:Verified="No"
      mdsol:Frozen="No" mdsol:Locked="No" mdsol:RequiresReview="No"
      mdsol:PendingReview="No"
      mdsol:Reviewed="No" mdsol:RequiresAnswerQuery="No"
      mdsol:RequiresPendingCloseQuery="No"
      mdsol:RequiresCloseQuery="No" mdsol:StickyPlaced="No" mdsol:Signed="No"
      mdsol:SignatureCurrent="No"
      mdsol:RequiresTranslation="No" mdsol:RequiresCoding="No"
      mdsol:RequiresPendingAnswerQuery="No"
      mdsol:RequiresSignature="No" mdsol:ReadyForFreeze="Yes"
      mdsol:ReadyForLock="Yes">
      <SiteRef LocationOID="12345" />
    </SubjectData>
  </ClinicalData>
  <ClinicalData StudyOID="Mediflex(Prod)" MetaDataVersionOID="30">
    <SubjectData SubjectKey="CVTest 0002" mdsol:Overdue="No" mdsol:Touched="Yes"
      mdsol:Empty="No"
      mdsol:Incomplete="Yes" mdsol:NonConformant="Yes" mdsol:RequiresSecondPass="No"
      mdsol:RequiresReconciliation="No" mdsol:RequiresVerification="No"
      mdsol:Verified="No"
      mdsol:Frozen="No" mdsol:Locked="No" mdsol:RequiresReview="No"
      mdsol:PendingReview="No"
      mdsol:Reviewed="No" mdsol:RequiresAnswerQuery="No"
      mdsol:RequiresPendingCloseQuery="No"
      mdsol:RequiresCloseQuery="No" mdsol:StickyPlaced="No" mdsol:Signed="No"
      mdsol:SignatureCurrent="No"
      mdsol:RequiresTranslation="No" mdsol:RequiresCoding="Yes"
      mdsol:RequiresPendingAnswerQuery="No"
      mdsol:RequiresSignature="No" mdsol:ReadyForFreeze="Yes"
      mdsol:ReadyForLock="Yes">
```

```

    <SiteRef LocationOID="333444" />
  </SubjectData>
</ClinicalData>
</ODM>
```

Scenarios

Scenario: Subject listing extract with subject status

When I make an HTTP GET request to the URL "/studies/{study name}/Subjects?status=all"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM elements:

```
"""
<ClinicalData StudyOID="{study name} (Prod)"
  MetaDataVersionOID="{MetaDataVersionOID}">
  <SubjectData SubjectKey="{subject name}" ... mdsol:NonConformant="Yes" ...>
    <SiteRef LocationOID="{site number}" />
  </SubjectData>
</ClinicalData>
"""
```

Scenario: Subject listing extract including inactive subjects

Given the background

But the subject has been inactivated

When I make an HTTP GET request to the URL "/studies/{study name}/Subjects?include=inactive"

Then I should receive a successful response

And the response should be a valid ODM 1.3 snapshot document

And should have the following ODM elements:

```
"""
<ClinicalData StudyOID="{study name} (Prod)"
  MetaDataVersionOID="{MetaDataVersionOID}">
  <SubjectData SubjectKey="{subject name}" ... mdsol:SubjectActive="No" ...>
  </SubjectData>
</ClinicalData>
"""
```

Scenario: Subject listing extract including inactive subjects or soft-deleted subjects

Given the background

But the subject has been inactivated

And there is another subject in the study having same name with following data:

form	form_transaction	form_repeat	field	value
field_transaction				
AE	Update	1	AE_STARTDATE	NON CONFORMANT
Insert				

And this subject has been soft deleted

When I make an HTTP GET request to the URL "/studies/{study name}/Subjects?"

```
include=inactiveAndDeleted"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:
"""

...
<SubjectData ... mdsol:SubjectActive="No" mdsol:Deleted="No" ...>
</SubjectData>
...
<SubjectData ... mdsol:Deleted="Yes" ...>
</SubjectData>
...
"""

Scenario: Subject listing extract including one inactive and soft-deleted subject
Given the background
But the subject has been inactivated
And this subject has been soft deleted
When I make an HTTP GET request to the URL "/studies/{study name}/Subjects?
include=inactiveAndDeleted"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:
"""

...
<SubjectData ... mdsol:SubjectActive="No" mdsol:Deleted="Yes" ...>
</SubjectData>
...
"""

Scenario: Subject listing extract with SubjectKeyType parameter
Given the background
And the subject has a UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" not visible in
Rave
When I make an HTTP GET request to the URL "<url>"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document

Scenario: Subject listing includes the Subject's UUID for inactive and soft-deleted
subjects
Given the background
And the subject with UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" has been
inactivated and soft-deleted
When I make an HTTP GET request to the URL "/studies/{study name}/Subjects?
SubjectKeyType=SubjectUUID&include=inactiveAndDeleted"
Then I should receive a successful response
And the response should be a valid ODM 1.3 snapshot document
And should have the following ODM elements:
"""


```

```
...
<SubjectData ... SubjectKey="c45f3bda-efd5-400f-aa72-b174dc0c93dd"
mdsol:SubjectName="ASUBJECT"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectActive="No"
mdsol:Deleted="Yes" ...>
</SubjectData>
...
"""

```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

1.5.10 Work With Study and Library Metadata

Service	Work With Study and Library Metadata
Category	Request Services
URI	GET https://{host}/RaveWebServices/metadata/studies/{study-name} {project-uuid}/drafts POST https://{host}/RaveWebServices/metadata/studies/{study-name}/drafts POST https://{host}/RaveWebServices/metadata/libraries/{library-name}/drafts
Description	Posts study or library metadata to a draft.

Use Case

As an integrating system

I want to post study or library metadata to a draft

So that I can manage my study or library metadata in Rave

Examples

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User
2. I can architect drafts in Rave
3. My Request body does not exceed the maximum request size of 2,000,000 bytes

[Back to top](#)

My POST Request

Header

My header contains the following elements:

1. [My authentication](#)
2. A valid Content-type of "text/xml" or "text/xml; charset=UTF-8;"

URI

```
GET https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}|{{project-uuid}}/drafts  
POST https://{{host}}/RaveWebServices/metadata/studies/{{study-name}}/drafts  
POST https://{{host}}/RaveWebServices/metadata/libraries/{{library-name}}/drafts
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
studies/{study-name}	The study name, example: "Mediflex"	Yes. See Important Warning note below.	a URL-escaped string
studies/{project-uuid}	The project UUID, for example: 2e0-9ff2-11e4-bcd8-0800200c9a66	Yes. See Important Warning note below.	 Note: To obtain the {project-uuid}, you need to contact Medidata Professional Services.
libraries/{library-name}	The global library name, example: "Mediflex"	Yes	a URL-escaped string



Important: The Metadata service has the option to use either studies/{study-name} or studies/{project-uuid} in the URL. You must use studies/{project-uuid} when your study name contains reserved characters, in order to avoid returning a 401 or 404 error message.

Examples:

```
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/studies/Mediflex/drafts
GET https://my-organisation.mdsol.com/RaveWebServices/metadata/studies/2e0-9ff2-11e4-
bcd8-0800200c9a66/drafts
POST https://my-organisation.mdsol.com/RaveWebServices/metadata/studies/Mediflex/
drafts
POST https://my-organisation.mdsol.com/RaveWebServices/metadata/libraries/Mediflex/
drafts
```

Body

My request Body is a valid ODM 1.3 snapshot document.

My Odm document is constructed as follows:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<?xml version="1.0" ?>
<ODM>
  <Study OID="{study-oid}">
    <GlobalVariables>
      <StudyName>{study-oid}</StudyName>
      <StudyDescription></StudyDescription>
      <ProtocolName>{study-oid}</ProtocolName>
    </GlobalVariables>
    <!-- "Measurement unit definitions" -->
    <MetaDataVersion
      OID="{metadata-oid}"
      Name="{draft-name}"
      mdsol:PrimaryFormOID="{primary-form}">
```

```
mdsol:DefaultMatrixOID="{default-matrix-oid}"
mdsol:SignaturePrompt="{signature-prompt}">

<!-- "Folder definitions" -->
<!-- "Form definitions" -->
<!-- "Log line definitions " -->
<!-- "Field definitions" -->
<!-- "Field Edit Check definitions" -->
<!-- "Medidata Coder definitions" -->
<!-- "Code list definitions" -->
<!-- "Confirmation Message definitions" -->
<!-- "Label definitions" -->
<!-- "Edit Check definitions" -->
<!-- "Derivation definitions" -->
<!-- "Custom Function definitions" -->

</MetaDataVersion>

</Study>
</ODM>
```

Attributes:

Attribute or element	Description	Mandatory?	Notes
Study OID	The study name	Yes	example: "Mediflex"
StudyName	The study name	Yes	example: "Mediflex"
ProtocolName	The study name	Yes	example: "Mediflex"
MetaDataVersion OID	The draft version number	Yes	example: 123
MetaDataVersion Name	The draft name	Yes	example: "Draft Number 1"
mdsol:PrimaryFormOID	The form OID of the primary form	No	example: "ENROL"
mdsol:DefaultMatrixOID	The default matrix OID	No	example: "BASE"
mdsol:SignaturePrompt	The signature prompt for the draft version	No	example: "Please sign..."

I can update any or all of the following items in my draft:

- [Measurement unit definitions](#)
- [Folder definitions](#)
- [Form definitions](#)
- [Log line definitions](#)
- [Field definitions](#)
- [Field edit check definitions](#)
- [Medidata Coder definitions](#)
- [Code list definitions](#)
- [Confirmation message definitions](#)
- [Label definitions](#)

Examples

[Back to top](#)

The Response

The whole transaction will either:

1. Succeed with a HTTP Response code of 200 OK.
2. Fail and roll back, with a 4xx or 5xx HTTP Response code and RWS error code in the response body.

Header	Reason	Body	Notes
200	The transaction was successful	XML representation of the items touched	See Response Notes for more information on the returned XML.
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See error responses - the complete list

Example success response

Header:

```
HTTP Response code : 200 OK
Content-Type : "text/xml; charset=utf-8"
```

Body:

```
<Response ReferenceNumber="fa63d085-629a-419e-a19d-21d6fb2bc93b"
InboundODMFileOID="InputODM.xml"
IsTransactionSuccessful="1"
SuccessStatistics="N/A"
NewRecords=""
DraftImported="1">
</Response>
```

Example error response

Header:

```
HTTP Response code : 403 Forbidden
Content-Type : "text/xml; charset=utf-8"
```

Body:

```
<Response ReferenceNumber=""
InboundODMFileOID="InputODM.xml"
IsTransactionSuccessful="0"
ReasonCode="RWS00102"
ErrorClientResponseMessage="Data posted to incorrect draft URL">
</Response>
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

Examples - working with study and library metadata

The following pages contain typical examples of ODM XML metadata for import into Rave using Rave Web Services (RWS), whilst also detailing the results of such imports. These examples are based on the Mediflex study, which is available to registered users at <https://innovate.mdsol.com>.

- [Importing Metadata \(Study Design\)](#)
- [Importing Lab Settings](#)
- [Importing Label Fields](#)
- [Importing Forms with Labels and No Fields](#)

[Back to top](#)

1.5.10.1 Example - Import Forms with Labels and No Fields

You can use RWS to import forms into Rave which have labels but no fields, using the mdsol:LabelRef and mdsol:LabelDef elements. The following example illustrates how to import a standard form with labels and no fields.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/Mediflex/drafts
```

Body

```
<ODM>  
...  
  <MetaDataVersion OID="1" Name="Draft 1">  
    <FormDef OID="FRM" Name="Example Form" Repeating="Yes" >  
      <ItemGroupRef ItemGroupOID="FRM" Mandatory="Yes" />  
    </FormDef>
```

```
<ItemGroupDef OID="FRM" Name="FRM" Repeating="No">
    <mdsol:LabelRef LabelOID="FRM.LBL1" OrderNumber="1" />
</ItemGroupDef>
<mdsol:LabelDef OID="FRM.LBL1" Name="LBL1" />
</MetaDataVersion>
...
</ODM>
```

[Back to top](#)

1.5.10.2 Example - Import Label Fields

You can use RWS to import label fields into Rave forms as independent items.

The OrderNumber attribute may be used to arrange labels in relation to their associated fields. This is possible as the attribute is shared by the mdsol:LabelRef element and the ItemRef element.

The mdsol:LabelRef element specifies the label's presence in the form and its position, and the ItemRef element specifies the presence and position of the related field.

Labels which have a higher OrderNumber than their corresponding fields appear first.

Starting with RWS v 1.4 the mdsol:Label extension is obsolete, but is supported for backward compatibility.

A request containing both the mdsol:Label and the mdsol:LabelRef/mdsol:LabelDef elements is not supported. The following example illustrates the use of the mdsol:LabelRef and the mdsol:LabelDef elements to import labels.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/Mediflex/drafts
```

Body

```
<ODM>
...
<MetaDataVersion OID="1" Name="Draft 1">
    <FormDef OID="VS" Name="Vital Signs" Repeating="Yes" >
        <ItemGroupRef ItemGroupOID="VS" Mandatory="Yes" />
    </FormDef>
    <ItemGroupDef OID="VS" Name="VS" Repeating="No">
        <ItemRef ItemOID="VS.HEART" OrderNumber="2" Mandatory="No" />
        <mdsol:LabelRef LabelOID="VS.LBL1" OrderNumber="1" />
    </ItemGroupDef>
    <ItemDef OID="VS.HEART" Name="HeartRate" DataType="text" Length="10">
        <Question>
            <TranslatedText>Heart rate</TranslatedText>
        </Question>
    </ItemDef>
    <mdsol:LabelDef OID="VS.LBL1" Name="HeartRateLabel">
        <TranslatedText>Please take resting heart rate</TranslatedText>
    </mdsol:LabelDef>
</MetaDataVersion>
...
</ODM>
```

[Back to top](#)

1.5.10.3 Example - Import Lab Settings

To import lab settings for an existing draft or import lab setting while creating a new study design in Rave Architect, post ODM XML for an existing draft with lab settings. The Rave account to be used should have appropriate permissions to upload a draft.

The sample ODM XML shown below illustrates how to import metadata into Rave to create or update a study design. The metadata elements in this example are as follows:

- The StudyName is Mediflex
- The ProtocolName is Mediflex
- The MetaDataVersion OID updates a draft study called Version 1.1 in project Mediflex
- The attributes for the LabSettings element are: StandardUnits="Conventional Unit", ReferenceLabs="Reference Range", and AlertLabs="Alert Range"

- The attributes for the LabVariableMapping element for the Sex variable are: StudyEventOID="SCREEN", FormOID="Demographics", ItemOID="Sex", and RangeLocation="Earliest date"
- The attributes for the LabVariableMapping element for the Age variable are: StudyEventOID="SCREEN", FormOID="Demographics", ItemOID="Age", and RangeLocation="Earliest date"

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/Mediflex/drafts
```

Body

```
<ODM FileType="Snapshot" Granularity="Metadata"  
CreationDateTime="2009-05-19T14:56:39.194-00:00"  
FileOID="PostMetaDataStudyImportDraft" ODMVersion="1.3"  
xmlns="http://www.cdisc.org/ns/odm/v1.3"  
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">  
<Study OID="Mediflex">  
  <GlobalVariables>  
    <StudyName>Mediflex</StudyName>  
    <StudyDescription>  
    </StudyDescription>  
    <ProtocolName>Mediflex</ProtocolName>  
  </GlobalVariables>  
  <MetaDataVersion OID="36" Name="Version 1.1">  
    <mdsol:LabSettings StandardUnits="Conventional Unit"  
      ReferenceLabs="Reference Range" AlertLabs="Alert Range">  
      <mdsol:LabVariableMapping LabVariable="Sex" StudyEventOID="SCREEN"  
        FormOID="Demographics" ItemOID="Sex" RangeLocation="Earliest date"/>  
      <mdsol:LabVariableMapping LabVariable="Age" StudyEventOID="SCREEN"  
        FormOID="Demographics" ItemOID="Age" RangeLocation="Earliest date"/>  
    </mdsol:LabSettings>  
  </MetaDataVersion>  
  </Study>  
</ODM>
```

1.5.10.4 Example - Importing Metadata (Study Design) in RWS

To create or update a study design in Rave Architect, post a valid ODM 1.3 document containing a single MetaDataVersion element. The Rave account which should be used, must have appropriate permissions to permit uploading a draft.

The sample ODM XML in Figure 52 illustrates how to import metadata into Rave to create or update a study design. The metadata elements in this example are as follows:

- The StudyName is Mediflex
- The ProtocolName is Mediflex
- The MetaDataVersion OID defines a draft study called 2 in project Mediflex
- The StudyEventDef creates a folder, with OID SCREEN and name Screen
- The FormDef OID DM creates a form, with OID DM and name DemographicThe ItemDef OID DM.BRTHDAT creates a field with OID BRTHDAT, name BRTHDAT and label Date of birth

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
https://innovate.mdsol.com/RaveWebServices/metadata/studies/Mediflex/drafts
```

Body

```
<ODM FileType="Snapshot" Granularity="Metadata"  
CreationDateTime="2009-05-19T14:56:39.194-00:00"  
FileOID="PostMetaDataStudyImportDraft" ODMVersion="1.3"  
xmlns="http://www.cdisc.org/ns/odm/v1.3"  
xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata">  
<Study OID="Mediflex">  
  <GlobalVariables>  
    <StudyName>Mediflex</StudyName>  
    <StudyDescription></StudyDescription>  
    <ProtocolName>Mediflex</ProtocolName>  
  </GlobalVariables>  
<MetaDataVersion OID="10" Name="2" mdsol:SignaturePrompt="Please sign">
```

```

<Protocol>
  <StudyEventRef StudyEventOID="SCREEN" Mandatory="No" />
</Protocol>
<StudyEventDef OID="SCREEN" Name="Screen" Type="Common"
  Repeating="Yes">
  <FormRef FormOID="DM" Mandatory="No" />
</StudyEventDef>
<FormDef OID="DM" Name="Demographic" Repeating="No"
  mdsol:OrderNumber="1" mdsol:SignatureRequired="No">
  <ItemGroupRef ItemGroupOID="DM" Mandatory="Yes" />
</FormDef>
  <ItemGroupDef OID="DM" Name="DM" Repeating="No">
    <ItemRef ItemOID="DM.BRTHDAT" Mandatory="No" />
  </ItemGroupDef>
    <ItemDef OID="DM.BRTHDAT" Name="BRTHDAT" DataType="date"
      mdsol:Active="Yes" mdsol:ControlType="DateTime">
      <Question>
        <TranslatedText xml:lang="en">Date of birth</TranslatedText>
      </Question>
    </ItemDef>
  </MetaDataVersion>
</Study>
</ODM>

```

[Back to top](#)

1.5.11 Work With Subject Data

Service	Work With Subject Data
Category	Request Services
URI	POST https://{host}/RaveWebServices/webservice.aspx?PostODMClinicalData POST https://{host}/RaveWebServices/webservice.aspx?PostODMClinicalData[&RequestIds={True}]
Description	Updates subject data.

Use Case

As an integrating system

I want to update my subjects data

So that I can manage my clinical data

[Examples](#)

[Best Practices](#)

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User.
2. I can update clinical data in my study.
3. My subject exists in my study and can be updated.
4. The fields pre-conditions for my role have been met.
5. My Request body does not exceed the maximum request size of 1000000 bytes.



Note: To record a change code for field updates, the first change code associated with a user's role in Rave is used. If the user's role does not have a change code, no change code is used.

[Back to top](#)

My POST Request

Header

My header contains the following elements:

1. [My authentication](#)
2. A valid Content-type of "text/xml" or "text/xml; charset=UTF-8;"

URIs

```
POST https://{{host}}/RaveWebServices/webservice.aspx?PostODMClinicalData
```

```
POST https://{{host}}/RaveWebServices/webservice.aspx?  
PostODMClinicalData[&RequestIds={True}]
```

URI Parameters:

Parameter	Description	Mandatory?	Notes
{host}	The host name	Yes	usually "{client}.mdsol.com"
[&RequestIds={True}]	Request Ids in response	No	Optional. When a subject is created, the SubjectId is returned. When records are touched, TouchedRecordId is returned - see below.

Examples:

```
POST https://{{host}}/RaveWebServices/webservice.aspx?PostODMClinicalData
```

```
POST https://{{host}}/RaveWebServices/webservice.aspx?  
PostODMClinicalData[&RequestIds={True}]
```

Note: PostODMClinicalData is case sensitive.

**Body**

My request Body is a valid ODM 1.3 transactional document.

My ODM document is constructed as follows:

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<?xml version="1.0" ?>  
<ODM>  
  
    <!-- "Field location Attributes" -->  
  
    <ClinicalData StudyOID="{{study-oid}}">  
        <SubjectData SubjectKey="{{subject-key}}" TransactionType="Update">  
            <SiteRef LocationOID="{{location-oid}}"/>  
            <StudyEventData StudyEventOID="{{folder-oid}}" StudyEventRepeatKey="{{folder-repeat-key}}" TransactionType="Update">  
                <FormData FormOID="{{form-oid}}" FormRepeatKey="{{form-repeat-key}}" TransactionType="Update">  
                    <ItemGroupData ItemGroupOID="{{record-oid}}" ItemGroupRepeatKey="{{item-group-repeat-key}}" TransactionType="Update">  
  
                        <!-- "Field level Attributes (with variations below)" -->  
  
                        <ItemData ItemOID="{{field-oid}}" Value="{{data}}"/>  
  
                    </ItemGroupData>  
                </FormData>  
            </StudyEventData>
```

```
</SubjectData>  
</ClinicalData>  
  
</ODM>
```



Note: I can only update one subject per request.

Field location attributes:

Attribute	Description	Mandatory?	Notes
StudyOID	The study identifier	Yes	See Locating my study
SubjectKey	The subject identifier	Yes	See Locating my subject
LocationOID	The site identifier	Yes	See Locating my site
StudyEventOID	The folder OID or “SUBJECT” for subject level forms	Yes	
StudyEventRepeatKey	The folder repeat key	No	See Locating my folder
FormOID	The form OID	Yes	
FormRepeatKey	The form repeat key	No	See Locating my form and Upserting my form
ItemGroupOID	The form OID or form OID + “_LOGLINE”	Yes	
ItemGroupRepeatKey	The log line repeat key or “1”	No	See Locating my log line and Upserting my log line

Field level attributes (with variations below):

Attribute	Description	Mandatory?	Notes
ItemOID	The field OID	Yes	See below
Value	The data	Yes	See below

Inserting a new subject

I can create a subject, and add data to it:

- [Creating my subject in a study and site](#)

Updating field data

I can update fields of these data types, control types and properties:

- [Text field](#)
- [Date field](#)
- [Unit dictionary field](#)
- [Data dictionary drop down list](#)
- [Data dictionary search list](#)
- [Data dictionary dynamic search list](#)

I can update these properties on a field:

- [Workflow properties](#) - Lock, Freeze, Verify

I can not update:

- Other workflow properties - Sign
- Derived fields, such as "Age".

Updating items belonging to a field

I can update these field items:

- [Review](#)
- [Queries](#) - Open, Forward, Answer, Cancel, Close
- [Translations](#)
- [Coding decisions](#)
- [External audits](#)

I cannot update:

- [Comments](#)
- [Stickies](#)

- Protocol Deviations

Examples

Best Practices

[Back to top](#)

The Response

My whole transaction will either:

1. Succeed with a HTTP Response code of 200 OK;
2. Fail and roll back, with a 4xx or 5xx HTTP Response code and RWS error code in the response body.

Header	Reason	Body	Notes
200	The transaction was successful	XML representation of the items touched	See Response Notes for more information on the returned XML.
X Y Z	Failure reason for X Failure reason for Y Failure reason for Z	response message	See error responses - the complete list

Examples of success responses

Header:

```
HTTP Response code : 200 OK
```

Body:

Case: Without "RequestIds" option.

```
<Response
    ReferenceNumber="43c40f30-9430-4147-9e2d-9dc902296301"
    InboundODMfileOID="2929--2"
    IsTransactionSuccessful = "1"
    SuccessStatistics="Rave objects touched: Subjects=1; Folders=0; Forms=0;
Fields=2; LogLines=0"
    NewRecords="" SubjectNumberInStudy="4956" SubjectNumberInStudySite="4955">
</Response>
```

Case: With "RequestIds = True" - Subject created and data updated; "SubjectId" and "TouchedRecordIds" are included in response.

```
<Response
    ReferenceNumber="43c40f30-9430-4147-9e2d-9dc902296301"
    InboundODMfileOID="2929--2"
    IsTransactionSuccessful = "1"
    SuccessStatistics="Rave objects touched: Subjects=1; Folders=0; Forms=0;
Fields=2; LogLines=0"
    NewRecords="" SubjectNumberInStudy="4956" SubjectNumberInStudySite="4955"
    SubjectId="6511" TouchedRecordIds="2220663,2220664">
</Response>
```

Case: With "RequestIds = True" - Only data is updated; "TouchedRecordIds" is included in response.

```
<Response
    ReferenceNumber="43c40f30-9430-4147-9e2d-9dc902296301"
    InboundODMfileOID="2929--2"
    IsTransactionSuccessful = "1"
    SuccessStatistics="Rave objects touched: Subjects=1; Folders=0; Forms=0;
Fields=2; LogLines=0"
    NewRecords="" SubjectNumberInStudy="4956" SubjectNumberInStudySite="4955"
    TouchedRecordIds="2220663,2220664">
</Response>
```

Case: With "RequestIds = True" - Only Subject is created; "SubjectId" is included in response.

```
<Response
    ReferenceNumber="43c40f30-9430-4147-9e2d-9dc902296301"
    InboundODMfileOID="2929--2"
    IsTransactionSuccessful = "1"
    SuccessStatistics="Rave objects touched: Subjects=1; Folders=0; Forms=0;
Fields=2; LogLines=0"
    NewRecords="" SubjectNumberInStudy="4956" SubjectNumberInStudySite="4955"
    SubjectId="6511">
</Response>
```

Example error response

Header:

403 HTTP Response code

Body:

```
<Response ReferenceNumber="fa63d085-629a-419e-a19d-21d6fb2bc93b"
    InboundODMfileOID="InputODM.xml"
    IsTransactionSuccessful="0"
    ReasonCode="RWS00042"
    ErrorOriginLocation="/ODM/ClinicalData[1]/SubjectData[1]/StudyEventData[1]/
FormData[1]/ItemGroupData[1]/ItemData[1]"
    SuccessStatistics="Rave objects touched: Subjects=0; Folders=0; Forms=0; Fields=0;
LogLines=0"
    ErrorClientResponseMessage="Field not authorised.">
</Response>
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

Examples - working with subject data

Rave Web Services (RWS) may be used to import clinical data into a Rave study. The result of importing clinical data through RWS is the same as making the equivalent change to clinical data through Rave's own EDC user

interface. This means any additional actions that are triggered by a clinical data change using the EDC user interface, are also triggered by a data import using RWS.

For example, in a typical Rave study, adding a new subject may trigger the automatic creation of a series of forms and folders related to that subject, without the need to run any additional commands. In such a study, adding a subject through RWS clinical data import also adds the same series of forms and folders. It is therefore important to understand the detailed structure of the Rave study you are working with before you use RWS to import clinical data.

The following pages include a sequence of clinical data import scenarios in ODM XML and equally illustrate the results of these imports in Rave. These scenarios are implemented with the Mediflex study which is available to registered users at <https://innovate.mdsol.com>. You can log in to this study using the user name and password provided by your Medidata representative. Taken together, these scenarios form an extended worked example of using RWS to accept ODM XML requests to update clinical data in a Rave study. Key points are explained in the text which accompanies each code extract.

See [How to Use the Operational Data Model \(ODM\)](#) for more information on how to correctly construct the ODM XML requests.

See the following examples:

Subject

- [Using a Secondary Subject Name](#)
- [Inserting a Subject](#)

Folder

- [Identifying Repeated Objects](#)
- [Updating a Form in a Nested Folder](#)
- [Locating Repeating Folders by Context](#)

Form

- [Updating a Form](#)
- [Inserting a Form](#)
- [Assigning a Lab to a Form](#)
- [Locating Repeating Forms by Context](#)
- [Updating or Inserting a Form Using TransactionType="Upsert"](#)

Logline

- Identifying Repeated Lines (Log Lines)
- Inserting and Removing Log Lines Across Multiple Forms
- Making a Record Inactive
- Reactivating an Inactive Log Line
- Locating Log Lines by Context
- Locating Repeating Log Lines and Repeating Forms by Context
- Updating or Creating a Logline Record

Field data

- Changing a Data Point's Status

Items belonging to a field

- Opening a Rave Query Using TransactionType-"Context"
- Coding a Data Point Using a Third-party Coding System
- Adding an External Audit Record
- Importing Translated Data

[Back to top](#)

1.5.11.1 Best Practices - Work With Subject Data

Scenario

I am updating clinical data and ...

I want to optimise my integrations across my studies

RWS Transfer Forms

A number of integrating clients are taking up the idea of "RWS Transfer Forms". A transfer form is a standard form, the same in every study, with a number of fixed fields (it might have log fields as well). Let's call these FieldA, FieldB, FieldC.

This form is used to make integration projects easier. Let's consider the case in which you need to integrate with an IVR vendor on 5 studies, all in different therapeutic areas. None of your forms are exactly the same, or even if they are alike, they are not always in the same place. For instance, the Demography form might be in a folder

called SCRN in one study, BASELN in another, SCREENING in a third and so on. Negotiating with your vendor and helping them understand the study structure so they can formulate an appropriate ODM message with the data they want to send you, can take a lot of time.

In this type of situation, you can use a transfer form, which must satisfy the following requirements:

- It is always at the subject level.
- It always has the same fields.
- It is always hidden from normal users via ViewRestrictions. This is important because it means that the data can never get frozen, locked, verified or otherwise involved in the workflow of your trial.

You, the study owner or builder, still have to do some work. This form is going to receive the data from the integration partner, but you are going to write edit checks or custom functions that decide what to do with that data, to "transfer" it to where you want it to go in your CRF, and under what circumstances. Using a CF you can check to see if the data from the vendor has changed from the original send, and whether the workflow is right to do the transfer of the data from the generic FieldA to the real field in your CRF. If it is not, at least you have the chance to log (via email or other mechanism) that this data arrived, and what your code decided to do with it.

Compared to the coordination work involved in setting up an integration custom for every study, the use of Transfer Forms makes it easier for the integrator, and brings control of the business logic for updating data back into the control of the study team.

[Back to top](#)

1.5.11.2 Example - Assign a Lab to a Form

You can assign a lab to a form using the vendor extension attributes mdsol:LaboratoryRef and mdsol:LaboratoryType. The lab must be defined in Rave before you can assign it to a form.

To assign a lab, use the following format of ODM.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
...  
  <FormData FormOID='DM' mdsol:LaboratoryRef='Uxbridge Lab Inc.'  
            mdsol:LaboratoryType='Central' transactionType="update">  
    </FormData>  
...  
</ODM>
```

[Back to top](#)

1.5.11.3 Example - Change the Data Point Status

RWS supports a number of custom vendor extension attributes for the ItemData element that can change the status of a data point in Rave. The ODM request in this example uses the attribute mdsol:Lock to lock a data point. Other attributes which are available are: freeze a datapoint (mdsol:Freeze), review a datapoint (mdsol:Review), and verify a datapoint (mdsol:Verify).

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>
...
<StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1"
TransactionType="Update">
    <FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="DM" TransactionType="Update">
            <ItemData ItemOID="BIRTHDT" Value="05 Jul 1986" mdsol:Lock="Yes">
                </ItemData>
            </ItemGroupData>
        </FormData>
    </StudyEventData>
...
</ODM>
```

[Back to top](#)

Response

The response below illustrates the data point has been locked.

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

Body

```
<Response ReferenceNumber="abc" InboundODMFileOID="LockStatus_To_1Yes"
IsTransactionSuccessful="1"
SuccessStatistics="Rave objects touched: Subjects=0; Folders=0; Forms=0; Fields=1;
LogLines=0" NewRecords="" >
</Response>
```

[Back to top](#)

1.5.11.4 Example - Identify Repeated Objects

The design of a study may permit multiple instances of an object, and in the Mediflex study you may have multiple folders representing multiple unscheduled visits. Rave identifies separate instances of these repeated objects with a special attribute, known as the [ObjectName]RepeatKey (for example, StudyEventRepeatKey). This identifies the position of this folder relative to other folders of the same type (folders that share the same OID in Rave). The value for the first instance of the [ObjectName]RepeatKey attribute must be 1, and this value must be incremented by 1 each time it is used, without spaces.

If the value of the StudyEventRepeatKey is incorrect, RWS returns the error response message: Non sequential or non consecutive folders are not permitted (RWS00030). Similar rules on sequential numbering apply in the case of repeating ItemGroup elements, which are used to represent log lines in Rave forms. In this instance, the attribute ItemGroupRepeatKey is used with the ItemGroupData element.

In the case of the ItemGroupData element, you can set the value of the ItemGroupRepeatKey attribute to LAST, which increments the sequence number of the repeat key. However the use of LAST is not recommended. The preferred method is to use the TransactionType of Upsert with the ItemGroupData element and to give the ItemGroupRepeatKey the special value of @context. See Updating or Creating a Logline Record for further information about using Upsert, and see Locating Repeating Forms by Context for further information about using the @context special value.

It is possible to use one ODM request to insert a folder and to insert a form within that folder for an existing subject. This is illustrated in the Mediflex study by inserting a folder for an Unscheduled Visit and inserting a form in that folder with data about the visit. The Mediflex study permits multiple unscheduled visit folders, which all share the same StudyEventOID, which is UNSCHEDULED. The request therefore includes the StudyEventRepeatKey attribute to identify a specific folder.

Setting the visit date triggers an edit check in Rave that renames the instance of the folder from its default value, Visit(1), to a name based on the visit date, such as Visit 21 Feb 2008.

The ODM request that creates the unscheduled visit is illustrated here:

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"  
FileType="Transactional" FileOID="Example-4" CreationDateTime="2008-01-01T00:00:00">  
  <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">  
    <SubjectData SubjectKey="123 ABC" TransactionType="Update">  
      <SiteRef LocationOID="4567"/>  
      <StudyEventData StudyEventOID="UNSCHEDULED" StudyEventRepeatKey="1"  
      TransactionType="Insert">  
        <FormData FormOID="VISIT">  
          <ItemGroupData ItemGroupOID="VISIT">  
            <ItemData ItemOID="DTC" Value="21 FEB 2008"/>  
          </ItemGroupData>  
        </FormData>  
      </StudyEventData>  
    </SubjectData>  
  </ClinicalData>  
</ODM>
```

[Back to top](#)

1.5.11.5 Example - Insert and Remove Log Lines Across Multiple Forms

You can use a single RWS request to insert and remove log line data across multiple forms relating to the same subject. Each form is identified by a unique value for its StudyEventOID attribute.

The Mediflex study permits multiple Pain Diary (ePRO) forms and each have log lines. In the example ODM request, these are identified as VISIT01 and VISIT02. In this example, these two forms already exist, consequently the request uses a TransactionType attribute value of Update, which is set at the SubjectData element level and is inherited by all the subsequent child elements.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"  
FileType="Transactional" FileOID="Example-7" CreationDateTime="2008-01-01T00:00:00">  
  <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">  
    <SubjectData SubjectKey="123 ABC" TransactionType="Update">  
      <SiteRef LocationOID="4567"/>  
      <StudyEventData StudyEventOID="VISIT01">  
        <FormData FormOID="FORM_PAIN_SI">  
          <ItemGroupData ItemGroupOID="FORM_PAIN_SI_LOG_LINE" ItemGroupRepeatKey="1">  
            <ItemData ItemOID="IT_DATE" Value="2008 01 01"/>  
            <ItemData ItemOID="IT_TIME" Value="12:00:00"/>  
            <ItemData ItemOID="IT_SEVERE" Value="50"/>  
            <ItemData ItemOID="IT_REC_ID" Value="12345678"/>  
          </ItemGroupData>  
        </FormData>  
      </StudyEventData>  
      <StudyEventData StudyEventOID="VISIT02">  
        <FormData FormOID="FORM_PAIN_SI">  
          <ItemGroupData ItemGroupOID="FORM_PAIN_SI_LOG_LINE" ItemGroupRepeatKey="1">  
            <ItemData ItemOID="IT_DATE" Value="2008 02 01"/>  
            <ItemData ItemOID="IT_TIME" Value="18:00:00"/>  
            <ItemData ItemOID="IT_SEVERE" Value="75"/>  
            <ItemData ItemOID="IT_REC_ID" Value="12345679"/>  
          </ItemGroupData>  
        </FormData>  
      </StudyEventData>
```

```
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>
```

[Back to top](#)

1.5.11.6 Example - Make a Record Inactive

You can make a record inactive by using a TransactionType attribute of Remove in an ODM request. In the ODM request shown in the example one log line is made inactive and another log line is added, in two separate ItemGroupData elements, each identified by its own ItemGroupRepeatKey attribute.

These two elements each have different settings for their TransactionType attributes, as illustrated in the following example.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"
FileType="Transactional" FileOID="Example-8" CreationDateTime="2008-01-01T00:00:00">
<ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">
<SubjectData SubjectKey="123 ABC" TransactionType="Update">
<SiteRef LocationOID="4567"/>
```

```
<StudyEventData StudyEventOID="VISIT01">
    <FormData FormOID="FORM_PAIN_SI">
        <ItemGroupData ItemGroupOID="FORM_PAIN_SI_LOG_LINE" ItemGroupRepeatKey="1"
TransactionType="Remove"/>
        <ItemGroupData ItemGroupOID="FORM_PAIN_SI_LOG_LINE" ItemGroupRepeatKey="2"
TransactionType="Insert">
            <ItemData ItemOID="IT_DATE" Value="2008 01 02"/>
            <ItemData ItemOID="IT_TIME" Value="18:00:00"/>
            <ItemData ItemOID="IT_SEVERE" Value="50"/>
            <ItemData ItemOID="IT_REC_ID" Value="12345680"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>
```

**Note:**

- If you try to insert a log line record in a form that does not have log line records, RWS returns the error response message: Non log line form, cannot insert multiple records (RWS00101).
- You cannot insert to or delete from a restricted log line. If you try to insert to a restricted log line, RWS returns the error response message: Record restricted by max limit (RWS00088). If you try to delete from a restricted log line, RWS returns the error response message Record inactivation not authorized (RWS00089).
- You cannot use a dynamic search list with RWS. If you try to use a dynamic search list, RWS returns the error response message Dynamic search list not supported (RWS00095).

[Back to top](#)

1.5.11.7 Example - Reactivate an Inactive Log Line

You can use the vendor extension attribute mdsol:submission="WholeItemGroup" with an ItemGroupData element as an alternative method to reactivate all the fields in a log line in one transaction.



Note: As of Classic Rave 2020.2.0, when a record is reactivated through a RWS POST request using the transaction type INSERT, all the data points are reactivated. As a result, the INSERT transaction type no longer allows you to reactivate and insert the data in the same POST; it reactivates the

loglines only. The UPSERT transaction type reactivates and inserts or updates the data in the same POST.

Example - Inactivate logline

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" FileType="Transactional" FileOID="2014-04-09 13:20:20 UTC" CreationDateTime="2014-04-09T13:20:20" ODMVersion="1.3">
  <ClinicalData StudyOID="RWS_study_with_folders4" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="JAN1601" TransactionType="Update">
      <SiteRef LocationOID="56789"/>
      <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1" TransactionType="Update">
        <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
          <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2" TransactionType="Remove" >
            <ItemData ItemOID="AEDESC" Value="HEADACHE1" TransactionType="Insert"/>
          </ItemGroupData>
        </FormData>
      </StudyEventData>
      </SubjectData>
    </ClinicalData>
  </ODM>
```

Example - Reactivate logline using Transaction Type = Insert

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" FileType="Transactional" FileOID="2014-04-09 13:20:20 UTC" CreationDateTime="2014-04-09T13:20:20" ODMVersion="1.3">
  <ClinicalData StudyOID="RWS_study_with_folders4" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="JAN1601" TransactionType="Update">
      <SiteRef LocationOID="56789"/>
      <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1" TransactionType="Update">
        <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
          <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2" TransactionType="Insert" >
            <ItemData ItemOID="AEDESC" Value="HEADACHE1" />
          </ItemGroupData>
        </FormData>
      </StudyEventData>
      </SubjectData>
    </ClinicalData>
  </ODM>
```

Example - Reactivate an inactive logline and Update field value using Transaction Type =Upsert

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata" FileType="Transactional" FileOID="2014-04-09 13:20:20 UTC" CreationDateTime="2014-04-09T13:20:20" ODMVersion="1.3">
  <ClinicalData StudyOID="RWS_study_with_folders4" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="JAN1601" TransactionType="Update">
      <SiteRef LocationOID="56789"/>
      <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1" TransactionType="Update">
        <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
          <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2" TransactionType="Upsert" >
            <ItemData ItemOID="AEDESC" Value="HEADACHEUpdated" />
          </ItemGroupData>
        </FormData>
      </StudyEventData>
      </SubjectData>
    </ClinicalData>
  </ODM>
```

1.5.11.8 Example - Use a Secondary Subject Name

In some clinical studies subjects are assigned additional or secondary names. RWS uses a vendor extension attribute mdsol:SecondarySubjectkey to address subjects by their secondary names.

Below is an example of a request that uses a TransactionType of Update to modify the record of an existing subject with an existing secondary name.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://  
www.mdsol.com/ns/odm/metadata"  
    ODMVersion="1.3" CreationDateTime="2008-01-01T00:00:00" FileOID="File1"  
    FileType="Transactional" >  
    <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">  
        <SubjectData SubjectKey="123 AB2ndName" mdsol:SecondarySubjectKey="Yes">  
            <SiteRef LocationOID="12345"></SiteRef>  
            <StudyEventData StudyEventOID="SCREEN">  
                <FormData FormOID="DM" TransactionType="Update">  
                    <ItemGroupData ItemGroupOID="DM" >  
                        <ItemData ItemOID="BRTHDTC" Value="23 Apr 1978"></ItemData>  
                    </ItemGroupData>  
                </FormData>  
                <StudyEventData>  
                </SubjectData>  
            </ClinicalData>  
</ODM>
```

[Back to top](#)

POST Request

Below is an example of a request that uses a TransactionType of Insert to create a new subject using a secondary subject name. In this case the subject is created with the default subject name of New Subject.

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:mdsol="http://  
www.mdsol.com/ns/odm/metadata  
"CreationDateTime="2008-05-20T05:00:19" FileOID="File1" FileType="Transactional"  
ODMVersion="1.3">  
  <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="28">  
    <SubjectData SubjectKey="2nd Subject Name" TransactionType="Insert"  
      mdsol:SecondarySubjectKey="Yes">  
      <SiteRef LocationOID="12345"/>  
    </SubjectData>  
  </ClinicalData>  
</ODM>
```

[Back to top](#)

1.6 Resources

The main RWS resources and their associated web services are:

Study

- [Retrieve Study and Library Metadata](#)
- [Work With Study and Library Metadata](#)
- [Retrieve Clinical Study Lists](#)
- [Retrieve Study and Library Metadata Lists](#)

Site

- Retrieve Admin Data with the Sites Dataset
- Locate a Site

Subject

- Locate a Subject
- Create a Subject in a Study and Site
- Retrieve the List of Subjects in a Study
- Work With Subject Data

Form

- Update a Whole Item Group
- Upsert a Form in RWS
- Locate a Form
- Retrieve Clinical View Form Datasets as CSV
- Form Definitions

Folder

- Locate a Folder
- Folder Definitions
- Navigating Repeated, Nested Folders

Log Line

- Upsert a Log Line
- Update a Whole Item Group
- Locate a Log Line
- Retrieve Custom Dataset Log Messages
- Log Line Definitions

Field

- Use Field Queries
- Update a Data Dictionary Drop Down List Field
- Update a Date Field
- Update a Unit Dictionary Field
- Add Coding Decisions to Fields
- Update Field Workflow Properties

- Add External Audits to Fields
- Add Translations to Fields
- Update a Dynamic Search List Field
- Update a Text Field
- Update a Data Dictionary Search List Field
- Review a Field
- Field Definitions

Lab

- Assign Labs to Forms

1.6.1 Create a Subject in a Study and Site

Use Case

I am updating clinical data and ...

I am creating my subject in a study and site



Note: Creating a new subject will also create the subjects folders and forms - as defined by the study default matrix.

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I am a valid Rave User.
2. I can update clinical data in my study.
3. My create request contains only one action, that is, <SubjectData> and <SiteRef> tags are listed only once in the ODM post.



Important: Prior to Rave 2018.1.0, RWS post requests supported subject creation with multiple <Subjectdata> elements in the same ODM post. However, from RaveX/Rave 2018.1.0 and onwards,

an RWS post request will fail if you attempt to insert/create a subject and update a separate event using multiple <Subjectdata> elements, on the same subject in a single ODM post.

For example, if you attempt to create a subject and populate the Demography form in the same RWS post request using multiple <Subjectdata> elements, the post fails and results in an error indicating “Subject doesn’t exist”.

If the <SubjectData> and <SiteRef> are listed only once in the ODM post, then the RWS post executes successfully. For more details, please refer to the [When does RWS create/insert a subject with no audits? FAQ](#).

My ODM snippet

```
...
<ClinicalData StudyOID="{study-oid}" >
  <SubjectData
    SubjectKey="{subject-key}"
    TransactionType="Insert"
  >
    <SiteRef LocationOID="{location-oid}"/>
  ...
</SubjectData>
</ClinicalData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
SubjectKey	The subject identifier	Yes	
StudyOID	The study identifier	Yes	
LocationOID	The location identifier	Yes	The LocationOID is case sensitive.
TransactionType	The transaction type	Yes	Upsert is not allowed when creating new subjects.

**Note:**

- I need to specify both the study and site when creating a subject.
- The site already exists in the study.
- I can create my subject with or without the primary form data (or any data).
- In the same request transaction, I can add data to any forms.
- When adding data in the same request, forms in the default matrix get created with the new subject, so an "Update" transaction type is specified for the folder and form.

Examples

Example: I am adding my subject - without any form data.

This example creates my subject named "123 ABC" in site "4567" in the "Mediflex(Dev)" study.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"
FileType="Transactional" FileOID="Example-1"
CreationDateTime="2008-01-01T00:00:00">
<ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">
  <SubjectData SubjectKey="123 ABC" TransactionType="Insert">
    <SiteRef LocationOID="4567"/>
  </SubjectData>
</ClinicalData>
</ODM>
```

Example: I am adding my subject - with default matrix form data.

This example creates my subject named "123 ABC" in site "4567" in the "Mediflex(Dev)" study. This example updates data in my subjects' "SCREENING" / "MH" form. Note that this folder and form is part of the default matrix set created with the inserted subject, so an "Update" transaction type is specified for the folder and form.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" Description="abc"
      CreationDateTime="2008-05-20T05:00:19" Granularity="SingleSubject"
FileType="Transactional"
      FileOID="LongOID1@xml" AsOfDateTime="2009-06-12T11:00:19" ODMVersion="1.3">
<ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">
  <SubjectData SubjectKey="123 ABC" TransactionType="Insert">
    <SiteRef LocationOID="4567" />
    <StudyEventData StudyEventOID="SCREENING" StudyEventRepeatKey="1"
```

```
TransactionType="Update">
    <FormData FormOID="MH" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="MH" TransactionType="Insert">
            <ItemData ItemOID="MH_DT" Value="06 Jan 2009" TransactionType="Insert">
            </ItemData>
        </ItemGroupData>
    </FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>
```

Example: Inserting a Subject in RWS.

1.6.1.1 Example - Insert a Subject

The following ODM XML requests illustrate how to use RWS to add a subject to Rave and subsequently how to update that subject by inserting the subject and then updating the corresponding demographic details form. These are two separate RWS requests.

 **Important:** Prior to Rave 2018.1.0, RWS post requests supported subject creation with multiple `<Subjectdata>` elements in the same ODM post. However, from RaveX/Rave 2018.1.0 and onwards, an RWS post request will fail if you attempt to insert/create a subject and update a separate event using multiple `<Subjectdata>` elements, on the same subject in a single ODM post.

For example, if you attempt to create a subject and populate the Demography form in the same RWS post request using multiple `<Subjectdata>` elements, the post fails and results in an error indicating “Subject doesn’t exist”.

If the `<SubjectData>` and `<SiteRef>` are listed only once in the ODM post, then the RWS post executes successfully. For more details, please refer to the [When does RWS create/insert a subject with no audits? FAQ](#).

In this request example, the Rave project name and environment for this study, Mediflex(Dev), is specified in the `StudyOID` attribute of the `ClinicalData` element. Although RWS ignores the `MetaDataVersionOID`, it must be present for this to be valid ODM XML.

The `TransactionType` attribute for the `SubjectData` element shown in the request has the value `Insert` because the data item (the subject) does not previously exist in the study. When you use a `TransactionType` of `Insert`, you must

also specify the site with the SiteRef element. This tells Rave which site the new subject belongs to. The site you are adding the subject to must already exist in your Rave study.

The ODM XML request shown here adds a subject named 123 ABC:

POST Request

Header

```
Authorization: Basic ZGV2OnBhc3N  
Content-type: "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"  
FileType="Transactional" FileOID="Example-1" CreationDateTime="2008-01-01T00:00:00">  
  <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">  
    <SubjectData SubjectKey="123 ABC" TransactionType="Insert">  
      <SiteRef LocationOID="4567"/>  
    </SubjectData>  
  </ClinicalData>  
</ODM>
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

1.6.2 Locate a Folder

Use Case

I am updating clinical data and ...

I am locating my folder by its OID

or

I am locating my repeating folder by its OID and repeat key

or

I am locating my nested repeating folder by its OID and repeat key

or

I am locating my repeating folder by its OID and a context value in one of its form fields

My ODM snippet

```
...
<SubjectData >
  <SiteRef />
  <StudyEventData StudyEventOID="{folder-oid}"
    StudyEventRepeatKey="{folder-repeat-key}"
    TransactionType="Update" >
    <FormData >
      ...
    </FormData>
  </StudyEventData>
</SubjectData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
StudyEventOID	The folder OID	Yes	
StudyEventRepeatKey	The folder repeat key	No	See below

StudyEventRepeatKey:

- For non-repeating folders I don't need this attribute
- For repeating folders I can use the numeric value or "@CONTEXT"
- For nested repeating folders I can use the Rave folder path `FolderX[m] /.../ FolderY[n]` *

* X and Y refer to Folder OIDs, and m and n refer to the relative ordinal number with respect to its parent folder.

See [Nested folder navigation](#) for more details.

Examples

Example: I am locating my non-repeating folder.

```
...
<StudyEventData StudyEventOID="FolderC" >
...
```

Example: I am locating my repeating folder.

```
...
<StudyEventData StudyEventOID="FolderC" StudyEventRepeatKey="2" >
...
```

Example: I am locating my nested repeating folder.

```
...
<StudyEventData StudyEventOID="FolderC" StudyEventRepeatKey="FolderA[2]/FolderB[1]/
FolderC[1]" >
...
```

Example: I am locating my repeating folder by a context value of a field in one of its forms.

```
...
<StudyEventData StudyEventOID="FolderC" StudyEventRepeatKey="@CONTEXT" >
<FormData >
  <ItemGroupData >
    <ItemData ItemOID="TEXT" Value="data" TransactionType="Context" />
  </ItemGroupData>
  ...
</FormData>
```

```
</StudyEventData>  
...
```

i Note: For StudyEventRepeatKey="@CONTEXT" - The form then folder is located where the field data value matches that supplied using the "Context" transaction type.

Example: Updating a form in a nested folder.

Example: Locating repeating folders by context.

i Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

```
Given I am a user  
And there is a study  
And the study has the following structure:  
| folder | form | form_view_restriction | form_entry_restriction |  
field | field_visible | field_order_number | field_view_restriction |  
field_entry_restriction |  
| VISIT | VISIT_INFO | | |  
VISIT_DATE | Yes | | 1 |  
|  
| VISIT | VISIT_INFO | | |  
VISIT_OUTCOME | Yes | | 2 |  
|  
| VISIT | VISIT_LOCATION | Data Manager | Investigator |  
LOCATION_OID | Yes | | 1 | Clinical Research Associate |  
Investigator |  
| VISIT | VISIT_LOCATION | Data Manager | Investigator |  
LOCATION_ADDRESS | No | | 2 |  
|  
And there is a subject in the study
```

Scenario: Update a field inside folder by locating the folder using one form as context

Given the subject has the following data:

```

| folder | folder_transaction | folder_repeat | form      | form_transaction |
form_repeat | field      | value      | field_transaction |
| VISIT   | Update     | 1          | VISIT_INFO | Update    |
1           | VISIT_DATE | 29 Jan 2009 | Insert     |           |
| VISIT   | Insert     | 2          | VISIT_INFO | Insert    |
1           | VISIT_DATE | 30 Jan 2009 | Insert     |           |
When I update the subject with the following ODM:
"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID "VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Remove folder by locating the folder using one form as context

Given the subject has the following data:

```

| folder | folder_transaction | folder_repeat | form      | form_transaction |
form_repeat | field      | value      | field_transaction |
| VISIT   | Update     | 1          | VISIT_INFO | Update    |
1           | VISIT_DATE | 29 Jan 2009 | Insert     |           |
| VISIT   | Insert     | 2          | VISIT_INFO | Insert    |
1           | VISIT_DATE | 30 Jan 2009 | Insert     |           |
When I update the subject with the following ODM:
"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Remove">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Context" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            </ItemGroupData>
        </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the folder with OID "VISIT" and repeat key "1" has been removed

Scenario: Update a field inside folder by locating the folder using more than one form as context

Given the subject has the following data:

```

| folder | folder_transaction | folder_repeat | form           | form_transaction
| form_repeat | field        | value       | field_transaction |
| VISIT    | Update      | 1          | VISIT_INFO   | Update
| 1         | VISIT_DATE  | 29 Jan 2009 | Insert       |
| VISIT    | Update      | 1          | VISIT_LOCATION | Update
| 1         | LOCATION_OID | London     | Insert       |
| VISIT    | Insert      | 2          | VISIT_INFO   | Insert
| 1         | VISIT_DATE  | 30 Jan 2009 | Insert       |
| VISIT    | Insert      | 2          | VISIT_LOCATION | Insert
| 1         | LOCATION_OID | Edinburgh  | Insert       |
When I update the subject with the following ODM:
"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" >
            <ItemData ItemOID="LOCATION_OID" Value="London" />
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID "VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Remove folder by locating the folder using more than one form as context
Given the subject has the following data:

```

| folder | folder_transaction | folder_repeat | form           | form_transaction
| form_repeat | field        | value       | field_transaction |
| VISIT    | Update      | 1          | VISIT_INFO   | Update
| 1         | VISIT_DATE  | 29 Jan 2009 | Insert       |
| VISIT    | Update      | 1          | VISIT_LOCATION | Update
| 1         | LOCATION_OID | London     | Insert       |
| VISIT    | Insert      | 2          | VISIT_INFO   | Insert
| 1         | VISIT_DATE  | 30 Jan 2009 | Insert       |
| VISIT    | Insert      | 2          | VISIT_LOCATION | Insert
| 1         | LOCATION_OID | Edinburgh  | Insert       |
When I update the subject with the following ODM:
"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Remove">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Context">
```

```

<ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Context" >
    <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
</ItemGroupData>
</FormData>
<FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
    <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
        <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
    </ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the folder with OID "VISIT" and repeat key "1" has been removed

Scenario: Update a field inside folder by locating the folder using an entry restricted form as context

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction
	form_repeat	field	value	field_transaction	
	VISIT	Update	1	VISIT_INFO	Update
1		VISIT_DATE	29 Jan 2009	Insert	
1		Update	1	VISIT_LOCATION	Update
1		LOCATION_OID	London	Insert	
1		Insert	2	VISIT_INFO	Insert
1		VISIT_DATE	30 Jan 2009	Insert	
1		Insert	2	VISIT_LOCATION	Insert
1		LOCATION_OID	Edinburgh	Insert	

And I am user with role "Investigator" on the study

When I update the subject with the following ODM:

"""

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the field with OID "VISIT_OUTCOME" in the form with OID "VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Update a field inside folder by locating the folder using an entry restricted field as context

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO
		VISIT_DATE	29 Jan 2009	Insert
1	VISIT	Update	1	VISIT_LOCATION
		LOCATION_OID	London	Insert
1	VISIT	Insert	2	VISIT_INFO
		VISIT_DATE	30 Jan 2009	Insert
1	VISIT	Insert	2	VISIT_LOCATION
		LOCATION_OID	Edinburgh	Insert

And I am user with role "Investigator" on the study

When I update the subject with the following ODM:

"""

```
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""
```

Then I should receive a successful XML response

And I should see the field with OID "VISIT_OUTCOME" in the form with OID "VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Update a field inside folder by locating the folder using a locked field as context

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	lock
1	VISIT	Update	1	VISIT_INFO
		VISIT_DATE	29 Jan 2009	Insert
1	VISIT	Update	1	VISIT_LOCATION
		LOCATION_OID	London	Insert
1	VISIT	Insert	2	VISIT_INFO
		VISIT_DATE	30 Jan 2009	Insert
1	VISIT	Insert	2	VISIT_LOCATION
		LOCATION_OID	Edinburgh	Insert

When I update the subject with the following ODM:

"""

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: StudyEventRepeatKey="@Context" should be case insensitive

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	
form_repeat	field	value		field_transaction		
1	VISIT	Update	1	VISIT_INFO	Update	
	VISIT_DATE	29 Jan 2009	Insert			
1	VISIT	Insert	2	VISIT_INFO	Insert	
	VISIT_DATE	30 Jan 2009	Insert			

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@ConText"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: No context fields provided

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	
form_repeat	field	value		field_transaction		
	VISIT	Update	1	VISIT_INFO	Update	

```

| 1 | VISIT_DATE | 29 Jan 2009 | Insert |
| VISIT | Update | 1 | VISIT_LOCATION | Update
| 1 | LOCATION_OID | London | Insert |
| VISIT | Insert | 2 | VISIT_INFO | Insert
| 1 | VISIT_DATE | 30 Jan 2009 | Insert |
| VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | Edinburgh | Insert |
When I update the subject with the following ODM:
"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Update"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Update" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Conflict"
And reason code "RWS00155"
And the message "No context provided"

Scenario: More than one matching folders found

Given the subject has the following data:

```

| folder | folder_transaction | folder_repeat | form | form_transaction
| form_repeat | field | value | field_transaction |
| VISIT | Update | 1 | VISIT_INFO | Update
| 1 | VISIT_DATE | 29 Jan 2009 | Insert |
| VISIT | Update | 1 | VISIT_LOCATION | Update
| 1 | LOCATION_OID | London | Insert |
| VISIT | Insert | 2 | VISIT_INFO | Insert
| 1 | VISIT_DATE | 29 Jan 2009 | Insert |
| VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | London | Insert |
When I update the subject with the following ODM:
"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>

```

```

        <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
    </ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Conflict"
And reason code "RWS00159"
And the message "Unique folder not found"

Scenario: No matching folders found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
VISIT	Update	1	VISIT_INFO	Update
1	VISIT_DATE	29 Jan 2009	Insert	
VISIT	Update	1	VISIT_LOCATION	Update
1	LOCATION_OID	London	Insert	
VISIT	Insert	2	VISIT_INFO	Insert
1	VISIT_DATE	30 Jan 2009	Insert	
VISIT	Insert	2	VISIT_LOCATION	Insert
1	LOCATION_OID	Edinburgh	Insert	

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="31 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00160"
And the message "No matching folder found"

Scenario: Deactivated folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
VISIT	Update	1	VISIT_INFO	Update
1	VISIT_DATE	29 Jan 2009	Insert	
VISIT	Update	1	VISIT_LOCATION	Update
1	LOCATION_OID	London	Insert	
VISIT	Insert	2	VISIT_INFO	Insert
1	VISIT_DATE	30 Jan 2009	Insert	

```

| VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | Edinburgh | Insert |
| VISIT | Remove | 1 | VISIT_INFO | Update
| 1 | VISIT_DATE | 29 Jan 2009 | Update |
When I update the subject with the following ODM:
"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: View restricted form causes folder not found

Given the subject has the following data:

```

| folder | folder_transaction | folder_repeat | form | form_transaction
| form_repeat | field | value | field_transaction |
| VISIT | Update | 1 | VISIT_INFO | Update
| 1 | VISIT_DATE | 29 Jan 2009 | Insert |
| VISIT | Update | 1 | VISIT_LOCATION | Update
| 1 | LOCATION_OID | London | Insert |
| VISIT | Insert | 2 | VISIT_INFO | Insert
| 1 | VISIT_DATE | 30 Jan 2009 | Insert |
| VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | Edinburgh | Insert |
And I am user with role "Data Manager" on the study
When I update the subject with the following ODM:
"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: View restricted field causes folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO
		VISIT_DATE	29 Jan 2009	Insert
1	VISIT	Update	1	VISIT_LOCATION
		LOCATION_OID	London	Insert
1	VISIT	Insert	2	VISIT_INFO
		VISIT_DATE	30 Jan 2009	Insert
1	VISIT	Insert	2	VISIT_LOCATION
		LOCATION_OID	Edinburgh	Insert

And I am user with role "Clinical Research Associate" on the study

When I update the subject with the following ODM:

"""

```
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
<FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
    <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
        <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
    </ItemGroupData>
</FormData>
</StudyEventData>
"""
```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: Deactivated form causes folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO
		VISIT_DATE	29 Jan 2009	Insert
1	VISIT	Update	1	VISIT_LOCATION
		LOCATION_OID	London	Insert
1	VISIT	Insert	2	VISIT_INFO
		VISIT_DATE	30 Jan 2009	Insert
1	VISIT	Insert	2	VISIT_LOCATION
		LOCATION_OID	Edinburgh	Insert
1	VISIT	Update	1	VISIT_INFO
		VISIT_DATE	29 Jan 2009	Update

When I update the subject with the following ODM:

"""

```
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
<FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
    <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
        <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
```

```

</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00160"
And the message "No matching folder found"

Scenario: Search on invisible field causes folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO
	VISIT_DATE	29 Jan 2009	Insert	
1	VISIT	Update	1	VISIT_LOCATION
	LOCATION_OID	London	Insert	

When I update the subject with the following ODM:

```

"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_ADDRESS" Value="" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00160"
And the message "No matching folder found"

Scenario: Empty folder should return folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO
	VISIT_DATE	29 Jan 2009	Insert	
1	VISIT	Update	1	VISIT_LOCATION
	LOCATION_OID	London	Insert	

When I update the subject with the following ODM:

```

"""

```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>

```

```
"""
Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00160"
And the message "No matching folder found"
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

1.6.2.1 Example - Locate Repeating Folders by Context

The special @CONTEXT value may be used for the StudyEventRepeatKey attribute in the StudyEventData element, to locate a folder when its repeat key is unknown and the values of fields within the folder are.

In the following example, the date of a visit is known (ItemOID="VISIT_DATE" Value="29 Jan 2019") and the repeat key of the folder containing the visit form is not. This permits the outcome field (ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN") to be updated.

POST Request1

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>
...
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2019"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
...
</ODM>
```

It is also possible to use the special @CONTEXT value for the FormRepeatKey attribute for a form at the same time as using it for the StudyEventRepeatKey attribute for a folder in a single operation.

In the following example, the @CONTEXT value is used for both the StudyEventRepeatKey and the FormRepeatKey.

POST Request2

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>
...
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="@CONTEXT" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update">
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2019"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="Pain killer given"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
...
</ODM>
```

The locate folders by context feature is only available for top-level folders and not for nested folders. The field or fields used to identify the folder must be standard fields, not log-line fields. This feature can be used to identify and remove a folder, as well as to perform operations on objects in the folder. If you use the special @CONTEXT value for a form to identify a folder, the TransactionType of Upsert is not permitted for that form.

1.6.2.2 Example - Locate My Folder

Locating My Folder Involves ...

Scenario

I am updating clinical data <update-subject-data> and ...

I am locating my folder by its OID

or

I am locating my repeating folder by its OID and repeat key

or

I am locating my nested repeating folder by its OID and repeat key

or

I am locating my repeating folder by its OID and a context value in one of its form fields

ODM

```
My ODM snippet
...
<SubjectData >
  <SiteRef />
  <StudyEventData StudyEventOID="{folder-oid}"
    StudyEventRepeatKey="{folder-repeat-key}"
    TransactionType="Update" >
    <FormData >
      ...
    </FormData>
  </StudyEventData>
</SubjectData>
...
```

ODM Attributes

Attribute	Description	Mandatory?	Notes
StudyEventOID	OID of the study event	Yes	
StudyEventRepeatKey	Key used for repeating the folder	Yes	
TransactionType	Type of transaction	Yes	Update
FormData	Form data associated with the study event	No	

StudyEventOID	The folder OID	Yes	
StudyEventRepeatKey	The folder repeat key	No	See below

StudyEventRepeatKey:

- For non-repeating folders I don't need this attribute
- For repeating folders I can use the numeric value or "["@CONTEXT](#)"
- For nested repeating folders I can use the Rave folder path `FolderX[m] /.../FolderY[n]` `^{1*}`

`1*` X and Y refer to Folder OIDs and m and n refer to the relative ordinal number with respect to its parent folder.

See Nested folder navigation `<nested-folder-navigation>` for more details.

Examples

I am locating my non-repeating folder [2]:

```
...
<StudyEventData StudyEventOID="FolderC" >
...
...
```

I am locating my repeating folder [3]:

```
...
<StudyEventData StudyEventOID="FolderC" StudyEventRepeatKey="2" >
...
...
```

I am locating my nested repeating folder [4]:

```
...
<StudyEventData StudyEventOID="FolderC" StudyEventRepeatKey="FolderA[2]/FolderB[1]/
FolderC[1]" >
...
...
```

I am locating my repeating folder by a context value of a field in one of its forms [5]:

```
...
<StudyEventData StudyEventOID="FolderC" StudyEventRepeatKey="@CONTEXT" >
<FormData >
  <ItemGroupData >
    <ItemData ItemOID="TEXT" Value="data" TransactionType="Context" />
  </ItemGroupData>
...
...
```

```
</FormData>
</StudyEventData>
...

```

Note

i For `StudyEventRepeatKey="@CONTEXT"` - The form then folder is located where the field data value matches that is supplied using the "Context" transaction type.

Scenarios

Background:

```
Given I am a user
And there is a study
And the study has the following structure:
```

Background:

```
Given I am a user
And there is a study
And the study has the following structure:
```

field	folder	form	form_view_restriction	form_entry_restriction	field_visible	field_order_number	field_view_restriction	field_entry_restriction
VISIT_DATE	VISIT	VISIT_INFO			Yes	1		
VISIT_OUTCOME	VISIT	VISIT_INFO			Yes	2		
LOCATION_OID	VISIT	VISIT_LOCATION	Data Manager		Yes	1	Investigator	
Investigator	LOCATION	VISIT_LOCATION	Clinical Research Associate					
LOCATION_ADDRESS	VISIT	VISIT_LOCATION	Data Manager		No	2	Investigator	
					And there is a subject in the study			

Scenario: Update a field inside folder by locating the folder using one form as context

Given the subject has the following data:

form_repeat	field	value	field_transaction	form	form_transaction
1	VISIT	Update	1	VISIT_INFO	Update
	VISIT_DATE	29 Jan 2009	Insert		
1	VISIT	Insert	2	VISIT_INFO	Insert
	VISIT_DATE	30 Jan 2009	Insert		

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

Then I should receive a successful XML response
```

And I should see the field with OID "VISIT_OUTCOME" in the form with OID "VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Remove folder by locating the folder using one form as context

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	
form_repeat	field		value	field_transaction		
1	VISIT	Update	1	VISIT_INFO	Update	
			VISIT_DATE	29 Jan 2009	Insert	
1	VISIT	Insert	2	VISIT_INFO	Insert	
			VISIT_DATE	30 Jan 2009	Insert	

When I update the subject with the following ODM:

```
"""<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Remove">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Context" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

Then I should receive a successful XML response
```

And I should see the folder with OID "VISIT" and repeat key "1" has been removed

Scenario: Update a field inside folder by locating the folder using more than one form as context

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	
form_repeat	field		value	field_transaction		
1	VISIT	Update	1	VISIT_INFO	Update	
			VISIT_DATE	29 Jan 2009	Insert	
	VISIT	Update	1	VISIT_LOCATION	Update	
1			LOCATION_OID	London	Insert	

```

| VISIT | Insert | 2 | VISIT_INFO | Insert
| 1 | VISIT_DATE | 30 Jan 2009 | Insert |
| VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | Edinburgh | Insert |
When I update the subject with the following ODM:
"""

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" >
            <ItemData ItemOID="LOCATION_OID" Value="London" />
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Remove folder by locating the folder using more than one form as context
Given the subject has the following data:
| folder | folder_transaction | folder_repeat | form | form_transaction
| form_repeat | field | value | field_transaction |
| VISIT | Update | 1 | VISIT_INFO | Update
| 1 | VISIT_DATE | 29 Jan 2009 | Insert |
| VISIT | Update | 1 | VISIT_LOCATION | Update
| 1 | LOCATION_OID | London | Insert |
| VISIT | Insert | 2 | VISIT_INFO | Insert
| 1 | VISIT_DATE | 30 Jan 2009 | Insert |
| VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | Edinburgh | Insert |
When I update the subject with the following ODM:
"""

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Remove">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Context" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">

```

```

<ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
    <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the folder with OID "VISIT" and repeat key "1" has been removed

Scenario: Update a field inside folder by locating the folder using an entry restricted form as context

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction		
1	VISIT	Update	1	VISIT_INFO	Update
	VISIT	Update	1	VISIT_LOCATION	Update
	VISIT	LOCATION_OID	London	Insert	
	VISIT	Insert	2	VISIT_INFO	Insert
	VISIT	VISIT_DATE	30 Jan 2009	Insert	
	VISIT	Insert	2	VISIT_LOCATION	Insert
	1	LOCATION_OID	Edinburgh	Insert	

And I am user with role "Investigator" on the study

When I update the subject with the following ODM:

```
"""
{.sourceCode .xml}
```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the field with OID "VISIT_OUTCOME" in the form with OID "VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Update a field inside folder by locating the folder using an entry restricted field as context

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction		
	VISIT	Update	1	VISIT_INFO	Update

```
| 1 | VISIT_DATE | 29 Jan 2009 | Insert |
| 1 | VISIT | Update | 1 | VISIT_LOCATION | Update
| 1 | LOCATION_OID | London | Insert |
| 1 | VISIT | Insert | 2 | VISIT_INFO | Insert
| 1 | VISIT_DATE | 30 Jan 2009 | Insert |
| 1 | VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | Edinburgh | Insert |

```

And I am user with role "Investigator" on the study

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Update a field inside folder by locating the folder using a locked field as context

Given the subject has the following data:

```
| folder | folder_transaction | folder_repeat | form | form_transaction
| form_repeat | field | value | field_transaction | lock |
| 1 | VISIT | Update | 1 | VISIT_INFO | Update
| 1 | VISIT_DATE | 29 Jan 2009 | Insert |
| 1 | VISIT | Update | 1 | VISIT_LOCATION | Update
| 1 | LOCATION_OID | London | Insert | Yes |
| 1 | VISIT | Insert | 2 | VISIT_INFO | Insert
| 1 | VISIT_DATE | 30 Jan 2009 | Insert |
| 1 | VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | Edinburgh | Insert | Yes |

```

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>

```

```

    </ItemGroupData>
</FormData>
<FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
    <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
        <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
    </ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: StudyEventRepeatKey="@Context" should be case insensitive

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	
form_repeat	field	value		field_transaction		
1	VISIT Update	1		VISIT_INFO Update		
	VISIT_DATE 29 Jan 2009 Insert					
1	VISIT Insert	2		VISIT_INFO Insert		
	VISIT_DATE 30 Jan 2009 Insert					

When I update the subject with the following ODM:

```

"""
{.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@ConText"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: No context fields provided

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction	
form_repeat	field	value		field_transaction		
1	VISIT Update	1		VISIT_INFO Update		
	VISIT_DATE 29 Jan 2009 Insert					
1	VISIT Update	1		VISIT_LOCATION Update		
	LOCATION_OID London Insert					
1	VISIT Insert	2		VISIT_INFO Insert		
	VISIT_DATE 30 Jan 2009 Insert					
1	VISIT Insert	2		VISIT_LOCATION Insert		

```
| 1           | LOCATION_OID | Edinburgh    | Insert          |
When I update the subject with the following ODM:
"""

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Update"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Update" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Conflict"

And reason code "RWS00155"

And the message "No context provided"

Scenario: More than one matching folders found

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	form_transaction
1	form_repeat	field	value	field_transaction	
		VISIT	Update	1	VISIT_INFO Update
		VISIT	Update	1	VISIT_LOCATION Update
		VISIT	Insert	1	VISIT_INFO Insert
		VISIT	Insert	2	VISIT_LOCATION Insert
		VISIT	Insert	2	VISIT_LOCATION Insert
		VISIT	Insert	1	VISIT_LOCATION Insert

When I update the subject with the following ODM:

```
"""

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Conflict"
 And reason code "RWS00159"
 And the message "Unique folder not found"

Scenario: No matching folders found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO Update
1	VISIT	Update	1	VISIT_LOCATION Update
1	VISIT	Insert	2	VISIT_INFO Insert
1	VISIT	Insert	30 Jan 2009	Insert
1	VISIT	Insert	2	VISIT_LOCATION Insert
1	VISIT	Insert	Edinburgh	Insert

When I update the subject with the following ODM:

```
"""
  {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="31 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""


```

Then I should receive an HTTP response with status code "Not Found"
 And reason code "RWS00160"
 And the message "No matching folder found"

Scenario: Deactivated folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO Update
1	VISIT	Update	1	VISIT_LOCATION Update
1	VISIT	Insert	2	VISIT_INFO Insert
1	VISIT	Insert	30 Jan 2009	Insert
1	VISIT	Insert	2	VISIT_LOCATION Insert
1	VISIT	Remove	1	VISIT_INFO Update
1	VISIT	Update	29 Jan 2009	Update

When I update the subject with the following ODM:

```
"""
  {.sourceCode .xml}
```

```

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: View restricted form causes folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO Update
	VISIT	Update	1	VISIT_LOCATION Update
1	LOCATION_OID	London	Insert	
1	Insert	2	VISIT_INFO Insert	
	VISIT_DATE	30 Jan 2009	Insert	
1	Insert	2	VISIT_LOCATION Insert	
1	LOCATION_OID	Edinburgh	Insert	

And I am user with role "Data Manager" on the study

When I update the subject with the following ODM:

```

"""
{.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: View restricted field causes folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO Update
	VISIT_DATE	29 Jan 2009	Insert	

```
| VISIT | Update | 1 | VISIT_LOCATION | Update
| 1 | LOCATION_OID | London | Insert |
| VISIT | Insert | 2 | VISIT_INFO | Insert
| 1 | VISIT_DATE | 30 Jan 2009 | Insert |
| VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | Edinburgh | Insert |

```

And I am user with role "Clinical Research Associate" on the study

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: Deactivated form causes folder not found

Given the subject has the following data:

```
| folder | folder_transaction | folder_repeat | form | form_transaction
| form_repeat | field | value | field_transaction |
| VISIT | Update | 1 | VISIT_INFO | Update
| 1 | VISIT_DATE | 29 Jan 2009 | Insert |
| VISIT | Update | 1 | VISIT_LOCATION | Update
| 1 | LOCATION_OID | London | Insert |
| VISIT | Insert | 2 | VISIT_INFO | Insert
| 1 | VISIT_DATE | 30 Jan 2009 | Insert |
| VISIT | Insert | 2 | VISIT_LOCATION | Insert
| 1 | LOCATION_OID | Edinburgh | Insert |
| VISIT | Update | 1 | VISIT_INFO | Remove
| 1 | VISIT_DATE | 29 Jan 2009 | Update |

```

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: Search on invisible field causes folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
VISIT	Update	1	VISIT_INFO	Update
1	VISIT_DATE	29 Jan 2009	Insert	
1	Update	1	VISIT_LOCATION	Update
	LOCATION_OID	London	Insert	

When I update the subject with the following ODM:

```
"""
{.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_ADDRESS" Value="" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00160"
And the message "No matching folder found"
```

Scenario: Empty folder should return folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
VISIT	Update	1	VISIT_INFO	Remove
1	VISIT_DATE	29 Jan 2009	Insert	
1	Update	1	VISIT_LOCATION	Remove
	LOCATION_OID	London	Insert	

When I update the subject with the following ODM:

```
"""
{.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00160"
And the message "No matching folder found"
```

folder	folder_transaction	folder_repeat	form	form_transaction	form_repeat	field	value	field_transaction	
VISIT	Update	1	VISIT_INFO	Update	1	VISIT_DATE	29Jan2009	Insert	
VISIT	Insert	2	VISIT_INFO	Insert	1	VISIT_DATE	30Jan2009	Insert	

Scenarios

Background:

```
Given I am a user
And there is a study
And the study has the following structure:
```

Background:

```
Given I am a user
And there is a study
And the study has the following structure:
| folder | form           | form_view_restriction | form_entry_restriction |
| field   | field_visible | field_order_number | field_view_restriction
| field_entry_restriction |
| VISIT   | VISIT_INFO    |                      |                      |
VISIT_DATE      | Yes          |                      |                      |
|                  |              |                      |                      |
| VISIT   | VISIT_INFO    |                      |                      |
VISIT_OUTCOME    | Yes          |                      |                      |
|                  |              |                      |                      |
| VISIT   | VISIT_LOCATION | Data Manager       | Investigator        |
LOCATION_OID     | Yes          |                      | Clinical Research Associate
| Investigator      |
| VISIT   | VISIT_LOCATION | Data Manager       | Investigator        |
LOCATION_ADDRESS | No           |                      |                      |
|                  |              |                      |                      |
And there is a subject in the study
```

Scenario: Update a field inside folder by locating the folder using one form as context

Given the subject has the following data:

```
| folder | folder_transaction | folder_repeat | form           | form_transaction |
form_repeat | field      | value       | field_transaction |
| VISIT   | Update      | 1           | VISIT_INFO | Update      |
1           | VISIT_DATE | 29 Jan 2009 | Insert      |             |
| VISIT   | Insert      | 2           | VISIT_INFO | Insert      |
1           | VISIT_DATE | 30 Jan 2009 | Insert      |             |
```

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
<FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
<ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
<ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
<ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
</ItemGroupData>
</FormData>
```

```
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Remove folder by locating the folder using one form as context

Given the subject has the following data:

form_repeat	field	value	field_transaction	form	form_transaction
1	VISIT	Update	1	VISIT_INFO	Update
		VISIT_DATE	29 Jan 2009	Insert	
1	VISIT	Insert	2	VISIT_INFO	Insert
		VISIT_DATE	30 Jan 2009	Insert	

When I update the subject with the following ODM:

```
"""

```

```
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Remove">
<FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Context">
<ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Context" >
<ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the folder with OID "VISIT" and repeat key "1" has been removed

Scenario: Update a field inside folder by locating the folder using more than one form as context

Given the subject has the following data:

form_transaction	form_repeat	field	value	field_transaction	form
Update	1	VISIT	1	VISIT_INFO	
		Update	VISIT_DATE	29 Jan 2009	Insert
Update	1	VISIT	1	VISIT_LOCATION	
		Update	LOCATION_OID	London	Insert
Insert	1	VISIT	2	VISIT_INFO	
		Insert	VISIT_DATE	30 Jan 2009	Insert
Insert	1	VISIT	2	VISIT_LOCATION	
		Insert	LOCATION_OID	Edinburgh	Insert

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}

```

```
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
<FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
<ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
<ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
```

```

        <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
    </ItemGroupData>
</FormData>
<FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
    <ItemGroupData ItemGroupOID="VISIT_LOCATION" >
        <ItemData ItemOID="LOCATION_OID" Value="London" />
    </ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Remove folder by locating the folder using more than one form as context
Given the subject has the following data:

form_transaction	form_repeat	field	value	field_transaction
Update	1	VISIT_DATE	29 Jan 2009	Insert
Update	1	LOCATION_OID	London	Insert
Insert	1	VISIT_DATE	30 Jan 2009	Insert
Insert	1	LOCATION_OID	Edinburgh	Insert

When I update the subject with the following ODM:

```

""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Remove">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Context" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the folder with OID "VISIT" and repeat key "1" has been removed

Scenario: Update a field inside folder by locating the folder using an entry restricted form as context

Given the subject has the following data:

```

| folder | folder_transaction | folder_repeat | form           |
form_transaction | form_repeat | field      | value       | field_transaction |
| VISIT   | Update        | 1          | VISIT_INFO  |
Update          | 1          | VISIT_DATE | 29 Jan 2009 | Insert      |
| VISIT   | Update        | 1          | VISIT_LOCATION |
Update          | 1          | LOCATION_OID | London     | Insert      |
| VISIT   | Insert         | 1          | VISIT_INFO  |
Insert          | 1          | VISIT_DATE | 30 Jan 2009 | Insert      |
| VISIT   | Insert         | 2          | VISIT_LOCATION |
Insert          | 1          | LOCATION_OID | Edinburgh | Insert      |
Insert          | 1          | LOCATION_OID | Edinburgh | Insert      |

And I am user with role "Investigator" on the study
When I update the subject with the following ODM:
"""

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""
Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Update a field inside folder by locating the folder using an entry
restricted field as context
Given the subject has the following data:
| folder | folder_transaction | folder_repeat | form           |
form_transaction | form_repeat | field      | value       | field_transaction |
| VISIT   | Update        | 1          | VISIT_INFO  |
Update          | 1          | VISIT_DATE | 29 Jan 2009 | Insert      |
| VISIT   | Update        | 1          | VISIT_LOCATION |
Update          | 1          | LOCATION_OID | London     | Insert      |
| VISIT   | Insert         | 1          | VISIT_INFO  |
Insert          | 1          | VISIT_DATE | 30 Jan 2009 | Insert      |
| VISIT   | Insert         | 2          | VISIT_LOCATION |
Insert          | 1          | LOCATION_OID | Edinburgh | Insert      |
Insert          | 1          | LOCATION_OID | Edinburgh | Insert      |

And I am user with role "Investigator" on the study
When I update the subject with the following ODM:
"""

<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">

```

```

<FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
    <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
        <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
    </ItemGroupData>
</FormData>
<FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
    <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
        <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
    </ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: Update a field inside folder by locating the folder using a locked field as context

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form			
form_transaction	form_repeat	field		value		field_transaction	lock
Update	Update	1		VISIT_DATE	29 Jan 2009	Insert	
Update	Update	1		LOCATION_OID	London	Insert	
Insert	Insert	1		VISIT_DATE	30 Jan 2009	Insert	
Insert	Insert	1		LOCATION_OID	Edinburgh	Insert	

When I update the subject with the following ODM:

```

"""
{.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>

```

```
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: StudyEventRepeatKey="@Context" should be case insensitive

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	form_transaction
form_repeat	field	value	field_transaction	
1	VISIT	Update	1	VISIT_INFO Update
		VISIT_DATE	29 Jan 2009	Insert
1	VISIT	Insert	2	VISIT_INFO Insert
		VISIT_DATE	30 Jan 2009	Insert

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@ConText"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the field with OID "VISIT_OUTCOME" in the form with OID
"VISIT_INFO" has the value "PAIN KILLER GIVEN"

Scenario: No context fields provided

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	
form_transaction	form_repeat	field	value	field_transaction
Update	1	1	VISIT_INFO	
1	VISIT	Update	VISIT_DATE	29 Jan 2009 Insert
			1	VISIT_LOCATION
Update	1	1	LOCATION_OID	London Insert
			2	VISIT_INFO
Insert	1	1	VISIT_DATE	30 Jan 2009 Insert
			2	VISIT_LOCATION
Insert	1	1	LOCATION_OID	Edinburgh Insert

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
```

```

        <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Update"/>
        <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
    </ItemGroupData>
</FormData>
<FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Update">
    <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Update" >
        <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Update"/>
    </ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Conflict"
And reason code "RWS00155"
And the message "No context provided"

Scenario: More than one matching folders found

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	
form_transaction	form_repeat	field		value	field_transaction
Update	VISIT	Update	1	1	VISIT_INFO
Update	VISIT	Update	1	VISIT_DATE	29 Jan 2009
Update	VISIT	Update	1	LOCATION_OID	London
Insert	VISIT	Insert	1	VISIT_DATE	29 Jan 2009
Insert	VISIT	Insert	1	LOCATION_OID	London

When I update the subject with the following ODM:

```

""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
            <ItemData ItemOID="VISIT_OUTCOME" Value="PAIN KILLER GIVEN"
TransactionType="Update"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Conflict"
And reason code "RWS00159"
And the message "Unique folder not found"

Scenario: No matching folders found

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	
--	--------	--------------------	---------------	------	--

```

form_transaction | form_repeat | field           | value      | field_transaction |
| VISIT   | Update          | 1             | VISIT_INFO | |
Update          | 1               | VISIT_DATE   | 29 Jan 2009 | Insert    |
| VISIT   | Update          | 1             | VISIT_LOCATION | |
Update          | 1               | LOCATION_OID | London     | Insert    |
| VISIT   | Insert          | 2             | VISIT_INFO | |
Insert          | 1               | VISIT_DATE   | 30 Jan 2009 | Insert    |
| VISIT   | Insert          | 2             | VISIT_LOCATION | |
Insert          | 1               | LOCATION_OID | Edinburgh | Insert
When I update the subject with the following ODM:
"""

```

```



```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: Deactivated folder not found

Given the subject has the following data:

```

| folder | folder_transaction | folder_repeat | form           |
form_transaction | form_repeat | field           | value      | field_transaction |
| VISIT   | Update          | 1             | VISIT_INFO | |
Update          | 1               | VISIT_DATE   | 29 Jan 2009 | Insert    |
| VISIT   | Update          | 1             | VISIT_LOCATION | |
Update          | 1               | LOCATION_OID | London     | Insert    |
| VISIT   | Insert          | 2             | VISIT_INFO | |
Insert          | 1               | VISIT_DATE   | 30 Jan 2009 | Insert    |
| VISIT   | Insert          | 2             | VISIT_LOCATION | |
Insert          | 1               | LOCATION_OID | Edinburgh | Insert
| VISIT   | Remove          | 1             | VISIT_INFO | |
Update          | 1               | VISIT_DATE   | 29 Jan 2009 | Update

```

When I update the subject with the following ODM:

```

"""


```

```

        TransactionType="Update"/>
    
```

Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00160"
And the message "No matching folder found"

Scenario: View restricted form causes folder not found

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	
form_transaction	form_repeat	field		value	field_transaction
Update	Update	VISIT	1	VISIT_INFO	
Update	1	VISIT_DATE	29 Jan 2009	Insert	
Update	Update	VISIT	1	VISIT_LOCATION	
Insert	1	LOCATION_OID	London	Insert	
Insert	Insert	VISIT	1	VISIT_INFO	
Insert	1	VISIT_DATE	30 Jan 2009	Insert	
Insert	Insert	VISIT	2	VISIT_LOCATION	
Insert	1	LOCATION_OID	Edinburgh	Insert	

And I am user with role "Data Manager" on the study

When I update the subject with the following ODM:

```

        """ { .sourceCode .xml }
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: View restricted field causes folder not found

Given the subject has the following data:

	folder	folder_transaction	folder_repeat	form	
form_transaction	form_repeat	field		value	field_transaction
Update	Update	VISIT	1	VISIT_INFO	
Update	1	VISIT_DATE	29 Jan 2009	Insert	
Update	Update	VISIT	1	VISIT_LOCATION	
Insert	1	LOCATION_OID	London	Insert	
Insert	Insert	VISIT	1	VISIT_INFO	
Insert	1	VISIT_DATE	30 Jan 2009	Insert	
Insert	Insert	VISIT	2	VISIT_LOCATION	
Insert	1	LOCATION_OID	Edinburgh	Insert	

And I am user with role "Clinical Research Associate" on the study

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_LOCATION" FormRepeatKey="1" TransactionType="Context">
        <ItemGroupData ItemGroupOID="VISIT_LOCATION" TransactionType="Context" >
            <ItemData ItemOID="LOCATION_OID" Value="London" TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: Deactivated form causes folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	field	value	field_transaction
form_transaction	form_repeat					
VISIT	Update			1	VISIT_INFO	
Update		1		VISIT_DATE	29 Jan 2009	Insert
VISIT	Update			1	VISIT_LOCATION	
Update		1		LOCATION_OID	London	Insert
VISIT	Insert			2	VISIT_INFO	
Insert		1		VISIT_DATE	30 Jan 2009	Insert
VISIT	Insert			2	VISIT_LOCATION	
Insert		1		LOCATION_OID	Edinburgh	Insert
VISIT	Update			1	VISIT_INFO	
Remove		1		VISIT_DATE	29 Jan 2009	Update

When I update the subject with the following ODM:

```
""" {.sourceCode .xml}
<StudyEventData StudyEventOID="VISIT" StudyEventRepeatKey="@CONTEXT"
TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemData ItemOID="VISIT_DATE" Value="29 Jan 2009"
TransactionType="Context"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00160"

And the message "No matching folder found"

Scenario: Search on invisible field causes folder not found

Given the subject has the following data:

folder	folder_transaction	folder_repeat	form	field	value	field_transaction
form_transaction	form_repeat					
VISIT	Update			1	VISIT_INFO	

```

Update | 1 | VISIT_DATE | 29 Jan 2009 | Insert |
| VISIT | Update | 1 | VISIT_LOCATION |
Update | 1 | LOCATION_OID | London | Insert |
When I update the subject with the following ODM:
"""


```

Footnotes

[1] Some attributes are omitted from the ODM document for clarity. See ODM Schema <odm-schema> for more details

[2] Some attributes are omitted from the ODM document for clarity. See ODM Schema <odm-schema> for more details

[3] Some attributes are omitted from the ODM document for clarity. See ODM Schema <odm-schema> for more details

[4] Some attributes are omitted from the ODM document for clarity. See ODM Schema <odm-schema> for more details

[5] Some attributes are omitted from the ODM document for clarity. See ODM Schema <odm-schema> for more details

1.6.2.3 Example - Update a Form in a Nested Folder

The way in which RWS addresses items in nested and repeated folders is explained in the section Folder Navigation for Repeated Objects. An example of this, specific to the Mediflex study, is to update a subject's Vital Signs form, which is in a nested folder. This requires a refinement of the use of the [ObjectName]RepeatKey as it needs to refer to a nested object. In this instance, this is the value of the StudyEventRepeatKey attribute, which is set to YEAR1[1]/CYCLE1[1], referring to the Cycle 01 sub-folder of the Year 01 folder in the Mediflex study. This request is illustrated below.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"
FileType="Transactional" FileOID="Example-4" CreationDateTime="2008-01-01T00:00:00">
<ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">
<SubjectData SubjectKey="123 ABC" TransactionType="Update">
<SiteRef LocationOID="4567"/>
<StudyEventData StudyEventOID="CYCLE1" StudyEventRepeatKey="YEAR1[1]/CYCLE1[1]">
<FormData FormOID="VS">
<ItemGroupData ItemGroupOID="VS">
<ItemData ItemOID="VSSTAT" Value="Y"/>
<ItemData ItemOID="VSDTC" Value="01 JAN 2008"/>
<ItemData ItemOID="SYSBP" Value="120"/>
<ItemData ItemOID="DIABP" Value="80"/>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>
```

[Back to top](#)

1.6.3 Locate a Form

Use Case

I am updating clinical data and ...

I am locating my form by its OID

or

I am locating my repeating form by its OID and repeat key

or

I am locating my nested repeating form by its OID and repeat key

or

I am locating my repeating form by its OID and a context value in one of its fields

My ODM snippet

```
...
<StudyEventData    >
  <FormData FormOID="{form-oid}"
            FormRepeatKey="{form-repeat-key}"   >
    ...
  </FormData>
</StudyEventData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
FormOID	The form OID	Yes	
FormRepeatKey	The form repeat key	No	See below

FormRepeatKey:

- For non-repeating forms I don't need this attribute
- For repeating forms I can use the numeric value or "@CONTEXT"

Examples

Example: I am tracking my non-repeating form.

```
...  
<FormData FormOID="FormC" >  
...
```

Example: I am locating my repeating form.

```
...  
<FormData FormOID="FormC" FormRepeatKey="2" >  
...
```

Example: I am locating my repeating form by a context value of a field.

```
...  
<FormData FormOID="FormC" FormRepeatKey="@CONTEXT" >  
  <ItemGroupData >  
    <ItemData ItemOID="TEXT" Value="data" TransactionType="Context" />  
  </ItemGroupData>  
</FormData>  
...
```

 **Note:** For FormRepeatKey="@CONTEXT", the form is located where the field data value matches the one supplied using the "Context" transaction type.

Example: Updating a form.

Example: Inserting a form.

Example: Locating repeating forms by context.

Example: Locating repeating log lines and repeating forms by context.



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

```
Given I am a user
And there is a study with repeating forms
```

Scenario Outline: Success

```
When I make a request to <perform an action> a repeating <form> <located> using
context on standard items
```

```
Then I should receive a successful response
And the body of the response should be XML indicating success
```

Examples:

<pre> perform an action form located </pre>			
locate mixed form at subject level			
locate standard form at subject level			
locate mixed form inside a folder			
locate standard form inside a folder			
update mixed form at subject level			
update standard form at subject level			
update mixed form inside a folder			
update standard form inside a folder			
remove mixed form at subject level			
remove standard form at subject level			
remove mixed form inside a folder			
remove standard form inside a folder			

Scenario Outline: Locate or remove entry restricted forms

```
Given I do not have permission to enter data on the form
```

```
When I make a request to <perform an action> an entry restricted repeating <form>
using context on standard items
```

```
Then I should receive a successful response
And the body of the response should be XML indicating success
```

Examples:

<pre> perform an action form </pre>		
locate mixed form		
locate standard form		
remove mixed form		
remove standard form		

Scenario: FormRepeatKey=@Context should be case-insensitive

When I make a request to update a repeating form located using FormRepeatKey=@Context on standard items

Then I should receive a successful response

And the body of the response should be XML indicating success

Scenario: Marking action on field used for locating repeating form

When I make a request to locate a repeating mixed form using context on standard items

And lock the field used for locating the form

Then I should receive a successful response

And the body of the response should be XML indicating success

Scenario Outline: Locate on a locked field

When I make a request to update a repeating <form> using context on a locked standard field

Then I should receive a successful response

And the body of the response should be XML indicating success

Examples:

form	
mixed form	
standard form	

Scenario: Locate a form without context items

When I make a request to locate a repeating form without context items

Then I should receive an HTTP response with status code "Conflict"

And reason code "RWS00155"

And the message "No context provided"

Scenario Outline: No Matching Form Found

When I make a request to locate a repeating <form> <located> that does not exist using context on standard items

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00154"

And the message "No matching form found"

Examples:

form	located	
standard form	inside a folder	
standard form	at subject level	
mixed form	inside a folder	
mixed form	at subject level	

Scenario Outline: Deactivated Form

When I make a request to locate on a repeating <form> <located> that is deactivated using context on standard items

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00154"

And the message "No matching form found"

Examples:

form	located	
standard form	inside a folder	
standard form	at subject level	
mixed form	inside a folder	
mixed form	at subject level	

Scenario Outline: More than one matching form

Given there are multiple forms with the same context values

When I make a request to locate a repeating <form> <located> using context

Then I should receive an HTTP response with status code "Conflict"

And reason code "RWS00153"

And the message "Unique form not found"

Examples:

form	located	
standard form	inside a folder	
standard form	at subject level	
mixed form	inside a folder	
mixed form	at subject level	

Scenario Outline: View Restricted Form

Given I do not have permission to view the form

When I make a request to locate a repeating <form> <located> using context

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00031"

And the message "Form not authorised"

Examples:

form	located	
standard form	inside a folder	
standard form	at subject level	
mixed form	inside a folder	
mixed form	at subject level	

Scenario Outline: Update entry restricted form

Given I do not have permission to enter data on the form

When I make a request to update an entry restricted repeating <form> using context on standard items

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00031"

And the message "Form not authorised"

Examples:

form	
mixed form at subject level	
standard form inside a folder	

Scenario Outline: more than 1 found but has view restriction

```
Given I do not have permission to view the form
And there are multiple forms with the same context values
When I make a request to locate a repeating <form> <located> using context
Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00031"
And the message "Form not authorised"
```

Examples:

form	located	
mixed form	at subject level	
standard form	inside a folder	



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

1.6.3.1 Example - Insert a Form

This example illustrates how to randomize the subject by adding a Randomization form to the Screening folder, and then setting the randomization date. In the Mediflex study, this form needs to be added manually each time it is used. Therefore, when it is created using an RWS request, the FormData element in the request needs a TransactionType of 'Insert', as it is adding a new form to the folder. In this example, the parent element, SubjectData, has a TransactionType of Update which is not suitable. The TransactionType value for FormData is inherited by its child elements.

If the Randomization form already existed and you wanted to add the randomization date, you would not need to add a separate TransactionType attribute as the inherited value would be suitable.

The ODM XML request in this example inserts the Randomization form in the FormData element, and in the subsequent child elements, it sets the date and assigns the subject to a study arm.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"  
FileType="Transactional" FileOID="Example-3" CreationDateTime="2008-01-01T00:00:00">  
  <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">  
    <SubjectData SubjectKey="123 ABC" TransactionType="Update">  
      <SiteRef LocationOID="4567"/>  
      <StudyEventData StudyEventOID="SCREEN">  
        <FormData FormOID="RAND" TransactionType="Insert">  
          <ItemGroupData ItemGroupOID="RAND">  
            <ItemData ItemOID="DSSTDT" Value="01 JAN 2008"/>  
            <ItemData ItemOID="ARMCD" Value="2"/>  
          </ItemGroupData>  
        </FormData>  
      </StudyEventData>  
    </SubjectData>  
  </ClinicalData>  
</ODM>
```

[Back to top](#)

1.6.3.2 Example - Locate Repeating Forms by Context

This example illustrates how to locate repeating forms by context.

- If you do not specify a value for the FormRepeatKey attribute of a repeating form, consequently the default behavior of RWS uses the value 1.
- If you do not know the FormRepeatKey value, you can find the correct instance of a repeating form by using a special value for the RepeatKey attribute for the form and known values as key data for one or more standard fields on the form with the TransactionType of Context. (A standard field on a Rave form will not be in a log line record.)

- The special value for the RepeatKey attribute is @CONTEXT.
- If RWS identifies one, and only one, form that matches the criteria indicated by the value or values with the TransactionType of Context, then actions in the ODM request are completed. Use of the TransactionType of Context can also locate, or remove fields as well as update them, and also works with locked fields and queries.
- If RWS cannot find the matching form or cannot complete the requested action, it returns an error. The reason code and error response message indicate why the request was not successful.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
  ...  
  <FormData FormOID="LAB" FormRepeatKey="@CONTEXT" TransactionType="Update">  
    <ItemGroupData ItemGroupOID="LAB" TransactionType="Update" >  
      <ItemData ItemOID="SAMPLE_ID" Value="875964" TransactionType="Context"/>  
      <ItemData ItemOID="HEMATOCRIT" Value="48.1" TransactionType="Update"/>  
    </ItemGroupData>  
  </FormData>  
  ...  
</ODM>
```

In the example, the FormRepeatKey attribute is not known for a lab form. This demonstrates how to use the FormRepeatKey of @CONTEXT to show that the value of this attribute is unknown, along with the

TransactionType of Context for a known value in the form (in the example, the sample ID). This permits the value for the analyte, HEMATOCRIT, to be updated.

[Back to top](#)

1.6.3.3 Example - Locate Repeating Log Lines and Repeating Forms by Context

This example illustrates how to combine the use of FormRepeatKey="@CONTEXT" and ItemGroupRepeatKey="@CONTEXT" in one ODM request to locate repeating log lines and folders by context.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
...  
    <FormData FormOID="LAB" FormRepeatKey="@CONTEXT" TransactionType="Update">  
        <ItemGroupData ItemGroupOID="LAB" TransactionType="Update" >  
            <ItemData ItemOID="SAMPLE_ID" Value="456987" TransactionType="Context"/>  
        </ItemGroupData>  
        <ItemGroupData ItemGroupOID="LAB" ItemGroupRepeatKey="@CONTEXT"  
        TransactionType="Update" >  
            <ItemData ItemOID="ANALYTE" Value="HEMATOCRIT" TransactionType="Context"/>  
            <ItemData ItemOID="VALUE" Value="47.9" TransactionType="Update"/>
```

```
</ItemGroupData>
</FormData>
...
</ODM>
```

In this example, the form is identified by the value of the Sample ID field, and the log line field to be updated is identified by the value of the ANALYTE field.

[Back to top](#)

1.6.3.4 Example - Update a Form

The next sample ODM request, illustrated here, updates the demographic information for the subject added in the previous request.

- The Mediflex study has been designed so that whenever a new subject is added, a set of folders and forms is created for the subject automatically. Therefore, the value of the TransactionType attribute for a request to add demographic information for the new subject is 'Update' as the form already exists.
- The folders and forms that are added in this study design include an initial screening folder (StudyEventOID="SCREEN") and a demographic information form (FormOID="DM"). In requests that have a TransactionType value of Update, you can omit the SiteRef element if the subject is unique across the whole study.
- Child elements of the SubjectData element automatically inherit its TransactionType value. You only need to specify the TransactionType attribute of a child element where the value is different from that of the parent element.
- The data points (field values) for the Demographics form are indicated by a series of ItemData elements, each with a different ItemOID attribute and a different value. The values for the RACE and COUNTRY ItemOID attributes use coded values from associated data dictionaries. Date values must be in the same format as the corresponding date field in the Rave study.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"  
FileType="Transactional" FileOID="Example-2" CreationDateTime="2008-01-01T00:00:00">  
  <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">  
    <SubjectData SubjectKey="123 ABC" TransactionType="Update">  
      <SiteRef LocationOID="4567"/>  
      <StudyEventData StudyEventOID="SCREEN">  
        <FormData FormOID="DM">  
          <ItemGroupData ItemGroupOID="DM">  
            <ItemData ItemOID="BRTHDT" Value="01 JAN 1980"/>  
            <ItemData ItemOID="SEX" Value="MALE"/>  
            <ItemData ItemOID="RACE" Value="3"/>  
            <ItemData ItemOID="COUNTRY" Value="GBR"/>  
          </ItemGroupData>  
        </FormData>  
      </StudyEventData>  
    </SubjectData>  
  </ClinicalData>  
</ODM>
```

[Back to top](#)

1.6.4 Locate a Log Line

Use Case

I am updating clinical data and ...

I am locating my form by its OID

or

I am locating my repeating log line by its OID and repeat key

or

I am locating my repeating log line by its OID and a context value in one of its fields

My ODM snippet

```
...
<FormData>
  <ItemGroupData ItemGroupOID="{record-oid}" ItemGroupRepeatKey="{log-line-repeat-
key}">
  ...
  </ItemGroupData>
</FormData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
ItemGroupOID	The log line OID	Yes	
ItemGroupRepeatKey	The log line repeat key	No	See below

ItemGroupRepeatKey value:

- For repeating log lines I can use the numeric value or "@CONTEXT"

Examples

Example: I am locating my repeating log line and I know the repeat number.

```
...
<ItemGroupData ItemGroupOID="LOGLINE_FormC" ItemGroupRepeatKey="2" />
...
```

Example: I am locating my repeating log line by a context value of a field.

```
...
<ItemGroupData ItemGroupOID="LOGLINE_FormC" ItemGroupRepeatKey="@CONTEXT" >
  <ItemData ItemOID="TEXT" Value="data" TransactionType="Context" />
  ...
</ItemGroupData>
...
```



Note: For ItemGroupRepeatKey="@CONTEXT", the log line is located where the field data value matches that supplied using the "Context" transaction type.

Example: Updating or creating a log line record.

Example: Identifying repeated log lines.

Example: Locating log lines by context.

Example: Locating repeating log lines and repeating forms by context.



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

```
Given I am a user
And there is a study with repeating forms
```

Scenario Outline: Success where ItemGroupRepeatKey is @CONTEXT

```
Given there is a <form> in the study
When I make a request to <perform an action> on a logline in the <form> with
ItemGroupRepeatKey=<repeat key>
Then I should receive a successful response
And the body of the response should be XML indicating success
```

Examples:

form	perform an action	repeat key
mixed form	locate	@CONTEXT
mixed form	update	@context
mixed form	remove	@Context
log form	locate	@ConText
log form	update	@CONtext
log form	remove	@conTEXT

Scenario: Locating a repeating form and locating a log line within the form

```
Given there are mixed forms in the study
When I make a request to update a logline with ItemGroupRepeatKey=@CONTEXT in the
mixed form located using context on standard field
Then I should receive a successful response
And the body of the response should be XML indicating success
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.4.1 Example - Identify Repeated Log Lines

Some forms in the Rave EDC module use a tabular format to record multiple sets of datapoints, which are known as log lines. Whenever you create an instance of a Rave form which uses log lines, the first line is created automatically (although initially this line does not contain data values). This occurs when you create the form manually through the Rave EDC module interface and when the form is created by applying a default matrix to another event. Similarly, if you use an RWS request to create a Rave form which uses log lines, the first line is created with the form.

Rave identifies each log line with an [ObjectName]RepeatKey attribute, which in this case is ItemGroupRepeatKey. The first line always has an ItemGroupRepeatKey attribute value of 1, and the value of this attribute is incremented by 1 for each subsequent line. As the first log line is always created when its parent form is created has implications for the value of the TransactionType attribute. Adding data to the first log line, which already exists, requires a TransactionType of Update, while adding data to any subsequent lines, which do not already exist, requires a TransactionType of Insert.

An example of a form which uses log lines is the Adverse Events form in the Mediflex study. In the ODM request example, a number of changes are made to the Adverse Events form. You do not need to enter values for all the data points (fields) in a log line when you create or update it.

The changes are as follows:

- * The question on the form; Were there Adverse Events? is set to Yes.
- * Four data points (fields) in the first log line are updated.
- * A second log line is added with three data points (fields) in it.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" ODMVersion="1.3"  
FileType="Transactional" FileOID="Example-6" CreationDateTime="2008-01-01T00:00:00">  
  <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">  
    <SubjectData SubjectKey="123 ABC" TransactionType="Update">  
      <SiteRef LocationOID="4567"/>  
      <StudyEventData StudyEventOID="SUBJECT">
```

```
<FormData FormOID="AE">
  <ItemGroupData ItemGroupOID="AE">
    <ItemData ItemOID="AEYN" Value="Y"/>
  </ItemGroupData>
  <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
    <ItemData ItemOID="AETERM" Value="Headache"/>
    <ItemData ItemOID="AESTDTC" Value="01 JAN 2008"/>
    <ItemData ItemOID="AEONG" Value="N"/>
    <ItemData ItemOID="AEENDTC" Value="01 JAN 2008"/>
  </ItemGroupData>
  <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2"
    TransactionType="Insert">
    <ItemData ItemOID="AETERM" Value="Joint pain"/>
    <ItemData ItemOID="AESTDTC" Value="01 JAN 2008"/>
    <ItemData ItemOID="AEONG" Value="Y"/>
  </ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>
```

[Back to top](#)

1.6.4.2 Example - Locate Log Lines by Context

When updating a log line entry in a Rave EDC form, you can identify the line by using a Repeatkey attribute such as ItemGroupRepeatKey, as described in the section. If you do not know the ItemGroupRepeatKey value for a log line item group on a Rave form, you can find the correct instance by using a special value for the RepeatKey attribute and known field values or key values for one or more other log line fields on the form using the TransactionType of Context.

The special value for the RepeatKey attribute is @CONTEXT.

If the line is not found, or if the identifiers do not allow RWS to find a unique line, RWS returns an error.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
...  
  <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">  
    <ItemGroupData ItemGroupOID="AE_LOGLINE" ItemGroupRepeatKey="@CONTEXT"  
      TransactionType="Update">  
      <ItemDataAny ItemOID="AETERM" TransactionType="Context">Headache</  
      ItemDataAny>  
      <ItemDataDate ItemOID="AESTDTC" TransactionType="Context">2020-03-14  
      </ItemDataDate>  
      <ItemDataDate ItemOID="AEENDTC">2020-03-20</ItemDataDate>  
    </ItemGroupData>  
  </FormData>  
...  
</ODM>
```

In this ODM request example, the log line to be located is identified as the line where the AETERM field has the value Headache and the AESTDTC field has the value 2020-03-14. If both these field values are found, this request updates the AEENDTC field with the value 2020-03-20.

[Back to top](#)

1.6.4.3 Example - Update or Create a Logline Record

You can use a single RWS request to either update an existing logline record, if it exists, or to create a new logline record, if it does not exist. To do this you must use the TransactionType of Upsert with both the ItemGroupData element, which is the parent of the ItemData element that represents the logline, and with the ItemData element itself. An example is illustrated below.

POST Request1

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
...  
  <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">  
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">  
      <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="3"  
        TransactionType="Upsert" >  
        <ItemData ItemOID="AEDESC" Value="Chest Pain"  
          TransactionType="Upsert"/>  
      </ItemGroupData>  
    </FormData>  
  </StudyEventData>  
...  
</ODM>
```

When you do not know the logline number you cannot set the correct value for the ItemGroupRepeatKey attribute. In this instance you can use the special value of @context for the ItemGroupRepeatKey attribute, together with the TransactionType of Upsert to help identify the logline. You must also have at least one ItemData element with the TransactionType of context to act as the key value to locate the logline.

The use of Upsert and @context together is the preferred method for incrementing the ItemGroupRepeatKey attribute, starting with RWS version 1.5.0. The alternative method of using LAST as the value for the ItemGroupRepeatKey is no longer recommended. Using LAST does not give you a record of the logline number used, which you need for subsequent updates, unless you capture the RWS response. We strongly recommend

that you create a unique identifier column (or composite column) and use the combination of ItemGroupRepeatKey="@Context" and TransactionType="Upsert" for the ItemGroupData element to locate the logline using key values and to update or insert the logline.

An example of this combined use is illustrated below.

[Back to top](#)

POST Request2

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
  ...  
  <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">  
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">  
      <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="@context"  
        TransactionType="Upsert">  
        <ItemData ItemOID="AEID" Value="101" TransactionType="Context"/>  
        <ItemData ItemOID="AEDESC" Value="Chest Pain" TransactionType="Upsert"/>  
      </ItemGroupData>  
    </FormData>  
  </StudyEventData>  
  ...  
</ODM>
```

[Back to top](#)

1.6.5 Locate a Site

Use Case

I am updating clinical data and ...

I am locating my site by its name

or

I am locating my site by its unique identifier



Note: A site's unique identifier is not related to iMedidata.

My ODM snippet

```
...
<ClinicalData  >
  <SubjectData  >
    <SiteRef
      LocationOID="{location-oid}"
      mdsol:LocationOIDType="{location-oid-type}" />
    <StudyEventData  >
      ...
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
LocationOID	The location identifier	Yes	The LocationOID is case sensitive.
mdsol:LocationOIDType	The location identifier type	No	See below

**Note:**

- When mdsol:LocationOIDType="SiteUUID" : The location identifier is the UUID
- When mdsol:LocationOIDType="SiteName" : The location identifier is the site number (default)
- When mdsol:LocationOIDType is missing : The location identifier is the site number (default)
- Every Site in Rave has a unique identifier, stored as [Sites].[UUID]

Examples

Example: I am locating my site by number.

```
...
<ClinicalData >
  <SubjectData >
    <SiteRef LocationOID="12345" mdsol:LocationOIDType="SiteName" />
    <StudyEventData >
      ...
    </StudyEventData>
  </SubjectData>
</ClinicalData>
...
```

```
...
<ClinicalData >
  <SubjectData >
    <SiteRef LocationOID="12345" />
    <StudyEventData >
      ...
    </StudyEventData>
  </SubjectData>
</ClinicalData>
...
```

Example: I am locating my site by UUID.

```
...
<ClinicalData >
  <SubjectData >
    <SiteRef LocationOID="fa63d085-629a-419e-a19d-21d6fb2bc93b"
      mdsol:LocationOIDType="SiteUUID" />
    <StudyEventData >
```

```

    ...
  </StudyEventData>
</SubjectData>
</ClinicalData>
...

```

i Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

```

Given I am a user
And there is a study
And the study has the following structure:
| form | is_primary | field | field_type | field_data_type | field_length |
field_control_type | field_order_number |
| AE | No | AEDESC | Standard | text | 20 |
Text | 1 |
And there is a subject in the study
And the subject has a UUID "c45f3bda-efd5-400f-bb72-b174dc0c93dd" not visible in
Rave
And the study has a unique identifier
And the study is associated with site UUID "efd5-400f-hj72-b174dc0c-g453bda-efd5"

```

Scenario Outline: Updating a subject in a site using the site number or unique identifier

When I post the following ODM ClinicalData to update the subject:

```

"""
<ClinicalData StudyOID="{unique study identifier}" mdsol:StudyOIDType="StudyUUID"
mdsol:StudyName="Mediflex(Dev)" MetaDataVersionOID="123">
  <SubjectData SubjectKey="c45f3bda-efd5-400f-bb72-b174dc0c93dd"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="AN 001"
TransactionType="Update">
    <SiteRef LocationOID=<location_oid>">
      mdsol:LocationOIDType=<location_oid_type>"<ignored_attributes>/>
        <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
          <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
            <ItemGroupData ItemGroupOID="AE" >
              <ItemData ItemOID="AEDESC" Value="NEW INSERTED TEST DATA"
TransactionType="Update"/>
            </ItemGroupData>
          </FormData>
        </StudyEventData>

```

```

    </SubjectData>
  </ClinicalData>
"""

```

Then I should receive a successful XML response
And I should see the following fields updated:

form	form_repeat	field	value
AE	1	AEDESC	NEW INSERTED TEST DATA

Examples:

location_oid	location_oid_type
ignored_attributes	
efd5-400f-hj72-b174dc0c-g453bda-efd5	SiteUUID
efd5-400f-hj72-b174dc0c-g453bda-efd5	SiteUUID
mdsol:SiteNumber="12345"	
efd5-400f-hj72-b174dc0c-g453bda-efd5	SiteUUID
mdsol:SiteNumber="12345" mdsol:StudySiteName="1234"	
{site oid}	SiteNumber
{site oid}	SiteNumber
mdsol:SiteUUID ="SiteUUID"	
{site oid}	SiteNumber
mdsol:SiteUUID ="SiteUUID" mdsol:StudySiteName="1234"	

Scenario Outline: Updating subject in a study using invalid site number or site Unique Identifier

When I post the following ODM ClinicalData to update the subject:

```

"""
<ClinicalData StudyOID="{unique study identifier}" mdsol:StudyOIDType="StudyUUID"
mdsol:StudyName="Mediflex(Dev)" MetaDataVersionOID="123">
  <SubjectData SubjectKey="c45f3bda-efd5-400f-bb72-b174dc0c93dd"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="AN 001"
TransactionType="Update">
    <SiteRef LocationOID=<location_oid>">
      mdsol:LocationOIDType=<location_oid_type>" mdsol:SiteNumber="12345"
      mdsol:StudySiteName="1234"/>
    <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
      <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE" >
          <ItemData ItemOID="AEDESC" Value="NEW INSERTED TEST DATA"
TransactionType="Insert"/>
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
"""

```

Then I should receive an HTTP response with status code "NotFound"
And reason code "RWS00017"
And the message "Site does not exist."

Examples:

	location_oid		location_oid_type
	invalid_site_uuid		SiteUUID
	invalid_site_number		SiteNumber

Scenario: Updating a subject with a valid study, site and subject uuids

When I post the following ODM ClinicalData to update the subject:

```
"""
<ClinicalData StudyOID="{unique study identifier}" mdsol:StudyOIDType="StudyUUID"
mdsol:StudyName="Mediflex(Dev)" MetaDataVersionOID="123">
    <SubjectData SubjectKey="c45f3bda-efd5-400f-bb72-b174dc0c93dd"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="AN 001"
TransactionType="Update">
        <SiteRef LocationOID="efd5-400f-hj72-b174dc0c-g453bda-efd5"
mdsol:LocationOIDType="SiteUUID" mdsol:SiteUUID ="efd5-400f-hj72-b174dc0c-g453bda-
efd5 " mdsol:StudySiteName="1234"/>
        <StudyEventData StudyEventID="SUBJECT" TransactionType="Update">
            <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
                <ItemGroupData ItemGroupOID="AE" >
                    <ItemData ItemOID="AEDESC" Value="NEW INSERTED TEST DATA"
TransactionType="Insert"/>
                </ItemGroupData>
            </FormData>
        </StudyEventData>
    </SubjectData>
</ClinicalData>
"""

Then I should receive a successful XML response
And I should see the following fields updated:
```

form form_repeat field value
AE 1 AEDESC NEW INSERTED TEST DATA



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.6 Locate a Study

Use Case

I am updating clinical data and ...

I am locating my study by its name,
or
I am locating my study by its unique identifier.

My ODM snippet

```
...
<ClinicalData
  StudyOID="{study-oid}"
  mdsol:StudyOIDType="{study-oid-type}" >
  <SubjectData >
    ...
  </SubjectData>
</ClinicalData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
StudyOID	The study identifier	Yes	Either the study UUID or name. See Specifying a study oid .
mdsol:StudyOIDType	The study identifier type	No	See below.

**Note:**

- When mdsol:StudyOIDType="StudyUUID" - The study identifier is the study UUID.
- When mdsol:StudyOIDType="Rave" - The study identifier is the study name (default).
- When mdsol:StudyOIDType="iMedidata" - The study identifier is the iMedidata study name.
- When mdsol:StudyOIDType is missing - The study identifier is the study name (default).

Examples

Example: I am locating my study by its name.

```
...
<ClinicalData StudyOID="Mediflex(Dev)" mdsol:StudyOIDType="Rave" >
  <SubjectData >
    ...
  </SubjectData>
</ClinicalData>
...
...
<ClinicalData StudyOID="Mediflex(Dev)" >
  <SubjectData >
    ...
  </SubjectData>
</ClinicalData>
...
```

Example: I am locating my study by its unique identifier.

```
...
<ClinicalData StudyOID="fa63d085-629a-419e-a19d-21d6fb2bc93b"
  mdsol:StudyOIDType="StudyUUID" >
  <SubjectData >
    ...
  </SubjectData>
</ClinicalData>
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

```
Given I am a user
And there is a study with unique study identifier
And the study has the following structure:
| folder | form | field | field_visible |
| VISIT | VISIT_INFO | VISIT_DATE | Yes |
And there is a subject in the study called "TheSubject"
```

Scenario Outline: Successful post using the study's unique identifier

When I post the following ODM:

```
"""
<ODM xmlns='http://www.cdisc.org/ns/odm/v1.3' xmlns:mdsol='http://
www.mdsol.com/ns/odm/metadata' ODMVersion='1.3' FileType='Transactional'
FileOID='Example-1' CreationDateTime='2008-01-01T00:00:00'>
  <ClinicalData StudyOID="{study_oid_type}" MetaDataVersionOID="MDV1.0" <ignored_attribute>>
    <SubjectData SubjectKey="TheSubject" TransactionType="Remove">
      <SiteRef LocationOID="{site number}"></SiteRef>
    </SubjectData>
  </ClinicalData>
</ODM>
"""
```

Then I should receive a successful XML response

And I should see that the subject "TheSubject" has been removed

Examples:

```
|           UUID           |   study_oid_type |
| ignored_attribute          |                   |
| {imediidata study identifier} |   imediidata
|                               |
| {study unique identifier}   |   StudyUUID       |
| mdsol:StudyName="Mediflex(Dev)" |
```

Scenario: Successful post with Rave study oid

When I post the following ODM:

```
"""
<ODM xmlns='http://www.cdisc.org/ns/odm/v1.3' xmlns:mdsol='http://
www.mdsol.com/ns/odm/metadata' ODMVersion='1.3' FileType='Transactional'
FileOID='Example-1' CreationDateTime='2008-01-01T00:00:00' >
  <ClinicalData StudyOID="{study oid}" MetaDataVersionOID="MDV1.0" >
    <SubjectData SubjectKey="TheSubject" TransactionType="Remove" >
```

```

    <SiteRef LocationOID="{site number}" />
  </SubjectData>
</ClinicalData>
</ODM>
"""

```

Then I should receive a successful XML response
 And I should see that the subject "TheSubject" has been removed

Scenario: Successful post with StudyOIDType value as Rave
 When I post the following ODM:

```

"""
<ODM xmlns='http://www.cdisc.org/ns/odm/v1.3' xmlns:mdsol='http://
www.mdsol.com/ns/odm/metadata' ODMVersion='1.3' FileType='Transactional'
FileOID='Example-1' CreationDateTime='2008-01-01T00:00:00'>
  <ClinicalData StudyOID="{study oid}" mdsol:StudyOIDType="Rave"
MetaDataVersionOID="MDV1.0">
    <SubjectData SubjectKey="TheSubject" TransactionType="Remove">
      <SiteRef LocationOID="{site number}"></SiteRef>
    </SubjectData>
  </ClinicalData>
</ODM>
"""

```

Then I should receive a successful XML response
 And I should see that the subject "TheSubject" has been removed

Scenario Outline: Successful post using the study's unique identifier and subject's unique identifier

Given the subject has a UUID "c45f3bda-efd5-400f-bb72-b174dc0c93dd" not visible in Rave

When I post the following ODM:

```

"""
<ODM xmlns='http://www.cdisc.org/ns/odm/v1.3' xmlns:mdsol='http://
www.mdsol.com/ns/odm/metadata' ODMVersion='1.3' FileType='Transactional'
FileOID='Example-1' CreationDateTime='2008-01-01T00:00:00'>
  <ClinicalData StudyOID=<study_uuid>" mdsol:StudyOIDType=<study_oid_type>">
    <MetaDataVersionOID="MDV1.0" <ignored_attribute>>
      <SubjectData SubjectKey="c45f3bda-efd5-400f-bb72-b174dc0c93dd"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="TheSubject"
TransactionType="Remove">
        <SiteRef LocationOID="{site number}"></SiteRef>
      </SubjectData>
    </ClinicalData>
</ODM>
"""

```

Then I should receive a successful XML response
 And I should see that the subject "TheSubject" has been removed

Examples:

study_uuid	study_oid_type
ignored_attribute	

```

| {imediadata study identifier} | iMedidata
|                               |
|   {study unique identifier} | StudyUUID           |
mdsol:StudyName="Mediflex(Dev)" |

Scenario Outline: Unique identifier does not exist
  When I post the following ODM:
  """
<ODM xmlns='http://www.cdisc.org/ns/odm/v1.3' xmlns:mdsol='http://
www.mdsol.com/ns/odm/metadata' ODMVersion='1.3' FileType='Transactional'
FileOID='Example-1' CreationDateTime='2008-01-01T00:00:00'>
  <ClinicalData StudyOID="" mdsol:StudyOIDType="">
    MetaDataVersionOID="MDV1.0" <ignored_attribute>
      <SubjectData SubjectKey="TheSubject" TransactionType="Remove">
        <SiteRef LocationOID="{site number}"></SiteRef>
      </SubjectData>
    </ClinicalData>
  </ODM>
"""
Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00014"
And the message "Study does not exist."

```

Examples:

```

|           study_uuid      | study_oid_type      |
| ignored_attribute          |                   |
|   {non existing study}   | iMedidata           |
|                           |                   |
|   {non existing study}   | StudyUUID           |
mdsol:StudyName="Mediflex(Dev)" |

```

Scenario Outline: User not authorised to view the study
 Given I am a user who is not authorised to view the study
 When I post the following ODM:
 """

```

<ODM xmlns='http://www.cdisc.org/ns/odm/v1.3' xmlns:mdsol='http://
www.mdsol.com/ns/odm/metadata' ODMVersion='1.3' FileType='Transactional'
FileOID='Example-1' CreationDateTime='2008-01-01T00:00:00'>
  <ClinicalData StudyOID="" mdsol:StudyOIDType="">
    MetaDataVersionOID="MDV1.0" <ignored_attribute>
      <SubjectData SubjectKey="TheSubject" TransactionType="Update">
        <SiteRef LocationOID="{site number}"></SiteRef>
      </SubjectData>
    </ClinicalData>
  </ODM>
"""

```

Then I should receive an HTTP response with status code "Not Found"
 And reason code "RWS00014"
 And the message "Study does not exist."

Examples:

```
|       study_uuid      |  study_oid_type |
| ignored_attribute    |                   |
| {imedata study identifier} | iMedidata
|
| {study unique identifier} | StudyUUID
|
mdsol:StudyName="Mediflex(Dev)" |
```

Scenario: Post with invalid mdsol:StudyOIDType value

When I post the following ODM:

```
"""
<ODM xmlns='http://www.cdisc.org/ns/odm/v1.3' xmlns:mdsol='http://
www.mdsol.com/ns/odm/metadata' ODMVersion='1.3' FileType='Transactional'
FileOID='Example-1' CreationDateTime='2008-01-01T00:00:00'>
    <ClinicalData StudyOID="{imedata study identifier}" MetaDataVersionOID="MDV1.0"
mdsol:StudyOIDType="InValidStudyIdentifier">
        <SubjectData SubjectKey="TheSubject" TransactionType="Remove">
            <SiteRef LocationOID="{site number}"></SiteRef>
        </SubjectData>
    </ClinicalData>
</ODM>
"""

```

Then I should receive an HTTP response with status code "BadRequest"

And reason code "RWS00056"

And the message "This is not a valid ODM 1.3 standard document. Syntactic Error:
The 'http://www.mdsol.com/ns/odm/metadata:StudyOIDType' attribute is invalid - The
value 'InValidStudyIdentifier' is invalid according to its datatype 'http://
www.mdsol.com/ns/odm/metadata:StudyOIDType' - The Enumeration constraint failed."



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.7 Locate a Subject

Use Case

I am updating clinical data and ...

I am locating my subject by its name
or
I am locating my subject by its unique identifier



Note: A subject's unique identifier is not related to iMedidata.

My ODM snippet

```
...
<ClinicalData >
  <SubjectData
    SubjectKey="{subject-key}"
    mdsol:SubjectKeyType="{subject-key-type}"
    mdsol:SubjectName="{subject-name}" >
    <SiteRef />
    ...
  </SubjectData>
</ClinicalData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
SubjectKey	The subject identifier	Yes	See below
mdsol:SubjectKeyType	The subject identifier type	Yes	See below
mdsol:SubjectName	The subject name	No	This is ignored

**Note:**

- When mdsol:SubjectKeyType="SubjectUUID" : The subject identifier is the UUID
- When mdsol:SubjectKeyType="SubjectName" : The subject identifier is the name (default)
- When mdsol:SubjectKeyType is missing : The subject identifier is the name (default)
- Every subject in Rave has a unique identifier, stored as [Subjects].[GUID]

Examples

Example: I am locating my subject by Name.

```
...
<ClinicalData >
  <SubjectData SubjectKey="ABAN8" mdsol:SubjectKeyType="SubjectName" >
    <SiteRef />
  ...
</SubjectData>
</ClinicalData>
...
```

```
...
<ClinicalData >
  <SubjectData SubjectKey="ABAN8" >
    <SiteRef />
  ...
</SubjectData>
</ClinicalData>
...
```

Example: I am locating my subject by UUID.

```
...
<ClinicalData >
  <SubjectData SubjectKey="fa63d085-629a-419e-a19d-21d6fb2bc93b"
mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="ABAN8" >
    <SiteRef />
  ...
</SubjectData>
```

```
</ClinicalData>
...

```

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

```
Given I am a user
And there is a study with the following structure:
| form | is_primary | field | field_type | field_data_type | field_length |
field_control_type | field_order_number |
| AE | No | AEDESC | Standard | text | 20 |
Text | 1 |
And there is a subject in the study
```

Scenario Outline: Update a subject's data using its unique identifier

Given the subject has a UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" not visible in Rave

When I post the following ODM ClinicalData to update the subject:

```
"""
<ClinicalData StudyOID="{study oid}" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="c45f3bda-efd5-400f-aa72-b174dc0c93dd" <subject data
attributes>
        TransactionType="Update">
            <SiteRef LocationOID="{site oid}" />
            <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
                <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
                    <ItemGroupData ItemGroupOID="AE" >
                        <ItemData ItemOID="AEDESC" Value="NEW INSERTED TEST DATA"
TransactionType="Insert"/>
                    </ItemGroupData>
                </FormData>
            </StudyEventData>
        </SubjectData>
    </ClinicalData>
"""

Then I should receive a successful XML response
And I should see the following fields updated:
```

form	form_repeat	field	value
AE	1	AEDESC	NEW INSERTED TEST DATA

Examples:

```
| subject data attributes
| mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="{subject name}" |
| mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="invalid" |
| mdsol:SubjectKeyType="SubjectUUID" |
```

Scenario Outline: Remove a subject using its unique identifier

Given the subject has a UUID "c45f3bda-efd5-400f-aa72-b174dc0c93dd" not visible in Rave

When I post the following ODM ClinicalData to update the subject:

```
"""
<ClinicalData StudyOID="{study oid}" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="c45f3bda-efd5-400f-aa72-b174dc0c93dd" <subject data
attributes>
        TransactionType="Remove">
            <SiteRef LocationOID="{site oid}"/>
        </SubjectData>
    </ClinicalData>
"""

```

Then I should receive a successful XML response

And the subject should be removed

Examples:

```
| subject data attributes
| mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="{subject name}" |
| mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="invalid" |
| mdsol:SubjectKeyType="SubjectUUID" |
```

Scenario Outline: Update a subject's data using its Name

When I post the following ODM ClinicalData to update the subject:

```
"""
<ClinicalData StudyOID="{study oid}" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="{subject name}" <subject data attributes>
        TransactionType="Update">
            <SiteRef LocationOID="{site oid}"/>
            <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
                <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
                    <ItemGroupData ItemGroupOID="AE" TransactionType="Update" >
                        <ItemData ItemOID="AEDESC" Value="NEW INSERTED TEST DATA"
        TransactionType="Insert"/>
                    </ItemGroupData>
                </FormData>
            </StudyEventData>
        </SubjectData>
    </ClinicalData>
"""

```

Then I should receive a successful XML response

And I should see the following fields updated:

form	form_repeat	field	value
AE	1	AEDESC	NEW INSERTED TEST DATA

Examples:

```
| subject data attributes
| mdsol:SubjectKeyType="SubjectName" mdsol:SubjectName="{subject name}" |
| mdsol:SubjectKeyType="SubjectName" |
```

Scenario: Remove a subject using its Name

When I post the following ODM ClinicalData to update the subject:

```
"""
<ClinicalData StudyOID="{study oid}" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="{subject name}" mdsol:SubjectKeyType="SubjectName"
mdsol:SubjectName="{subject name}" TransactionType="Remove">
        <SiteRef LocationOID="{site oid}"/>
    </SubjectData>
</ClinicalData>
"""

Then I should receive a successful XML response
And the subject should be removed
```

Scenario: Create a subject using a unique identifier will fail

When I post the following ODM ClinicalData:

```
"""
<ClinicalData StudyOID="{study oid}" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="80851ca2-8be5-4c7c-b48e-d99bc8f5f05d"
mdsol:SubjectKeyType="SubjectUUID"
    mdsol:SubjectName="ASUBJECT" TransactionType='Insert'>
        <SiteRef LocationOID='{site oid}'></SiteRef>
    </SubjectData>
</ClinicalData>
"""

Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00171"
And the message "Using the Subject UUID is not authorised for this action"
```

Scenario Outline: Update a subject's data using invalid unique identifier

When I post the following ODM ClinicalData to update the subject:

```
"""
<ClinicalData StudyOID="{study oid}" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="invalid UUID" <subject data attributes>
TransactionType="Update">
    <SiteRef LocationOID="{site oid}"/>
    <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
        <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
            <ItemGroupData ItemGroupOID="AE" >
                <ItemData ItemOID="AEDESC" Value="NEW INSERTED TEST DATA"/>
            </ItemGroupData>
        </FormData>
    </StudyEventData>
</SubjectData>
</ClinicalData>
```

```
"""
Then I should receive an HTTP response with status code "NotFound"
And reason code "RWS00023"
And the message "Subject does not exist."
```

Examples:

```
| subject data attributes
| mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="{valid subject name}"
| mdsol:SubjectKeyType="SubjectUUID"
```

Scenario Outline: Remove a subject using invalid unique identifier

When I post the following ODM ClinicalData to update the subject:

```
"""
<ClinicalData StudyOID="{study oid}" MetaDataVersionOID="study_version">
    <SubjectData SubjectKey="invalid UUID" <subject data attributes>
        TransactionType="Remove">
            <SiteRef LocationOID="{site oid}"/>
        </SubjectData>
    </ClinicalData>
```

```
"""
Then I should receive an HTTP response with status code "NotFound"
And reason code "RWS00023"
And the message "Subject does not exist."
```

Examples:

```
| subject data attributes
| mdsol:SubjectKeyType="SubjectUUID" mdsol:SubjectName="{subject name}"
| mdsol:SubjectKeyType="SubjectUUID"
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.8 Review a Field

Use Case

I am updating clinical data and ...

I am Reviewing or Un-reviewing a field

Assumptions

To succeed, I have satisfied this pre-requisite:

I belong to the Marking Group name the Review is open to.

My ODM snippet

```
...
<ItemData ItemOID="{field-oid}" Value="{data}" TransactionType="Update"  >
  <mdsol:Review
    Reviewed="{reviewed}"
    GroupName="{group-name}"/>
</ItemData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
Reviewed	"Yes" or "No"	Yes	
GroupName	The Review Group name	Yes	

Examples

To REVIEW:

```
...  
<mdsol:Review Reviewed="Yes" GroupName="Data Management"/>  
...
```

To UN-REVIEW:

```
...  
<mdsol:Review Reviewed="No" GroupName="Data Management"/>  
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.9 Update a Field

- [Updating a Data Dictionary Drop Down List Field](#)
- [Updating a Data Dictionary Search List Field](#)
- [Updating a Date Field](#)
- [Updating a Dynamic Search List Field](#)
- [Updating a Text Field](#)
- [Updating a Unit Dictionary Field](#)

1.6.10 Update a Data Dictionary Drop Down List Field

Use Case

I am updating clinical data and ...

I am selecting a value from a drop down list or specifying my own value.

My ODM snippet

```
...
<ItemData
  ItemOID="{field-oid}"
  Value="{list-item}"
  mdsol:SpecifyValue="{data}"
  TransactionType="{Insert|Update|Remove|Upsert}">
</ItemData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
ItemOID	The field OID	Yes	
Value	The list item identifier	Yes	
mdsol:SpecifyValue	User entered data	No	Included when the Value is the "other" option.

Examples

Example: Update with data selected from the dictionary list.

```
...
<ItemData
  ItemOID="RACE"
  Value="5"
  TransactionType="Update">
</ItemData>
...
```

Example: Update with data not in the list (list item 6 is the option for "other please specify").

```
...
<ItemData
  ItemOID="RACE"
  Value="6" mdsol:SpecifyValue="Oriental"
  TransactionType="Update">
</ItemData>
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.11 Update a Data Dictionary Search List Field

Use Case

I am updating clinical data and ...

I am selecting a value from a search list or specifying my own value

My ODM snippet

```
...
<ItemData
  ItemOID="{field-oid}"
```

```
Value="{list-item}"  
TransactionType="{Insert|Update|Remove|Upsert}">  
</ItemData>  
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
ItemOID	The field OID	Yes	
Value	The field value	Yes	Accepts user entered values or data dictionary coded values if selecting from the list

i **Note:** The difference between a drop down list and a search list is evident when specifying a value not found in the list.

Examples

Example: Update with data selected from the dictionary list.

```
...
<ItemData
  ItemOID="MEDICINES"
  Value="PAN1"
  TransactionType="Update">
</ItemData>
...
```

Example: Update with data not found in the list.

```
...
<ItemData
  ItemOID="MEDICINES"
  Value="ASPIRIN"
  TransactionType="Update">
</ItemData>
...
```

i **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

Given I am a Rave user

And there is a study

And the study has the following data dictionaries:

data_dict_oid	data_type	coded_value	decoded_value	specify
COUNTRIES	text	UK	UNITED KINGDOM	No

COUNTRIES text AUS AUSTRALIA No
COUNTRIES text OTH OTHER Yes

And the study has the following structure:

folder form field field_type field_data_type field_control_type
data_dictionary

SCREEN DM COUNTRY Standard text SearchList
--

COUNTRIES

And there is a subject in the study

Scenario: Post a field value from the dictionary list

When I update the subject with the following ODM:

"""

```
<StudyEventData StudyEventOID='SCREEN' TransactionType='Update'>
  <FormData FormOID='DM' FormRepeatKey='1' TransactionType='Update'>
    <ItemGroupData ItemGroupOID='DM' TransactionType='Update' >
      <ItemData ItemOID="COUNTRY" Value="UK" TransactionType="Update"/>
    </ItemGroupData>
  </FormData>
</StudyEventData>
"""
```

Then I should receive a successful XML response

And I should see the following fields updated:

folder form field value
SCREEN DM COUNTRY UK

Scenario Outline: Post a localized field value not from the dictionary list

When I <field_action_type> the field "COUNTRY" with the following itemdata ODM:

"""

```
<ItemData ItemOID="COUNTRY" Value="Mordor モルдор" TransactionType="<itemdata_tt>"/>
"""
```

Then I should receive a successful XML response

And I should see the following fields updated:

folder form field value
SCREEN DM COUNTRY MORDOR モルдор

Examples:

field_action_type itemdata_tt
update Update
insert Insert

Scenario Outline: Post a field value not from the dictionary list using TransactionType of context

Given the field "COUNTRY" has a value of "<item_data_value>"

When I post the following ODM to update the field "COUNTRY":

"""

```
<ItemData ItemOID="COUNTRY" Value="<item_data_value>" TransactionType="Context"/>
"""
```

Then I should receive a successful XML response

Examples:

item_data_value

MORDOR	

Scenario Outline: Post a non-dictionary field value using Specify dictionary entry
 When I <field_action_type> the field "COUNTRY" with the following itemdata ODM:
 """

```
<ItemData ItemOID="COUNTRY" Value="OTH" mdsol:SpecifyValue="">
  TransactionType="

```

Then I should receive a successful XML response
 And I should see the following fields updated:

folder	form	field	value
SCREEN	DM	COUNTRY	<specify_value>

Examples:

field_action_type	itemdata_tt	specify_value
update	Update	MORDOR
update	Update	MORDOR မြန်မာ
insert	Insert	MORDOR

Scenario Outline: Context search on an empty or non-empty non-dictionary field value
 Given the field "COUNTRY" has "OTH" specify value of "<field_value>"
 When I post the following ODM to update the field "COUNTRY":
 """

```
<ItemData ItemOID="COUNTRY" Value="OTH" mdsol:SpecifyValue="">
  TransactionType="Context"/>
```

Then I should receive a successful XML response

Examples:

field_value
MORDOR

Scenario Outline: Post a non-dictionary field value using empty or missing Specify dictionary entry

When I update the field "COUNTRY" with the following itemdata ODM:
 """

```
<ItemData ItemOID="COUNTRY" <item_data_value_attributes>
  TransactionType="Update"/>
```

Then I should receive a successful XML response

And I should see the following fields updated:

folder	form	field	value
SCREEN	DM	COUNTRY	<value>

Examples:

item_data_value_attributes	value

```
| Value="OTH" mdsol:SpecifyValue="" | |  
| Value="OTH" | |
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.12 Update a Date Field

Use Case

I am updating clinical data and ...

I am updating my field with "Date" typed data

My ODM snippet

```
...  
<ItemDataDate ItemOID="{field-oid}" TransactionType="Update">{data}</ItemDataDate>  
<ItemDataAny ItemOID="{field-oid}" TransactionType="Update">{data}</ItemDataAny>  
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
ItemOID	The field OID	Yes	

i **Note:** When an "ItemDataDate" element is used, the other non-date fields are represented as "ItemDataAny" elements.

Example

```
<ItemDataDate ItemOID="VISIT_DATE" TransactionType="Update">2009-01-30</ItemDataDate>
<ItemDataAny ItemOID="VISIT_OUTCOME" TransactionType="Update">PAIN KILLER GIVEN</ItemDataAny>
```

i **Note:** If "ItemDataDate" is used but the Rave field is not a date type, then the value will be entered as received.

i **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

Given I am a user
And there is a study
And the study has the following structure:

form	field	field_data_type	field_length	
field_control_type	field_order_number			
DateTime	VISIT_INFO	VISIT_DATE	date	20
	Text	1		
	VISIT_INFO	VISIT_OUTCOME	text	20
	Text	2		

And there is a subject in the study

Scenario: Successfully Insert clinical data for a subject using Date datatype
When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
```

```

<ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
    <ItemDataDate ItemOID="VISIT_DATE" TransactionType="Insert">2009-01-29</
ItemDataDate>
    <ItemDataAny ItemOID="VISIT_OUTCOME" TransactionType="Insert">PAIN KILLER
GIVEN</ItemDataAny>
</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the following fields updated:

form	form_repeat	field	value
VISIT_INFO	1	VISIT_DATE	29 JAN 2009
VISIT_INFO	1	VISIT_OUTCOME	PAIN KILLER GIVEN

Scenario: Successfully Update clinical data for a subject using Date datatype

Given the subject has the following data:

form	form_transaction	form_repeat	field	value
field_transaction				
VISIT_INFO	Update	1	VISIT_DATE	29 Jan 2009
Insert				
VISIT_INFO	Update	1	VISIT_OUTCOME	NOT KNOWN
Insert				

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
            <ItemDataDate ItemOID="VISIT_DATE" TransactionType="Update">2009-01-30</
ItemDataDate>
            <ItemDataAny ItemOID="VISIT_OUTCOME" TransactionType="Update">PAIN KILLER
GIVEN</ItemDataAny>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the following fields updated:

form	form_repeat	field	value
VISIT_INFO	1	VISIT_DATE	30 JAN 2009
VISIT_INFO	1	VISIT_OUTCOME	PAIN KILLER GIVEN

Scenario: Context on ItemDataDate

Given the subject has the following data:

form	form_transaction	form_repeat	field	value
field_transaction	field_order_number			
VISIT_INFO	Update	1	VISIT_DATE	29 Jan 2009
Insert				
VISIT_INFO	Update	1		

When I update the subject with the following ODM:

"""

```
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="VISIT_INFO" FormRepeatKey="1" TransactionType="Update">
    <ItemGroupData ItemGroupOID="VISIT_INFO" TransactionType="Update" >
      <ItemDataDate ItemOID="VISIT_DATE" TransactionType="Context">2009-01-29</
    ItemDataDate>
      <ItemDataAny ItemOID="VISIT_OUTCOME" TransactionType="Insert">PAIN KILLER
    GIVEN</ItemDataAny>
    </ItemGroupData>
  </FormData>
</StudyEventData>
"""
Then I should receive a successful XML response
And I should see the following fields updated:
| form | form_repeat | field | value |
| VISIT_INFO | 1 | VISIT_OUTCOME | PAIN KILLER GIVEN |
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.13 Update a Dynamic Search List Field

Use Case

I am updating clinical data and ...

I am selecting a value from a dynamic search list or entering my own value

My ODM snippet

```
...
<ItemData
  ItemOID="{field-oid}"
  Value="{list-item}"
  TransactionType="{ Insert|Update|Remove|Upsert }">
</ItemData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
ItemOID	The field OID	Yes	
Value	The field value	Yes	



Note: For dynamic search list: the field value should always be the user entered string, even when selecting from the list.

Examples

Example: Update with data selected from the dynamic search list.

```
...
<ItemData
  ItemOID="MEDICINES"
  Value="PANADOL"
  TransactionType="Update">
</ItemData>
...
```

Example: Update with data not in the dynamic search list.

```
...
<ItemData
  ItemOID="MEDICINES"
  Value="ASPIRIN"
  TransactionType="Update">
</ItemData>
...
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

```
Given I am a Rave user
And there is a study
And the study has a data dictionary "COUNTRIES" with the following entries:
| data_type | coded_value | decoded_value |
| text      | J           | JAPAN       |
```

text S SENEGL
text Z ZANZIBAR

And the study has the following structure:

folder form field field_type field_data_type field_control_type
SCREEN DM COUNTRY Standard text Dynamic SearchList

And the study sets up the Dynamic SearchList in the "DM" form for the "COUNTRY" field

And there is a subject in the study

Scenario Outline: Select a value from the drop down list

When I update the subject with the following ODM:

"""

```
<StudyEventData StudyEventOID='SCREEN' TransactionType='Update'>
  <FormData FormOID='DM' FormRepeatKey='1' TransactionType='Update'>
    <ItemGroupData ItemGroupOID='DM' TransactionType='Update' >
      <ItemData ItemOID='COUNTRY' Value='JAPAN'
TransactionType='<transaction_type>' />
    </ItemGroupData>
  </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the following fields updated:

folder form field value
SCREEN DM COUNTRY JAPAN

Examples:

transaction_type
Insert
Update

Scenario Outline: Enter a value not listed in the drop down

When I post the following ODM to update the field "COUNTRY":

"""

```
<ItemData ItemOID="COUNTRY" Value="" TransactionType="<transaction_type>"/>
"""

```

Then I should receive a successful XML response

And I should see the following fields updated:

folder form field value
SCREEN DM COUNTRY <value>

Examples:

value transaction_type
MORDOR Insert
モードル Update

Scenario: Select a value from the drop down list with a transaction type of Context

Given the field "COUNTRY" has a value of "ZANZIBAR"

When I post the following ODM to update the field "COUNTRY":

"""

```
<ItemData ItemOID="COUNTRY" Value="ZANZIBAR" TransactionType="Context"/>
"""

```

Then I should receive a successful XML response
And all the field values should be unchanged

Scenario: Enter a value not listed in the drop down with a transaction type of Context
Given the field "COUNTRY" has a value of "SHIRE"
When I post the following ODM to update the field "COUNTRY":

```
"""
<ItemData ItemOID="COUNTRY" Value="SHIRE" TransactionType="Context"/>
"""

```

Then I should receive a successful XML response
And all the field values should be unchanged



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.14 Update a Text Field

Use Case

I am updating clinical data and ...

I am updating my text field

My ODM snippet

```
...
<ItemData
  ItemOID="{field-oid}"
  Value="{data}"
  TransactionType="Update"  >
</ItemData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
ItemOID	The field OID	Yes	
Value	The field data	Yes	

Example

```
<ItemData
  ItemOID="DOB"
  Value="29 JAN 2011"
  TransactionType="Update">
</ItemData>
```

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

```
Given I am a user
And there is a study with the following structure:
| form | is_primary | field | field_data_type | field_length | field_control_type
| field_order_number | require_review | review_group |
| AE | No | AEDTC | date | 20 | DateTime
| 2 | Yes | | Data Management |
```

Scenario Outline: Successful update with positive pre-condition

```
Given I am assigned to the EDC Role "Power User"
And there is a subject in the study with the following data:
| form | form_transaction | field | value | field_transaction |
| AE | Update | AEDTC | 06 Jan 2010 | Update |
And the field "AEDTC" is "<pre_condition>"
And the role has the following Pre-conditions set for "Entry" permission:
| mask | status | pre_condition |
| true | true | <pre_condition> |
When I update the field "AEDTC"
Then I should receive a successful response
```

Examples:

pre_condition	
IsNonConformant	
IsReviewed	
RequiresAnswerQuery	

Scenario Outline: Successful update with negative pre-condition

```
Given I am assigned to the EDC Role "Power User"
And there is a subject in the study with the following data:
```

```

| form | form_transaction | field | value      | field_transaction |
| AE   | Update           | AEDTC | 06 Jan 2010 | Update          |
And the field "AEDTC" is not "<pre_condition>"  

And the role has the following Pre-conditions set for "Entry" permission:  

| mask | status | pre_condition |  

| true | false | <pre_condition> |  

When I update the field "AEDTC"  

Then I should receive a successful response

```

Examples:

```

| pre_condition |  

| IsNonConformant |  

| IsReviewed |  

| RequiresAnswerQuery |

```

Scenario: Unsuccessful update with positive pre-condition for IsNonConformant
Given I am assigned to the EDC Role "Power User"
And there is a subject in the study with the following data:
| form | form_transaction | field | value | field_transaction |
| AE | Update | AEDTC | 06 Jan 2010 | Update |
And the field "AEDTC" is not "IsNonConformant"
And the role has the following Pre-conditions set for "Entry" permission:
| mask | status | pre_condition |
| true | true | IsNonConformant |
When I update the field "AEDTC"
Then I should receive an HTTP response with status code "NotFound"
And reason code "RWS00042"
And the message "Field not authorised."

Scenario: Unsuccessful update with positive pre-condition for IsReviewed
Given I am assigned to the EDC Role "Power User"
And there is a subject in the study with the following data:
| form | form_transaction | field | value | field_transaction |
| AE | Update | AEDTC | 06 Jan 2010 | Update |
And the field "AEDTC" is not "IsReviewed"
And the role has the following Pre-conditions set for "Entry" permission:
| mask | status | pre_condition |
| true | true | IsReviewed |
When I update the field "AEDTC"
Then I should receive an HTTP response with status code "NotFound"
And reason code "RWS00042"
And the message "Field not authorised."

Scenario: Unsuccessful update with positive pre-condition for RequiresAnswerQuery
Given I am assigned to the EDC Role "Power User"
And there is a subject in the study with the following data:
| form | form_transaction | field | value | field_transaction |
| AE | Update | AEDTC | 06 Jan 2010 | Update |
And the field "AEDTC" is not "RequiresAnswerQuery"
And the role has the following Pre-conditions set for "Entry" permission:

```

| mask | status | pre_condition |
| true | true   | RequiresAnswerQuery |
When I update the field "AEDTC"
Then I should receive an HTTP response with status code "NotFound"
And reason code "RWS00042"
And the message "Field not authorised."

```

Scenario Outline: Unsuccessful update with negative pre-condition

```

Given I am assigned to the EDC Role "Power User"
And there is a subject in the study with the following data:
| form | form_transaction | field | value           | field_transaction |
| AE   | Update            | AEDTC | 06 Jan 2010 | Update          |
And the field "AEDTC" is "<pre_condition>"
And the role has the following Pre-conditions set for "Entry" permission:
| mask | status | pre_condition |
| true | false  | <pre_condition> |
When I update the field "AEDTC"
Then I should receive an HTTP response with status code "NotFound"
And reason code "RWS00042"
And the message "Field not authorised."

```

Examples:

```

| pre_condition      |
| IsNonConformant  |
| IsReviewed        |
| RequiresAnswerQuery |

```

Scenario Outline: Unsuccessful update with status only

```

Given I am assigned to the EDC Role "Power User"
And there is a subject in the study with the following data:
| form | form_transaction | field | value           | field_transaction |
| AE   | Update            | AEDTC | 06 Jan 2010 | Update          |
And the field "AEDTC" is "<pre_condition>" or the field "AEDTC" is not "<pre-
condition>"
And the role has the following Pre-conditions set for "Entry" permission:
| mask | status | pre_condition |
| false| true  | <pre_condition> |
When I update the field "AEDTC"
Then I should receive an HTTP response with status code "NotFound"
And reason code "RWS00042"
And the message "Field not authorised."

```

Examples:

```

| pre_condition      |
| IsNonConformant  |
| IsReviewed        |
| RequiresAnswerQuery |

```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.15 Update a Unit Dictionary Field

Use Case

I am updating clinical data and ...

I am updating my field and specifying the measurement unit

My ODM snippet

```
...
<ItemData ItemOID="{field-oid}" Value="{data}" TransactionType="Update" >
  <MeasurementUnitRef MeasurementUnitOID="{unit-oid}" />
</ItemData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
MeasurementUnitOID	The selected unit identifier	Yes	

Example

```
<ItemData ItemOID="VSORRESHT" Value="160" >
  <MeasurementUnitRef MeasurementUnitOID="1" />
</ItemData>
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.16 Update a Whole Item Group

Use Case

I am updating clinical data and ...

I am updating the whole standard form or logline



Note: This feature is used to update the standard form or logline to reflect the submitted ODM ItemGroupData. Pre-existing fields get updated if they are included in the ItemGroupData, or cleared (value set to an empty string) if they are not. Fields in the submitted ItemGroupData which do not exist will be inserted.

My ODM snippet

```
...
<ItemGroupData ItemGroupOID="{record-oid}" mdsol:Submission="WholeItemGroup" >
  <ItemData ItemOID="{field-oid}" Value="{data}" TransactionType="Insert"/>
  ...
</ItemGroupData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
mdsol:Submission	Specify "WholeItemGroup"	No	

Example

```
...
<FormData FormRepeatKey='1' FormOID='DEMOG' TransactionType='Update'>
  <ItemGroupData ItemGroupOID='DEMOG' mdsol:Submission='WholeItemGroup' >
    <ItemData ItemOID='FIELD1' Value='abc' TransactionType='Insert' />
    <ItemData ItemOID='FIELD2' Value='def' TransactionType='Insert' />
  </ItemGroupData>
</FormData>
...

```

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Scenario Outline: Successfully submit itemdata and touch all fields with
mdsol:Submission=WholeItemGroup

When I make a request to <ItemGroupData action> <specified itemdata elements> of
the form with WholeItemGroup submission

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see that all the fields in the form are submitted

Examples:

ItemGroupData action	specified itemdata elements
Insert	with no itemdata elements specified
Insert	with one itemdata elements specified
Insert	with few itemdata elements specified
Insert	with all itemdata elements specified
Update	with no itemdata elements specified
Update	with one itemdata elements specified
Update	with few itemdata elements specified
Update	with all itemdata elements specified

Scenario Outline: Successfully submit and touch only specified itemdata with
mdsol:Submission=SpecifiedItemsOnly

When I make a request to <ItemGroupData action> <specified itemdata elements> of
the form with SpecifiedItemsOnly submission

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see that only specified fields of the form are submitted

Examples:

ItemGroupData action specified itemdata elements		
Insert with no itemdata elements specified		
Insert with few itemdata elements specified		
Update with no itemdata elements specified		
Update with few itemdata elements specified		

Scenario Outline: Successfully submit and touch only specified itemdata with Context and Remove transactiontype

When I make a request to <ItemGroupData action> <specified itemdata elements> of the form with <submission value> submission

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see that only specified fields of the form are submitted

Examples:

ItemGroupData action specified itemdata elements submission value			
Context with no itemdata elements specified WholeItemGroup			
Remove with no itemdata elements specified WholeItemGroup			
Context with few itemdata elements specified WholeItemGroup			
Remove with few itemdata elements specified WholeItemGroup			
Context with no itemdata elements specified SpecifiedItemsOnly			
Context with few itemdata elements specified SpecifiedItemsOnly			
Remove with no itemdata elements specified SpecifiedItemsOnly			
Remove with few itemdata elements specified SpecifiedItemsOnly			

Scenario: All fields should be submitted when TransactionType attribute is not given at ItemGroup data level

When I post the following ODM:

```
"""
<ODM xmlns='http://www.cdisc.org/ns/odm/v1.3' xmlns:mdsol='http://
www.mdsol.com/ns/odm/metadata' CreationDateTime='2011-06-23T10:23:46'
ODMVersion='1.3' FileType='Transactional' FileOID='test'>
  <ClinicalData StudyOID='{study oid}' MetaDataVersionOID='study_version'>
    <SubjectData SubjectKey='ASUBJECT' TransactionType='Insert'>
      <SiteRef LocationOID='12345'></SiteRef>
      <StudyEventData StudyEventRepeatKey='1' StudyEventOID='SUBJECT'>
        <FormData FormRepeatKey='1' FormOID='STANDARD' TransactionType='Update'>
          <ItemGroupData ItemGroupOID='STANDARD' mdsol:Submission='WholeItemGroup' >
            <ItemData Value='SAMEVALUE' ItemOID='FIELD1' TransactionType='Insert'></
ItemData>
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
</ODM>
"""

Then I should receive a successful response
```

And the body of the response should be XML indicating success
 And I should see that all the fields in the form "STANDARD" are submitted

Scenario: All fields should be submitted when the TransactionType attribute is not given at ItemGroup data level for standard form with Form action of Insert

Given there is a study with the following structure:

form	is_primary	field	field_data_type	field_length	field_control_type
field_order_number					
1	DM	No	SEX	text	20
2	DM	No	DOB	date	20

And there is a subject in the study

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="DM" FormRepeatKey="2" TransactionType="Insert">
    <ItemGroupData ItemGroupOID="DM" mdsol:Submission='WholeItemGroup'>
      <ItemData ItemOID="SEX" Value="MALE" />
    </ItemGroupData>
  </FormData>
</StudyEventData>
""""
```

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see that all the fields in the form "DM" are submitted

Scenario: All fields should be submitted when the TransactionType attribute is not given at ItemGroup data level for log line

Given there is a study with the following structure:

form	is_primary	field	field_type	field_data_type	field_length	field_control_type	field_order_number
Text	AE	No	AEDESC	Log	text	20	1
Date	AE	No	AEDATE	Log	date	20	2

And there is a subject in the study with the following data:

form	form_transaction	form_repeat	logline_repeat	logline_transaction	field	value	field_transaction
AEDESC	AE	Update		1			Update
AEDESC	AE	Update		2			Insert
AEDATE	AEDESC	test data 1	Insert				
AEDATE	AEDESC	test data 2	Insert				
AEDATE	AEDATE	Update		2			Insert
AEDATE	AEDATE	29 JAN 2011	Insert				

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
    <ItemGroupData ItemGroupOID="AE" ItemGroupRepeatKey="1"
```

```

mdsol:Submission='WholeItemGroup'>
    <ItemData ItemOID="AEDESC" Value="test data 1 new" />
</ItemGroupData>
</FormData>
</StudyEventData>
"""

Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see that all the log fields at the record position "1" for form "AE"
should be submitted

```

Scenario: Reactivate log line

Given there is a study with the following structure:

form	is_primary	field	field_type	field_data_type	field_length	field_control_type	field_order_number
Text	No	AEDESC	Log	text	20		1
Date/Time	No	AEDATE	Log	date	20		2

And there is a subject in the study with the following data:

form	form_transaction	form_repeat	logline_repeat	logline_transaction	field	value	field_transaction
AEDESC	Update		1	Update	AE	test data 1	Insert
AEDESC	Update		2	Insert	AE	test data 2	Insert
AEDATE	Update		2	Insert	AE	29 JAN 2011	Insert
AEDESC	Update		1	Remove	AE	test data 1	Update

When I update the subject with the following ODM:

```

"""

<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE" ItemGroupRepeatKey="1"
TransactionType="Insert" mdsol:Submission='WholeItemGroup'>
            <ItemData ItemOID="AEDESC" Value="test data 1 new" />
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

Then I should receive a successful response

```

And the body of the response should be XML indicating success

And I should see that all the log fields at the record position "1" for form "AE" should be submitted

Scenario: Reactivate log line with inactive field

Given there is a study with the following structure:

form	is_primary	field	field_type	field_data_type	field_length	field_control_type	field_order_number
Text	No	AEDESC	Log	text	20		1

	AE	No	AEDESC	Log	text	20	
Text			1				
	AE	No	AEDATE	Log	date	20	
Datetime			2				

And there is a subject in the study with the following data:

	form	form_transaction	form_repeat	logline_repeat	logline_transaction		
field	value		field_transaction				
	AE	Update		1		Update	
AEDESC	test data 1		Insert				
	AE	Update		2		Insert	
AEDESC	test data 2		Insert				
	AE	Update		2		Insert	
AEDATE	29 JAN 2011		Insert				
	AE	Update		1		Remove	
AEDESC	test data 1		Remove				

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE" ItemGroupRepeatKey="1"
TransactionType="Insert" mdsol:Submission='WholeItemGroup'>
            </ItemGroupData>
        </FormData>
    </StudyEventData>
"""

```

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see that all the log fields at the record position "1" for form "AE" should be submitted and active

Scenario: Submit WholeItemGroup without any ItemData elements for log fields

Given there is a study with the following structure:

	form	is_primary	field	field_type	field_data_type	field_length		
				field_control_type	field_order_number			
Text				AEDESC	Log	text	20	
				1				
				AEDATE	Log	date	20	
Datetime				2				

And there is a subject in the study with the following data:

	form	form_transaction	form_repeat	logline_repeat	logline_transaction		
field	value		field_transaction				
	AE	Update		1		Update	
AEDESC	test data 1		Insert				
	AE	Update		1		Update	
AEDATE	29 JAN 2011		Insert				

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE" ItemGroupRepeatKey="2">
```

```

TransactionType="Insert" mdsol:Submission='WholeItemGroup'>
    </ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see that all the log fields at the record position "1" for form "AE"
should be submitted

Scenario: Submit WholeItemGroup without any ItemData elements for non log fields

Given there is a study with the following structure:

form	is_primary	field	field_data_type	field_length	field_control_type	field_order_number
DateTime	No	DOB	date	20	DM	1
Text	No	GENDER	text	20	DM	2

And there is a subject in the study
When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="DM" TransactionType="Insert">
mdsol:Submission='WholeItemGroup'>
    </ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see that all the fields in the form "DM" are submitted

Scenario: WholeItemGroup submission re-inserting an inactive field with specified
ItemData

Given there is a study with the following structure:

form	is_primary	field	field_data_type	field_length	field_control_type	field_order_number
1	No	SEX	text	20	Text	1
2	No	DOB	date	20	DateTime	2

And there is a subject in the study with the following data:

form	form_transaction	form_repeat	field	value	field_transaction
DM	Update		SEX	FEMALE	Insert
DM	Update		SEX	FEMALE	Remove

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">

```

```

<ItemGroupData ItemGroupOID="DM" TransactionType="Insert"
mdsol:Submission='WholeItemGroup'>
    <ItemData ItemOID="SEX" Value="MALE" TransactionType="Insert"/>
</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see that all the fields in the form "DM" are submitted

Scenario: WholeItemGroup submission with inactive field without any ItemData

Given there is a study with the following structure:

form	is_primary	field	field_data_type	field_length	field_control_type
field_order_number					
1	DM	No	SEX	text	20
2	DM	No	DOB	date	20
					DateTime

And there is a subject in the study with the following data:

form	form_transaction	form_repeat	field	value	field_transaction
DM	Update		SEX	FEMALE	Insert
DM	Update		SEX	FEMALE	Remove

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="DM" TransactionType="Insert"
mdsol:Submission='WholeItemGroup'>
            </ItemGroupData>
        </FormData>
    </StudyEventData>
"""

```

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see only active fields in the form "DM" submitted

Scenario: Log line WholeItemGroup submission with ItemGroupRepeatKey LAST

Given there is a study with the following structure:

form	is_primary	field	field_type	field_data_type	field_length	field_control_type	field_order_number
Text	AE	No	ADESC	Log	text	20	1
Date	AE	No	AEDATE	Log	date	20	2

And there is a subject in the study

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">

```

```
<ItemGroupData ItemGroupOID="AE" ItemGroupRepeatKey="LAST"
TransactionType="Insert" mdsol:Submission='WholeItemGroup'>
</ItemGroupData>
</FormData>
</StudyEventData>
"""

Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see that all the log fields at the record position "2" for form "AE"
should be submitted
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.17 Upsert a Form in RWS

Use Case

I am updating clinical data and ...

I am upserting my repeating form - located by its OID and a context value in one of its fields



Note: The upsert will insert a form with no matches; update a form for one match, and fail for multiple matches.

My ODM snippet

```
...
<StudyEventData    >
<FormData FormOID="{form-oid}" FormRepeatKey="@CONTEXT" TransactionType="Upsert"  >
  <ItemGroupData ItemGroupOID="LAB">
    <ItemData ItemOID="{field-oid}" Value="{match-data}" TransactionType="Context"/>
    <ItemData ItemOID="{field-oid}" Value="{data}"/>
  </ItemGroupData>
</FormData>
</StudyEventData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
FormOID	The form OID	Yes	
FormRepeatKey	use "@CONTEXT"	No	"@CONTEXT" indicates there is a corresponding field with TransactionType="Context"

Examples

Example:

```
...
<FormData FormOID="LAB" FormRepeatKey="@CONTEXT" TransactionType="Upsert">
  <ItemGroupData ItemGroupOID="LAB">
    <ItemData ItemOID="SAMPLEID" Value="123456" TransactionType="Context"/>
    <ItemData ItemOID="HEMACRIT" Value="78"/>
  </ItemGroupData>
</FormData>
...
```

Example: Updating or Inserting a Form Using TransactionType "Upsert"



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

Given I am a user
And there is a study with repeating forms

Scenario: Upsert an existing form (with one context field)

When I make a request to upsert(update) a repeating mixed form at subject level using context on standard items

Then I should receive a successful response
And the body of the response should be XML indicating success

Scenario Outline: Upsert a new form (with one context field)

When I make a request to upsert(insert) a repeating <form> at subject level using context on standard items

Then I should receive a successful response
And the body of the response should be XML indicating success

Examples:

form	
standard form	
mixed form	

Scenario: upsert with entry restriction on form

Given I do not have permission to enter data on the form

When I make a request to upsert an entry restricted repeating mixed form at subject level using context on standard items

Then I should receive an HTTP response with status code "NotFound"

And reason code "RWS00031"

And the message "Form not authorised"

Scenario: upsert with view restriction on form

Given I do not have permission to view the form

When I make a request to upsert a view restricted repeating mixed form at subject level using context on standard items

Then I should receive an HTTP response with status code "NotFound"

And reason code "RWS00031"

And the message "Form not authorised"

Scenario: upsert with entry restriction on field

Given I have entry restriction on form field

When I make a request to upsert a repeating standard form with field entry restriction at subject level using context

Then I should receive an HTTP response with status code "NotFound"

And reason code "RWS00042"

And the message "Field not authorised."

Scenario: upsert with view restriction on field

Given I have view restriction on form field

When I make a request to upsert a repeating standard form with field view restriction at subject level using context

Then I should receive an HTTP response with status code "NotFound"

And reason code "RWS00042"

And the message "Field not authorised."

Scenario: failed to insert due to subject admin permission not assigned

Given I have restriction on adding new forms

When I make a request to upsert(insert) a repeating standard form at subject level using context on standard items with no subject admin permission

Then I should receive an HTTP response with status code "NotFound"

And reason code "RWS00031"

And the message "Form not authorised"



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

[Back to top](#)

1.6.17.1 Example - Update or Insert a Form Using TransactionType Upsert

In some instances, for example, when updating lab data, you may not be aware if the form you want to update exists in Rave. Hence using FormRepeatKey="@CONTEXT" on its own would prove unsuccessful.

However, you can use TransactionType="Upsert" to instruct Rave to update a form if it exists, and to insert a new form if one does not already exist. Such a scenario is illustrated on this page. If Rave creates a new form, the data specified in the request is added to the new form.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
...  
  <FormData FormOID="LAB" FormRepeatKey="@CONTEXT" TransactionType="Upsert">  
    <ItemGroupData ItemGroupOID="LAB" TransactionType="Update" >  
      <ItemData ItemOID="SAMPLE_ID" Value="875964" TransactionType="Context"/>  
      <ItemData ItemOID="HEMATOCRIT" Value="48.1" TransactionType="Update"/>  
    </ItemGroupData>  
  </FormData>  
...  
</ODM>
```

[Back to top](#)

1.6.18 Upsert a Log Line

Use Case

I am updating clinical data and ...

I am upserting my log line - located by its OID and a context value in one of its fields



Note: The upsert will insert a log line for no matches; update a log line for one match, and fail for multiple matches.

My ODM snippet

```
...
<ItemGroupData ItemGroupOID="{record-oid}" ItemGroupRepeatKey="@CONTEXT"
TransactionType="Upsert" >
  <ItemData ItemOID="{field-oid}" Value="{match-data}" TransactionType="Context"/>
  <ItemData ItemOID="{field-oid}" Value="{data}"/>
</ItemGroupData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
FormOID	The form OID	Yes	
ItemGroupRepeatKey	use "@CONTEXT"	No	"@CONTEXT" indicates there is a corresponding field with TransactionType="Context"

Example

```
...
<FormData>
  <ItemGroupData ItemGroupOID="LOGLINE_LAB" ItemGroupRepeatKey="@CONTEXT"
TransactionType="Upsert">
    <ItemData ItemOID="SAMPLEID" Value="123456" TransactionType="Context"/>
    <ItemData ItemOID="HEMACRIT" Value="78"/>
  </ItemGroupData>
</FormData>
...

```

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

Given I am a user
 And there is a study with the following structure:

form	is_primary	field	field_type	field_data_type	field_length	field_control_type	field_order_number
Text	No	AEDESC	Log	text	20	AE	1
Date/Time	No	AEDATE	Log	date	20	AE	2

Scenario: TransactionType of Upsert at ItemGroupData with ItemGroupRepeatKey specified

Given there is a subject in the study with the following data:

form	form_transaction	logline_repeat	logline_transaction	field	value	field_transaction		
JAN 2011	Insert	1	Update	AEDESC	test	AE	Update	1
JUL 2011	Insert	2	Insert	AEDESC	test	AE	Update	2
				AEDATE	29			
				AEDATE	09			

 When I update the subject with the following ODM:
 """
 <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">

```

<FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
    <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="3"
TransactionType="Upsert" >
        <ItemData ItemOID="AEDESC" Value="NEW INSERTED TEST DATA"
TransactionType="Upsert"/>
    </ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response
And the response should have NewRecords with value "New Logline is added with
Repeat Key:[3] for FormOID:[AE]"

And I should see the following fields updated:

form	form_repeat	field	logline	value	
AE	1	AEDATE	3		
AE	1	AEDESC	3	NEW INSERTED TEST DATA	

Scenario: TransactionType of Upsert at ItemGroupData for 1st logline - Insert case

Given there is a subject in the study

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1"
TransactionType="Upsert" >
            <ItemData ItemOID="AEDATE" Value="25 OCT 2010"/>
            <ItemData ItemOID="AEDESC" Value="NEW INSERTED TEST DATA"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the following fields updated:

form	form_repeat	field	logline	value	
AE	1	AEDATE	1	25 OCT 2010	
AE	1	AEDESC	1	NEW INSERTED TEST DATA	

Scenario: TransactionType of Upsert at ItemGroupData for 1st log line - Update case

Given there is a subject in the study with the following data:

value	form	form_transaction	logline_repeat	logline_transaction	field	
data 1	AE	Update	1	Update	AEDESC	test
JAN 2011	AE	Insert			AEDATE	29
	AE	Update	1	Update		

And I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1"

```

```

TransactionType="Upsert" >
    <ItemData ItemOID="AEDESC" Value="NEW UPDATED TEST DATA" />
</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the following fields updated:

form	field	logline	value
AE	AEDESC	1	NEW UPDATED TEST DATA

Scenario: TransactionType of Upsert at ItemGroupData level for n'th log line Insert case

Given there is a subject in the study with the following data:

form	form_transaction	logline_repeat	logline_transaction	field	value	field_transaction
AE	Update	1		Update	AEDESC	test
data 1	Insert					
	AE	Update	1	Update	AEDATE	29
JAN 2011	Insert					
data 2	Update	2		Insert	AEDESC	test
	Insert					
JUL 2011	Update	2		Insert	AEDATE	09
	Insert					

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="3"
TransactionType="Upsert" >
            <ItemData ItemOID="AEDESC" Value="NEW UPDATED TEST DATA" />
</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And the response should have NewRecords with value "New Logline is added with Repeat Key:[3] for FormOID:[AE]"

And I should see the following fields updated:

form	field	logline	value
AE	AEDATE	3	
AE	AEDESC	3	NEW UPDATED TEST DATA

Scenario: TransactionType of Upsert at ItemGroupData level for nth log line Update case

Given there is a subject in the study with the following data:

form	form_transaction	logline_repeat	logline_transaction	field	value	field_transaction
AE	Update	1		Update	AEDESC	test
data 1	Insert					

AE Update	1	Update	AEDATE 29
JAN 2011 Insert			
AE Update	2	Insert	AEDESC test
data 2 Insert			
AE Update	2	Insert	AEDATE 09
JUL 2011 Insert			

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2"
TransactionType="Upsert" >
            <ItemData ItemOID="AEDESC" Value="NEW UPDATED TEST DATA" />
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""
```

Then I should receive a successful XML response

And I should see the following fields updated:

form field logline value	
AE AEDESC 2	NEW UPDATED TEST DATA

Scenario: TransactionType of Upsert at ItemGroupData with ItemGroupRepeatKey of @Context for Insert

Given there is a subject in the study with the following data:

value	field_transaction		
AE Update	1	Update	AEDESC test
data 1	Insert		
AE Update	1	Update	AEDATE 29
JAN 2011	Insert		
AE Update	2	Insert	AEDESC
existing context test data	Insert		
AE Update	2	Insert	AEDATE 09
JUL 2011	Insert		

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="@Context"
TransactionType="Upsert" >
            <ItemData ItemOID="AEDESC" Value="CONTEXT DATA NOT FOUND"
TransactionType="Context"/>
            <ItemData ItemOID="AEDATE" Value="31 JAN 2011"
TransactionType="Upsert"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""
```

Then I should receive a successful XML response

And the response should have NewRecords with value "New Logline is added with Repeat Key:[3] for FormOID:[AE]"

And I should see the following fields updated:

form	field	logline	value
AE	AEDATE	3	31 JAN 2011
AE	AEDESC	3	CONTEXT DATA NOT FOUND

Scenario: TransactionType of Upsert at ItemGroupData with ItemGroupRepeatKey of @Context for Update

Given there is a subject in the study with the following data:

form	form_transaction	logline_repeat	logline_transaction	field	value		field_transaction
data	1						
AE	Update	1		Update		AEDESC	test
JAN 2011		Insert				AEDATE	29
	AE	Update	1	Update		AEDESC	
			Insert				
	AE	Update	2	Insert		AEDESC	
	existing context test data	Insert					
JUL 2011		2		Insert		AEDATE	09
	AE	Update					
			Insert				

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="@Context"
TransactionType="Upsert" >
            <ItemData ItemOID="AEDESC" Value="EXISTING CONTEXT TEST DATA"
TransactionType="Context"/>
            <ItemData ItemOID="AEDATE" Value="31 JAN 2011"
TransactionType="Upsert"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
""""
```

Then I should receive a successful XML response

And I should see the following fields updated:

form	field	logline	value
AE	AEDATE	2	31 JAN 2011

Scenario: TransactionType of Upsert and ItemGroupRepeatKey of @CONTEXT at ItemGroupData without any ItemData context

Given there is a subject in the study with the following data:

form	form_transaction	logline_repeat	logline_transaction	field	value		field_transaction
data	1						
AE	Update	1		Update		AEDESC	test
JAN 2011		Insert				AEDATE	29
	AE	Update	1	Update		AEDESC	
			Insert				
	AE	Update	2	Insert		AEDESC	
	existing context test data	Insert					

	AE Update	2	Insert	AEDATE 09
JUL 2011		Insert		

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="@CONTEXT"
TransactionType="Upsert" >
            </ItemGroupData>
        </FormData>
    </StudyEventData>
"""

```

Then I should receive a successful XML response

And the response should have NewRecords with value "New Logline is added with
Repeat Key:[3] for FormOID:[AE]"

And I should see a new record of 3 inserted in the form "AE"

Scenario: TransactionType of Upsert and ItemGroupRepeatKey of @CONTEXT for a new log
form with one unmatched ItemData context

Given there is a subject in the study

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="@CONTEXT"
TransactionType="Upsert" >
            <ItemData ItemOID="AEDESC" Value="NEW CONTEXT TEST DATA"
TransactionType="Context"/>
            <ItemData ItemOID="AEDATE" Value="31 JAN 2011"
TransactionType="Upsert"/>
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful XML response

And I should see the following fields updated:

form	field	logline	value
AE	AEDATE	1	31 JAN 2011
AE	AEDESC	1	NEW CONTEXT TEST DATA

Scenario: TransactionType of Upsert and ItemGroupRepeatKey of @CONTEXT for a new
mixed form with more than one unmatched ItemData context

Given there is a study with the following structure:

form	is_primary	field	field_type	field_data_type	field_length	field_control_type	field_order_number
Text	No	AEDESC	Log	text	20		1
DateTime	No	AEDATE	Log	date	20		2
	No	AETERM	Standard	text	20		

Text | 3 |

And there is a subject in the study
When I update the subject with the following ODM:
"""

```
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
    <ItemGroupData ItemGroupOID="AE" TransactionType="Update" >
      <ItemData ItemOID="AETERM" Value="NEW AE TERM"
TransactionType="Update"/>
    </ItemGroupData>
    <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="@CONTEXT"
TransactionType="Upsert" >
      <ItemData ItemOID="AEDESC" Value="NEW CONTEXT TEST DATA 1"
TransactionType="Context"/>
      <ItemData ItemOID="AEDATE" Value="30 JAN 2011"
TransactionType="Upsert"/>
    </ItemGroupData>
    <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="@Context"
TransactionType="Upsert" >
      <ItemData ItemOID="AEDESC" Value="NEW CONTEXT TEST DATA 2"
TransactionType="Context"/>
      <ItemData ItemOID="AEDATE" Value="31 JAN 2011"
TransactionType="Upsert"/>
    </ItemGroupData>
  </FormData>
</StudyEventData>
"""
```

Then I should receive a successful XML response
And the response should have NewRecords with value "New Logline is added with Repeat Key:[2] for FormOID:[AE]"

And I should see a new record of 2 inserted in the form "AE"

And I should see the following fields updated:

form	field	logline	value
AE	AEDATE	0	
AE	AEDESC	0	
AE	AETERM	0	NEW AE TERM
AE	AEDATE	1	30 JAN 2011
AE	AEDESC	1	NEW CONTEXT TEST DATA 1
AE	AETERM	1	NEW AE TERM
AE	AEDATE	2	31 JAN 2011
AE	AEDESC	2	NEW CONTEXT TEST DATA 2
AE	AETERM	2	NEW AE TERM

Scenario: TransactionType of Upsert for a new log form with more than one unmatched ItemData context where the ItemGroupRepeatKey is specified fo the first one.

Given there is a subject in the study

When I update the subject with the following ODM:
"""

```
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="AE" FormRepeatKey="1" TransactionType="Update">
```

```
<ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="@Context"
TransactionType="Upsert" >
    <ItemData ItemOID="AEDESC" Value="NEW CONTEXT TEST DATA 1"
TransactionType="Context"/>
    <ItemData ItemOID="AEDATE" Value="31 JAN 2011"
TransactionType="Upsert"/>
</ItemGroupData>
<ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="2"
TransactionType="Upsert" >
    <ItemData ItemOID="AEDESC" Value="NEW CONTEXT TEST DATA 2"
TransactionType="Context"/>
    <ItemData ItemOID="AEDATE" Value="30 JAN 2011"
TransactionType="Upsert"/>
</ItemGroupData>

</FormData>
</StudyEventData>
"""
Then I should receive a successful XML response
And the response should have NewRecords with value "New Logline is added with
Repeat Key:[2] for FormOID:[AE]"
And I should see a new record of 2 inserted in the form "AE"
And I should see the following fields updated:
| form | field | logline | value |
| AE | AEDATE | 1 | 31 JAN 2011 |
| AE | AEDESC | 1 | NEW CONTEXT TEST DATA 1 |
| AE | AEDATE | 2 | 30 JAN 2011 |
| AE | AEDESC | 2 | NEW CONTEXT TEST DATA 2 |
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.19 Add Coding Decisions to Fields

Use Case

I am updating clinical data and ...

I am adding my coding decision to a field

My ODM snippet

```
...
<ItemData OID="{field-oid}" Value="{value}" TransactionType="Context">
  <mdsol:CodingData
    CodingDictionaryOID="{coding-dictionary-oid}"
    CodingDictionaryVersion="{coding-dictionary-version}"
    TransactionType="Insert">
    <mdsol:DictionaryLevelData
      DictionaryLevelOID="{dictionary-level-oid}"
      Value="{code}"
      Term="{term}"/>
    ...
  </mdsol:CodingData>
</ItemData>
...

```

mdsol:CodingData Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
mdsol:DictionaryLevelData	Dictionary Level Data	No	
CodingDictionaryOID	Coding Dictionary OID	Yes	
CodingDictionaryVersion	Coding Dictionary Version	Yes	
CoderUsername	Coder User name	No	
DateTimeStamp	Date-TimeStamp	No	
TransactionType	TransactionType	Yes	
CodingMode	Coding Mode	No	

Examples

Example:

```
<ItemData ItemOID="TERM" Value="HEADACHE123" TransactionType="Context">
  <mdsol:CodingData CodingDictionaryOID="MedDRA/E" CodingDictionaryVersion="V11.1">
    TransactionType="Insert">
    <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
      Value="10034567" Term="Endocrine System"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
      Value="10047635" Term="Vitamin related disorders"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Term"
      Value="10020914 "Term="Hypervitaminoses NEC"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="Preferred Term" Value="10020916"
      Term="Hypervitaminosis A"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="Low-Level Term" Value="10000033"
      Term="A-hypervitaminosis"/>
  </mdsol:CodingData>
</ItemData>
```

Note: The field data value must match that supplied using the “Context” transaction type.



Example: Coding a Data Point Using a Third-Party Coding System



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

Given I am a user
 And there is a coding dictionary "Test_Dictionary" with version "14.1" and following columns and terms:

column_name	Term	Code
System Organ Class	Nervous system disorders	1
High-Level Group Term	Headaches	2

And there is a study with the following structure:

form	field	field_data_type	field_length	field_order_number	field_control_type	coding_dictionary	coding_dictionary_version
------	-------	-----------------	--------------	--------------------	--------------------	-------------------	---------------------------

	AE	AE_DESC	text	500	1	
Text			Test_Dictionary	14.1		
	AE	AE_STARTDATE	date	500	2	
DateTime						
And there is a subject in the study with the following data:						
	form	form_transaction	form_repeat	field	value	
field_transaction						
	AE	Update	1	AE_DESC	CLUSTER HEADACHE	
Insert						
	AE	Update	1	AE_STARTDATE	29 JAN 2009	
Insert						

Scenario: Code a datapoint using terms already exist in the dictionary

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Insert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class" Value="1"
Term="Nervous system disorders"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""
```

Then I should receive a successful XML response

And I should see the coding applied to the following fields:

	form	form_repeat	field	column_name	term
code					
	AE	1	AE_DESC	System Organ Class	Nervous system disorders
1					
	AE	1	AE_DESC	High-Level Group Term	Headaches
2					

Scenario: Code a datapoint where the term does not exist in the Rave coding dictionary definition

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Insert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class" Value="1"
Term="Nervous system problems"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="3" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""
```

Then I should receive a successful XML response

And I should see the coding applied to the following fields:

	form	form_repeat	field	column_name	term
code					
1	AE 1		AE_DESC	System Organ Class	Nervous system problems
3	AE 1		AE_DESC	High-Level Group Term	Headaches

Scenario: Code a datapoint with identical dictionary level

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Insert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="1" Term="Nervous system disorders"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""
```

Then I should receive a successful XML response

And I should see the coding applied to the following fields:

	form	form_repeat	field	column_name	term
code					
1	AE 1		AE_DESC	High-Level Group Term	Nervous system disorders
2	AE 1		AE_DESC	High-Level Group Term	Headaches

Scenario: Update the coding for a datapoint

Given I code the field "AE_DESC" with following data:

	coding_dictionary	coding_dictionary_version	column_name	term	
			code		
system disorders	Test_Dictionary	14.1		System Organ Class	Nervous
Headaches	Test_Dictionary	14.1		High-Level Group Term	
		2			

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Update">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
Value="1" Term="Nervous system problems"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="3" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""
```

Then I should receive a successful XML response

And I should see the coding applied to the following fields:

form	form_repeat	field	column_name	term
code				
AE	1	AE_DESC	System Organ Class	Nervous system problems
1		AE_DESC	High-Level Group Term	Headaches
3				

Scenario: Code a datapoint using transaction type of Upsert on mdsol:CodingData

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Upsert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class" Value="1"
Term="Nervous system problems"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="3" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""
```

Then I should receive a successful XML response

And I should see the coding applied to the following fields:

form	form_repeat	field	column_name	term
code				
AE	1	AE_DESC	System Organ Class	Nervous system problems
1		AE_DESC	High-Level Group Term	Headaches
3				

Scenario: Update the coding for a datapoint using transaction type of Upsert on mdsol:CodingData

Given I code the field "AE_DESC" with following data:

coding_dictionary	coding_dictionary_version	column_name	
term		code	
Test_Dictionary	14.1	System Organ Class	Nervous
system disorders	1		
Test_Dictionary	14.1	High-Level Group Term	
Headaches	2		

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Upsert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
Value="1" Term="Nervous system problems"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="3" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
```

```
"""
Then I should receive a successful XML response
And I should see the coding applied to the following fields:
| form | form_repeat | field | column_name | term
| code |
| AE | 1 | AE_DESC | System Organ Class | Nervous system problems
| 1 |
| AE | 1 | AE_DESC | High-Level Group Term | Headaches
| 3 |

```

Scenario: Remove coding for a datapoint

Given I code the field "AE_DESC" with following data:

term	coding_dictionary	coding_dictionary_version	column_name	
			code	
system disorders	Test_Dictionary	14.1	System Organ Class	Nervous
Headaches	Test_Dictionary	14.1	High-Level Group Term	
		2		

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Remove">
    <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
Value="1" Term="Nervous system problems"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""


```

Then I should receive a successful XML response

And I should see no coding values for the field "AE_DESC"

Scenario: Do not accept more than one CodingData elements in the ItemData element

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Insert">
    <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class" Value="1"
Term="Nervous system problems"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Remove">
    <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
Value="1" Term="Nervous system problems"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>

```

```
    </mdsol:CodingData>
</ItemData>
"""

```

Then I should receive an HTTP response with status code "Conflict"
And reason code "RWS00110"
And the message "Can not accept more than one element for Coding Data"

Scenario: Coding can only be applied to the items with transaction type of Context
When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Update">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Insert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="Column does not exist"
Value="1" Term="Nervous system disorders"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""

```

Then I should receive an HTTP response with status code "Bad Request"
And reason code "RWS00166"
And the message "Coding can only be applied to the items with transaction type of Context"

Scenario: Field not enabled for coding

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_STARTDATE" Value="29 JAN 2009" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Insert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class" Value="1"
Term="Nervous system disorders"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""

```

Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00096"
And the message "Field not enabled for coding"

Scenario: User not authorised to do the coding

Given I am not authorised to do the coding

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Insert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"

```

```

Value="1" Term="Nervous system disorders"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""

```

Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00097"
And the message "User not authorised for coding"

Scenario: Coding dictionary not associated with the field

When I update the field with the following ODM:

```

"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="DictionaryNotAssociatedToField"
CodingDictionaryVersion="14.1" TransactionType="Insert">
    <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
Value="1" Term="Nervous system disorders"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""

```

Then I should receive an HTTP response with status code "Conflict"
And reason code "RWS00098"
And the message "Coding dictionary not associated with the field"

Scenario: Coding dictionary column does not exist

When I update the field with the following ODM:

```

"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Insert">
    <mdsol:DictionaryLevelData DictionaryLevelOID="Column does not exist"
Value="1" Term="Nervous system disorders"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""

```

Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00099"
And the message "Coding dictionary level/column does not exist"

Scenario: Cannot recode the datapoint using transaction type of insert

Given I code the field "AE_DESC" with following data:

coding_dictionary	coding_dictionary_version	column_name	
term	code		
Test_Dictionary	14.1	System Organ Class	Nervous
system disorders	1		

```

| Test_Dictionary | 14.1 | High-Level Group Term |
Headaches | 2 |
When I update the field with the following ODM:
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Insert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
Value="1" Term="Nervous system problems"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""

Then I should receive an HTTP response with status code "Conflict"
And reason code "RWS00114"
And the message "Coding value already exists"

Scenario: Transaction Type of Context is not supported on mdsol:CodingData
Given I code the field "AE_DESC" with following data:
| coding_dictionary | coding_dictionary_version | column_name |
term | code |
| Test_Dictionary | 14.1 | System Organ Class | Nervous
system disorders | 1 |
| Test_Dictionary | 14.1 | High-Level Group Term |
Headaches | 2 |
When I update the field with the following ODM:
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Context">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
Value="1" Term="Nervous system disorders"/>
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="2" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""

Then I should receive an HTTP response with status code "Conflict"
And reason code "RWS00107"
And the message "Transaction type not supported for this element"

Scenario: Update transaction on CodingData element will apply the coding to the field
if it is not applied
When I update the field with the following ODM:
"""
<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Update">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class">

```

```
Value="1" Term="Nervous system problems"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="3" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""

Then I should receive a successful XML response
And I should see the coding applied to the following fields:
| form | form_repeat | field      | column_name           | term
| code |
|   | AE      | 1          | AE_DESC   | System Organ Class   | Nervous system problems
| 1  |
|   | AE      | 1          | AE_DESC   | High-Level Group Term | Headaches
| 3  |

Scenario: Remove transaction on mdsol:CodingData does not throw an error if the
coding is not applied
When I update the field with the following ODM:
"""

<ItemData ItemOID="AE_DESC" Value="CLUSTER HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="Test_Dictionary"
CodingDictionaryVersion="14.1" TransactionType="Remove">
    <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
Value="1" Term="Nervous system problems"/>
    <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
Value="3" Term="Headaches"/>
    </mdsol:CodingData>
</ItemData>
"""

Then I should receive a successful XML response
And I should see no coding values for the field "AE_DESC"
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.19.1 Example - Code a Data Point Using a Third-Party Coding System

It is possible to code a data point with values from a third-party coding system, using the vendor extension elements mdsol:CodingData and mdsol:DictionaryLevelData to map the required terms. For example, to create a new coding term, use the ODM format shown below.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
...  
    <ItemData ItemOID="TESTMEDDRA" Value="HEADACHE123" TransactionType="Context">  
        <mdsol:CodingData CodingDictionaryOID="MedDRA/E"  
CodingDictionaryVersion="V11.1"  
            TransactionType="Insert">  
                <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"  
Value="10034567" Term="Endocrine System"/>  
                <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"  
Value="10047635" Term="Vitamin related disorders"/>  
                <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Term"  
Value="10020914" Term="Hypervitaminoses NEC"/>  
                <mdsol:DictionaryLevelData DictionaryLevelOID="Preferred Term"  
Value="10020916"  
                    Term="Hypervitaminosis A"/>  
                    <mdsol:DictionaryLevelData DictionaryLevelOID="Low-Level Term"  
Value="10000033"  
                    Term="A-hypervitaminosis"/>  
            </mdsol:CodingData>  
    </ItemData>  
...  
</ODM>
```

[Back to top](#)

1.6.20 Add External Audits to Fields

Use Case

I am updating clinical data and ...

I am adding my external audits to a field



Note: The audit order is the original Rave audit before any external audits.

My ODM snippet

```
...
<ItemData OID="{field-oid}" Value="{value}" TransactionType="Context">
    <AuditRecord EditPoint="DataManagement">
        <UserRef UserOID="{user}" />
        <LocationRef LocationOID="{location-oid}" />
        <DateTimeStamp>{time-stamp}</DateTimeStamp>
        <ReasonForChange>{reason}</ReasonForChange>
    </AuditRecord>
</ItemData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
UserOID	audited user's identifier	Yes	
LocationOID	Location OID	Yes	The LocationOID is case sensitive.
IncludeFileOID	Provide the option of include ODM file OID in the default audit text	No	The value is 'Yes' or 'No'.

Examples

Example:

```
...
<ItemData ItemOID="RACE" Value="1" TransactionType="Update">
  <AuditRecord EditPoint="DataManagement" mdsol:IncludeFileOID="Yes">
    <UserRef UserID="noadminauth" />
    <LocationRef LocationOID="RWS" />
    <DateTimeStamp>2009-02-04T14:10:32-05:00</DateTimeStamp>
    <ReasonForChange>Datapoint manually edited 1</ReasonForChange>
  </AuditRecord>
</ItemData>
...
...
```

Example: Adding an External Audit Record



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Scenario: Post an external Audit to an existing field should succeed

Given there is a study with the following structure:

And there is a subject in the study with the following data:

When I update the subject with the following ODM:

"""

```
<StudyEventData StudyEventOID="SCREEN" TransactionType="Update"
StudyEventRepeatKey="1">
  <FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">
    <ItemGroupData ItemGroupOID="DM" TransactionType="Update" >
      <ItemData ItemOID="BIRTHDATE" Value="12 Jan 2001" TransactionType="Update">
        <AuditRecord EditPoint="DataManagement">
          <UserRef UserID="noadminauth" />
          <LocationRef LocationOID="RWS" />
          <DateTimeStamp>2009-02-04T14:10:32-05:00</DateTimeStamp>
          <ReasonForChange>Datapoint manually edited 1</ReasonForChange>
```

```

        </AuditRecord>
    </ItemData>
</ItemGroupData>
</FormData>
</StudyEventData>
"""

Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see the audit entries against the following fields:
| form | field | value |
audit
|
| DM | BRTHDT | 12 JAN 2001 | External Audit Record. Reason for change:
'Datapoint manually edited 1', Location OID: 'RWS', Time:
'2009-02-04T14:10:32-05:00', User OID: 'noadminauth' |

Scenario: Post an external Audit to a new field should succeed
Given there is a study with the following structure:
| folder | form | field | field_data_type | field_control_type |
| SCREEN | DM | BRTHDT | date | DateTime |
And there is a subject in the study
When I update the subject with the following ODM:
"""

<StudyEventData StudyEventOID="SCREEN" TransactionType="Update"
StudyEventRepeatKey="1">
<FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">
<ItemGroupData ItemGroupOID="DM" TransactionType="Update" >
<ItemData ItemOID="BRTHDT" Value="12 Jan 2001" TransactionType="Insert">
<AuditRecord EditPoint="DataManagement">
<UserRef UserOID="noadminauth" />
<LocationRef LocationOID="RWS" />
<DateTimeStamp>2009-02-04T14:10:32-05:00</DateTimeStamp>
<ReasonForChange>Datapoint manually edited 1</ReasonForChange>
</AuditRecord>
</ItemData>
</ItemGroupData>
</FormData>
</StudyEventData>
"""

Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see the audit entries against the following fields:
| form | field | value |
audit
|
| DM | BRTHDT | 12 JAN 2001 | External Audit Record. Reason for change:
'Datapoint manually edited 1', Location OID: 'RWS', Time:
'2009-02-04T14:10:32-05:00', User OID: 'noadminauth' |

Scenario: Post an external Audit whilst updating an existing field should succeed

```

Given there is a study with the following structure:

folder	form	field	field_data_type	field_control_type
SCREEN	DM	BRTHDTC	date	DateTime

And there is a subject in the study with the following data:

folder	form	field	value
SCREEN	DM	BRTHDTC	12 May 1990

When I update the subject with the following ODM:

"""

```
<StudyEventData StudyEventOID="SCREEN" TransactionType="Update"
StudyEventRepeatKey="1">
    <FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="DM" TransactionType="Update" >
            <ItemData ItemOID="BRTHDTC" Value="10 May 1990" TransactionType="Update">
                <AuditRecord EditPoint="DataManagement">
                    <UserRef UserOID="noadminauth" />
                    <LocationRef LocationOID="RWS" />
                    <TimeStamp>2012-02-04T14:10:32-05:00</TimeStamp>
                    <ReasonForChange>Datapoint manually edited 1</ReasonForChange>
                </AuditRecord>
            </ItemData>
        </ItemGroupData>
    </FormData>
</StudyEventData>
```

"""

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see the audit entries in the correct order

And I should see the audit entries against the following fields:

form	field	value	
audit			

"""

```
| DM | BRTHDTC | 10 MAY 1990 | External Audit Record. Reason for change:
'Datapoint manually edited 1', Location OID: 'RWS', Time:
'2012-02-04T14:10:32-05:00', User OID: 'noadminauth' |
```

Scenario: Post an external Audit whilst removing an existing field should succeed

Given there is a study with the following structure:

folder	form	field	field_data_type	field_control_type
SCREEN	DM	BRTHDTC	date	DateTime

And there is a subject in the study with the following data:

folder	form	field	value
SCREEN	DM	BRTHDTC	10 May 1990

When I update the subject with the following ODM:

"""

```
<StudyEventData StudyEventOID="SCREEN" TransactionType="Update"
StudyEventRepeatKey="1">
    <FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">
        <ItemGroupData ItemGroupOID="DM" TransactionType="Update" >
            <ItemData ItemOID="BRTHDTC" Value="10 May 1990" TransactionType="Remove">
                <AuditRecord EditPoint="DataManagement">
```

```
<UserRef UserOID="noadminauth" />
<LocationRef LocationOID="RWS" />
<DateTimeStamp>2012-02-04T14:10:32-05:00</DateTimeStamp>
<ReasonForChange>Datapoint manually edited 1</ReasonForChange>
</AuditRecord>
</ItemData>
</ItemGroupData>
</FormData>
</StudyEventData>
"""
Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see the audit entries in the correct order
And I should see the audit entries against the following fields:
```

form	field	value
audit		
DM	BRTHDT	10 MAY 1990 External Audit Record. Reason for change: 'Datapoint manually edited 1', Location OID: 'RWS', Time: '2012-02-04T14:10:32-05:00', User OID: 'noadminauth'



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.20.1 Example - Add an External Audit Record

You can add a reference to an external audit record to a data point in Rave by including the AuditRecord element in an ODM request.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?
PostODMClinicalData
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>
  ...
  <ItemData ItemOID="RACE" Value="1" TransactionType="Update">
    <AuditRecord EditPoint="DataManagement" mdsol:IncludeFileOID="Yes">
      <UserRef UserID="noadminauth" />
      <LocationRef LocationOID="RWS" />
      <DateTimeStamp>2009-02-04T14:10:32-05:00</DateTimeStamp>
      <ReasonForChange>Datapoint manually edited 1</ReasonForChange>
    </AuditRecord>
  </ItemData>
  ...
</ODM>
```

[Back to top](#)

1.6.21 Use Field Queries

Use Case

I am updating clinical data and ...

I am opening, forwarding, answering, cancelling, or closing my query.

 **Note:**

- All manual queries require a manual close. For more information, see [Closing Manual and System Queries](#).
- From Rave Core 2025.2.0 onwards, you can use a new ItemData optional attribute, mdsol:SubmitMarkingOnly.
 - mdsol:SubmitMarkingOnly="Yes" allows you to open a query without submitting any data.

- If mdsol:SubmitMarkingOnly="No" or if you omit it, the functionality remains the same as before.

Assumptions

To succeed, I have satisfied these pre-requisites:

1. I belong to the Marking Group name the Query is open to.
2. I am able to manage queries.
3. I know the current field data value.

My ODM snippet

```
...
<ItemData ItemOID="{field-oid}" Value="{data}" TransactionType="Context"
mdsol:SubmitMarkingOnly="Yes">
  <mdsol:Query
    QueryRepeatKey="{query id}"
    Response="{response}"
    Recipient="{recipient}"
    Value="{value}"
    Status="{status}"/>
</ItemData>
...
```

ODM Attributes

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
Recipient	Recipient Marking Group name	No	
Value	The query text	No	
Status	Status	Yes	Required value: Open / Answered / Forwarded / Canceled / Closed.
QueryRepeatKey	The Database ID of the Query record	No	
RequiresResponse	RequiresResponse	No	Required value: yes / no
Response	The response data	No	
PrecedingQueryRepeatKey	The Database ID of the preceding Query record	No	
CoderUsername	CoderUsername	No	Coding specific
CodingDictionaryVersion	CodingDictionaryVersion	No	Coding specific
RequireManualClose	RequireManualClose	No	<ul style="list-style-type: none"> • Required value: yes / no • See Open Query for more details.

Examples

To OPEN:

```
...  
<mdsol:Query Recipient="Site from CRA" Value="Analyse the race value" Status="Open"/>  
...
```

To ANSWER:

```
...  
<mdsol:Query Recipient="Site from CRA" QueryRepeatKey="123" Response="None  
identified" Status="Answered"/>  
...
```

To FORWARD:

```
...  
<mdsol:Query Recipient="Site from DM" QueryRepeatKey="123" Status="Forwarded"/>  
...
```

To RE-QUERY:

```
...  
<mdsol:Query Value="Re-Analyse the race value" PrecedingQueryRepeatKey="123"  
Status="Open"/>  
...
```

To CANCEL:

```
...  
<mdsol:Query Recipient="Site from CRA" QueryRepeatKey="123" Status="Cancelled"/>  
...
```

To CLOSE:

```
...  
<mdsol:Query Recipient="Site from CRA" QueryRepeatKey="123" Status="Closed"/>  
...
```



Note: The field data value must match that supplied using the "Context" transaction type.

Example: Opening a Rave Query Using TransactionType="Context"



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Open Query

Background:

Given I am a user associated with the following marking groups:

number	name
1	Site from System
2	Site from CRA
3	Site from DM
4	Site from Site

Scenario Outline: Open query against a field with valid TransactionType

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="
```

```
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value" Status="Open"/>
```

```
</ItemData>
```

"""

Then I should receive a successful XML response

And the response should have the following query elements:

"""

```
<Query ID="{query id}" FieldOID="AE_DESC" />
```

"""

And I should see the query opened against the following fields:

form	field	value	query_id	query	marking_group
	AE	<field value>	{query id}	Analyse the race value	Site from CRA

Examples:

transaction type	field value

Context	HEADACHE	
Update	MIGRAINE	

Scenario Outline: Open query with valid RequiresResponse attribute

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" RequiresResponse="

```

"""

Then I should receive a successful XML response

And the response should have the following query elements:

"""

```
<Query ID="{query id}" FieldOID="AE_DESC" />
```

"""

And I should see the query opened against the following fields:

form	field	value	query_id	query	marking_group
------	-------	-------	----------	-------	---------------

AE	AE_DESC	HEADACHE	{query id}	Analyse the race value	Site from CRA
----	---------	----------	------------	------------------------	---------------

And I should see the following query updated in Rave:

query_id	query_status	require response
----------	--------------	------------------

{query id}	Open	Yes
------------	------	-----

Examples:

option	value	
Yes	True	
No	False	

Scenario: Opening a Query against a frozen field should succeed

Given the background

But the field "AE_DESC" status is "Frozen"

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value" Status="Open"/>
</ItemData>
```

"""

Then I should receive a successful XML response

And the response should have the following query elements:

"""

```
<Query ID="{query id}" FieldOID="AE_DESC" />
```

"""

And I should see the query opened against the following fields:

form	field	value	query_id	query	marking_group
------	-------	-------	----------	-------	---------------

AE	AE_DESC	HEADACHE	{query id}	Analyse the race value	Site from CRA
----	---------	----------	------------	------------------------	---------------

Scenario: Open a Query and locking a field in one transaction should succeed
Given the background

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" mdsol:Lock="Yes"
TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value"
Status="Open"/>
</ItemData>
```

"""

Then I should receive a successful XML response

And the response should have the following query elements:

"""

```
<Query ID="{query id}" FieldOID="AE_DESC" />
```

"""

And I should see the query opened against the following fields:

form	field	value	query_id	query	marking_group
AE	AE_DESC	HEADACHE	{query id}	Analyse the race value	Site from CRA

Scenario: Multiple queries can be opened on a single field

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value 1"
Status="Open"/>
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value 2"
Status="Open"/>
</ItemData>
```

"""

Then I should receive a successful XML response

And the response should have the following query elements:

"""

```
<Query ID="{query id 1}" FieldOID="AE_DESC" />
```

```
<Query ID="{query id 2}" FieldOID="AE_DESC" />
```

"""

And I should see the query opened against the following fields:

form	field	value	query_id	query	marking_group
AE	AE_DESC	HEADACHE	{query id1}	Analyse the race value 1	Site from CRA
AE	AE_DESC	HEADACHE	{query id2}	Analyse the race value 2	Site from CRA

Scenario: Open queries on two separate fields in a single request

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
```

```

<mdsol:Query Recipient="Site from CRA" Value="Analyse the race value"
Status="Open"/>
</ItemData>
<ItemData ItemOID="AE_STARTDATE" Value="29 JAN 2009" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value"
Status="Open"/>
</ItemData>
"""
Then I should receive a successful XML response
And the response should have the following query elements:
"""
<Query ID="{query id 1}" FieldOID="AE_DESC" />
<Query ID="{query id 2}" FieldOID="AE_STARTDATE" />
"""
And I should see the query opened against the following fields:
| form | field           | value        | query_id      | query          |
marking_group |
| AE   | AE_DESC          | HEADACHE    | {query id1}  | Analyse the race value | Site
from CRA |
| AE   | AE_STARTDATE     | 29 JAN 2009 | {query id2}  | Analyse the race value | Site
from CRA |

Scenario: Open multiple queries on two separate fields in a single request
When I update the field with the following ODM:
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value 1"
Status="Open"/>
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value 2"
Status="Open"/>
</ItemData>
<ItemData ItemOID="AE_STARTDATE" Value="29 JAN 2009" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value 3"
Status="Open"/>
</ItemData>
"""

Then I should receive a successful XML response
And the response should have the following query elements:
"""
<Query ID="{query id 1}" FieldOID="AE_DESC" />
<Query ID="{query id 2}" FieldOID="AE_DESC" />
<Query ID="{query id 3}" FieldOID="AE_STARTDATE" />
"""
And I should see the query opened against the following fields:
| form | field           | value        | query_id      | query          |
marking_group |
| AE   | AE_DESC          | HEADACHE    | {query id1}  | Analyse the race value 1 | Site
from CRA |
| AE   | AE_DESC          | HEADACHE    | {query id2}  | Analyse the race value 2 | Site
from CRA |

```

```
| AE      | AE_STARTDATE | 29 JAN 2009 | {query id3} | Analyse the race value 3 |
Site from CRA |
```

Scenario: Failure when opening a query without a valid role permission

Given the background

But I am not authorised to open a query

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value"
Status="Open"/>
</ItemData>
```

"""

Then I should receive an HTTP response with status code "Forbidden"

And reason code "RWS00068"

And the message "Open query not authorized"

Scenario: Failure when opening a query when Pre-conditions are not met

Given the background

But the role has the following Pre-conditions set for "OpenQuery" permission:

mask	status	pre_condition
true	true	IsNonConformant

And the field "AE_STARTDATE" is not "NonConformant"

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_STARTDATE" Value="29 Jan 2009" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value"
Status="Open"/>
</ItemData>
```

"""

Then I should receive an HTTP response with status code "Forbidden"

And reason code "RWS00068"

And the message "Open query not authorized"

Scenario Outline: Open query supports localized strings in query text

When I update the field with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="{query text}" Status="Open"/>
</ItemData>
```

"""

Then I should receive a successful XML response

And the response should have the following query elements:

"""

```
<Query ID="{query id}" FieldOID="AE_DESC" />
```

"""

And I should see the query opened against the following fields:

form	field	value	query_id	query	marking_group
AE	AE_DESC	HEADACHE	{query id}	<query text>	Site from CRA

Examples:

```
| query text
| ┌─────────┐
| |         |
| |         |
| └─────────┘
```

Scenario: Failure when opening a query against a locked field

Given the background

But the field "AE_DESC" status is "Locked"

When I update the field with the following ODM:

```
"""

```

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value"
Status="Open"/>
</ItemData>
```

```
"""

```

Then I should receive an HTTP response with status code "Conflict"

And reason code "RWS00044"

And the message "Field locked."

Scenario Outline: Failure when opening a query with invalid TransactionType

When I update the field with the following ODM:

```
"""

```

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="<type>">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value"
Status="Open"/>
</ItemData>
```

```
"""

```

Then I should receive an HTTP response with status code "<response code>"

And reason code "<reason code>"

And the response should have the message beginning with "<response_message>"

Examples:

type	response code	reason code	
response_message			
Delete	Bad Request	RWS00056	This is not a valid ODM 1.3 standard document
Upsert	Conflict	RWS00107	Transaction type not supported for this element

Scenario Outline: Failure when opening a query with invalid Status

When I update the field with the following ODM:

```
"""

```

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Update">
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value"
Status="<query_status>"/>
</ItemData>
```

```
"""

```

Then I should receive an HTTP response with status code "Bad Request"

And reason code "RWS00056"

And the response should have the message beginning with "This is not a valid ODM

1.3 standard document"

Examples:

query_status
Opened
OPEN
New
Delete

Scenario Outline: Failure when opening a query to recipient that does not exist

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Update">
    <mdsol:Query Recipient="" Value="Analyse the race value"
Status="Open"/>
</ItemData>
"""
```

Then I should receive an HTTP response with status code "Not Found"

And reason code "RWS00071"

And the message "Query marking group name not found"

Examples:

marking group
Site from
CRA Site

Scenario Outline: Failure when opening a query with invalid RequiresResponse attribute

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" RequiresResponse="<option>" 
Value="Analyse the race value" Status="Open"/>
</ItemData>
"""
```

Then I should receive an HTTP response with status code "Bad Request"

And reason code "RWS00056"

And the response should have the message beginning with "This is not a valid ODM

1.3 standard document"

Examples:

option
YES
NO
True
False
1
0

Scenario: Failure when opening a query with QueryRepeatKey

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" QueryRepeatKey="1" Value="Analyse the
race value" Status="Open"/>
</ItemData>
"""
```

Then I should receive an HTTP response with status code "Bad Request"

And reason code "RWS00120"

And the message "Query cannot be opened with query repeat key"

Scenario: Failure when opening a query with empty query text

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Value="" Status="Open"/>
</ItemData>
"""
```

Then I should receive a successful XML response

And I should see no query opened against the field "AE_DESC"

Scenario: Failure when opening a query with missing Value attribute

When I update the field with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Status="Open"/>
</ItemData>
"""
```

Then I should receive a successful XML response

And I should see no query opened against the field "AE_DESC"

Answer Query

Scenario: Answer a query with response text

When I answer the query with the following ODM:

```
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" QueryRepeatKey="{query id}" Response="None
identified" Status="Answered"/>
</ItemData>
"""
```

Then I should receive a successful XML response

And I should see my query has been "Answered" with response "None identified" in Rave

Scenario Outline: Answer query supports localized strings in response text

When I answer the query with the following ODM:

```
"""
```

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" QueryRepeatKey="{query id}"
Response="<query text>" Status="Answered"/>
</ItemData>
"""
Then I should receive a successful XML response
And I should see my query has been "Answered" with response "<query text>" in Rave
```

Examples:

```
| query text  
| □□□□□□□  
| □□□□
```

Scenario: Failure when answering a query which does not require a response

Given there is an open query with following data:

When I answer the query with the following ODM:

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Response="None identified"
QueryRepeatKey="{query id}" Status="Answered"/>
</ItemData>
"""
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00074"
And the message "Query not does not require response"
```

Scenario: Failure when answering a query which requires response with empty string
When I answer the query with an empty response
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00129"
And the message "Query requires response"

Scenario: Failure when answering a query without a valid role permission
Given the background
But I am not authorised to answer a query
When I answer the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00073"
And the message "Answer query not authorized"

```

Scenario: Failure when answering a query when Pre-conditions are not met
  Given the background
  And there is an open query with following data:
    | form | form_repeat | field           | value        | query
marking_group | query_id   | requires_response |
    | AE      | 1           | AE_STARTDATE | 29 Jan 2009 | Analyse the race

```

```
from CRA | {query id} | Yes           |
But my role has the following Pre-conditions set for "AnswerQuery" permission:
| mask | status | pre_condition   |
| true | true  | IsNonConformant |
And the field "AE_STARTDATE" is not "NonConformant"
When I answer the query with the following ODM:
"""
<ItemData ItemOID="AE_STARTDATE" Value="29 Jan 2009" TransactionType="Context">
  <mdsol:Query Recipient="Site from CRA" QueryRepeatKey="{query id}" Response="None
identified" Status="Answered"/>
</ItemData>
"""

Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00073"
And the message "Answer query not authorized"

Scenario: Failure when answering a cancelled query
Given the background
But the query is cancelled
When I answer the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00094"
And the message "Query is already closed or cancelled"

Scenario: Failure when answering a closed query
Given the background
But the query is closed
When I answer the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00094"
And the message "Query is already closed or cancelled"

Scenario: Failure when answering a query with invalid marking group
When I answer the query with the following ODM:
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
  <mdsol:Query Recipient="Site from" QueryRepeatKey="{query id}" Response="None
identified" Status="Answered"/>
</ItemData>
"""

Then I should receive an HTTP response with status code "Not Found"
And reason code "RWS00071"
And the message "Query marking group name not found"
```

Forward Query

```
Scenario: Forward an open query
When I forward the query with the following ODM:
```

```
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
  <mdsol:Query Recipient="Site from DM" QueryRepeatKey="{query id}"
Status="Forwarded"/>
</ItemData>
"""

Then I should receive a successful XML response
And I should see my query has been forwarded to "Site from DM" in Rave

Scenario: Forward multiple open query
Given there are multiple open queries
When I forward all my queries to "Site from DM"
Then I should receive a successful XML response
And I should see the following query updated in Rave:
| query_id      | query_status | marking_group |
| {query id1}   | Open          | Site from DM  |
| {query id2}   | Open          | Site from DM  |

Scenario: Failure when forwarding a query without a valid role permission
Given the background
But I am not authorised to forward a query
When I forward the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00075"
And the message "Forward query not authorized"

Scenario: Failure when forwarding a query when Pre-conditions are not met
Given the background
And there is an open query with following data:
| form | form_repeat | field           | value        | query      |
marking_group | requires_response |
| AE    | 1            | AE_STARTDATE | 29 JAN 2009 | Analyse the race value | Site
from CRA | Yes          |               |
But my role has the following Pre-conditions set for "ForwardQuery" permission:
| mask | status | pre_condition |
| true | true  | IsNonConformant |
And the field "AE_STARTDATE" is not "NonConformant"
When I forward the query with the following ODM:
"""

<ItemData ItemOID="AE_STARTDATE" Value="29 JAN 2009" TransactionType="Context">
  <mdsol:Query Recipient="Site from DM" QueryRepeatKey="{query id}"
Status="Forwarded"/>
</ItemData>
"""

Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00075"
And the message "Forward query not authorized"

Scenario: Failure when forwarding a cancelled query
Given the background
```

```
But the query is cancelled
When I forward the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00094"
And the message "Query is already closed or cancelled"
```

```
Scenario: Failure when forwarding a closed query
Given the background
But the query is closed
When I forward the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00094"
And the message "Query is already closed or cancelled"
```

```
Scenario: Failure when forwarding an answered query
Given the background
But the query is answered
When I forward the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00121"
And the message "Query already answered"
```

```
Scenario: Failure when forwarding a query to same marking group
When I forward the query to the same marking group
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00119"
And the message "Cannot Forward Query to the same Marking Group"
```

Cancel Query

```
Scenario: Cancel an open query
When I cancel the query with the following ODM:
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" QueryRepeatKey="{query id}"
Status="Cancelled"/>
</ItemData>
"""
Then I should receive a successful XML response
And I should see my query has been "Cancelled" in Rave
```

```
Scenario: Failure when cancelling a query without a valid role permission
Given the background
But I am not authorised to cancel a query
When I cancel the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00069"
And the message "Cancel query not authorized"
```

Scenario: Failure while cancelling a query when Pre-conditions are not met
Given the background
And there is an open query with following data:

form	form_repeat	field	value	query
marking_group	query id	requires_response		
AE	1	AE_STARTDATE	29 JAN 2009	Analyse the race value Site from CRA {query id} Yes

But my role has the following Pre-conditions set for "CancelQuery" permission:

mask	status	pre_condition
true	true	IsNonConformant

And the field "AE_STARTDATE" is not "NonConformant"
When I cancel the query with the following ODM:
"""

```
<ItemData ItemOID="AE_STARTDATE" Value="29 JAN 2009" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" QueryRepeatKey="{query id}">
        Status="Cancelled"/>
</ItemData>
```

"""
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00069"
And the message "Cancel query not authorized"

Scenario: Failure when cancelling an answered query
Given the background
But the query is answered
When I cancel the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00121"
And the message "Query already answered"

Scenario: Failure when cancelling a closed query
Given the background
But the query is closed
When I cancel the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00094"
And the message "Query is already closed or cancelled"

Scenario Outline: Failure when cancelling a query with invalid QueryRepeatkey attribute
When I cancel the query with the following ODM:
"""

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" Response="None identified">
        QueryRepeatKey="" Status="Cancelled"/>
</ItemData>
```

"""
Then I should receive an HTTP response with status code "<response>"
And reason code "<reason code>"

And the response should have the message beginning with "<response_message>"

Examples:

value	response	reason code	
response_message			
	Bad Request RWS00056	This is not a valid ODM 1.3 standard document	
1010101 Not Found	RWS00072	Query repeat key not found	

Re-query Query

Scenario: Re-query an answered query

Given the background

But the query is answered

When I re-query with the following ODM:

"""

```
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
  <mdsol:Query PrecedingQueryRepeatKey="{query id}" Value="Re-Analyse the race
value" Status="Open"/>
</ItemData>
"""

```

Then I should receive a successful XML response

And the response should have the following query elements:

"""

```
<Query ID="{query id}" FieldOID="AE_DESC" />
"""

```

And I should see my query has been "Open" in Rave

Scenario: Failure when opening re-query without a valid role permission

Given the background

But the query is answered

And I am not authorised to open a query

When I re-query the field

Then I should receive an HTTP response with status code "Forbidden"

And reason code "RWS00068"

And the message "Open query not authorized"

Scenario: Failure when opening a re-query when Pre-conditions are not met

Given the background

And there is an answered query with following data:

form	form_repeat	field	value	query	
marking_group		query id	requires_response		
AE	1	AE_STARTDATE	29 Jan 2009	Analyse the race value Site from CRA {query id} Yes	

But my role has the following Pre-conditions set for "OpenQuery" permission:

mask	status	pre_condition	
true	true	IsNonConformant	

And the field "AE_STARTDATE" is not "NonConformant"
When I re-query with the following ODM:
"""
<ItemData ItemOID="AE_STARTDATE" Value="29 Jan 2009" TransactionType="Context">
 <mdsol:Query PrecedingQueryRepeatKey="{query id}" Value="Re-Analyse the race
value" Status="Open"/>
</ItemData>
"""

Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00068"
And the message "Open query not authorized"

Scenario Outline: Failure when opening a re-query with invalid
PrecedingQueryRepeatKey attribute
Given the background
But the query is answered
When I re-query with the following ODM:
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
 <mdsol:Query PrecedingQueryRepeatKey="" Value="Re-Analyse the race
value" Status="Open"/>
</ItemData>
"""

Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00117"
And the message "Preceding query does not exist"

Examples:

key_value
0
10101010

Scenario: Failure when re-opening a cancelled query
Given the background
But the query is cancelled
When I re-query the field
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00118"
And the message "Cannot re-query an unanswered query"

Close Query

Scenario: Close an answered query
Given the query is answered
When I close the query with the following ODM:
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
 <mdsol:Query Recipient="Site from CRA" QueryRepeatKey="{query id}"
Status="Closed"/>

```
</ItemData>
"""

Then I should receive a successful XML response
And I should see the following query updated in Rave:
| query_id | query_status |
| {query id} | Closed |


Scenario: Failure when closing a query without CloseQuery permission
Given the background
And the query is answered
But I am not authorised to Close a query
When I close the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00070"
And the message "Close query not authorized"

Scenario: Close query should fail when Pre-conditions are not met
Given the background
And there is an answered query with following data:
| form | form_repeat | field | value | query |
marking_group | requires_response |
| AE | 1 | AE_STARTDATE | 29 Jan 2009 | Analyse the race value | Site
from CRA | Yes |
And the field "AE_STARTDATE" is not "NonConformant"
But my role has the following Pre-conditions set for "CloseQuery" permission:
| mask | status | pre_condition |
| true | true | IsNonConformant |
When I close the query with the following ODM:
"""

<ItemData ItemOID="AE_STARTDATE" Value="29 Jan 2009" TransactionType="Context">
    <mdsol:Query QueryRepeatKey="{query id}" Status="Closed"/>
</ItemData>
"""

Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00070"
And the message "Close query not authorized"

Scenario: Failure when closing an unanswered query
When I close the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00116"
And the message "Query cannot be closed without answer"

Scenario: Failure when closing an already closed query
Given the background
And the query is closed
When I close the query with the following ODM:
"""

<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" QueryRepeatKey="{query id}">
```

```
Status="Closed"/>
</ItemData>
"""
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00094"
And the message "Query is already closed or cancelled"
```

Scenario: Failure when closing an already cancelled query

```
Given the background
And the query is cancelled
When I close the query
Then I should receive an HTTP response with status code "Forbidden"
And reason code "RWS00094"
And the message "Query is already closed or cancelled"
```

Scenario Outline: Failure when closing query with invalid QueryRepeatkey attribute

```
Given the background
And the query is answered
When I close the query with the following ODM:
```

```
"""
<ItemData ItemOID="AE_DESC" Value="HEADACHE" TransactionType="Context">
    <mdsol:Query Recipient="Site from CRA" QueryRepeatKey="">
Status="Closed"/>
</ItemData>
"""
Then I should receive an HTTP response with status code "<response>"
And reason code "<reason code>"
And the response should have the message beginning with "<response_message>"
```

Examples:

value	response	reason code	
response_message			
	Bad Request RWS00056		This is not a valid ODM 1.3 standard document
1010101	Not Found RWS00072		Query repeat key not found



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.21.1 Example - Open a Rave Query Using TransactionType Context

You can post an RWS request which opens a Query in Rave, using the custom vendor extension element mdsol:Query.

A request to open a query can use a TransactionType attribute of Context in the parent element. This checks the existing data and makes adding the query conditional (the query is only added if the data has not changed). If necessary, a request to open a query can use a TransactionType attribute of Update in the parent element. This resets the value of the parent element and opens the query. In this example, the value Context is used for the TransactionType attribute of the ItemData element. This makes opening the query conditional on the values for the ItemOID and Value attributes for the ItemData element being the same in the database as they are in the request.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
...  
  <ItemData ItemOID="RACE" Value="1" TransactionType="Context">  
    <mdsol:Query Recipient="Site from CRA" Value="Analyse the race value"  
    Status="Open">  
    </ItemData>
```

```
...  
</ODM>
```

1.6.21.2 Example - Open a Rave Query Using ItemData SubmitMarkingsOnly

You can post an RWS request which opens a Query in Rave, using the custom vendor extension element mdsol:Query.

From Rave Core 2025.2.0 onwards, you can use a new ItemData optional attribute, mdsol:SubmitMarkingOnly. In the following example, mdsol:SubmitMarkingOnly="Yes", which means that you can open a query without submitting any data. If mdsol:SubmitMarkingOnly="No" or if you omit it, then the functionality remains the same as before.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N  
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<FormData FormOID="AE">  
    <ItemGroupData ItemGroupOID="AE">  
        <ItemData ItemOID="ONE" Value="" TransactionType="Update"  
        mdsol:SubmitMarkingOnly="Yes">  
            <mdsol:Query Recipient="Site" Value="This is query text."  
        Status="Open"/>  
    </ItemData>
```

```
</ItemGroupData>  
</FormData>
```

1.6.22 Add Translations to Fields

Use Case

I am updating clinical data and ...

I am adding my translations to a field

My ODM snippet

```
...  
<ItemData OID="{field-oid}" Value="{value}" TransactionType="Context">  
  <mdsol:TranslatedData Value="{value}" TransactionType="Insert"/>  
</ItemData>  
...
```

mdsol:TranslatedData Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
Value	Value	Yes	
TransactionType	TransactionType	Yes	

Examples

Example:

```
...
<ItemData OID="MEDICATION" Value="paracétamol" TransactionType="Context">
<mdsol:TranslatedData Value="paracetamol" TransactionType="Insert"/>
</ItemData>
...
```



Note: The field data value must match that supplied using the "Context" transaction type.

Example: Importing Translated Data



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.22.1 Example - Import Translated Data

You can import translated data using the vendor extension element mdsol:TranslatedData. This method is used to translate from the language of the data entry to the default locale language of the Rave instance URL. You must use the TransactionType of Context with this element. The text to be translated and the translation text are the values of the Value attribute.

For example, to translate the data point value from French in to English, use the following format of ODM.

POST Request

Header

```
Authorization : Basic ZGV2OnBhc3N
Content-type : "text/xml"
```

URI

```
POST https://my-organisation.mdsol.com/RaveWebServices/webservice.aspx?  
PostODMClinicalData
```

Body

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
<ODM>  
...  
    <ItemData OID="MEDICATION" Value="paracétamol" TransactionType="Context">  
        <mdsol:TranslatedData Value="paracetamol"></mdsol:TranslatedData>  
    </ItemData>  
...  
</ODM>
```

[Back to top](#)

1.6.23 Update Field Workflow Properties

Use Case

I am updating clinical data and ...

I am updating the Lock status on my field

or

I am updating the Freeze status of my field

or

I am updating the Verify status of my field

My ODM snippet

```
...
<ItemData
  ItemOID="{field-oid}"
  Value="{data}"
  TransactionType="Update"
  mdsol:Lock="{Yes|No|AsFound}"
  mdsol:Freeze="{Yes|No}"
  mdsol:Verify="{Yes|No}" >
</ItemData>
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
ItemOID	The field OID	Yes	
Value	The field value	Yes	
mdsol:Lock	The lock status	No	See mdsol:Lock
mdsol:Freeze	The freeze status	No	
mdsol:Verify	The verify status	No	

mdsol:Lock

- If the attribute value is specified as 'Yes' and the field is not locked, RWS will lock it.
- If the attribute value is specified as 'No' and the field is locked, RWS will unlock it, after which the field may be updated.
- If the attribute value is specified as 'AsFound', the field can be updated irrespective of its lock status. If locked, RWS will subsequently unfreeze, update then re-freeze and re-lock the field in a single transaction. The same process is applicable to frozen fields.
- When using 'AsFound', if the user role does not have sufficient permission to execute the workflow back to it's original state, then RWS will return a 'Forbidden' error.
- The Audits page in Rave will display every change made by RWS.

Examples

```
<ItemData ItemOID="BIRTHDTC" Value="05 Jul 1986" mdsol:Lock="Yes"
TransactionType="Update"/>
<ItemData ItemOID="BIRTHDTC" Value="05 Jul 1986" mdsol:Lock="No"
TransactionType="Update"/>
<ItemData ItemOID="BIRTHDTC" Value="05 Jul 1986" mdsol:Lock="AsFound"
TransactionType="Update"/>
<ItemData ItemOID="BIRTHDTC" Value="05 Jul 1986" mdsol:Freeze="Yes"
TransactionType="Update"/>
<ItemData ItemOID="BIRTHDTC" Value="05 Jul 1986" mdsol:Freeze="No"
TransactionType="Update"/>
<ItemData ItemOID="BIRTHDTC" Value="05 Jul 1986" mdsol:Verify="Yes"
TransactionType="Update"/>
<ItemData ItemOID="BIRTHDTC" Value="05 Jul 1986" mdsol:Verify="No"
TransactionType="Update"/>
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

Given I am a Rave user

And there is a study with the following structure:

form	field	field_type	field_data_type	field_control_type
AE	AEDATE	Log	date	DateTime

And there is a subject in the study with the following data:

form	form_transaction	logline_repeat	logline_transaction	field_value	field_transaction
AE	Update	1	Update	AEDATE	29 JAN 2011

Scenario Outline: Successfully lock a log field for a positive pre-condition

Given I am assigned to the EDC Role "Power User"

And the role has the following Pre-conditions set for "Lock" permission:

mask	status	pre_condition
true	true	IsNotEmpty

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
            <ItemData ItemOID="AEDATE" Value="29 JAN 2011" mdsol:Lock="<lock_status>" TransactionType="<transaction_type>" />
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

Then I should receive a successful response
And the field status should be <field_lock_status>
```

Examples:

lock_status	transaction_type	field_lock_status
Yes	Update	Locked
Yes	Context	Locked
AsFound	Update	UnChanged
AsFound	Context	UnChanged

Scenario Outline: Successfully unlock a log field for a positive pre-condition

Given I am assigned to the EDC Role "Power User"

And the role has the following Pre-conditions set for "Unlock" permission:

mask	status	pre_condition
true	true	IsNotEmpty

And the following fields have been locked:

form	field	logline_repeat
AE	AEDATE	1

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
```

```

<ItemData ItemOID="AEDATE" Value="29 JAN 2011" mdsol:Lock="<lock_status>" TransactionType="<transaction_type>" />
</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful response
And the field status should be <field_lock_status>

Examples:

lock_status	transaction_type	field_lock_status
No	Update	Unlocked
No	Context	Unlocked

Scenario Outline: Successfully lock a log field for a negative pre-condition
Given I am assigned to the EDC Role "Power User"

And the role has the following Pre-conditions set for "Lock" permission:

mask	status	pre_condition
true	false	IsNonConformant

When I update the subject with the following ODM:

```

<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
<FormData FormOID="AE">
<ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
<ItemData ItemOID="AEDATE" Value="29 JAN 2011" mdsol:Lock="<lock_status>" TransactionType="<transaction_type>" />
</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful response
And the field status should be <field_lock_status>

Examples:

lock_status	transaction_type	field_lock_status
Yes	Update	Locked
Yes	Context	Locked
AsFound	Update	UnChanged
AsFound	Context	UnChanged

Scenario Outline: Successfully unlock a log field for a negative pre-condition
Given I am assigned to the EDC Role "Power User"

And the role has the following Pre-conditions set for "Unlock" permission:

mask	status	pre_condition
true	false	IsNonConformant

And the following fields have been locked:

form	field	logline_repeat
AE	AEDATE	1

When I update the subject with the following ODM:

```

<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
<FormData FormOID="AE">

```

```

<ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
    <ItemData ItemOID="AEDATE" Value="29 JAN 2011" mdsol:Lock="<lock_status>" TransactionType="<transaction_type>" />
</ItemGroupData>
</FormData>
</StudyEventData>
"""

```

Then I should receive a successful response
And the field status should be <field_lock_status>

Examples:

lock_status	transaction_type	field_lock_status	
No	Update	Unlocked	
No	Context	Unlocked	

Scenario: Successfully unlock and update a log field using AsFound with pre-condition to lock not met

Given I am assigned to the EDC Role "Power User"

And the role has the following Pre-conditions set for "Lock" permission:

mask	status	pre_condition	
true	true	IsNonConformant	

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
            <ItemData ItemOID="AEDATE" Value="02 FEB 1988" mdsol:Lock="AsFound" TransactionType="Update" />
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

Then I should receive a successful response

And the field status should be Unlocked

Scenario: Failure to lock and update a log field using AsFound with pre-condition to unlock not met

Given I am assigned to the EDC Role "Power User"

And the following fields have been locked:

form	field	logline_repeat	
AE	AEDATE	1	

And the role has the following Pre-conditions set for "Unlock" permission:

mask	status	pre_condition	
true	true	IsNonConformant	

When I update the subject with the following ODM:

```

"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="AE">
        <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
            <ItemData ItemOID="AEDATE" Value="02 FEB 2000" mdsol:Lock="AsFound" TransactionType="Update" />
        </ItemGroupData>
    </FormData>
</StudyEventData>
"""

```

```
    </ItemGroupData>
  </FormData>
</StudyEventData>
"""

```

Then I should receive an error response with status code "Forbidden"
And reason code "RWS00058"
And the message "Unlock not authorized"

Scenario: Failure to lock a log field for a positive pre-condition not met
Given I am assigned to the EDC Role "Power User"

And the role has the following Pre-conditions set for "Lock" permission:

mask	status	pre_condition
true	true	WasSigned

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="AE">
    <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
      <ItemData ItemOID="AEDATE" Value="02 FEB 1981" mdsol:Lock="Yes"
               TransactionType="Update" />
    </ItemGroupData>
  </FormData>
</StudyEventData>
"""

```

Then I should receive an error response with status code "Forbidden"
And reason code "RWS00057"
And the message "Lock not authorized"

Scenario Outline: Failure to unlock a log field for a positive pre-condition not met
Given I am assigned to the EDC Role "Power User"

And the following fields have been locked:

form	field	logline_repeat
AE	AEDATE	1

And the role has the following Pre-conditions set for "Unlock" permission:

mask	status	pre_condition
true	true	IsNonConformant

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="AE">
    <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
      <ItemData ItemOID="AEDATE" Value="29 JAN 2011" mdsol:Lock="<lock_status>"
               TransactionType="<transaction_type>" />
    </ItemGroupData>
  </FormData>
</StudyEventData>
"""

```

Then I should receive an error response with status code "Forbidden"
And reason code "RWS00058"
And the message "Unlock not authorized"

Examples:

lock_status	transaction_type
No	Update
AsFound	Update

Scenario: Failure to lock a log field for a negative pre-condition not met
 Given I am assigned to the EDC Role "Power User"

And the role has the following Pre-conditions set for "Lock" permission:

mask	status	pre_condition
1	true	false

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="AE">
    <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
      <ItemData ItemOID="AEDATE" Value="02 FEB 2001" mdsol:Lock="Yes"
        TransactionType="Update" />
    </ItemGroupData>
  </FormData>
</StudyEventData>
"""
```

Then I should receive an error response with status code "Forbidden"

And reason code "RWS00057"

And the message "Lock not authorized"

Scenario: Failure to unlock a log field for a negative pre-condition not met
 Given I am assigned to the EDC Role "Power User"

And the following fields have been locked:

form	field	logline_repeat
AE	AEDATE	1

And the role has the following Pre-conditions set for "Unlock" permission:

mask	status	pre_condition
1	true	false

And the field "GENDER" is "Frozen"

When I update the subject with the following ODM:

```
"""
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
  <FormData FormOID="AE">
    <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
      <ItemData ItemOID="AEDATE" Value="02 FEB 2001" mdsol:Lock="No"
        TransactionType="Update" />
    </ItemGroupData>
  </FormData>
</StudyEventData>
"""
```

Then I should receive an error response with status code "Forbidden"

And reason code "RWS00058"

And the message "Unlock not authorized"

Scenario: Failure to lock a log field for status only pre-condition not met

Given I am assigned to the EDC Role "Power User"
 And the role has the following Pre-conditions set for "Lock" permission:
 | mask | status | pre_condition |
 | false | true | IsNotFrozen |
 When I update the subject with the following ODM:
 """

```
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
<FormData FormOID="AE">
  <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
    <ItemData ItemOID="AEDATE" Value="02 FEB 2001" mdsol:Lock="Yes"
      TransactionType="Update" />
  </ItemGroupData>
</FormData>
</StudyEventData>
"""
```

Then I should receive an error response with status code "Forbidden"
 And reason code "RWS00057"
 And the message "Lock not authorized"

Scenario Outline: Failure to unlock a log field for status only pre-condition not met
 Given I am assigned to the EDC Role "Power User"

And the following fields have been locked:

form	field	logline_repeat	
AE	AEDATE	1	

And the role has the following Pre-conditions set for "Unlock" permission:

mask	status	pre_condition	
false	true	IsNotFrozen	

When I update the subject with the following ODM:
 """

```
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
<FormData FormOID="AE">
  <ItemGroupData ItemGroupOID="AE_LOG_LINE" ItemGroupRepeatKey="1">
    <ItemData ItemOID="AEDATE" Value="29 JAN 2000" mdsol:Lock=""
      TransactionType="" />
  </ItemGroupData>
</FormData>
</StudyEventData>
"""
```

Then I should receive an error response with status code "Forbidden"
 And reason code "RWS00058"
 And the message "Unlock not authorized"

Examples:

lock_status	transaction_type	
No	Update	
AsFound	Update	



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

1.6.24 Assign Labs to Forms

Use Case

I am updating clinical data and ...

I want to assign a central or local lab to a form
So that I can post clinical data for a subject

My ODM snippet

```
...
<StudyEventData    >
  <StudyEventData StudyEventOID="{study-event-oid}" TransactionType="Update">
    <FormData FormOID="{form-oid}" FormRepeatKey="{form-repeat-key}">
      mdsol:LaboratoryRef="{lab-name}" mdsol:LaboratoryType="{lab-type}">
        </FormData>
    </StudyEventData>
...
...
```

ODM Attributes



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
StudyEventOID	The study identifier	Yes	
FormRepeatKey	The form repeat key	No	
mdsol:LaboratoryRef	The lab name	No	
mdsol:LaboratoryType	The lab type	No	Two values central or local

Example

```
...
<FormData FormOID="LAB_TEST" FormRepeatKey="1" mdsol:LaboratoryRef="my_local_lab_2"
mdsol:LaboratoryType="Local">
</FormData>
...
...
```

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Scenarios

Background:

```
Given I am a Rave user
And there is a valid lab configuration
And there is a study with the following structure:
| form           | field                  | field_data_type      | field_length |
field_control_type |
| LAB_TEST       | MY_VARIABLE_RWS_101 | integer              | 3            |
Text               |
And there is a subject in the study
And the local labs have been setup for the site
```

Scenario: Successfully update the lab associated with the form

```
Given the form "LAB_TEST" is associated with "my_local_lab" lab
When I update the subject with the following ODM:
```

```
"""
<StudyEventData StudyEventID="SUBJECT" TransactionType="Update">
  <FormData FormOID="LAB_TEST" FormRepeatKey="1"
mdsol:LaboratoryRef="my_local_lab_2" mdsol:LaboratoryType="Local">
    </FormData>
  </StudyEventData>
"""

```

Then I should receive a successful XML response

And the lab assigned to the form is "my_local_lab_2"

And the analyte ranges assigned to the fields on the form are present

Scenario: Updating the lab with the same value does not clear the analyte ranges

Given the form "LAB_TEST" is associated with "my_local_lab" lab

When I update the subject with the following ODM:

```
"""

```

```
<StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
    <FormData FormOID="LAB_TEST" FormRepeatKey="1"
mdsol:LaboratoryRef="my_local_lab" mdsol:LaboratoryType="Local">
    </FormData>
</StudyEventData>
"""

Then I should receive a successful XML response
And the lab assigned to the form is "my_local_lab"
And the analyte ranges assigned to the fields on the form are present
```



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

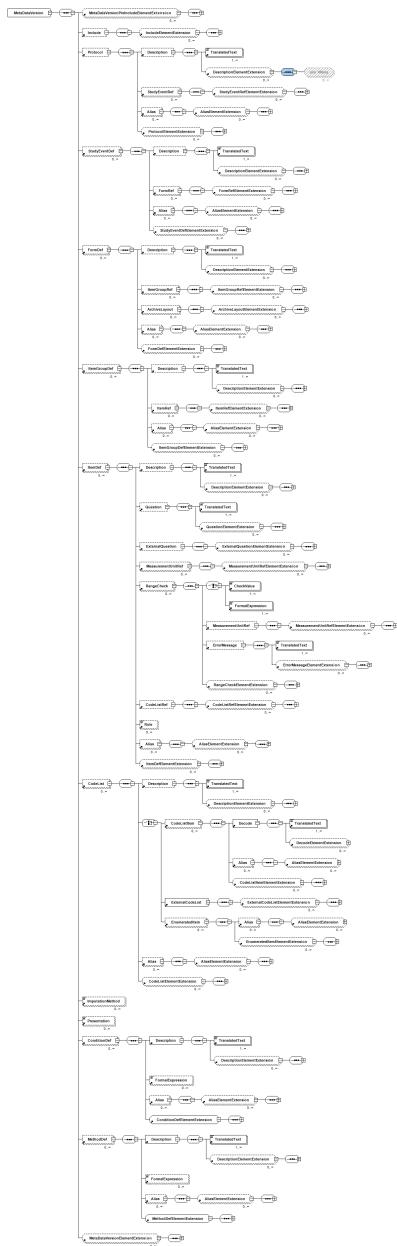
1.7 References

The following references provide you with background information and details regarding the use of RWS [request services](#):

- [Operational Data Model \(ODM\) Schema](#)
- [ODM Adapter Guidelines](#)
- [ODM Adapter Included Audit Subcategories](#)
- [Clinical View Service Limitations](#)
- [Configurable Datasets Authorization Best Practices](#)
- [Navigating Repeated, Nested Folders](#)
- [Definitions](#)
- [Error Responses - Complete List](#)
- [Specify a Study OID](#)
- [URL Special Character Escaping](#)
- [Product Mapping Terminology](#)

1.7.1 Operational Data Model (ODM) Schema

The Operational Data Model (ODM) is represented as an XML document and consists of a hierarchical tree of elements, as shown below:



These elements can have attributes that define their features. Elements and attributes in the Rave Web Services API follow the CDISC ODM 1.3 standard and the ODM 1.3 rules on the hierarchy of elements. The order of child elements must be followed.



Note: Some vendor extensions to the ODM 1.3 standard are also used.

- [How to Use the Operational Data Model \(ODM\)](#)
- [ODM Schema Downloads and Releases History](#)
- [RWS ODM Header Requirements](#)
- [ODM Study Metadata Schema Support](#)
- [ODM Subject Data Schema Support](#)
- [A Valid ODM 1.3 Snapshot Document](#)
- [A Valid ODM 1.3 Transactional Document](#)

1.7.1.1 How to Use the Operational Data Model (ODM)

This page provides information regarding the correct configuration of requests that use the Rave Web Services (RWS) API. The rules and procedures listed apply to all elements and attributes described in this guide.

About Elements and Attributes

The [Operational Data Model \(ODM\)](#) is represented as an XML document and consists of a hierarchical tree of elements. These elements may have attributes that define their features. Elements and attributes in the Rave Web Services API follow the CDISC ODM 1.3 standard, and the ODM 1.3 rules on the hierarchy of elements. You must follow the given order of child elements.



Note: RWS also uses some vendor extensions to the ODM 1.3 standard.

Elements

An element in ODM can contain one or more elements or element groups. The sub-elements or sub-element groups are listed in the Body section of the element description. An element group consists of one or more element names (or element group names) enclosed in parentheses and separated by commas or vertical bars.

The following rules follow the [Syntax Notation specified in section 2.12 of the CDISC ODM 1.3 standard](#) and apply to element groups:

Rule	Description
Elements or element groups separated by commas	The elements or element groups must occur in the ODM document in the order listed in the group.
Elements or element groups separated by a pipe character ()	Exactly one of the elements or element groups separated by vertical lines must occur in the ODM document.
Element or element group followed by a question mark (?)	The element or element group is optional.
Element or element group followed by an asterisk (*)	The element or element group may occur zero or more times.
Element or element group followed by a plus sign (+)	The element or element group must occur one or more times.

Element Order

The order of elements in requests processed by RWS is significant. Always follow the order of elements in element groups. Requests that present elements in an incorrect order may fail.

Attributes

The attributes for each element are listed in the element definitions. The following rules apply to attributes:

Rule	Description
Attribute options displayed in curly braces & separated by a pipe character (), for example, { True False }	Select one of the possible options.
Attributes displayed with the notation N/A in the Rave Mapping column	The attributes are not imported into Rave. However, some of these attributes may be required for conformance with the ODM standard.
'Optional' in the 'Notes' column	The attribute is optional.

Rule	Description
Attributes displayed with the notation 'Required' in the 'Notes' column	The attribute is required for RWS processing or for ODM document validation.

About Adding New Vendor Extensions

Periodically, Medidata may introduce new **mdsol**: vendor extension attributes and elements to the the Rave Web Services API. In such instances, we take care to maintain backwards compatibility. Vendor extension attributes are, by definition, optional components. If we add any new elements or make any changes to element ordering, we always ensure that these are defined as optional behaviors. Incompatible, optional behavior is triggered by a parameter in the URL. If the parameter is omitted, the original compatible behavior persists.

ODM XML Version Support

RWS supports [ODM version 1.3](#) for clinical data POST and metadata POST and GET requests. We only recommend the use of ODM version 1.3.

RWS does support ODM version 1.2 for clinical data POST requests only; however, use ODM version 1.2 only where absolutely necessary.



Note: Vendor extensions are not supported with ODM version 1.2.

Vendor Extensions

Medidata uses the CDISC vendor extension convention to add attributes and elements that are not defined in the CDISC ODM. The namespace is: `xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata"`. Vendor extensions are shown in the following examples:

Attribute example

```
<FormDef OID="FRM001" Name="Patient" Repeating="No" mdsol:Active="Yes">
  ...
</FormDef>
```

Element example

```
<FormDef OID="FRM001" Name="Patient" Repeating="No">
  <mdsol:ViewRestriction>Role 1</mdsol:ViewRestriction>
  <mdsol:ViewRestriction>Role 2</mdsol:ViewRestriction>
</FormDef>
```

When a vendor extension attribute is used with a vendor extension element, the mdsol: namespace indicator is used with the element only - not with the attribute.

Error Reporting

RWS generates a response code for every transaction. The [Error Responses Listing](#) provides a list of error response codes. Error details are returned in the body of the response.

The error details may include the error response message and error location in XPath notation. The body format may or may not be in CDISC ODM format, depending on the request service.

Security

The Rave user account used by the external system calling RWS must have the relevant permissions. See the [Rave Role Permissions](#) for more information on Rave security roles.

Maximum Length for Requests

There are limits to the length of request messages. The table below provides the default size limit for each request type in bytes. An administrator can change these default values in the WebServicesRequestProcessors database table.

Type of Request	Default Maximum Length (bytes)
Clinical Data POST	1,000,000
Metadata GET	2,000,000
Metadata POST	2,000,000



Note: The limit does not apply to GET requests.

1.7.1.2 ODM Schema Downloads and Releases History

ODM Schema Download Files

Rave Version	Filename	Schema Version	GA Date
2024.1.0	ODMSchema2024.1.0.zip	ODMSchema2024.1.0	19th Apr 2024
2020.3.0	ODMSchema2020.3.0.zip	ODMSchema2020.3.0	20th Nov 2020
2019.2.0	ODMSchema2019.2.0.zip	ODMSchema2019.2.0	22nd Nov 2019
2018.1.0	ODMSchema2018.1.0.zip	ODMSchema2018.1.0	29th Jun 2018
2015.3.0	ODMSchema2015.3.0.zip	ODMSchema2015.3.0	16th Oct 2015
2015.1.0	ODMSchema2015.1.0.zip	ODMSchema2015.1.0	26th Jan 2015
2014.2.0	ODMSchema2014.1.0.zip	ODMSchema2014.1.0	26th Sep 2014
2014.1.0	ODMSchema2014.1.0.zip	ODMSchema2014.1.0	25th Apr 2014
2013.4.0	ODMSchema2013.3.0.zip	ODMSchema2013.3.0	30th Dec 2013
2013.3.0	ODMSchema2013.2.0.zip	ODMSchema2013.2.0	27th Sep 2013
2013.2.0	ODMSchema2013.1.0.zip	ODMSchema2013.1.0	28th Jun 2013
2013.1.0.1	ODMSchema2013.1.0.zip	ODMSchema2013.1.0	30th May 2013
2013.1.0	ODMSchema2012.2.0.zip	ODMSchema2012.2.0	20th Mar 2013
2012.1.0.1	ODMSchema2012.2.0.zip	ODMSchema2012.2.0	08th Feb 2013

Rave Version	Filename	Schema Version	GA Date
2012.1.0	ODMSchema2012.2.0.zip	ODMSchema2012.2.0	21st Dec 2012
(earlier)	ODMSchema2012.1.0.zip	ODMSchema2012.1.0	N/A

ODM Schema Release History

ODM Schema Version 2024.1.0

[ODMSchema2024.1.0.zip](#)

- Rave Release 2024.1.0

Release Date: 19th Apr 2024

- Added optional attributes SubjectDate, PropertyValue, InitialMetaDataVersionOID, and MigratedMetaDataVersionOID to odm:SubjectData
- Added optional attribute DataPageDate to odm:FormData
- Added optional attributes PreviousLocationOID, SharedLocationOID, PreviousSharedLocationOID, SubjectCreatedAtSite to odm:SiteRef
- Added optional attributes RecordDate, Description to odm:ItemGroupData
- Added optional attribute MissingCodeDescription to odm:ItemData

ODM Schema Version 2020.3.0

[ODMSchema2020.3.0.zip](#)

- Rave Release 2020.3.0

Release Date: 20th Nov 2020

- Added group UserElementExtension to support new element mdsol:UserGroups
- Added complexType mdsol-UserGroups
- Added new element UserGroups of type mdsol-UserGroups
- Added new element UserGroupName of type odm:text

ODM Schema Version 2019.2.0

[ODMSchema2019.2.0.zip](#)

- Rave Release 2019.2.0

Release Date: 22nd Nov 2019

- Added optional attribute mdsol:StudyEnvSiteNumber to SiteRefAttributeExtension
- Added optional attribute mdsol:StudyEnvSiteNumber to LocationRefAttributeExtension
- Added enumeration value StudyEnvSiteNumber to LocationOIDType simpleType
- Added attribute StudyEnvSiteNumber to mdsol extensions

ODM Schema Version 2018.1.0

[ODMSchema2018.1.0.zip](#)

- Rave Release 2018.1.0

Release Date: 29th Jun 2018

- Added optional attribute mdsol:Active to odm.LocationRef
- Added optional attributes mdsol:RespectUniqueFileOID to ODMAbstractExtension.
- Added optional attribute mdsol:RunCleaningEngineInBatchMode to odm:ClinicalDataAttributeExtension
- Added optional attribute mdsol:RequireManualClose to odm:ItemData
- Added optional attribute mdsol:ChangeCode
- Added optional CommentTransactionType and modified mdsol-Comment

ODM Schema Version 2015.3.0

[ODMSchema2015.3.0.zip](#)

- Rave Release 2015.3.0

Release Date: 16th Oct 2015

- Added optional mdsol:InstanceOverDue attribute to odm:StudyEventData element
- Added optional mdsol:InstanceStartWindow attribute to odm:StudyEventData element

- Added optional mdsol:InstanceEndWindow attribute to odm:StudyEventData element
- Added optional mdsol:InstanceClose attribute to odm:StudyEventData element
- Added optional mdsol:InstanceAccess attribute to odm:StudyEventData element

ODM Schema Version 2015.1.0

[ODMSchema2015.1.0.zip](#)

- Rave Release 2015.1.0

Release Date: 26th Jan 2015

- Added optional mdsol:InstanceId attribute to odm:StudyEventData element
- Added optional mdsol:DataPageId attribute to odm:FormData element
- Added optional mdsol:RecordId attribute to odm:ItemGroupData element
- Added optional mdsol:ClinicalSignificance element to odm:ItemData element
- Added optional mdsol:LabData element to odm:ItemData element

ODM Schema Version 2014.1.0

[ODMSchema2014.1.0.zip](#)

- Rave Release 2014.1.0

Release Date: 25 Apr 2014

- Added optional mdsol:IncludeFileOID attribute to odm:AuditRecord element
- Added optional mdsol:MatrixOID attribute to MetadataVersion element
- Added optional TransactionType attribute to mdsol:ProtocolDeviation element
- Added optional mdsol:ProtocolDeviation element to ItemData Element extension

ODM Schema Version 2013.3.0

[ODMSchema2013.3.0.zip](#)

- Rave Release 2013.4.0

Release Date: 30 Dec 2013

- Added mdsol:PartialClinicalData as second-level element - same as ClinicalData

- Added UUID, CancelURI, CancelVERB, DateTime and Username attributes to mdsol:Query
- Made mdsol:Query - Status attribute optional
- Removed CoderUsername and CodingDictionaryVersion from mdsol:Query
- Added mdsol:CodingContextURI, mdsol:CodingTermUUID, mdsol:UpdatedTimeStamp to ItemData
- Made mdsol:Query - QueryRepeatKey attribute accept floats
- Added mdsol:LocationName to SiteRef

ODM Schema Version 2013.2.0

[ODMSchema2013.2.0.zip](#)

- Rave Release 2013.3.0

Release Date: 27 Sep 2013

- Added mdsol:SubjectActive and mdsol:Deleted attribute extensions to SubjectData element

ODM Schema Version 2013.1.0

[ODMSchema2013.1.0.zip](#)

- Rave Release 2013.2.0
- Rave Release 2013.1.0.1

Release Date: 30 May 2013

- Added mdsol:AuditSubCategoryName
- Removed mdsol:AuditSubCategoryId from ODM ClinicalData
- Added mdsol:CodingApprovalType, mdsol:CodingModeType, mdsol:CodingDictionaryVersion to ODM metadata in mdsol:CodingRequestDef
- Added mdsol:UserRole attribute extension to User element
- Added mdsol:InstanceName and mdsol:DataPageName to StudyEventData and FormData element respectively

ODM Schema Version 2012.2.0

[ODMSchema2012.2.0.zip](#)

- Rave Release 2013.1.0
- Rave Release 2012.1.0.1
- Rave Release 2012.1.0

Release Date: 21 Dec 2012

- Added mdsol:<Comment/@CommentRepeatKey> and mdsol:Comment/Value
- Renamed typo in mdsol schema type from TransactionTypeType to TransactionTypeType
- Made TransactionTypeType optional whenever used.
- Added mdsol:DerivationDef element extension
- Added mdsol:CustomFunctionDef element extension
- Added mdsol:EditCheckDef element extension
- Added mdsol:FileName attribute extension to ItemDataAny
- Added mdsol:Comment element extension
- Added mdsol:AuditSubCategoryId attribute extension to ClinicalData element
- Added mdsol:UserGroup and mdsol:SiteGroup attribute extensions to User element

1.7.1.3 RWS ODM Header Requirements

RWS supports ODM version 1.3 for clinical data POST and metadata POST and GET requests.

Medidata strongly recommends using ODM version 1.3 only. RWS does support ODM version 1.2 for clinical data POST requests only; however, use ODM version 1.2 only where absolutely necessary.

Vendor extensions are not supported with ODM version 1.2. [Click here to download the ODM 1.3.1 schema with mdsol vendor extensions.](#)

Note: View the CDISC ODM 1.3 Schema Definition

To view the CDISC definition for the CDISC ODM 1.3 schema: go to the [CDISC Website](#) > select the Standards menu > select Operational Data Model.

ODM Element

Rave Web Services only supports a single ODM element per transaction.

Body
(Study*, AdminData*, ReferenceData*, ClinicalData*, Association*, ds:Signature*)

The AdminData, ReferenceData, Association, and ds:Signature elements can have zero occurrences in ODM. RWS does not use any of the previously stated elements, and, therefore, ignores their definition if included in the Body section. For this reason, the table below does not contain information regarding the AdminData, ReferenceData, Association, and ds:Signature elements.

Attribute	Value	Notes
Description	text (255 characters max)	Optional
FileType	{Snapshot Transactional}	Required. RWS supports Transactional for clinical data, Snapshot for metadata
FileOID	oid	Required. FileOID is a unique identifier for the file. It is used for logging purposes.
Granularity	{ All Metadata AdminData ReferenceData AllClinicalData SingleSite SingleSubject}	The SingleSubject value is used for clinical data or Metadata. RWS supports the Metadata value for metadata.
Archival	{ Yes }	Optional
CreationDateTime	datetime	Required
PriorFileOID	oidref	Optional
AsOfDateTime	datetime	Optional
ODMVersion	text	Required. RWS only supports ODM versions "1.3" or "1.2". RWS checks the ODM version number, and validates the XML against the ODM 1.3 schema. RWS continues to provide backward compatibility for ODM 1.2 clinical data post only.
Originator	text (255 characters max)	Optional. Originator stores the value for the service provider business

Attribute	Value	Notes
		name. It is used for logging purposes.
SourceSystem	text (255 characters max)	Optional
SourceSystemVersion	text (255 characters max)	Optional
ID	ID	Optional

Example:

```
<ODM xmlns='http://www.cdisc.org/ns/odm/v1.3'
      ODMVersion='1.3'
      FileType='Transactional'
      FileOID='Example-1'
      CreationDateTime='2008-01-01T00:00:00'>
  <ClinicalData >
    ...
  </ClinicalData>
</ODM>
```

[Back to top](#)

1.7.1.4 ODM Study Metadata Schema Support

This article details the Study related ODM Schema elements supported in RWS.

Study

RWS only supports a single study element. If more than one study element is received, only the first study element is processed.

Body
GlobalVariables, BasicDefinitions?, MetaDataVersion*

Attribute	Value	Notes	Rave mapping
OID	oid	Required	Not mapped in Rave, but should be the same as Protocol Name.
mdsol:ProjectType	{ Project (default) GlobalLibrary Volume}	Optional	Maps to New Project or New Global Library Volume in Rave.

Contained In

ODM

Example:

```
<ODM>
  <Study OID="XEmplxy001" mdsol:ProjectType="Project">
    <GlobalVariables>
      ...
    </GlobalVariables>
    <BasicDefinitions>
      ...
    </BasicDefinitions>
    <MetaDataVersion>
      ...
    </MetaDataVersion>
  </Study>
</ODM>
```

GlobalVariables

Body

StudyName, StudyDescription, ProtocolName

Attribute

NONE

Contained In

Study

The GlobalVariables element includes general summary information about the study.

Example:

```
<ODM>
  <Study OID="XEmplxy001" mdsol:ProjectType="Project">
    <GlobalVariables>
      <StudyName>...</StudyName>
```

```
<StudyDescription>...</StudyDescription>
    <ProtocolName>XEmplyx001</ProtocolName>
</GlobalVariables>
<BasicDefinitions>
    ...
</BasicDefinitions>

<MetaDataVersion>
    ...
</MetaDataVersion>
</Study>
</ODM>
```

StudyName

StudyName specifies a short external name for the study. The content of the StudyName element is not imported by RWS. However, a valid ODM XML requires StudyName to be specified under the GlobalVariables section.

Body
name
Attribute
NONE
Contained In
Global Variables

Example:

```
...
<GlobalVariables>
    <StudyName>Not Used</StudyName>
    <StudyDescription>...</StudyDescription>
    <ProtocolName>...</ProtocolName>
</GlobalVariables>
...
```

StudyDescription

StudyDescription represents a free-text study description. The content of the StudyDescription element is not imported by RWS. However, a valid ODM XML requires StudyDescription to be specified under the GlobalVariables section.

Body
text
Attribute
NONE
Contained In
GlobalVariables

Example:

```
...
<GlobalVariables>
    <StudyName>Not Used.</StudyName>
    <StudyDescription>Not Used</StudyDescription>
    <ProtocolName>ProjectName</ProtocolName>
</GlobalVariables>
...
```

ProtocolName

The ProtocolName element represents the project name, and it maps to the Name field within the Project Settings in Rave. The size of ProtocolName is limited to maximum 50 characters.

Body
text
Attribute
NONE

Contained In

GlobalVariables

Example:

```
...
<GlobalVariables>
    <StudyName>Not Used</StudyName>
    <StudyDescription>Not Used</StudyDescription>
    <ProtocolName>Mediflex ODM</ProtocolName>
</GlobalVariables>
...
```

BasicDefinitions

Body

(MeasurementUnit*)

Attribute

NONE

Contained In

Study

Example:

```
<BasicDefinitions>
    <MeasurementUnit OID='...' Name='...'>
        <Symbol>
            <TranslatedText xml:lang="en">KG</TranslatedText>
        </Symbol>
    </MeasurementUnit>
</BasicDefinitions>
...
```

MeasurementUnit

Body

Symbol

Attribute	Value	Notes	Rave mapping
OID	oid	Required	Not mapped in Rave.
Name	text (255 characters max)	Required	Maps to Coded Unit within unit dictionary entries in Rave. See Note 1 .
mdsol:UnitDictionaryName	text (255 characters max)	Optional	Maps to unit dictionary Name in Rave. See Note 2 . If fixed units are used and this attribute is not specified, see Note 4 . (Also see the MeasurementUnitRef)
mdsol:ConstantA	integer	Optional	Maps to the unit dictionary Constant A in Rave (defaults to 1 if not present).
mdsol:ConstantB	integer	Optional	Maps to the unit dictionary Constant B in Rave (defaults to 1 if not present).
mdsol:ConstantC	integer	Optional	Maps to the unit dictionary Constant C in Rave (defaults to 0 if not present).
mdsol:ConstantK	integer	Optional	Maps to the unit dictionary Constant K in Rave (defaults to 0 if not present).
mdsol:OrderNumber	integer	Optional	Maps to the unit dictionary ordinal positioning in Rave.

Attribute	Value	Notes	Rave mapping
mdsol:StandardUnit	{Yes No (default)}	Optional	Yes = Standard checked within the unit dictionary entry in Rave. No = Standard unchecked within the unit dictionary entry in Rave. See Note 3 .

Contained In**BasicDefinitions**

MeasurementUnit is the physical unit of measure for a data item or value. The mdsol:ConstantA , mdsol:ConstantB, mdsol:ConstantC, and mdsol:ConstantK are used in the following formula to convert the user-entered value to the standard measurement unit equivalent:

Standard Value = mdsol:ConstantA / mdsol:ConstantB * (User Value + mdsol:ConstantC) + mdsol:ConstantK

Example:

```
...
<BasicDefinitions>
    <MeasurementUnit OID="MSU00001" Name="KG" mdsol:UnitDictionaryName="UN1"
        mdsol:ConstantA="1" mdsol:ConstantB="1" mdsol:ConstantC="0"
        mdsol:ConstantK="0"
        mdsol:StandardUnit="Yes">
        <Symbol>
            <TranslatedText xml:lang="en">KG</TranslatedText>
        </Symbol>
    </MeasurementUnit>
</BasicDefinitions>
...
```

Symbol

Symbol is a human-readable name for a measurement unit. The TranslatedText element within the Symbol element maps to the User Data String value for the unit dictionary entry in Rave.

Body**TranslatedText****Attribute****NONE****Contained In****MeasurementUnit**

Example:

```
...
<BasicDefinitions>
    <MeasurementUnit >
        <Symbol>
            <TranslatedText xml:lang="en">KG</TranslatedText>
        <Symbol>
    </MeasurementUnit>
</BasicDefinitions>
...
```

TranslatedText

The TranslatedText element contains human-readable text that is appropriate for a particular language. The use of the TranslatedText element shown in the table below varies depending on its parent element.

Body
text

Attribute	Value	Notes	Rave mapping
xml:lang	languageTag	Optional	Not mapped in Rave.

ContainedIn

Decode, Question, Symbol

You can ignore the `xml:lang` attribute when the `xml:lang` code does not exist for the language in the XML or the `TranslatedText`. [XML Specification RFC1766](#) lists the languages with `xml:lang` code.

- If the parent element is `Decode`, the `TranslatedText` element maps to the User Data String value for the data dictionary entry in Rave.
- If the parent element is `Question`, the `TranslatedText` element maps to the Field Label in Rave.
- If the parent element is `Symbol`, the `TranslatedText` element maps to the User Data String value for the unit dictionary entry in Rave (see [Note 5](#)).

Example:

```
...
<BasicDefinitions>
    <MeasurementUnit OID="...">
        <Symbol>
            <TranslatedText xml:lang="en">KG</TranslatedText>
        <Symbol>
    </MeasurementUnit>
</BasicDefinitions>
...
```

MetaDataVersion

RWS only supports a single `MetaDataVersion` element for metadata POST requests.

The `Include` element is optional and the `ImputationMethod` and `Presentation` elements can have zero occurrences in ODM. These elements are not used by RWS in the import and export processes. Hence, element definitions for `ImputationMethod` and `Presentation` are not included in this document.

A metadata version defines the types of study events, forms, item groups, and items that form the study data.

Attribute	Value	Notes	Rave mapping
OID	oid	Required	Not mapped in Rave.
Name	name (limited to OID + (Name) 254 characters maximum length)	Required	Maps to the CRF Draft Settings Draft Name in Rave.
Description	text	Optional	Not mapped in Rave.
mdsol:PrimaryFormOID	oid	Optional	Should contain a Form OID. Maps to the Primary Form within the CRF Draft Settings in Rave (see Note 6).
mdsol:DefaultMatrixOID	oid (limited to a maximum of 50 characters)	Optional	Should contain a Matrix name. Maps to the Default Matrix within the CRF Draft Settings in Rave. If not present, this defaults to Base.
mdsol:DeleteExisting	{Yes (default) No}	Optional	Yes means the current draft in Rave is replaced with the new draft created by RWS. No means the content of the new draft created by RWS is merged with any existing draft with the same name. Does not appear in metadata export.

Attribute	Value	Notes	Rave mapping
mdsol:SignaturePrompt	text (maximum 2000 characters)	Optional	Optional text for string displayed at signature field.

Contained In

Study

Example:

```
...
<MetaDataVersion mdsol:DeleteExisting="Yes" OID="MDV.INITIAL" Name="Draft1.0"
    mdsol:PrimaryFormOID="F.AE" mdsol:DefaultMatrixOID="F.AE">
...
</MetaDataVersion>
...
```

Protocol

Body

(StudyEventRef+)

Attribute

NONE

Contained In

MetaDataVersion

The protocol lists the kinds of study events that can occur within a specific study version. All clinical data must occur within one of these study events.

Example:

```
...
<Protocol>
    <StudyEventRef StudyEventOID="EVT00001" Mandatory="Yes" OrderNumber="1" />
</Protocol>
...
```

StudyEventRef

Body

EMPTY

Attribute	Value	Notes	Rave mapping
StudyEventOID	oidref (limited to maximum 50 characters)	Required	OID Reference to the StudyEventDef
OrderNumber	integer	Optional	Maps to Folder Order in Rave (see Note 7).
Mandatory	{ Yes No}	Required	Not mapped in Rave.

Contained In**Protocol**

The StudyEventRef is a reference to a StudyEventDef as it occurs within a specific study version. The list of StudyEventRefs identifies the types of study events that are allowed to occur within the study. The StudyEventRefs within a single MetaDataVersion must not have duplicate StudyEventOIDs nor duplicate OrderNumbers.

Example:

```
...
<Protocol>
    <StudyEventRef StudyEventOID="EVT00001" Mandatory="Yes" OrderNumber="1" />
    <StudyEventRef StudyEventOID="EVT00002" Mandatory="Yes" OrderNumber="2" />
</Protocol>
...
```

StudyEventDef

Body

(FormRef+)

Attribute	Value	Notes	Rave mapping
OID	oid (limited to maximum 50 characters)	Required	Maps to Folder OID in Rave. See Note 8
Name	name	Required	Maps to Folder Name in Rave. See Note 9
Repeating	{Yes No(default) }	Required	Maps to Folder Reusable in Rave. Yes means Reusable is unchecked in Rave. No means Reusable is checked in Rave.
Type	{Scheduled Unscheduled Common}	Required	Not mapped in Rave.
Category	text (limited to maximum 255 characters)	Optional	Not mapped in Rave.
mdsol:AccessDays	integer	Optional	Maps to Folder Access field in Rave.
mdsol:StartWinDays	integer	Optional	Maps to Folder Start field in Rave.
mdsol:TargetDays	integer	Optional	Maps to Folder Target field in Rave.
mdsol:EndWinDays	integer	Optional	Maps to Folder End field in Rave.
mdsol:OverDueDays	integer	Optional	Maps to Folder Overdue field in Rave.

Attribute	Value	Notes	Rave mapping
mdsol:CloseDays	integer	Optional	Maps to Folder Close field in Rave.

Contained In

MetaDataVersion

A StudyEventDef contains a set of forms. Each StudyEventDef maps to a column in the default matrix in Rave.

Example:

```
...
<StudyEventDef mdsol:AccessDays="20" mdsol:StartWinDays="10" mdsol:TargetDays="5"
    mdsol:OverDueDays="10" mdsol:CloseDays="20" OID="SE.SCR" Name="Screening Visit"
    Repeating="No" Type="Scheduled" Category="Screening">
    <FormRef FormOID="FRM00001" OrderNumber="1" Mandatory="Yes" />
    <FormRef FormOID="FRM00002" OrderNumber="2" Mandatory="Yes" />
</StudyEventDef>
...
```

FormRef

Body

(FormRef+)

Attribute	Value	Notes	Rave mapping
FormOID	oidref (limited to maximum of 50 characters)	Required	Reference to the FormDef.
OrderNumber	integer	Optional	Not mapped in Rave.
Mandatory	{ Yes No }	Required	Not mapped in Rave.

Contained In**StudyEventDef**

A FormRef represents a reference to a FormDef as it occurs within a specific StudyEventDef. The list of FormRefs identifies the types of forms that are allowed to occur within this type of study event. Where a FormRef exists within a StudyEventDef, the corresponding cell in the default matrix is marked with an X - unless there is no corresponding StudyEventRef in the protocol.

Example:

```
...
<StudyEventDef OID="EVT00001" Name="Visit 1" Repeating="No" Type="Scheduled"
Category="">
    <FormRef FormOID="FRM00001" OrderNumber="1" Mandatory="Yes" />
    <FormRef FormOID="FRM00002" OrderNumber="2" Mandatory="Yes" />
</StudyEventDef>
...
```

FormDef

Attribute	Value	Notes	Rave mapping
OID	oid (limited to maximum of 50 characters)	Required	Maps to Form OID in Rave. This may be used as a PrimaryFormOID (see MetaDataTableVersion).
Name	name (limited to OID + (Name) 254 characters maximum length)	Required	Maps to Form Name in Rave. See Note 10.
Repeating	{Yes(default) No }	Optional	Not mapped in Rave.
mdsol:OrderNumber	integer	Optional	Maps to Form ordinal positioning in Rave. See Note 11 .
mdsol:Active	{Yes(default) No }	Optional	Yes means Active is checked for this Form. No means Active is unchecked.
mdsol:Template	{Yes No(default) }	Optional	Yes means Template is checked (this is a template form). No means Template is unchecked (this is a standard form).
mdsol:SignatureRequired	{Yes No(default) }	Optional	Yes means Signature Required is checked for this form. No means Signature Required is unchecked.
mdsol:LogDirection	{Portrait(default) Landscape}	Optional	Portrait = Log Direction is Portrait for this form. Landscape = Log Direction

Attribute	Value	Notes	Rave mapping
			is Landscape for this form. See Note 12 .
mdsol:DoubleDataEntry	{MustNotDDE (default MayDDE MustDDE)}	Optional	Maps to the Forms DDE field in Rave. MustNotDDE = Never, MayDDE = As per Site and MustDDE = Always.
mdsol:ConfirmationStyle	{LinkNext NoLink (default) LinkCustom None}	Optional	Maps to Forms Redirect in Rave. See Note 13 .
mdsol:LinkStudyEventOID	oid	Optional	Maps to the Folder dropdown list in Rave when Redirect = Link Custom.
mdsol:LinkFormOID	oid	Optional	Maps to the Form dropdown list in Rave when Redirect = Link Custom.

Contained In

MetaDataVersion

Example:

```
<FormDef OID="F.CONSENT" Name="Informed Consent" Repeating="No" mdsol:OrderNumber="1"
mdsol:Active="Yes" mdsol:Template="Yes" mdsol:SignatureRequired="Yes"
mdsol:LogDirection="Portrait" mdsol:DoubleDataEntry="MustDDE"
mdsol:ConfirmationStyle="NoLink" mdsol:LinkStudyEventOID="SE.2"
mdsol:LinkFormOID="F.AE">
    <ItemGroupRef ItemGroupOID="ITG00001" Mandatory="Yes" />
</FormDef>
```

Each FormDef element maps to a row in the default matrix in Rave.

Example:

```
...
<FormDef OID="F.CONSENT" Name="Informed Consent" Repeating="No" mdsol:OrderNumber="1"
mdsol:Active="Yes" mdsol:Template="Yes" mdsol:SignatureRequired="Yes"
mdsol:LogDirection="Portrait"
    mdsol:DoubleDataEntry="MustDDE" mdsol:ConfirmationStyle="NoLink"
    mdsol:LinkStudyEventOID="SE.2"
    mdsol:LinkFormOID="F.AE">
        <ItemGroupRef ItemGroupOID="ITG00001" Mandatory="Yes" />
</FormDef>
...
```

ItemGroupRef

ItemGroupRef represents a reference to an ItemGroupDef as it occurs within a specific FormDef. The list of ItemGroupRefs identifies the types of item groups that are allowed to occur within this type of form. The ItemGroupRefs within a single FormDef must not have duplicate ItemGroupIDs or OrderNumbers.

Body

EMPTY

Attribute	Value	Notes	Rave mapping
ItemGroupOID	oidref	Required	Reference to the ItemGroupDef
OrderNumber	integer	Optional	Not mapped in Rave.
Mandatory	{Yes No}	Required	Not mapped in Rave.

Contained In**FormDef****Example:**

```
...
<FormDef >
    <ItemGroupRef ItemGroupOID="IG.ADDITIONAL_CATS" Mandatory="Yes" OrderNumber="1"/>
</FormDef>
...
```

mdsol:HelpText**Body****EMPTY**

Attribute	Value	Notes	Rave mapping
xml:lang	languageTag	Optional	Not mapped in Rave.

Contained In

FormDef, ItemDef

The mdsol:HelpText element is optional and can appear as a child element of FormDef and ItemDef. The mdsol:HelpText element maps to Help Text in Rave. The content of mdsol:HelpText is limited to maximum 4000 characters.

 **Important:** The sequence of child elements of ItemDef and FormDef is significant. The child elements Question, mdsol:HeaderText and mdsol:HelpText must always appear in this order.

Example as child of ItemDef:

```
...
<ItemDef OID="SC.SCREENDATE" Name="SCREENDATE">
  ...
    <mdsol:HelpText>This should be the date of the screening visit, not the first
scheduled visit</mdsol:HelpText>
  ...
</ItemDef>
...
```

Example as child of FormDef:

```
...
<MetaDataVersion>
  <FormDef>
    <ItemGroupRef ItemGroupOID="IG.ADDITIONAL_CATS" Mandatory="Yes"
OrderNumber="1" />
    <mdsol:HelpText xml:lang="en">Form Help Text</mdsol:HelpText>
  </FormDef>
</MetaDataVersion>
...
```

mdsol:ViewRestriction

The mdsol:ViewRestriction element maps to Restrict View within Form Restrictions (under Architect/Restrictions) or Field View Restrictions in Rave (depending on the parent element in Rave).

The mdsol:ViewRestriction element is optional, and can appear as a child element of FormDef and ItemDef.

Body
EMPTY
Attribute
NONE
Contained In
FormDef, ItemDef

To enter more than one restriction, you may either use multiple instances of this element or use the element mdsol:ViewRestrictions with multiple values separated by commas.

Permitted values in the list of view restrictions are the EDC Roles defined in the core configuration for Rave. Other values are ignored.

Example with FormDef:

```
...
<MetaDataVersion>
    <FormDef>
        <ItemGroupRef ItemGroupOID="IG.ABSCESS" Mandatory="Yes" OrderNumber="1" />

        <mdsol:ViewRestriction>CDC </mdsol:ViewRestriction>
        <mdsol:ViewRestriction> PI</mdsol:ViewRestriction>
    </FormDef>
</MetaDataVersion>
...
```

Example with ItemDef:

```
...
<ItemDef>
    ...
    <mdsol:ViewRestriction>PI</mdsol:ViewRestriction>
    ...
</ItemDef>
...
```

mdsol:EntryRestriction

The mdsol:EntryRestriction is an optional element that can appear as a child element of FormDef and ItemDef. mdsol:EntryRestriction maps to Restrict Entry within the restrictions for the form (under Restrictions) in Rave. This element is optional. The mdsol:EntryRestriction element maps to Field Entry Restrictions in Rave. To enter more than one restriction, you may either use multiple instances of this element or use the element mdsol:EntryRestrictions with multiple values separated by commas.

Body
EMPTY

Attribute
NONE
Contained In
FormDef, ItemDef

Permitted values in the list of entry restrictions are the EDC Roles defined in the core configuration for Rave. Other values are ignored.

Example with FormDef:

```
...
<MetaDataVersion>
    <FormDef>
        <ItemGroupRef ItemGroupOID="IG.ABSCESS" Mandatory="Yes" OrderNumber="1" />
        <mdsol:EntryRestriction>CDC </mdsol:EntryRestriction>
        <mdsol:EntryRestriction> PI</mdsol:EntryRestriction>
    </FormDef>
</MetaDataVersion>
...
```

Example with ItemDef:

```
...
<ItemDef >
  ...
    <mdsol:EntryRestriction>PI</mdsol:EntryRestriction>
  ...
</ItemDef>
...
```

ItemGroupDef

Body

(ItemRef+, Alias*, mdsol:LabelRef*)

An ItemGroupDef describes a type of item group that can occur within a study. An ItemGroupDef must be referenced by an ItemGroupRef within a FormDef to be included in the study. The Alias element can have zero occurrences in ODM. The Alias element is not used by RWS, so an element definition for Alias is not included in this document.

Attribute	Value	Notes	Rave mapping
OID	oid	Required	Defines the ItemRef OID referred to by ItemGroupRef elements. Not imported into Rave.
Name	name	Required	Not mapped in Rave.
Repeating	{Yes No(default) }	Required	Yes means all fields within the ItemGroupDef are log line fields in Rave. No means all fields within the ItemGroupDef are standard fields.
IsReferenceData	{ Yes No }	Optional	Not mapped in Rave.
SASDatasetName	sasName	Optional	Not mapped in Rave.
Domain	text (255 characters max)	Optional	Not mapped in Rave.
Origin	text (255 characters max)	Optional	Not mapped in Rave.
Role	name	Optional	Not mapped in Rave.
Purpose	text (255 characters max)	Optional	Not mapped in Rave.
Comment	text	Optional	Not mapped in Rave.

Contained In**MetaDataVersion**

Within a FormDef, only the first appearing ItemGroupDef (referred by ItemGroupRef) with attribute Repeating="Yes" is imported. All the remaining ItemGroupDef elements with Repeating="Yes" are ignored. all ItemGroupDef elements with Repeating="No" are imported.

Example:

```
...
<ItemGroupDef OID="ITG00002" Domain="PA" Name="Chest X-Ray" Repeating="No">
  <ItemRef ItemOID="ITD00006" Mandatory="Yes" />
  <ItemRef ItemOID="ITD00007" Mandatory="Yes" />
</ItemGroupDef>
...
```

mdsol:LabelRef

mdsol:LabelDef allows labels to be imported and exported as independent items. From RWS 1.4 onwards, the use of the mdsol:LabelRef and mdsol:LabelDef elements replace the mdsol:Label element, which is obsolete except for backwards compatibility. Requests that include both mdsol:Label and mdsol:LabelRef/ mdsol:LabelDef are not supported.

Body

NONE

Attribute	Value	Notes	Rave mapping
LabelOID	oid (Limited to a maximum of 50 characters)	Required	LabelOID
OrderNumber	integer	Required	Maps to Item order as listed in Rave Architect. See Note 15 .

Contained In

ItemGroupDef

Example:

```
...
<MetaDataVersion OID="1" Name="Draft 1">
    <FormDef OID="VS" Name="Vital Signs" Repeating="Yes">
        <ItemGroupRef ItemGroupOID="VS" Mandatory="Yes"/>
    </FormDef>

    <ItemGroupDef OID="VS" Name="VS" Repeating="No">
        <ItemRef ItemOID="VS.HEART" OrderNumber="2" Mandatory="No" />
        <mdsol:LabelRef LabelOID="VSLBL1" OrderNumber="1"/>
    </ItemGroupDef>

    <ItemDef OID="VS.HEART" Name="HeartRate" DataType="text" Length="10">
        <Question>
            <TranslatedText>Heart rate</TranslatedText>
        </Question>
    </ItemDef>

    <mdsol:LabelDef OID="VSLBL1" Name="HeartRateLabel">
        <TranslatedText>Please take resting heart rate</TranslatedText>
    </mdsol:LabelDef>
</MetaDataVersion>
...
```

ItemRef

ItemRef represents a reference to an ItemDef as it occurs within a specific ItemGroupDef. The list of ItemRefs identifies the types of items that are allowed to occur within this type of item group. The ItemRefs within a single ItemGroupDef must not have duplicate ItemOIDs or OrderNumbers.

Body

mdsol:Attribute

Attribute	Value	Notes	Rave mapping
ItemOID	oidref (limited to a maximum of 50 characters)	Required	Maps to Field OID and VarOID in Rave. Reference to the ItemDef. See Note 14 .
OrderNumber	integer	Optional	If present, maps to Item order as listed in Rave Architect. See Note 15 .
Mandatory	{Yes No(default) }	Required	Yes means Auto Query for Required data is checked in Rave. No means Auto Query for Required is unchecked in Rave.
KeySequence	integer	Optional	Not mapped in Rave.
ImputationMethodOID	oidref	Optional	Not mapped in Rave.
Role	name	Optional	Not mapped in Rave.
RoleCodeListOID	oidref	Optional	Not mapped in Rave.

Contained In**ItemGroupDef****Example:**

```
...
<ItemGroupDef OID="ITG00002" Domain="PA" Name="Chest X-Ray" Repeating="No">
    <ItemRef ItemOID="ITD00006" Mandatory="Yes" />
    <ItemRef ItemOID="ITD00007" Mandatory="Yes" />
</ItemGroupDef>
...
```

mdsol:Attribute

The mdsol:Attribute element allows you to import data items from an external system.

Body**EMPTY**

Attribute	Value	Notes	Rave mapping
Namespace	text	Identifies an external system.	Not mapped in Rave.
Name	text	Identifies data item from external system.	Not mapped in Rave.
Value	text	Value of the data item from the external system.	Not mapped in Rave.
TransactionType	{Insert Update Remove}		Not mapped in Rave.

Contained In

ItemGroupRef

Example:

```
<ItemGroupDef OID="DM" Name="DM" Repeating="No">
    <ItemRef ItemOID="DM.BRTHDAT" Mandatory="No" />
    <ItemRef ItemOID="AE.ACTIONCD1">
        <mdsol:Attribute Namespace="MyIntegration" Name="Ignore" Value="Yes" TransactionType="Insert"/>
    </ItemRef>
</ItemGroupDef>
```

ItemDef**Body**

(Question?, ExternalQuestion?, MeasurementUnitRef*, RangeCheck*, CodeListRef?, Role*, Alias*, mdsol:HeaderText?, mdsol:HelpText?, mdsol:ViewRestriction* (or mdsol:ViewRestrictions*), mdsol:EntryRestriction* (or mdsol:EntryRestrictions*), mdsol:ReviewGroup* (or mdsol:ReviewGroups*), mdsol:Label?)

The ExternalQuestion element is optional. The Role and Alias elements can have zero occurrences in ODM. None of these elements are used by RWS in the import process.

An ItemDef element describes a type of item that can occur within a study. This maps to a field in Rave. Only ItemDef elements referenced from an ItemGroupDef, which is referenced by a FormDef, is processed.

Attribute	Value	Notes	Rave mapping
OID	oid (limited to maximum 50 characters)	Required	Maps to Field OID and VarOID in Rave. See Note 16.
Name	Name (limited to OID + (Name) 254 characters)	Required	Maps to Field Name in Rave. See Note 16 and Note 17.
DataType	{text integer float date datetime time }	Required	Maps to Format and Control Type in Rave. text = \$ + Length text (for length > 50) = \$+Length integer = Length date = dd MMM yyyy time = HH:nn:ss.
Length	integer	Required by Rave for data types text, integer, and float (see DataType, above); Optional for other data types.	Maps to Format in Rave (See DataType above).
SignificantDigits	integer	Optional	Maps to Format Significant Digits (See DataType above).
SASFieldName	sasName	Optional	Not mapped in Rave.
SDSVarName	sasName	Optional	Not mapped in Rave.
Origin	text (255 characters max)	Optional	Not mapped in Rave.

Attribute	Value	Notes	Rave mapping
Comment	text (255 characters max)	Optional	Not mapped in Rave.
mdsol:Active	{Yes (default) No }	Optional	Maps to Active in Rave.
mdsol:ControlType	{CheckBox Text DateTime DropDownList SearchList RadioButton RadioButton (vertical) File Upload LongText Signature page Signature folder Signature subject Dynamic SearchList}	Optional	Maps to Control Type in Rave. See Note 18.
mdsol:AcceptableFileExtensions	text (255 characters max. If mdsol:ControlType = File Upload, should contain file types delimited by commas, for example: ,doc,xml,jpeg)	Optional	Maps to Accept files with extensions in Rave.
mdsol:IndentLevel	{0 (default) 1 2}	Optional	Maps to Indent level in Rave.
mdsol:SourceDocument	{Yes No (default) }	Optional	Maps to Requires Verification (under Field Verification and Reviews). Yes = checked, No = unchecked.
mdsol:DefaultValue	text (255 characters max)	Optional	Maps to Default Value in Rave.
mdsol:SASFormat	text (255 characters max)	Optional	Maps to SAS Format in Rave.

Attribute	Value	Notes	Rave mapping
mdsol:SASLabel	text (255 characters max)	Optional	Maps to SAS Label in Rave.
mdsol:QueryFutureDate	{Yes No (default) }	Optional	Maps to Auto-Query for future Date/time in Rave. Yes = checked, No = unchecked.
mdsol:Visible	{Yes(default) No }	Optional	Maps to Is visible in Rave. Yes = checked, No = unchecked.
mdsol:TranslationRequired	{Yes No (default) }	Optional	Maps to Requires Translation in Rave. Yes = checked, No = unchecked. See Note 19 .
mdsol:QueryNonConformance	{Yes No (default) }	Optional	Maps to Auto-Query for non-conformant data. Yes = checked, No = unchecked.
mdsol:OtherVisits	{Yes No (default) }	Optional	Maps to Show Previous Visit Values in Rave. Yes = checked, No = unchecked.
mdsol:CanSetItemGroupDate	{Yes No (default) }	Optional	Maps to Can Set Record Date in Rave. Yes = checked, No = unchecked.

Attribute	Value	Notes	Rave mapping
mdsol:CanSetFormDate	{Yes No (default) }	Optional	Maps to Can Set DataPage Date in Rave. Yes = checked, No = unchecked.
mdsol:CanSetStudyEventDate	{Yes No (default) }	Optional	Maps to Can Set Instance Date in Rave. Yes = checked, No = unchecked.
mdsol:CanSetSubjectDate	{Yes No (default) }	Optional	Maps to Can Set Subject Date in Rave. Yes = checked, No = unchecked.
mdsol:VisualVerify	{Yes No (default) }	Optional	Maps to Visual Verify in Second Pass in Rave. Yes = checked, No = unchecked. Only appears on fields of DDE forms.
mdsol:DoesNotBreakSignature	{Yes No (default) }	Optional	Maps to Does not participate in Signature in Rave. Yes = checked, No = unchecked.
mdsol:DateTimeFormat	Any valid Rave date-time variable format	Optional	Maps to the field's variable format. See Note 20 .
mdsol:FieldNumber	text (maximum 50 characters)	Optional	Maps to Num in Rave. See Note 21 .

Attribute	Value	Notes	Rave mapping
mdsol:VariableOID	oid (limited to 50 characters)	Optional	Maps to Variable OID in Rave. If not present, the Field OID maps to the Variable OID in Rave.

Contained In

MetaDataVersion

Example:

```
...
<FormDef OID="VS" Name="Vital Signs" Repeating="Yes ">
    <ItemGroupRef ItemGroupOID="VS" Mandatory="Yes" />
</FormDef>
<ItemGroupDef OID="VS" Name="VS" Repeating="No">
    <ItemRef ItemOID="VS.WEIGHT" Mandatory="No" />
    <ItemRef ItemOID="VS.VSDate" Mandatory="No" />
</ItemGroupDef>
<ItemDef OID="VS.WEIGHT" Name="WEIGHT" DataType="float" Length="3"
SignificantDigits="3" mdsol:ControlType="Text" />
<ItemDef OID="VS.VSDate" Name="Date of Vitals" DataType="date-time"
mdsol:DateTimeFormat="dd MM yy" mdsol:ControlType="DateTime" />
<ItemDef OID="VS.HEART" Name="HeartRate" DataType="text" Length="10"
mdsol:FieldNumber="1"/>
...
```

Question**Body**

(TranslatedText+)

The TranslatedText element within this element maps to the field Label in Rave. Question is a human-readable label used to name an item on paper or on a screen.

Attribute

NONE

Contained In

ItemDef

Example:

```
...
<ItemDef >
  ...
  <Question>
    <TranslatedText>Ethnicity</TranslatedText>
    <TranslatedText xml:lang="fr">Appartenance ethnique</TranslatedText>
    <TranslatedText xml:lang="de">Volksgruppe</TranslatedText>
  </Question>
  ...
</ItemDef>
...
```

MeasurementUnitRef**Body**

EMPTY

Attribute	Value	Notes	Rave mapping
MeasurementUnitOID	oidref	Required	Maps to a unit dictionary entry in Rave. If only one MeasurementUnitRef exists, maps to Fixed Unit in Rave. This is a reference to a MeasurementUnit definition. See Note 4 .
mdsol:OrderNumber	integer	Optional	Maps to the position of the unit dictionary entry within the list, that is, the unit dictionary ordinal position in Rave. See Note 22 .

Contained In**ItemData, ItemDef**

MeasurementUnitRef maps to unit dictionary entries in Rave.

Example:

```
...
<ItemDef >
  ...
    <MeasurementUnitRef MeasurementUnitOID="MU.ML" mdsol:OrderNumber="1"/>
  ...
</ItemDef>
...
```



Note: For multiple MeasurementUnitRef elements, the value within the attribute mdsol:UnitDictionaryName of the MeasurementUnit associated with the first referenced MeasurementUnitRef is assigned as the unit dictionary in Rave.

Example:

```
...
<MeasurementUnit OID="MU_INCHES" Name="Inches" mdsol:UnitDictionaryName="Height">
  <Symbol>
    <TranslatedText xml:lang="en">in</TranslatedText>
  </Symbol>
</MeasurementUnit>
<MeasurementUnit OID="MU_METRES" Name="Metres" mdsol:UnitDictionaryName="Metres">
  <Symbol>
    <TranslatedText xml:lang="en">mt</TranslatedText>
  </Symbol>
</MeasurementUnit>
...
<ItemDef OID="I_HEIGHT">
  ...
    <MeasurementUnitRef MeasurementUnitOID="MU_INCHES"/>
    <MeasurementUnitRef MeasurementUnitOID="MU_METRES"/>
  ...
</ItemDef>
...
```

For multiple MeasurementUnitRef elements in which the first instance has an unqualified MeasurementUnitOID (and it has no mdsol:UnitDictionaryName attribute), the parent ItemDef OID attribute value is used as a unit dictionary name for a unit dictionary specific to the field.

Example:

```
...
<BasicDefinitions>
    <MeasurementUnit OID="MSU001" Name="m">
        ...
    </MeasurementUnit>
    <MeasurementUnit OID="MSU002" Name="cm">
        ...
    </MeasurementUnit>
</BasicDefinitions>
...
<ItemDef OID="IT001">
    ...
    <MeasurementUnitRef MeasurementUnitOID="MSU001">
        <MeasurementUnitRef MeasurementUnitOID="MSU002">
            ...
    </ItemDef>
    ...

```

If an ItemDef contains only one unqualified MeasurementUnitRef (and the referenced MeasurementUnit has no mdsol:UnitDictionaryName attribute), the first TranslatedText element value under the referenced MeasurementUnit is used as the Fixed Unit in Rave.

Example:

```
...
<MeasurementUnit OID="MU_MMHG" Name="mm Hg">
    <Symbol>
        <TranslatedText xml:lang="en">mm Hg</TranslatedText>
        <TranslatedText xml:lang="fr">mm Hg</TranslatedText>
        <TranslatedText xml:lang="de">mm Hg</TranslatedText>
    </Symbol>
</MeasurementUnit>
...
<ItemDef >
    ...
    <MeasurementUnitRef MeasurementUnitOID="MU_MMHG"/>
    ...
</ItemDef>
...
```

When an ItemDef contains multiple MeasurementUnitRefs with custom attribute StandardUnit on more than one MeasurementUnit, then the first referenced MeasurementUnit with StandardUnit="Yes" is used as standard for the unit dictionary.

The TranslatedText element contents for this MeasurementUnit is used as the standard unit name.

Example:

```
...
<MeasurementUnit OID="MU_INCHES" Name="Inches"...mdsol:StandardUnit="No">
    <Symbol>
        <TranslatedText xml:lang="en">in</TranslatedText>
        <TranslatedText xml:lang="fr">pouce</TranslatedText>
        <TranslatedText xml:lang="de">Zoll</TranslatedText>
    </Symbol>
</MeasurementUnit>
<MeasurementUnit OID="MU_CENTIMETRES" Name="Centimetres" mdsol:StandardUnit="Yes">
    <Symbol>
        <TranslatedText xml:lang="en">cm</TranslatedText>
        <TranslatedText xml:lang="fr">cm</TranslatedText>
        <TranslatedText xml:lang="de">cm</TranslatedText>
    </Symbol>
</MeasurementUnit>
...
<ItemDef >
    ...
    <MeasurementUnitRef MeasurementUnitOID="MU_INCHES"/>
    <MeasurementUnitRef MeasurementUnitOID="MU_CENTIMETRES"/>
    ...
</ItemDef>
...

```

When an ItemDef contains multiple MeasurementUnitRef elements without the custom attribute mdsol:StandardUnit="Yes" on the associated MeasurementUnit, then the TranslatedText element value for the MeasurementUnit (referenced by MeasurementUnitRef) with constant A=1, constant B=1, constant C=0 and constant K=0 becomes the standard unit.

In all other cases, the TranslatedText for the first referenced MeasurementUnit is used as the standard unit for the unit dictionary.

Example:

```
...
<MeasurementUnit OID="MU_INCHES" Name="Inches" mdsol:ConstantA="1"
```

```
mdsol:ConstantB="1" mdsol:ConstantC="0" mdsol:ConstantK="0">
  <Symbol>
    <TranslatedText xml:lang="en">in</TranslatedText>
    <TranslatedText xml:lang="fr">pouce</TranslatedText>
    <TranslatedText xml:lang="de">Zoll</TranslatedText>
  </Symbol>
</MeasurementUnit>
<MeasurementUnit OID="MU_CENTIMETRES" Name="Centimetres">
  <Symbol>
    <TranslatedText xml:lang="en">cm</TranslatedText>
    <TranslatedText xml:lang="fr">cm</TranslatedText>
    <TranslatedText xml:lang="de">cm</TranslatedText>
  </Symbol>
</MeasurementUnit>
...
<ItemDef>
  ...
  <MeasurementUnitRef MeasurementUnitOID="MU_INCHES"/>
  <MeasurementUnitRef MeasurementUnitOID="MU_CENTIMETRES"/>
  ...
</ItemDef>
...
```

RangeCheck

Body

(CheckValue+, MeasurementUnitRef?, ErrorMessage?)

The ErrorMessage element is optional in ODM and is not used by RWS.

Attribute	Value	Notes	Rave mapping
Comparator	{LT LE GT GE EQ NE IN NOTIN}	Required	Only LE and GE are supported by RWS. Both Comparator and SoftHard map to the low/high range values for Auto-Query for data out of range or Mark non-conformant data out of range in Rave. See Note 23.
SoftHard	{Soft Hard}	Required	SoftHard and Comparator map to the low/high range values for Auto-Query for data out of range or Mark non-conformant data out of range in Rave.

Contained In**ItemDef**

A RangeCheck defines a constraint on the value of the enclosing item. RangeCheck maps to the Low and High range values of Field Edit Checks. If an ItemDef that contains at least one MeasurementUnitRef also contains RangeChecks, those RangeChecks should themselves contain MeasurementUnitRefs to indicate which unit is being used for those ranges. RWS uses the MeasurementUnitRef in a RangeCheck to convert the values in the range to the standard unit.

Example:

```
...
<ItemDef>
  ...
  <Question>
    <TranslatedText xml:lang="en">Weight</TranslatedText>
  </Question>
  <RangeCheck Comparator="GE" SoftHard="Hard">
    <CheckValue>40</CheckValue>
    <MeasurementUnitRef MeasurementUnitOID="MU_KILOGRAMMES"/>
  </RangeCheck>
  <RangeCheck Comparator="LE" SoftHard="Hard">
    <CheckValue>100</CheckValue>
    <MeasurementUnitRef MeasurementUnitOID="MU_KILOGRAMMES"/>
  </RangeCheck>
  ...
</ItemDef>
...
```

CheckValue**Body**

value

Attribute

NONE

Contained In**RangeCheck**

CheckValue is a comparison value used in a range check. CheckValue maps to Low or High range values in the range. See the RangeCheck section for more information. This value is used in conjunction with the mdsol:ConstantA - K attributes on the MeasurementUnit element for the calculation of Low and High range values of Field Edit Checks: Auto_Query for data out of range and Mark non-conformant data out of range.

CheckValue is used in the following calculation to convert the CheckValue to a standard measurement unit if the current ItemDef contains a MeasurementUnitRef that has a corresponding MeasurementUnit with the mdsol:ConstantA - K attributes present.

Standard Unit = mdsol:ConstantA/mdsol:ConstantB * (CheckValue + mdsol:ConstantC) + mdsol:ConstantK

If there are no corresponding MeasurementUnitRef elements, CheckValue is used as the High or Low range value as applicable.

Example:

```
<ItemDef>
  ...
  <Question>
    <TranslatedText xml:lang="en">Weight</TranslatedText>
  </Question>
  <RangeCheck Comparator="GE" SoftHard="Hard">
    <CheckValue>0</CheckValue>
  </RangeCheck>
  <RangeCheck Comparator="LE" SoftHard="Hard">
    <CheckValue>500</CheckValue>
  </RangeCheck>
  ...
</ItemDef>
...
```

CodeListRef**Body**

EMPTY

Attribute	Value	Notes	Rave mapping
CodeListOID	oidref (255 characters max)	Required	Maps to a data dictionary Name in Rave. Reference to the CodeList definition. See Note 24 and Note 25 .

Contained In**ItemDef****Example:**

```
...
<ItemDef >
    ...
    <Question>
        <TranslatedText>Was this clinically significant?</TranslatedText>
    </Question>
    <CodeListRef CodeListOID="YES_NO" />
    ...
</ItemDef>
...
```

mdsol:HeaderText

The mdsol:HeaderText element is optional and maps to Fields Header Text in Rave.

Body**EMPTY**

Attribute	Value	Notes	Rave mapping
xml:lang	languageTag	Optional	Not mapped in Rave.

Contained In

ItemDef

Example:

```
<ItemDef >
  ...
  <mdsol:HeaderText>Only 2 initials are permissible</mdsol:HeaderText>
  ...
</ItemDef>
...
```

mdsol:ReviewGroup

The mdsol:ReviewGroups element is optional and maps to Required Manual Reviews (under Verification and Reviews) in Rave.

Body

EMPTY

Attribute

NONE

Contained In

ItemDef

Example:

```
...
<ItemDef >
  ...
  <mdsol:ReviewGroup>CDC</mdsol:ReviewGroup>
  ...
  <mdsol:ReviewGroup>PI</mdsol:ReviewGroup>
  ...
</ItemDef>
...
```

mdsol:Label

The element mdsol:Label is optional and maps to Field Label in Rave for fields that are not associated with a variable (that is, label fields). RWS does not support forms that only contain labels. mdsol:Label element is obsolete from RWS 1.4 and supported only for backwards compatibility. Use mdsol:LabelRef and mdsol:LabelDef. Requests that include both mdsol:Label and mdsol:LabelRef/mdsol:LabelDef are not supported.

Body

EMPTY

Attribute	Value	Notes	Rave mapping
Position	{Above Below}	Required if this element is used.	Specifies the relative position of the custom element mdsol:Label to be displayed above or below the parent ItemDef. See Note 26 .
xml:lang	languageTag	Optional	Not mapped in Rave.

Contained In**ItemDef****Example:**

```
...
<ItemDef name="Weight">
  ...
    <mdsol:Label Position="Above">Please enter the patient weight:</mdsol:Label>
  ...
</ItemDef>
...
```

A new field Label is created in Rave above Weight (that is, Label + Position attribute value + ItemDef Name attribute) that is inserted above the Weight field for the form in Architect. The Label field value is "Please enter the patient weight:". For metadata export, all field labels are included in the ItemDef for the variable field they refer to, and are given the Position attribute of Above - unless the label is the last item in a form, in which case it is given the Position attribute of Below.

mdsol:LabelDef

mdsol:LabelDef is used with mdsol:LabelRef to allow labels to be imported and exported as independent items. From RWS 1.4, the use mdsol:LabelRef and mdsol:LabelDef replaces the mdsol:Label element, which is obsolete except for backwards compatibility. Requests that include both mdsol:Label and mdsol:LabelRef or mdsol:LabelDef are not supported.

Body

TranslatedText*, mdsol:ViewRestriction*

Attribute	Value	Notes	Rave mapping
OID	oid (Limited to a maximum of 50 characters)	Required	OID
Name	text	Required	Maps to the field's name label in Rave.
FieldNumber	text (maximum 50 characters)	Optional	Maps to Num field in Rave. See Note 21 .

Contained In

MetaDataVersion

Example:

```
...
<MetaDataVersion OID="1" Name="Draft 1">
    <FormDef OID="VS" Name="Vital Signs" Repeating="Yes">
        <ItemGroupRef ItemGroupOID="VS" Mandatory="Yes"/>
    </FormDef>

    <ItemGroupDef OID="VS" Name="VS" Repeating="No">
        <ItemRef ItemOID="VS.HEART" OrderNumber="2" Mandatory="No" />
        <mdsol:LabelRef LabelOID="VSLBL1" OrderNumber="1"/>
    </ItemGroupDef>

    <ItemDef OID="VS.HEART" Name="HeartRate" DataType="text" Length="10">
        <Question>
            <TranslatedText>Heart rate</TranslatedText>
        </Question>
    </ItemDef>

    <mdsol:LabelDef OID="VSLBL1" Name="HeartRateLabel" FieldNumber="1">
        <TranslatedText>Please take resting heart rate</TranslatedText>
    </mdsol:LabelDef>
</MetaDataVersion>
...
```

CodeList**Body**

(CodeListItem* | ExternalCodeList)

The ExternalCodeList element is not used by RWS. A CodeList defines a discrete set of permitted values for an item and maps to a data dictionary in Rave. The entries in a dictionary are defined with CodeListItems.

Attribute	Value	Notes	Rave mapping
OID	OID (255 characters max)	Required	Maps to data dictionary Name in Rave. See Note 27.
Name	name	Required	Not mapped in Rave.
DataType	{integer float text string }	Required	Not mapped in Rave. See Note 28.
SASFormatName	sasFormat	Optional	Not mapped in Rave.

Contained In

MetaDataVersion

Example:

```
...
<CodeList Name="GenderCodes" DataType="text" OID="CDL00001">
    <CodeListItem CodedValue="F">
        <Decode>
            <TranslatedText xml:lang="en">Female</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="M">
        <Decode>
            <TranslatedText xml:lang="en">Male</TranslatedText>
        </Decode>
    </CodeListItem>
</CodeList>
...
```

CodeListItem

Each CodeListItem element maps to a data dictionary entry within a data dictionary in Rave.

Body

(Decode)

Attribute	Value	Notes	Rave mapping
CodedValue	value	Required	Maps to the Coded Data value of the data dictionary entry in Rave.
mdsol:OrderNumber	integer	Optional	Maps to the position of the data dictionary entry within the list; that is, data dictionary ordinal positioning in Rave. See Note 29 .
mdsol:Specify	{Yes No (default) }	Optional	Maps to the data dictionary Specify field in Rave. Yes = checked, No = unchecked.

Contained In

(CodeList)

CodeListItem defines an individual member value of a CodeList including the display format. There can be more than one CodeListItem element in a CodeList element. Each one is imported into Rave as a data dictionary entry.

Example:

```
...
<CodeList Name="GenderCodes" DataType="text" OID="CDL00001">
    <CodeListItem CodedValue="F" mdsol:OrderNumber="1" mdsol:Specify="No">
        <Decode>
            <TranslatedText xml:lang="en ">Female</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="M" mdsol:OrderNumber="2" mdsol:Specify="No">
        <Decode>
            <TranslatedText xml:lang="en ">Male</TranslatedText>
        </Decode>
    </CodeListItem>
</CodeList>
...
```

Decode**Body**

(TranslatedText+)

The first instance of the ODM element TranslatedText under element Decode of CodeListItem maps to the User Data String value for the data dictionary entry in Rave. This value is imported into the default locale.

Attribute

NONE

Contained In

CodeListItem

Example:

```
...
<CodeList Name="GenderCodes" DataType="text" OID="CDL00001">
    <CodeListItem CodedValue="F" mdsol:OrderNumber="1" mdsol:Specify="No">
        <Decode>
            <TranslatedText xml:lang="en ">Female</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="M" mdsol:OrderNumber="2" mdsol:Specify="No">
        <Decode>
            <TranslatedText xml:lang="en ">Male</TranslatedText>
        </Decode>
    </CodeListItem>
</CodeList>
...
```

mdsol:ConfirmationMessage

The mdsol:ConfirmationMessage element is optional, maps to the Draft Message, and displays as the Confirmation Message in the CRF Draft Settings section.

If mdsol:ConfirmationMessage is missing or empty, the default value of Data saved is used.

Body

EMPTY

Attribute	Value	Notes	Rave mapping
xml:lang	languageTag	Optional	Not mapped in Rave.

Contained In

MetaDataVersion

Example:

```
<MetaDataVersion>
...
    <mdsol:ConfirmationMessage>Form has been submitted.</mdsol:ConfirmationMessage>
...
</MetaDataVersion>
```

1.7.1.5 ODM Subject Data Schema Support

This article details the Subject related ODM Schema elements supported in RWS.

ClinicalData

RWS only supports a single ClinicalData element per transaction. RWS supports updates to active and inactive studies. If RWS receives clinical data for more than one study, it returns the error response message: 'Data not imported', as more than one study was identified in the document (RWS00013).

If RWS receives clinical data for

- a study that does not exist or
- a study that is not assigned to the user account and supplied by the import transaction,

then you receive the following error message: "Study does not exist" (RWS00014).

Body

SubjectData*

Attribute	Value	Notes	Rave mapping
StudyOID	oidref	Required. Study name / project name. See Note 30 .	Rave Project name (Environment).
mdsol:StudyOIDType	{Rave iMedidata StudyUUID}	Rave (default) indicates that the StudyOID is a Rave study name. StudyUUID (preferred) indicates the unique identifier for a study. iMedidata is the same as StudyUUID. Optional when using Rave. Required when using iMedidata and StudyUUID.	N/A
MetaDataVersionOID	oidref	Required. References the MetaDataVersion that governs the data nested within this element.	Not mapped in Rave.

Contained In

ODM

Example:

```
<ODM>
  <ClinicalData StudyOID="Mediflex (Dev)" MetaDataVersionOID="1">
    <SubjectData>
      ...
    </SubjectData>
  </ClinicalData>
</ODM>
```

Example with mdsol:StudyOIDType:

```
<ODM>
  <ClinicalData StudyOID="{study unique identifier}" mdsol:StudyOIDType="{Study OID Type}" MetaDataVersionOID="MDV1.0">
    <SubjectData>
      ...
    </SubjectData>
  </ClinicalData>
</ODM>
```

[Back to top](#)

SubjectData

SubjectData contains clinical data for a single subject. RWS supports one or more SubjectData elements in one transaction, but only one subject. If more than one SubjectData element is included in a transaction, the SubjectKey for each must be the same (see the example below). RWS will accept clinical data from a single subject containing one or more forms (data pages). If RWS receives a request to create a subject and there is a deactivated subject with the same name, it creates a new subject with that subject name.

The AuditRecord, Signature, and InvestigatorRef elements are optional. The Annotation element can have zero occurrences in ODM. RWS only supports AuditRecord. RWS does not use - and thus, ignores - all other elements.

Attribute	Value	Notes	Rave mapping
SubjectKey	SubjectKey	Required. Used to name a specific subject and is unique within the parent study.	Rave subject name. If mdsol:SecondarySubjectKey is 'Yes', SubjectKey maps to Secondary Subject Name in Rave.
mdsol:SubjectKeyType	{SubjectUUID SubjectName}	Optional when using SubjectName. Required when using SubjectUUID.	N/A
mdsol:SubjectName	Text	Ignored. Include mdsol:SubjectName when SubjectUUID is used to assist with ODM file readability.	Subject Name
TransactionType	{ Insert Update Remove Upsert Context }	Optional. May be Insert, Update, Remove, Upsert, or Context. See Note 31.	N/A
mdsol:SecondarySubjectKey	{Yes No}	Indicates whether the subject key is a secondary subject name.	N/A

Contained In
ClinicalData

Error response messages related to SubjectData

Error Condition	Error Response Message	Reason Code
RWS receives clinical data for more than one subject.	Data for more than one subject was found in the document.	RWS00020
RWS receives clinical data for an inactive subject.	Subject not active.	RWS00019
RWS receives clinical data matching more than one active subject within a site.	Transaction not processed as more than one subject has the same identifier at the given site.	RWS00021
RWS receives clinical data for creating a subject that already exists.	Subject already exists.	RWS00024
RWS receives clinical data for creating a subject without site details.	Subjects must have site details to be created.	RWS00025
RWS receives clinical data for a subject that does not exist.	Subject does not exist.	RWS00023
RWS receives clinical data to create a subject using a unique identifier.	Subject UUID is not authorized for this action.	RWS00171

Example:

```

<ODM>
  <ClinicalData>
    <SubjectData SubjectKey="123 ABC" mdsol:SubjectKeyType="SubjectUUID"
mdsol:SubjectName="AN 001" TransactionType="Update">
      ...
    </SubjectData>
  </ClinicalData>
</ODM>

```

```
<SubjectData SubjectKey="123 ABC" TransactionType="Update">
  ...
</SubjectData>
</ClinicalData>
</ODM>
```

[Back to top](#)

SiteRef

RWS supports zero or one SiteRef element per subject:

- SiteRef is required for certain actions in RWS, such as ODM POST requests.
- SiteRef is optional in ODM:SubjectData.
- Site Ref must be provided for any clinical data transactions.

SiteRef refers to the Rave Site Number, which you can use across multiple studies, and not to the Study Site Number, which you can use for a specific study. RWS uses the LocationOID attribute as a Rave site number to identify sites in Rave. In addition, RWS uses the mdsol:LocationOIDType attribute to locate a subject site by name or a unique identifier. This allows you to update an existing subject within a study and site.

Body

(LocationOID*, mdsol:LocationOIDType*, mdsol:SiteNumber*, mdsol:StudySiteName*, mdsol:SiteUUID)

Attribute	Value	Notes	Rave mapping
LocationOID	oidref	Required.	
mdsol:LocationOIDType	{SiteUUID SiteNumber}	Specifies if the LocationOID is a Rave site number or site unique identifier. If you do not provide the mdsol:LocationOIDType, RWS maps to the Rave site number by default. Optional when using SiteNumber. Required when using SiteUUID.	N/A
mdsol:SiteNumber	Text	Ignored and the GET request succeeds with or without it. You should include mdsol:SiteNumber when you use SiteUUID to assist with ODM file readability.	Site Number
mdsol:StudySiteName	Text	Ignored and the GET request succeeds with or without it.	N/A
mdsol:SiteUUID	Text	Ignored and the GET request succeeds with or without it.	N/A
mdsol:LocationOIDType	{SiteUUID SiteNumber StudyEnvSiteNumber}		StudySites.StudySiteNumber

Contained In
SubjectData

Error response messages related to SiteRef

Error Condition	Error Response Message	Reason Code
RWS receives clinical data for a site without a CRF version pushed to it.	Site has no CRF version assigned.	RWS00015
RWS receives clinical data for an inactive site.	Site not active.	RWS00016
RWS receives clinical data for a site that does not exist or a site that the user is not authorized to access.	Site does not exist.	RWS00017
RWS receives clinical data without site details for more than one active subject across all sites in a study.	Transaction not processed as more than one subject without site details identified in the response for this study.	RWS00022

Example:

```

<ODM>
  <ClinicalData>
    <SubjectData>
      <SiteRef LocationOID="4567" />
      ...
    </SubjectData>
  </ClinicalData>
</ODM>

```

[Back to top](#)

StudyEventData

Body

(AuditRecord?, Signature?, Annotation*, FormData*)

The AuditRecord and Signature elements are optional. The Annotation element can have zero occurrences in ODM. RWS only supports AuditRecord. RWS does not use (ignores) all other elements.

If RWS receives a request to create a folder (Instance) that is already deactivated, it reactivates it. You must separately reactivate any associated forms and fields. If RWS receives a request to import subject level forms for the study, RWS maps all the FormRefs under the StudyEventDef with StudyEventOID="SUBJECT" to Subject level form in Rave. That is, the respective forms must be checked under the subject column in the default matrix.

For StudyEventRef StudyEventOID="SUBJECT" OrderNumber="1", "OrderNumber" will be ignored.

Attribute	Value	Notes	Rave mapping
StudyEventOID	oidref	Required	Rave folder OID
StudyEventRepeatKey	repeatKey	Optional. Number for top level folders only; Rave folder path for nested folders (see Folder Navigation for Repeated Objects)	Rave folder (Instance) repeat number, or Rave folder path
TransactionType	{ Insert Update Remove Upsert Context }	Optional. May be Insert, Update, Remove, Upsert, or Context. See Note 33 .	N/A

Contained In
SubjectData

Error response messages related to StudyEventData

Error Condition	Error Response Message	Reason Code
RWS receives a request to create a folder (Instance), but the associated folder OID is not found.	Folder does not exist.	RWS00027
RWS receives a request to create a folder (Instance) that already exists.	Folder already exists.	RWS00029
RWS receives a request to create a folder (Instance) with a non-sequential repeat number.	Non-sequential or non-consecutive folders are not permitted.	RWS00030
RWS receives clinical data to update/remove a folder (Instance) that does not exist.	Folder not found.	RWS00028

Example:

```

<ODM>
  <ClinicalData>
    <SubjectData>
      <SiteRef/>
      <StudyEventData StudyEventOID="SCREEN">
        <FormData>
          ...
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
</ODM>

```

[Back to top](#)

FormData

The AuditRecord, Signature, and ArchiveLayoutRef elements are optional. The Annotation element can have zero occurrences in ODM. RWS only supports AuditRecord. RWS does not use (ignores) all other elements.

If you send a request to RWS to create a form (data page) that is already deactivated, it reactivates only the form (data page). If you send a request to RWS to remove a form, it deactivates the form, associated records and fields (data points). If you reactivate a form, RWS will not reactivate the fields. However, you can reactivate individual fields on a specific request. Until you reactivate fields, forms and fields are inactive or locked. If you send a request to RWS to create a record that is already deactivated, it reactivates it.

Attribute	Value	Notes	Rave mapping
FormOID	oidref	Required.	Rave form OID.
FormRepeatKey	repeatKey	Optional.	Rave form (data page) repeat number.
TransactionType	{ Insert Update Remove Upsert Context }	Optional. May be Insert, Update, Remove, Upsert or Context. See Note 34 and Note 35 .	N/A
mdsol:LaboratoryRef	text	Associates a form with a lab.	Rave Laboratory name.
mdsol:LaboratoryType	{Central Local}	Distinguishes between local lab and central lab.	Rave Laboratory type.

Contained In
StudyEventData

Error response messages related to FormData

Error Condition	Error Response Message	Reason Code
RWS receives clinical data for a form (data page) that is entry restricted for the user.	Form not authorized.	RWS00031
RWS receives a request to create a form that already exists.	Form already exists.	RWS00034
RWS receives a request to create a form with a non-sequential repeat number.	Non sequential or non consecutive forms are not permitted.	RWS00035
RWS receives clinical data for a locked form.	Form locked.	RWS00036
RWS receives a request to create a form but the associated form OID is not found.	Form does not exist.	RWS00033
RWS receives clinical data to update or remove a form that does not exist, is deactivated, or has viewing restrictions.	Form does not exist.	RWS00033

Example:

```
<ODM>
  <ClinicalData>
    <SubjectData>
      <SiteRef/>
    <StudyEventData>
      <FormData FormOID="DM">
        <ItemGroupData>
```

```
    ...
  </ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>
```

[Back to top](#)

ItemGroupData

Body

(AuditRecord?, Signature?, Annotation*, ItemData*, ItemDataDate*, ItemDataAny*)

The AuditRecord and Signature elements are optional. The Annotation element can have zero occurrences in ODM. RWS only supports AuditRecord. RWS does not use (ignores) all other elements.

Attribute	Value	Notes	Rave mapping
ItemGroupOID	oidref	Required. For log line items, RWS ignores ItemGroupOID. For non-log line items, ItemGroupOID must be identical to the FormOID.	N/A
ItemGroupRepeatKey	{repeatKey @context }	Optional. See Note 36 .	Rave log-line record number
TransactionType	{ Insert Update Remove Upsert Context }	Optional. May be Insert, Update, Remove, Upsert, or Context. See Note 37 and Note 38 .	N/A
mdsol:Submission	{WholeItemGroup SpecifiedItemsOnly}	Optional. May be WholeItemGroup or SpecifiedItemsOnly (default). See Note 39 . Only valid with a TransactionType of Insert or Update. Ignored if TransactionType is Upsert or Remove.	

Contained In
FormData, ReferenceData

RWS does not use ReferenceData in the import process.

Usage notes for ItemGroupData

- If an integer value is given for an Insert transaction, that value must be one greater than the number of log records that already exist on that form. If an integer value is given for an Update or Remove transaction, that value must be less than or equal to the number of log records that exist on that form. You can use LAST to insert a new Rave log record without specifying a specific integer value.
- If RWS receives clinical data that contains the ODM element ItemGroupData with the ItemGroupRepeatKey attribute, it imports it as a log line record into Rave and RWS ignores the ODM ItemGroupOID attribute.
- If RWS receives clinical data that contains ODM element ItemGroupData without the ItemGroupRepeatKey attribute, it imports it as a standard record into Rave and RWS ignores the ODM ItemGroupOID attribute.
- If RWS receives a request to create a record (log line record) that is already deactivated, it reactivates it.

Error response messages related to ItemGroupData

The following table illustrates the error conditions related to the ItemGroupData element.

Error Condition	Error Response Message	Reason Code
RWS receives clinical data for a locked record.	Record locked.	RWS00040
RWS receives clinical data for a record (log line record) that does not exist or a deactivated record.	Record does not exist.	RWS00037
RWS receives a request to create a record that already exists.	Record already exists.	RWS00038
RWS receives a request to create a record with a non sequential repeat number.	Non-sequential or non-consecutive records are not permitted.	RWS00039

Example:

```
<ODM>
  <ClinicalData>
    <SubjectData>
      <SiteRef/>
      <StudyEventData>
        <FormData>
          <ItemGroupData ItemGroupOID="DM">
            <ItemData>
              ...
              <ItemData/>
            </ItemGroupData>
          </FormData>
        </StudyEventData>
      </SubjectData>
    </ClinicalData>
</ODM>
```

[Back to top](#)

ItemData

Body

(AuditRecord?, Signature?, MeasurementUnitRef?, Annotation*, mdsol:Review, mdsol:Query, mdsol:CodingData, mdsol:TranslatedData)

The AuditRecord and Signature elements are optional and the Annotation element can have zero occurrences in ODM. RWS only supports AuditRecord. RWS does not use (ignores) all other elements.

Attribute	Value	Notes	Rave mapping
ItemOID	oidref	Required Values qualified by the Rave form OID are supported to ensure consistency with Rave Data Exporter.	or + "." +
TransactionType	{ Insert Update Remove Upsert Context }	Optional. May be Insert, (for new or re-activated field (data point) values), Update (for changed field values), Remove (for inactivated field values), Upsert, or Context. See Note 40 . The value Upsert can only be used if TransactionType for ItemGroupData is also Upsert.	N/A
Value	text	Optional. Only present where the field value is not a Remove transaction.	N/A
IsNull	"Yes"	Optional. Only Yes is accepted by RWS. Only used where the value attribute is not present. If the value attribute is present, the IsNull setting is ignored.	Rave field value (NULL).
mdsol:SpecifyValue	text	Used where RWS receives clinical data for a data point associated with a data dictionary that allows the option	

Attribute	Value	Notes	Rave mapping
		to specify an unlisted value (the Specify option is selected in Rave Architect). (See Error response messages related to ItemData.)	
mdsol:Lock	{Yes No}	Used when a request is received to lock or unlock a data point.(See Error response messages related to ItemData.)	Yes maps to lock checked in Rave. No maps to lock unchecked in Rave.
mdsol:Freeze	{Yes No}	Used when a request is received to freeze or unfreeze a data point. (See Error response messages related to ItemData.)	Yes maps to freeze checked in Rave. No maps to freeze unchecked in Rave.
mdsol:Verify	{Yes No}	Used when a request is received to verify or unverify a data point. (See Error response messages related to ItemData.)	Yes maps to verify checked in Rave. No maps to verify unchecked in Rave.

Contained In

ItemGroupData

RWS supports two types of ItemData elements, ItemDataDate and ItemDataAny, as an alternative to using the ItemData element with the Value attribute. See [ItemData\[Type\]](#) below.

**Note:**

- RWS records changes by using the first change code associated with the user's role in Rave, unless the user's role does not have a change code. Subsequently, no change code will be used. See [Change Codes](#) for more information.
- For fields (data points) associated with data dictionaries, the coded data value must be used, for example, "1" for male or "2" for female.
- RWS accepts data for a hidden field only if the field is made visible by an earlier import transaction, that is, if the field is unhidden by an edit check triggered by an earlier transaction.
- RWS accepts non-conformant values for fields.

Error response messages related to ItemData

Error Condition	Error Response Message	Reason Code
RWS receives clinical data for a field (data point) that does not exist, is deactivated or is view restricted.	Field does not exist.	RWS00041
RWS receives clinical data to update or remove a field that is entry restricted for the user.	Field not authorized.	RWS00042
RWS receives clinical data for a locked field.	Field locked.	RWS00044

Error Condition	Error Response Message	Reason Code
RWS receives a request to create a field that already exists.	Field already exists.	RWS00043
RWS receives clinical data for a hidden field.	Hidden field/ field not visible.	RWS00045
RWS receives clinical data for a file upload or picture field.	Picture/file upload field cannot be updated.	RWS00046
RWS receives clinical data for a frozen field.	Transaction on frozen field is not permitted.	RWS00048
RWS receives clinical data for a derived field.	Transaction on derived field is not permitted.	RWS00049
RWS receives clinical data for a data dictionary with the value not in the dictionary.	Data not in dictionary.	RWS00047
RWS receives clinical data with mdsol:SpecifyValue for a data point which is not associated with a data dictionary entry with Specify checked.	Dictionary item does not require specify value.	RWS00112
RWS receives a request to lock a data point when the user is not authorized to lock.	Lock not authorized.	RWS00057
RWS receives a request to unlock a data point when the user is not authorized to unlock.	Unlock not authorized.	RWS00058
RWS receives a request to freeze a data point when the user is not authorized to freeze.	Freeze not authorized.	RWS00059

Error Condition	Error Response Message	Reason Code
RWS receives a request to unfreeze a data point when the user is not authorized to unfreeze.	Unfreeze not authorized.	RWS00060
RWS receives a request to verify a data point when the user is not authorized to verify.	Verify not authorized.	RWS00065
RWS receives a request to unverify a data point when the user is not authorized to unverify.	Unverify not authorized.	RWS00066
RWS receives a request with ItemData for a data point where the TransactionType=Context, but the data point context value does not match the current value for the data point.	Datapoint context value does not match current value.	RWS00100

Example:

```

<ODM>
  <ClinicalData>
    <SubjectData>
      <SiteRef/>
      <StudyEventData>
        <FormData>
          <ItemGroupData>
            <ItemData ItemOID="BRTHDTG" Value="01-JAN-1980"/>
            <ItemData ItemOID="SEX" Value="MALE"/>
            <ItemData ItemOID="RACE" Value="3"/>
            <ItemData ItemOID="COUNTRY" Value="GBR"/>
            <ItemData ItemOID="DOB" IsNull="Yes"/>
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
</ODM>

```

[Back to top](#)

ItemData[Type]

Starting with version 1.3, the ODM specification supports two versions of the ItemData element, which is used for clinical data item values. The untyped form, ItemData, is supported by RWS and the details of this element and its attributes are listed above (see ItemData). ODM version 1.3 supports a range of typed forms of the ItemData element, and two of these are supported by RWS: ItemDataDate and ItemDataAny. You can use either the typed or untyped forms of ItemData within any ODM message. However, you should not mix them in the same message.

[Back to top](#)

ItemDataDate

ItemDataDate is the typed version of the ItemData element for date values. Use this element for date values that need to be converted automatically to conform with Rave EDC formats. All the attributes listed above for ItemData, except the Value attribute, can be used with ItemDataDate.

Example:

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" Description="support ItemDataDate"
CreationDateTime="2008-05-20T05:00:19" Granularity="SingleSubject"
FileType="Transactional" FileOID="f001" AsOfDateTime="2009-06-12T11:00:19"
ODMVersion="1.3">
    <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">
        <SubjectData SubjectKey="IDD01" TransactionType="Insert">
            <SiteRef LocationOID="12345" />
            <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1"
TransactionType="Update">
                <FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">
                    <ItemGroupData ItemGroupOID="DM" TransactionType="Insert">
                        <ItemDataDate ItemOID="BIRTHDTC"
TransactionType="Insert">1977-07-14</ItemDataDate>
                        <ItemDataAny ItemOID="SEX" TransactionType="Insert">FEMALE</
ItemDataAny>
                    </ItemGroupData>
                </FormData>
            </StudyEventData>
        </SubjectData>
    </ClinicalData>
</ODM>
```

[Back to top](#)

ItemDataAny

ItemDataAny is the typed version of the ItemData element for transmitting clinical data values that do not have the correct DataType for the corresponding ItemDef. All the attributes listed above for ItemData, except the Value attribute, can be used with ItemDataAny.

Example:

```
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" Description="support ItemDataDate"
CreationDateTime="2008-05-20T05:00:19" Granularity="SingleSubject"
FileType="Transactional" FileOID="ItemDataDate@xml"
AsOfDateTime="2009-06-12T11:00:19" ODMVersion="1.3">
  <ClinicalData StudyOID="Mediflex(Dev)" MetaDataVersionOID="1">
    <SubjectData SubjectKey="IDD01" TransactionType="Insert">
      <SiteRef LocationOID="12345" />
      <StudyEventData StudyEventOID="SCREEN" StudyEventRepeatKey="1"
TransactionType="Update">
        <FormData FormOID="DM" FormRepeatKey="1" TransactionType="Update">
          <ItemGroupData ItemGroupOID="DM" TransactionType="Insert">
            <ItemDataDate ItemOID="BIRTHDTC"
TransactionType="Insert">1977-07-14</ItemDataDate>
            <ItemDataAny ItemOID="SEX" TransactionType="Insert">FEMALE</
ItemDataAny>
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
</ODM>
```

[Back to top](#)

AuditRecord

An AuditRecord carries information pertaining to the creation, deletion, or modification of clinical data. This information includes who performed that action, where, when, and why that action was performed. When AuditRecord information is imported into Rave, it is identified with an External Audit Record prefix. AuditRecord is currently supported for ItemData only.

Body

(UserRef, LocationRef, DateTimeStamp, ReasonForChange?, SourceID?)

Attribute	Value	Notes	Rave mapping
EditPoint	{Monitoring DataManagement DBAudit}	Identifies the data processing phase in which the action occurred (optional).	N/A
UsedImputationMethod	{Yes No}	Indicates whether the action involved the use of a Method (optional).	N/A
ID	ID	Optional	N/A
mdsol:IncludeFileOID	{Yes No}	Indicates whether to include the ODM File OID in the audit readable text (optional). Default is 'No'.	N/A

[Back to top](#)

MeasurementUnitRef

Body

Empty

Attribute	Value	Notes	Rave mapping
MeasurementUnitOID	oidref	Required. Values qualified by the Rave unit dictionary name are supported to ensure consistency with Rave Data Exporter.	or + "." +

Contained In
ItemData, ItemDef, RangeCheck

ItemDef and RangeCheck are not used by RWS in the import process.

Error response messages related to MeasurementUnitRef

Error Condition	Error Response Message	Reason Code
RWS receives clinical data that refers to a unit dictionary that does not exist.	Unit Dictionary does not exist.	RWS00050
RWS receives clinical data that refers to a unit dictionary entry that does not exist.	Unit Dictionary entry does not exist.	RWS00051

Example:

```

<ODM>
  <ClinicalData>
    <SubjectData>
      <SiteRef/>
      <StudyEventData>
        <FormData>
          <ItemGroupData>
            <ItemData ItemOID="MU1">
              <MeasurementUnitRef MeasurementUnitOID="kg"/>
            </ItemData>
            <ItemData ItemOID="MU2">
              <MeasurementUnitRef MeasurementUnitOID="height.in"/>
            </ItemData>
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
</ODM>

```

[Back to top](#)

mdsol:Review

The mdsol:Review element is optional and maps to the Review check box of the Review Group specified in a Rave field.

Body

EMPTY

Attribute	Value	Notes	Rave mapping
Reviewed	{Yes No}	Action subject to user role having appropriate permissions in Rave.	Yes maps to Review checked in Rave. No maps to Review unchecked in Rave.
GroupName	text		Specifies the name of the Review Group.

Contained In
ItemData

Error response messages related to mdsol:Review

Error Condition	Error Response Message	Reason Code
RWS receives a request to review a data point. However, the user is not authorized to review the data point assigned to the review group.	Review not authorized.	RWS00061
RWS receives a request to unreview a data point. However, the user is not authorized to unreview the data point assigned to the review group.	Unreview not authorized.	RWS00062
RWS receives a request to review/unreview a data point with a non existing mdsol:GroupName attribute.	Review group name not found.	RWS00063
RWS receives a request to review or unreview a data point. However, the data point is not associated with the review group.	Field not associated with review group.	RWS00064
RWS receives a request with a duplicate review group name for a data point.	Duplicate review group name.	RWS00067

[Back to top](#)

mdsol:Query

The mdsol:Query element is optional and maps to a query opened on a data point (field) in Rave.

Body

Empty

Attribute	Value	Notes	Rave mapping
Value	text		Query text
Recipient	text		Marking group
QueryRepeatKey	RepeatKey	Used to uniquely identify the query for the data point.	Query repeat number (not available in Rave UI).
RequiresResponse	{Yes No}		Yes maps to response checked in Rave. No (default value) maps to response unchecked in Rave.
Response	text		Maps to query response in Rave.
Status	{Open Answered Cancelled Closed Forwarded}	All actions are subject to the user role having the necessary permission in Rave.	Open opens manual query. Answered answers the open query (uses mdsol:QueryRepeatKey). Cancelled cancels the open query (uses mdsol:QueryRepeatKey). Closed closes the open query (uses mdsol:QueryRepeatKey). Forwarded forwards the open query (uses mdsol:QueryRepeatKey) to the marking group specified in mdsol:Recipient.

Contained In

ItemData

[Back to top](#)

mdsol:CodingData

The mdsol:CodingData element is optional and allows you to code an item in Rave. It requires the TransactionType attribute of the parent ItemData element to be Context, so that it can match and check OIDs.

Body

mdsol:DictionaryLevelData

Attribute	Value	Notes	Rave mapping
CodingDictionaryOID	text		Coding dictionary name.
CodingDictionaryVersion	text		Coding dictionary version.
TransactionType	{Insert Update Upsert Remove Context}	Required	Insert inserts the underlying coding dictionary level data into Rave. Update updates the underlying coding dictionary level data in Rave. Upsert updates the underlying coding dictionary level data in Rave, or inserts it if no coding data previously existed. Remove removes the underlying coding dictionary level data from Rave. See Note 41 .

Contained In
ItemData

Error response messages related to mdsol:CodingData

Error Condition	Error Response Message	Reason Code
RWS receives a request with CodingData for a data point. However, the data point does not require coding.	Field does not require coding.	RWS00096
RWS receives a request with CodingData for a data point. However, the user is not authorized for coding.	User not authorized for coding.	RWS00097
RWS receives a request with CodingData for a data point, However, the coding dictionary is not associated with the data point.	Coding dictionary not associated with the field.	RWS00098
RWS receives a request with CodingData for a data point. However, the coding dictionary level or the coding dictionary column does not exist.	Coding dictionary level/column does not exist.	RWS00099
RWS receives a request with CodingData for a data point. However, the TransactionType for the parent ItemData element is not Context.	Coding can only be applied to the items with transaction type of Context.	RWS00166
RWS receives a request with CodingData for a data point where the ItemData TransactionType	Datapoint context value does not match current value.	RWS00100

Error Condition	Error Response Message	Reason Code
=Context. However, the data point context value does not match the current value for the data point.		
RWS receives a request with multiple CodingData elements for a data point.	Multiple coding on a field not authorized.	RWS00110
RWS receives a request to simultaneously code and translate a data point within a transaction.	Simultaneous coding and translating not supported.	RWS00113
RWS receives a request to code a data point that is already coded.	Coding value already exists for the field.	RWS00114

The following example illustrates the use of mdsol:CodingData and of mdsol:DictionaryLevelData. Note that the parent ItemData element has the TransactionType of Context.

```
<ItemData ItemOID="TESTMEDDRA" Value="HEADACHE" TransactionType="Context">
    <mdsol:CodingData CodingDictionaryOID="MedDRA" CodingDictionaryVersion="V11.1"
    TransactionType="Insert">
        <mdsol:DictionaryLevelData DictionaryLevelOID="System Organ Class"
        Value="10034567" Term="Testing 123" />
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Group Term"
        Value="10047635" Term="Testing Vitamin related disorders" />
        <mdsol:DictionaryLevelData DictionaryLevelOID="High-Level Term"
        Value="10020914" Term="Testing Hypervitaminoses NEC" />
        <mdsol:DictionaryLevelData DictionaryLevelOID="Preferred Term" Value="10020916"
        Term="Testing Hypervitaminosis A" />
        <mdsol:DictionaryLevelData DictionaryLevelOID="Low-Level Term" Value="10000033"
        Term="Testing A-hypervitaminosis" />
    </mdsol:CodingData>
</ItemData>
```

[Back to top](#)

mdsol:DictionaryLevelData

The mdsol:DictionaryLevelData element provides details of the coding dictionary level. Required when coding dictionary levels are used.

Body

EMPTY

Attribute	Value	Notes	Rave mapping
DictionaryLevelOID	text		Dictionary level name.
Value	text		Dictionary level value.
Term	text		Coding term.

Contained In

mdsol:CodingData

[Back to top](#)

mdsol:TranslatedData

The mdsol:TranslatedData is optional and allows RWS to translate a data point in Rave.

Body

EMPTY

Attribute	Value	Notes	Rave mapping
mdsol:Value	text		Translated text.
TransactionType	{Insert Update Remove Context}	Optional	Insert inserts the underlying translated data into Rave. Update updates the underlying translated data in Rave. Remove removes the underlying translated data from Rave. See Note 42 .

Contained In
ItemData

Error response messages related to mdsol:TranslatedData

Error Condition	Error Response Message	Reason Code
RWS receives a request with TranslatedData for a data point where the ItemData element has TransactionType=Context. However, the data point context value does not match the current value for the data point.	Datapoint context value does not match current value.	RWS00100
RWS receives a request to translate the data point value. However, the requesting user is not authorized to translate.	Translation not authorized.	RWS00086
RWS receives a request to translate the data point value. However, the item does not require translation.	Item does not require translation.	RWS00087
RWS receives a request for a user in default locale to update the data point requiring translation (that is, data point value inserted or updated in non-default locale).	Field requiring translation update in default locale not authorized.	RWS00090
RWS receives a request with multiple TranslatedData elements for a data point.	Multiple translation on a field not authorized.	RWS00115

[Back to top](#)

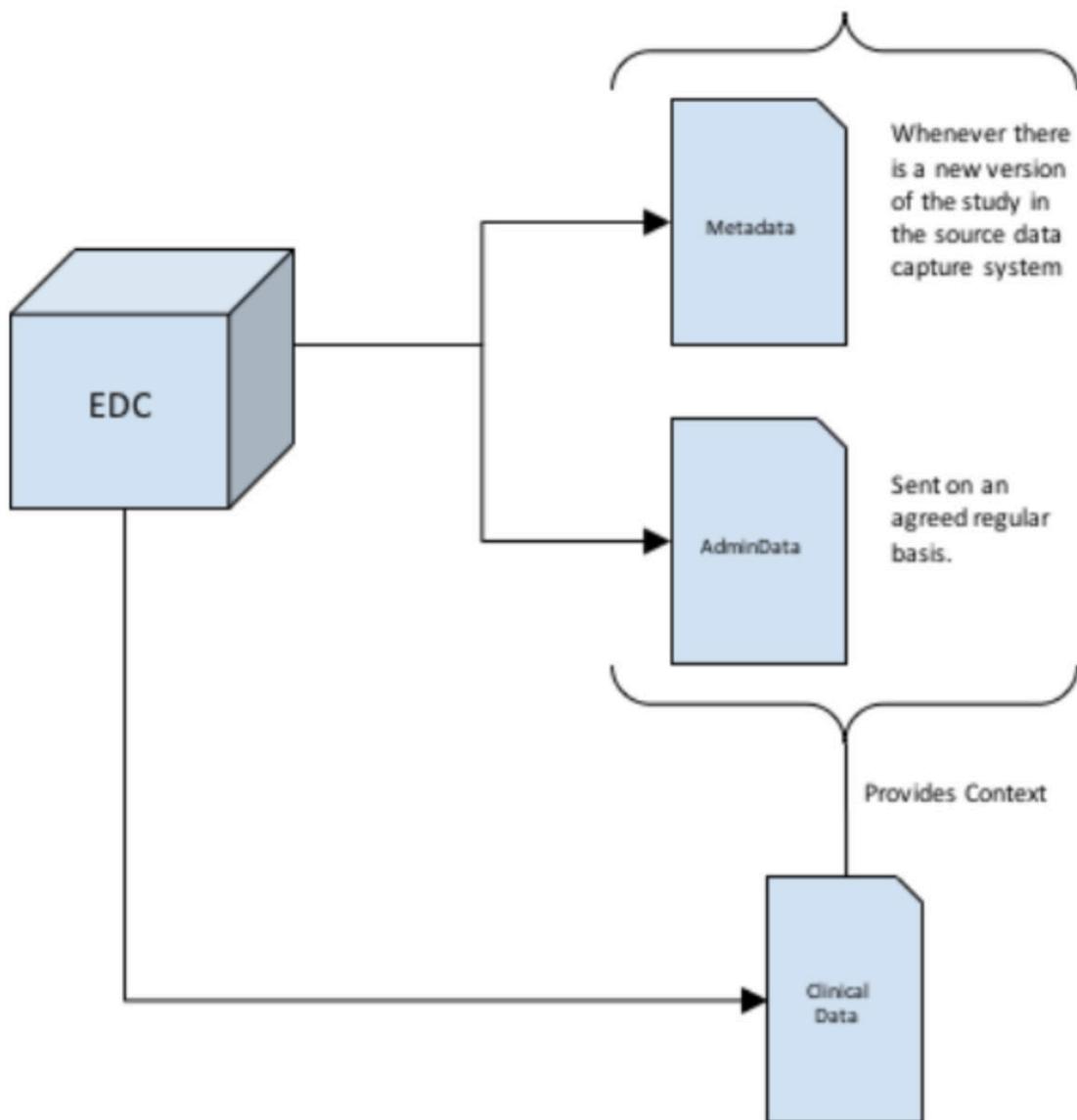
1.7.1.6 ODM Implementation

The CDISC Operational Data Model (ODM) has three main implementation scopes:

- Study Metadata - study designs,
- Administrative Data - including information about sites/locations and users, and
- Clinical Data - all entered data in the data capture system.

Given ODM's heritage as a Study Archive format, there are extra elements that Medidata is not using at this point.

The ODM workflow is illustrated as follows:



1.7.1.7 ODM Notes

The following table provides the contents of the notes referred to on the RWS [ODM Study Metadata Schema Support](#) and [ODM Subject Data Schema Support](#) articles.

Note Number	Comments
1	A coded unit must exist within unit dictionary entries in order to appear in the unit dictionary list. This is taken from the Name attribute of the MeasurementUnit element. These entries are ordered by mdsol:OrderNumber attribute on the corresponding MeasurementUnitRef element(s).
2	To create a custom unit dictionary, place the mdsol:UnitDictionaryName attribute on all relevant MeasurementUnit elements.
3	The mdsol:StandardUnit attribute indicates the unit dictionary entry which represents the standard unit for the unit dictionary. See MeasurementUnitRef section for more information on standard units.
4	If you use fixed units and you do not use the mdsol:UnitDictionaryName attribute, then the value is taken from the first translated text element in the MeasurementUnitRef/Symbol element defined in BasicDefinitions .
5	The value of the TranslatedText element when within the Symbol element could be used as the Fixed Unit in Rave. See the MeasurementUnitRef section for more information.
6	The Primary Form, identified by mdsol:PrimaryFormOID , is removed from the folder matrix since it gets used at the subject level automatically by Rave. If the mdsol:PrimaryFormOID custom extension is not present, the OID attribute of

Note Number	Comments
	the first FormDef occurrence within the MetaDataTable element is used as the Primary Form.
7	If no OrderNumber is present, folders are ordered in the same order as the ODM request.
8	The term SUBJECT is a reserved word for an OID to identify forms at the subject level.
9	Where Rave finds a duplicate value for the Name field, it automatically adds the OID in parentheses to the original name to create a unique name. For example, Name=Folder1, OID=FolderOIDABC, and Name=Folder1, OID=FolderOIDFGH become Name=Folder1 (FolderOIDABC) and Name=Folder1 (FolderOIDFGH) respectively. Any full stops or periods in the OID are replaced by underscores (_).
10	Where Rave finds a duplicate value for the Name field, it automatically adds the OID in parentheses to the original name to create a unique name. For example, Name=Form1, OID=FormOIDABC, and Name=Form1, OID=FormOIDFGH become Name=Form1 (FormOIDABC) and Name=Form1 (FormOIDFGH) respectively. Any full stops or periods in the OID are replaced by underscores (_).
11	If no mdsol:OrderNumber is specified, forms are ordered in the same order as the ODM document.
12	If the mdsol:LogDirection attribute does not exist, and if this is a form with log fields, then it defaults to Portrait (that is, if the referenced ItemGroupDef/OID has the attribute Repeating="Yes").

Note Number	Comments
13	If <code>mdsol:ConfirmationStyle = LinkCustom</code> , <code>mdsol:LinkStudyEventOID</code> should contain the unique identifier (StudyEvent OID) for the custom link folder and <code>mdsol:LinkFormOID</code> should contain the unique identifier (Form OID) for the custom link form.
14	When more than one <code>ItemGroupDef/ItemRef</code> refers to the same <code>ItemDef/ItemOID</code> , Rave Architect Loader returns an error. This is reported by RWS with the error response message: Architect Loader error RWS00091.
15	The <code>OrderNumber</code> attribute is combined across both <code>mdsol:LabelRef</code> and <code>ItemRef</code> elements. To display labels above fields on a form, specify a lower value for <code>OrderNumber</code> for the label than for the corresponding field. To display labels below fields on a form, specify a higher value for <code>OrderNumber</code> for the label than for the corresponding field. If no <code>OrderNumber</code> is specified, fields are ordered in the same order as the ODM message.
16	When more than one <code>ItemGroupDef/ItemRef</code> has the same <code>ItemDef/ItemOID</code> , Rave Architect Loader returns an error. This is reported by RWS with the error response message: Architect Loader error (RWS00091).
17	When more than one <code>ItemDef</code> has the same value in its <code>Name</code> attribute, Rave Architect Loader returns an error. This is reported by RWS with the error response message: Architect Loader error (RWS00091).
18	If <code>mdsol:ControlType</code> is not present or populated, and where a <code>CodeListRef</code> exists in this <code>ItemDef</code> , the Control Type is set to a default of <code>DropDownList</code> . Otherwise, the <code>DataType</code> attribute is checked and is set as follows:

Note Number	Comments
	If DataType = "text" and Length > 50, Control Type = "LongText". If DataType = "integer", "float" or "text", Control Type = "Text". If DataType = "date", "datetime" or "time", Control Type = "DateTime".
19	If the locale is not supported within XML standards, the field contains the content as it appears in Rave, but does not include the <code>xml:lang</code> attribute.
20	mdsol:DateTimeFormat attribute can use any Rave date-time format. This attribute is only valid when <code>DataType</code> attribute is either date, time, or date-time, and is ignored in all other cases. Default format for date: dd MMM yyyy Default format for time: HH:nn:ss Default format for date-time: dd MMM yyyy HH:nn:ss
21	mdsol:FieldNumber maps to the Field Num field in Rave. This is an optional, additional identifier and does not affect the position of fields on a form.
22	If no mdsol:OrderNumber is specified, unit dictionary entries are given in the same order as in the ODM message.
23	When the Comparator attribute is GE and the SoftHard attribute is Soft, the range check value represents the low range value of the Auto-Query for data out of range field option. When the Comparator attribute is LE and the SoftHard attribute is Soft, the range check value represents the high range value of the Auto-Query for data out of range field option. When the Comparator attribute is GE and the SoftHard attribute is Hard, the range check value represents the low range value of the Mark non-conformant data out of range field option. When the Comparator attribute is LE and the SoftHard attribute is Hard, the range check

Note Number	Comments
	value represents the high range value of the Mark non-conformant data out of range field option.
24	The DataType attributes of the referenced CodeList and the containing ItemDef must be the same. Where there is a CodeListRef element, the corresponding field in Rave is associated to the data dictionary created by the associated CodeList element.
25	If an ItemDef contains a CodeListRef element, the ItemDef element should not have an mdsol:ControlType attribute value of Checkbox. It can be one of three options: DropDownList, RadioButton or SearchList.
26	RWS ignores any mdsol:Label elements without Position attributes or with Position attributes containing any values other than Above or Below.
27	A CodeList value is also used to connect the data dictionary entries (that is, CodeListRef) with the data dictionary within Rave.
28	The DataType is not imported into Rave, but this value must be the same as the DataType attribute for ItemDef that contains the referenced CodeListRef.
29	If no mdsol:OrderNumber is specified, data dictionary entries are in the same order as they appear in the ODM message.
30	If localized strings exist for the environment name, consequently, the locale of the user account under which the transaction is being processed is used to determine the relevant environment name . If the environment is not specified, then the name of the

Note Number	Comments
	production environment for that project is used as a default. This is usually (but not always) PROD.
31	TransactionType is optional for SubjectData. If TransactionType is not specified, RWS uses the default of Update.
32	SiteRef is required even if it is optional according to ODM:SubjectData.
33	TransactionType is optional for StudyEventData. If TransactionType is not specified, RWS uses the TransactionType of the parent element. If the parent element does not specify the TransactionType, RWS uses a default of Update.
34	TransactionType is optional for FormData. If TransactionType is not specified, RWS uses the TransactionType of the parent element. If none is found, RWS uses a default of Update.
35	Upsert is supported only when FormRepeatKey is equal to @CONTEXT.
36	ItemGroupRepeatKey = @context is used when the RepeatKey value is unknown. This can only be used with TransactionType of Upsert or Context.
37	TransactionType = Context is used to locate a log line by data point value (field values), see Locating Log Lines by Context for more details.
38	TransactionType is optional for ItemGroupData. If TransactionType is not specified, RWS uses the TransactionType of the parent element. If none is found, RWS uses a default of Update.

Note Number	Comments
39	If mdsol:Submission = 'WholeItemGroup', then Rave marks all data items (data points) associated with the data page - including those that are not updated - as touched. This triggers any edit checks based on data points being touched (for example, to ensure that significant data points are not left blank). This is equivalent to using Save in the Rave EDC user interface. If mdsol:Submission = 'SpecifiedItemsOnly', only those data points included in the ODM request are marked as touched. Data points not included in the ODM are not marked as touched, which is the equivalent of submitting data points with empty values.
40	TransactionType is optional for ItemData . If TransactionType is not specified, RWS uses the TransactionType of the parent element. If the parent element does not specify the TransactionType , RWS uses a default of Update.
41	If a user submits an mdsol:CodingData element with TransactionType =Remove for a data point where coding does not exist, RWS returns a success code and ignores the fact that the data point was not currently coded.
42	TransactionType is optional for mdsol:TranslatedData . If TransactionType is not specified, RWS uses the TransactionType of the parent element. If the parent element does not specify the TransactionType , RWS uses a default of Update.

1.7.1.8 A Valid ODM 1.3 Snapshot Document



Note:

- A Snapshot ODM document has ODM FileType="Snapshot" and contains no TransactionType attributes.
- See also: [ODM Schema Mapping](#).

My ODM document contains the following code snippet:

```
<?xml version="1.0" ... ?>
<ODM
  xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata"
  ODMVersion="1.3"
  Granularity="Metadata"
  FileType="Snapshot"
  FileOID="{file-oid}"
  Description="{description}"
  CreationDateTime="{creation-date-time}"
  AsOfDateTime="{as-of-date-time}"
>
  <ClinicalData>
    ...
  </ClinicalData>
</ODM>
```

ODM Attributes:

Attribute	Description	Mandatory?	Notes
FileOID	The file identifier.	Yes	
Description	The description.	No	
CreationDateTime	The file creation time stamp.	Yes	
AsOfDateTime	The data time stamp.	No	

[Back to top](#)

1.7.1.9 A Valid ODM 1.3 Transactional Document

**Note:**

- A Transactional ODM document has ODM FileType="Transactional" and contains TransactionType attributes.
- See also: [ODM Schema Mapping](#).

My Odm document contains the following code snippet:

```
<?xml version="1.0" ... ?>
<ODM
  xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:mdsol="http://www.mdsol.com/ns/odm/metadata"
  ODMVersion="1.3"
  Granularity="SingleSubject"
  FileType="Transactional"
  FileOID="{file-oid}"
  Description="{description}"
  CreationDateTime="{creation-date-time}"
  AsOfDateTime="{as-of-date-time}"
>
  <ClinicalData ... TransactionType="{Insert|Update|Remove|Context}">
    ...
  </ClinicalData>
</ODM>
```

ODM Attributes:

Attribute	Description	Mandatory?	Notes
FileOID	The file identifier.	Yes	
Description	The description.	No	
CreationDateTime	The file creation time stamp.	Yes	
AsOfDateTime	The data time stamp.	No	
TransactionType	The transaction type.	No	Child elements inherit the transaction type from the parent.

1.7.1.10 ODM Unicode Character Conversion List



Important:

- Extended ASCII characters like 'á' are encoded and represented using XML character references, so that they are preserved in the transmission. It is your responsibility to convert those numeric character references back to characters in your application.
- Prior to Classic Rave 2019.2.0, entering Unicode characters in Rave EDC was supported but retrieving the same Unicode characters in ODM extracts through RWS was not supported. The Unicode characters supported in RWS were equivalent to Latin ASCII characters and special characters like '?', '!', '!', and so on. Many Unicode characters were converted to question mark ?, as specified in the table below.
- From Classic Rave 2019.2.0 onwards, the ODM Adapter V2 [Clinical Audit Records \(CAR\) service](#) has introduced an optional Boolean query string parameter, `unicode`, to support Unicode characters.
- Add `unicode=true` to the query string to return Unicode data in RWS, for example:

```
{ {url}}/datasets/ClinicalAuditRecords.odm?studyoid=<Study  
OID>&startid=<Start Audit ID>&per_page=<Per Page>&unicode=true
```

- If you do not require Unicode data, either omit specifying the `unicode` query parameter or add `unicode=false` to the query string, for example:

```
{ {url}}/datasets/ClinicalAuditRecords.odm?studyoid=<Study  
OID>&startid=<Start Audit ID>&per_page=<Per Page>&unicode=false  
  
{ {url}}/datasets/ClinicalAuditRecords.odm?studyoid=<Study  
OID>&startid=<Start Audit ID>&per_page=<Per Page>
```

Code	Actual	Converted
256	Ã	A
257	â	a

Code	Actual	Converted
258	Ă	A
259	ă	a
260	Ā	A
261	ā	a
262	Ć	C
263	ć	c
264	Ĉ	C
265	ĉ	c
266	Ĉ	C
267	ć	c
268	Č	C
269	č	c
270	Đ	D
271	đ	d
272	Đ	Đ
273	đ	d
274	Ē	E
275	ē	e
276	Ě	E
277	ě	e

Code	Actual	Converted
278	É	E
279	é	e
280	Œ	E
281	œ	e
282	Ě	E
283	ě	e
284	Ĝ	G
285	ĝ	g
286	Ğ	G
287	ğ	g
288	Ġ	G
289	ġ	g
290	Ģ	G
291	ǵ	g
292	Ĥ	H
293	ĥ	h
294	Ҥ	H
295	ҥ	h
296	Ӣ	I
297	ӣ	i

Code	Actual	Converted
298	ı	ı
299	ı	i
300	ſ	ı
301	ſ	i
302	ł	l
303	ł	i
304	ł	l
305	ł	i
308	Ĵ	J
309	Ĵ	j
310	Ķ	K
311	ķ	k
313	Ľ	L
314	ĺ	l
315	ľ	L
316	ļ	l
317	ļ	L
318	ŕ	l
321	Ŗ	L
322	Ŗ	l

Code	Actual	Converted
323	Ń	N
324	ń	n
325	Ņ	N
326	Ņ	n
327	Ň	N
328	ň	n
332	Ō	O
333	ō	o
334	Ŏ	O
335	ŏ	o
336	Ő	O
337	ő	o
340	Ŕ	R
341	ŕ	r
342	Ŗ	R
343	ŗ	r
344	Ř	R
345	ř	r
346	Ś	S
347	ś	s

Code	Actual	Converted
348	Ş	S
349	ş	s
350	Ş	S
351	§	s
354	Ț	T
355	ț	t
356	Ț	T
357	ť	t
358	Ƒ	T
359	ť	t
360	Ӯ	U
361	ӹ	u
362	Ӱ	U
363	ӻ	u
364	Ӯ	U
365	ӹ	u
366	Ӹ	U
367	ӻ	u
368	Ӯ	U
369	ӹ	u

Code	Actual	Converted
370	ȝ	U
371	ȝ	u
372	Ŵ	W
373	ŵ	w
374	Ŷ	Y
375	ŷ	y
377	Ź	Z
378	ź	z
379	Ž	z
380	ž	z
384	ƀ	b
393	Đ	Đ
401	Ƒ	f
407	ҭ	l
410	ҭ	l
415	Ө	O
416	Ӯ	O
417	ӫ	o
427	ҭ	t
430	Ҭ	T

Code	Actual	Converted
431	U'	U
432	u'	u
438	ż	z
448	ł	ł
451	!	!
461	Ā	Ā
462	ă	ă
463	Ĩ	Ĩ
464	ĩ	ĩ
465	Ő	Ő
466	ő	ő
467	Ŭ	Ŭ
468	ű	ű
469	Ū	Ū
470	ū	ū
471	Ú	Ú
472	ú	ú
473	Ӯ	Ӯ
474	ӹ	ӹ
475	Ӱ	Ӱ

Code	Actual	Converted
476	Ù	u
478	Ā	A
479	ā	a
484	Ğ	G
485	ğ	g
486	Ě	G
487	ě	g
488	Ќ	K
489	ќ	k
490	Ѡ	O
491	ѻ	o
492	Ӯ	O
493	ӻ	o
496	҂	j
609	g	g
697	'	'
698	"	700 '' 708 ^ ^ 712 '' 713 -- 714 .. 715 ..

Code	Actual	Converted
		717 _— 730 °° 768 `' 769 `' 770 ^^ 771 ~ 772 -- 773 — 776 778 °° 782 "
807	,	,
817	-	—
818	—	—
894	:	:
915	Γ	G
920	Θ	T
931	Σ	S
934	Φ	F
937	Ω	O
945	α	a
946	β	ß
948	δ	d
949	ε	e
956	μ	μ

Code	Actual	Converted
960	π	p
963	σ	s
964	τ	t
966	φ	f
1211	h	h
1417	:	:
1642	%	%
8192		
8193		
8194		
8195		
8196		
8197		
8198		
8208	-	-
8209	-	-
8215	=	=
8228	.	.
8242	,	,
8245	\`	\`

Code	Actual	Converted
8260	/	/
8304	º	º
8353	¢	¢
8356	£	£
8359	Pts	P
8450	©	C
8455	€	E
8458	¤	g
8459	‰	H
8460	¤	H
8461	¤	H
8462	h	h
8464	ƒ	I
8465	ñ	I
8466	Ł	L
8467	ł	I
8469	₦	N
8472	₽	P
8473	₱	P
8474	₴	Q

Code	Actual	Converted
8475	ꝑ	R
8476	ꝑ	R
8477	ꝑ	R
8484	ꝑ	Z
8488	ꝑ	Z
8490	K	K
8492	ꝑ	B
8493	ꝑ	C
8494	e	e
8495	e	e
8496	ꝑ	E
8497	ꝑ	F
8499	ꝑ	M
8500	o	o
8709	ꝝ	Ø
8722	-	-
8723	ꝑ	±
8725	/	/
8726	\	\
8727	*	*

Code	Actual	Converted
8728	◦	◦
8729	·	·
8730	√	√
8734	∞	8
8739		
8745	∩	∩
8758	:	:
8764	~	~
8776	≈	≈
8801	≡	≡
8804	≤	≤
8805	≥	≥
8810	<<	<<
8811	>>	>>
8901	.	.
8962	△	△
8963	^	^
8976	⊓	⊓
8992	{	(
8993	}	

Code	Actual	Converted
9001	□	<
9002	□	>
9472	—	-
9474		
9484	Γ	+
9488	⊍	+
9492	L	+
9496	J	+
9500	†	+
9508	‡	
9516	T	-
9524	⊥	-
9532	+	+
9552	=	-
9553		
9554	F	+
9555	Γ	+
9556	Ḟ	+
9557	⊍	+
9558	⊎	+

Code	Actual	Converted
9559	¶	+
9560	£	+
9561	₩	+
9562	₪	+
9563	£	+
9564	₩	+
9565	£	+
9566	ƒ	
9567	₩	
9568	₩	
9569	≠	
9570	₩	
9571	₩	
9572	〒	-
9573	〒	-
9574	〒	-
9575	±	-
9576	₩	-
9577	₩	-
9578	≠	+

Code	Actual	Converted
9579	‡	+
9580	‡†	+†
9600	■	—
9604	■	—
9608	■	
9612	█	
9616	█	
9617	●●●	
9618	●●●●	
9619	●●●●●	
9632	■	
9788	☀	¤
10072		
12296	<	<
12297	>	>
12298	<<	<<
12299	>>	>>
12314	[[
12315]]
12539	·	·

1.7.2 ODM Adapter Guidelines

Medidata's Operational Data Model (ODM) Adapter is a Rave Web Services capability which provides users with transactional ODM extracts of clinical transaction records for fields, field workflow status, field comments and queries for a given study. This solution enables partners and sponsors to load Rave EDC data into their enterprise warehousing systems.

- The solution is comprised of a set of pre-installed *custom datasets*.
- The data is returned in [CDISC ODM v1.3.1 format with Medidata vendor extensions](#).

The table below summarizes the main ODM Adapter best practices and related rationale.

#	ODM Adapter Best Practice	Rationale
1.	Always ensure that the opening and closing ODM tags are present in a received transmission.	If opening and closing ODM tags are not found, it is likely that the transmission has been interrupted and the package misses data.
2.	Always parse and check the ODM XML transaction header and, if you can, persist the header data.	The header section contains useful data such as what is the next StartId, and whether there is any more data to pull.
3.	At minimum, you should store the Record SourceId. It is also recommended to replicate the received data.	Essential when comparing the records that you receive to what Medidata has in the URL, when you suspect missing audit records.
4.	Try and run the ODM Adapter during off peak hours - either out of working hours, or during lower utilization on the URL.	The current and previous versions of the ODM Adapter can be server resource intensive, which in turn reduces page turn responsiveness and other Rave operations.

#	ODM Adapter Best Practice	Rationale
5.	Use the Study Metadata service in conjunction with the Clinical Audit Record service.	When you build receiving structures, for example, a clinical data warehouse, you may need to denormalize your data to make it easier to report. Using the metadata from the Study Metadata service helps keep your data structures in sync with Rave.
6.	Do not create duplicate queries on the same data point.	Sometimes, edit checks and RWS POST activities can produce identical queries on the same data point. Since the ODM Adapter cannot distinguish between these queries, it may result in unpredictable processing.
7.	Do not have more than two ODM Adapter sessions running at the same time, against the same URL. For optimal use, Medidata recommends a single ODM Adapter session.	It has been proven in tests that three ODM Adapter sessions pulling 470,000 records per hour will increase Rave page turn time by 100%.
8.	Restrict the number of users who can access ODM Adapter.	The fewer staff have access to the ODM Adapter, the better it will perform.
9.	Only the Default Matrix is present in the Study Metadata service.	Although the other matrices are not present, all the form metadata is present in the Study Metadata service. Hence, you can create all the tables you need for your data warehouse using the form

#	ODM Adapter Best Practice	Rationale
		metadata, and you will not miss any data in the study.
10.	Make your web service receiving process flexible enough to handle new Audit Record Subcategories .	New Audit Record Subcategories will be added over time.
11.	Expect timeouts and make your web service receiving process tolerant of this behaviour. Wait one hour before restarting your next call.	<p>ODM Adapter uses the REST protocol. As such, there may be periods whereby it appears that the tool has hung. This is expected due to the distributed method by which the ODM XML file is transmitted between servers, across the internet.</p> <ul style="list-style-type: none"> • Make your receiver system adjust to potential timeouts. • Do not launch another ODM Adapter session without knowing if a previous session has timed out, or not.
12.	If you persist the ODM XML transactions and use them to compose a real-time snapshot, your process must be able to insert new data, change existing data and delete data.	If your system has the ability to persist data, but cannot delete data points and metadata, this will compromise your reports.
13.	If you encounter many timeouts when accessing the same data for the same study, on the same URL,	Occasionally, the distance between Audit records is too great and the ODM Adapter process takes too long to return data. In this

#	ODM Adapter Best Practice	Rationale
	please contact Medidata Customer Support.	circumstance, Medidata Customer Support can provide a StartId that bridges the gap, and allows your web service receiver process to continue.
14.	Always use the previously supplied StartId as the bookmark for your next ODM Adapter Clinical Audit Record service call. The StartId is given in the header of the previous transaction.	The StartId in the header gives you the Audit record start position for your next call. It is not mandatory to follow it, but it will save you time.
15.	When you get timeout, you should decrease the output from the request service by using the available parameters. For example, when using the the ODM Adapter Clinical Audit Records service, you should reduce the page size.	To avoid timeout, in general, you should reduce the request output by using the available parameters for the service.

1.7.3 ODM Adapter Included Audit Subcategories

The following are the audit subcategories included in the [ClinicalAuditRecords Custom Data Set](#) for ODM Adapter Version 1, and the audit subcategories for the default mode of the ODM Adapter Version 2.

For additional details, see [Rave Audit Categories and Subcategories](#).



Note: The following Subject, Instance, DataPage, and Record statuses are created through a roll-up mechanism:

- Freeze
- Lock

- UnFreeze
- Unlock

Subject

Audit Category	Audit Subcategory	Description
Data	MigDPDeleteBrokeSignature	Signature that was broken when a data point was deleted during a CRF Migration.
Freeze	Freeze	Subjects that were frozen (no data entry allowed).
Freeze	UnFreeze	Subjects that were unfrozen.
General	Activated	Subjects who were activated.
General	Deleted	Subjects who were deleted.
General	Inactivated	Subjects that were inactivated.
HardLock	Lock	Subjects that were hard-locked (no changes allowed).
HardLock	Unlock	Subjects that were released from a hard lock.
Sign	SignatureBrokenIndirectly	Data whose signature is broken indirectly.
Sign	SubjectSignatureBroken	Subjects whose signature is broken (non-conformant). If the subject details are signed by an investigator and then data is changed, updated, or deleted, then the subject details are no longer signed.

Audit Category	Audit Subcategory	Description
Subject	SubjectCopied	Subject that was copied.
Subject	SubjectCreated	Subject that was created.
Subject	SubjectMoved	Subject whose study site was changed.
Subject	SubjectNameChanged	Subject whose name changed.
Subject	SubjectShared	Subject that is shared between sites.
Subject	SubjectStatusChanged	Subject whose status has changed.

Instance

Audit Category	Audit Subcategory	Description
Data	InstanceAccess	Instances for which Access date has been set.
Data	InstanceClose	Instances for which Close date has been set.
Data	InstanceEndWindow	Instances for which End Window date has been set.
Data	InstanceOverdue	Instances for which Overdue date has been set.
Data	InstanceStartWindow	Instance for which Instance Start window date has been set.
Data	InstanceTarget	Instances for which target date has been set.

Audit Category	Audit Subcategory	Description
Data	MigDPDeleteBrokeSignature	Signature that was broken when a data point was deleted during a CRF Migration.
Freeze	Freeze	Instances that were frozen (no data entry allowed).
Freeze	UnFreeze	Instances that were unfrozen.
General	Activated	Instances that were activated.
General	Deleted	Instances that were deleted.
General	Inactivated	Instances that were inactivated.
General	ObjectNameChanged	Instances that have been renamed.
HardLock	Lock	Instances that were hard-locked (no changes allowed).
HardLock	Unlock	Instances that were released from a hard lock.
Sign	FolderSignatureBroken	Instances whose signature is broken (non-conformant). If a folder is signed by an investigator and then data is changed, updated, or deleted, then the folder is no longer signed.
Sign	SignatureBrokenIndirectly	Data whose signature is broken indirectly.
Sign	Signed	Data that a user successfully signed electronically.

Audit Category	Audit Subcategory	Description
Sign	ValidESigCredential	Pages that have a successful eSignature attempt due to valid password credentials.
Subject	MigInstanceDeleted	Instances deleted during a CRF Migration.
Subject	MigInstanceMoved	Instances moved during a CRF Migration.

DataPage

Audit Category	Audit Subcategory	Description
Data	ActivatedWithReasonCode	Data pages activated with Reason Code.
Data	InactivatedWithReasonCode	Data pages the user inactivated with Reason Code.
Data	MigDPDeleteBrokeSignature	Signature that was broken when a data page was deleted during a CRF Migration.
Freeze	Freeze	Data pages that were frozen (no data entry allowed).
Freeze	UnFreeze	Data pages that were unfrozen.
General	Deleted	Data pages that were deleted.
General	Inactivated	Data pages that were inactivated.
General	ObjectNameChanged	Data pages that have been renamed.

Audit Category	Audit Subcategory	Description
General	Activated	Data pages that were activated.
HardLock	Lock	Data pages that were hard-locked (no changes allowed).
HardLock	Unlock	Data pages that were released from a hard lock.
Labs	LabAssigned	Lab results entered into form.
Labs	LabRemoved	Lab results removed from form.
Sign	PageSignatureBroken	Pages whose signature is broken (non-conformant). If a form is signed by the investigator and then data is changed, updated or deleted, then the form is no longer signed.
Sign	SignatureBrokenIndirectly	Data whose signature is broken indirectly.
Sign	Signed	Data that a user successfully signed electronically.
Sign	ValidESigCredential	Pages that have a successful eSignature attempt due to valid password credentials.
Subject	MigDataPageDeleted	Data pages that were deleted during a CRF Migration.
Subject	MigDataPageMoved	Data pages that were moved during a CRF Migration.

Record

Audit Category	Audit Subcategory	Description
Data	ActivatedWithReasonCode	Records activated with Reason Code.
Data	InactivatedWithReasonCode	Records the user inactivated with Reason Code.
Freeze	Freeze	Records that were frozen (no data entry allowed).
Freeze	UnFreeze	Records that were unfrozen.
General	Activated	Records that were activated.
General	Deleted	Records that were deleted.
General	Inactivated	Records that were inactivated.
HardLock	Lock	Records that were hard-locked (no changes allowed).
HardLock	Unlock	Records that were released from a hard lock.
Subject	MigRecordDeleted	Records deleted during a CRF Migration.

DataPoint

Audit Category	Audit Subcategory	Description
Comment	CommentActivate	Data points with activated comments.
Comment	CommentAdd	Data points with added comments.

Audit Category	Audit Subcategory	Description
Comment	CommentEdit	Data points that have a comment that has been edited.
Comment	CommentInactivate	Data points with an inactivated comment.
Data	AcceptedDefaultValue	Data points with default values that are entry-restricted (but not view-restricted) from the user.
Data	ActivatedWithReasonCode	Data points activated with Reason Code.
Data	Entered	Data points where the user has entered data for the first time.
Data	EnteredByDdeAutoReconcile	Data that was entered for Pass 1 and Pass 2 of a DataPage, and the data was equivalent. As a result, the system autoreconciled the DataPage.
Data	EnteredEmpty	Data points that have been entered empty by a user.
Data	EnteredEmptyWithChangeCode	Data points that have been submitted empty with Change Code by the user.
Data	EnteredInForeignLocale	Data points that have information entered in a foreign language.
Data	EnteredInForeignLocaleWithChangeCode	Data points that have information entered in a foreign language with Change Code.
Data	EnteredNonConformant	Data points that were made nonconformant by the user.

Audit Category	Audit Subcategory	Description
Data	EnteredWithChangeCode	Data points the user entered with Change Code.
Data	EnteredWithMissingCode	Data points that the user entered with Change and Missing Code.
Data	InactivatedWithReasonCode	Data points the user inactivated with Reason Code.
Data	MigCodedUnitChanged	Data points where a Coded Unit was changed by a CRF Migration.
Data	MigDataChanged	Data points whose data was changed by a CRF migration.
Data	MigDataDeleted	Data points whose data was deleted by a CRF Migration.
Data	MigDataPointInserted	Data points that were created with empty data by a CRF Migration.
Data	MigDateChanged	Data points whose date format was changed during a CRF Migration.
Data	MigDDEStringChanged	Data points that have data that was deleted during CRF Migration.
Data	MigDPDeleteBrokeSignature	Signature that was broken when a data point was deleted during a CRF Migration.
Data	MigDPDeleted	Data points that have data that was deleted during CRF Migration.
Data	MigUDEDDefChanged	Data points whose Unit Dictionary definitions have changed during a CRF Migration.

Audit Category	Audit Subcategory	Description
Data	SetConformant	Data points that were set to conformant.
Data	SetInvisible	Data points set to invisible. Certain items on a page can be hidden or revealed until a question is answered. The corresponding question becomes visible or invisible.
Data	SetNonConformant	Data points that were set to nonconformant.
Data	Translated	Data points that have been translated.
Data	SetVisible	Data points set to visible.
Freeze	Freeze	Data points that were frozen (no data entry allowed).
Freeze	UnFreeze	Data points that were unfrozen.
General	Activated	Data points that were activated.
General	Created	Data points that were created.
General	Deleted	Data points that were deleted.
General	Inactivated	Data points that were inactivated.
General	ObjectNameChanged	Data points that have been renamed.
HardLock	Lock	Data points that were hard-locked (no changes allowed).

Audit Category	Audit Subcategory	Description
HardLock	Unlock	Data points that were released from a hard lock.
Labs	ClinicalSignificanceEmpty	Lab data for which the clinical significance is empty.
Labs	ClinicalSignificanceRemoved	Lab data for which the clinical significance has been deleted.
Labs	ClinicalSignificanceSet	Lab data for which the clinical significance has been set.
Labs	ClinicalSignificanceSetWithComment	Lab data for which clinical significance has been set with comment.
Labs	DataEntryOutOfRange	Lab data that was out of range.
Labs	LabRangeChanged	Lab analyte ranges that were changed.
Labs	LabRangeChangedToUnits	Lab analyte ranges whose units have changed.
Labs	LabRangeStatusChanged	Lab ranges whose status has changed.
Labs	LabRangeStatusNotChangedOnLocked	Lab ranges whose status did not change when the data points were hard-locked.
ProtocolDeviation	ActivatePD	Protocol Deviation that was activated.
ProtocolDeviation	CreatePD	Protocol Deviation that was created.

Audit Category	Audit Subcategory	Description
ProtocolDeviation	EditClassPD	Protocol Deviation class that was edited.
ProtocolDeviation	EditCodePD	Protocol Deviation code that was edited.
ProtocolDeviation	EditTextPD	Protocol Deviation text that was edited.
ProtocolDeviation	InActivatePD	Protocol Deviation that was activated.
Query	MigQueryClosed	A query that was closed during CRF Migration because it no longer exists in the target version.
Query	MigQueryStringChanged	A query string that has changed as a result of a CRF Migration.
Query	QueryAnswer	A query that has been answered.
Query	QueryAnswerByChange	A query that has been answered by data change.
Query	QueryCancel	A query that has been cancelled.
Query	QueryClose	A query that has been closed.
Query	QueryForward	A query that has been forwarded to another Marking Group.
Query	QueryOpen	A query that has been opened.
Review	Review	Data that was reviewed for a Review Group.
Review	ReviewRequired	Data that is pending review by Review Group.

Audit Category	Audit Subcategory	Description
Review	UnReview	Data that was unreviewed for Review Group.
Sign	SignatureBroken	Data whose signature is broken.
Sign	SignatureBrokenIndirectly	Data whose signature is broken indirectly.
Sign	Signed	Data that a user successfully signed electronically.
Sign	ValidESigCredential	Pages that have a successful eSignature attempt due to valid password credentials.
User	PasswordSet	Users whose passwords have been set.
Verification	UnVerify	Data points that were unverified.
Verification	Verify	Data points that were verified.
Verification	VerifyRequired	Data points that are pending verification. A datapoint must be checked according to a Rave workflow. For example, if you enter an age, then that age must be verified by a monitor.

1.7.4 Clinical View Service Limitations

The RWS Clinical View service extracts the same information as a Rave Data Listing report, but in ODM or CSV format.

The data extracted either by RWS or by a data listing report is not real-time data as viewable in Rave EDC, due to the slight delay required for the data to appear in the clinical view, and, therefore, in the RWS Clinical View service.

This time delay varies depending on the following factors:

- The configured time interval between Clinical View refreshes (this value is set in the clinical view configuration).
- The volume of data changes in Rave since the last Clinical View refresh.
- The volume of data in the Rave database for the study.

Because of this delay, there may be instances when data updates in Rave do not appear in the Clinical View data set when you extract incremental data sets with RWS, depending on the frequency of the extract and start date/time parameter used in the extract.



Note: An "X-MWS-CV-Last-Updated" time stamp of when the clinical view was last updated is returned in the response header.

1.7.5 Configurable Datasets Authorization Best Practices

Configurable Datasets (CDS) are a feature of RWS that allows you to create arbitrary data extracts with the aid of REST requests. RWS CDS supports GET only requests and CSV, XML, and JSON file formats.

Create a New CDS

Define the data extract SQL as well as the [Dot Liquid template](#) to render the document that will be returned. For more details, see the RWS documentation below.

Create a New User

Each integrating system is responsible for creating its own users. If you are creating a user as part of an integration, the user must be created at the same time the CDS is being installed. This can be done in:

- DailyChanges, if installing as part of Rave, or
- as part of your installation process, if part of a Medidata integration with its own installation procedure (such as Imaging, Subject Service, ePro, and so on).

[Back to top](#)

CDS Authorization

Since the CDS feature was designed to allow you to extract out of Rave any data you want without limitations, RWS has no way of knowing if you actually have access to the data you are returning within Rave.

Therefore, except under rare circumstances (such as returning non-authorized data), you must always provide authorization inside your stored procedure by checking if the UserID of the user passed into that stored procedure actually has access to the data you are returning.

To set up CDS authorization, perform the following steps:

1. On the dataset, set the **RequiresUserFiltering** field to 1. This tells RWS to accept a **UserID** parameter in the **SQL Stored** procedure of the CDS.
2. Add a **UserID** integer parameter to your stored procedure.

[Back to top](#)

Example spRWSCDSPProjectFetchByStudyUUID

Important: If you extract the project name and UUID for a given studyUUID in a CDS in a simple way, like shown here:

```
SELECT LOWER(P.UUID) AS uuid,
       dbo.fnLocalDefault(P.ProjectName) AS name
  FROM dbo.Studies S
 WHERE S.UUID = @studyUUID
```

then you open yourself up to a **security risk** in which anyone with Rave access could now call this stored procedure.

Important: To avoid security risks, follow these steps:

1. Check that the **UserID** that was sent in actually has access to this data.
2. Ensure that the user:
 1. has access to the project sent in,

2. has access to all the projects for the role they have, and
3. can perform read actions on that project.

The **correct code** appears in the following snippet:

```
IF OBJECT_ID('dbo.spRWSProjectFetchByStudyUUID') IS NULL
    EXEC ('CREATE PROCEDURE dbo.spRWSProjectFetchByStudyUUID AS RETURN')

GO
ALTER PROCEDURE dbo.spRWSProjectFetchByStudyUUID
(
    @UserID INT,
    @StudyUUID VARCHAR(36)
)

AS
BEGIN
SET NOCOUNT ON
DECLARE @ProjectID INT = NULL
DECLARE @DefaultLocale CHAR(3) = 'eng'
CREATE TABLE #UserProjects (
    ProjectID INT,
    ProjectName INT,
    ExternalID INT,
    AbbrID INT,
    Description INT,
    ProjectActive BIT,
    Updated DATETIME,
    Created DATETIME,
    ServerSyncDate DATETIME,
    PageRedirection INT,
    ProjectTypeID INT,
    IsAllCopySources BIT,
    IconID INT,
    CreatedBy INT,
    UUID VARCHAR(36),
    ProjectSourceSystemId INT,
    EnableVerifyReviewOnInactive BIT,
    PName VARCHAR(4000)
)
INSERT INTO #UserProjects
EXEC dbo.spAccessibleProjectsLoadByUserID
    @UserID = @UserID,
    @Locale = @DefaultLocale
```

```
-- Verify user has access to the project
SELECT @ProjectID = P.ProjectID
    FROM dbo.Studies S
    JOIN #UserProjects P ON P.ProjectID = S.ProjectID
    AND S.UUID = @StudyUUID

IF (@projectId IS NOT NULL)
BEGIN
    DECLARE @UserTypeID INT
    DECLARE @ProjectTypeID INT
    DECLARE @AllProjectsID INT
    SELECT @UserTypeID = ObjectTypeID
        FROM dbo.ObjectTypeR
        WHERE ObjectName = 'Medidata.Core.Objects.User'
    SELECT @ProjectTypeID = ObjectTypeID
        FROM dbo.ObjectTypeR
        WHERE ObjectName = 'Medidata.Core.Objects.Project'
    SELECT @AllProjectsID = ConfigValue
        FROM dbo.Configuration
        WHERE Tag = 'SecurityAllProjectsID'
CREATE TABLE #UserRoles (
    RoleID SMALLINT,
    RolePermissions INT,
    Active BIT,
    Created DATETIME,
    Updated DATETIME,
    ServerSyncDate DATETIME,
    ViewAllSites BIT,
    CreateSubjects BIT,
    ModifyPrimaryRecord BIT,
    IsInvestigator BIT,
    SubjectAdmin BIT,
    SeeAudits BIT,
    ChangeInvestigator BIT,
    ShareSubject BIT,
    CanViewStudyGrid BIT,
    CanBatchSign BIT,
    CanInactivateForm BIT,
    CanReactivateForm BIT,
    UsePrimarySubjectName BIT,
    CanUseTemplates BIT,
    CanModifyTemplates BIT,
    SetClinicalSignificance BIT,
    UseLabMaintenance BIT,
    ViewAllSubjects BIT,
    Pass1DataEntry BIT,
    Pass2DataEntry BIT,
    Reconcile BIT,
    RoleNameID INT,
    CanUseDCFs BIT,
```

```
CanUseAdvancedDCFs BIT,
ModuleId TINYINT,
UUID VARCHAR(36)
)

-- Fetch role associated with all-projects access
INSERT INTO #UserRoles
EXEC dbo.spRoleFetchByUserObject
    @UserObjectID = @userID,
    @UserObjectTypeID = @UserTypeID,
    @GrantOnObjectID = @AllProjectsID,
    @GrantOnObjectTypeID = @ProjectTypeID

-- Fetch role associated with this specific project
INSERT INTO #UserRoles
EXEC dbo.spRoleFetchByUserObject
    @UserObjectID = @userID,
    @UserObjectTypeID = @UserTypeID,
    @GrantOnObjectID = @ProjectID,
    @GrantOnObjectTypeID = @ProjectTypeID

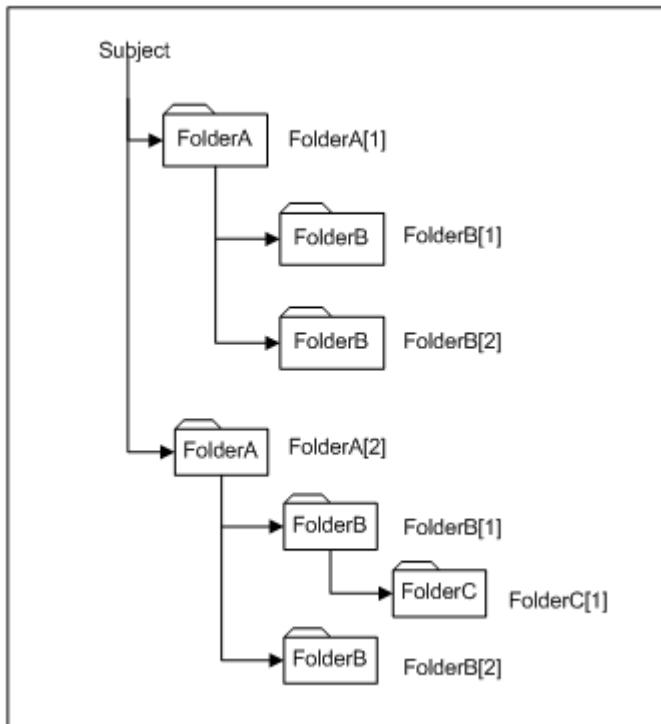
-- Verify the user can perform a read action on the project
IF ((SELECT COUNT(*)
     FROM #UserRoles R
     JOIN dbo.RoleActions RA ON RA.RoleID = R.RoleID
     JOIN dbo.ImpliedActionTypes IA ON IA.ActionType = RA.ActionType
     WHERE IA.Implied ActionType = 50 /* READ */) > 0)
BEGIN
    SELECT LOWER(P.UUID) AS uuid,
           dbo.fnLocalDefault(P.ProjectName) AS name
    FROM dbo.Studies S
    JOIN #UserProjects P ON P.ProjectID = S.ProjectID
    AND S.UUID = @studyUUID
END
END
END
```

1.7.6 Navigating Repeated, Nested Folders

Navigating repeated, nested folders applies to updating or retrieving subject data.

In a Rave study, folders can contain repeated child folders with identical OIDs. For example, a subject record may contain multiple visit folders, and similarly, each visit folder may contain multiple subfolders.

In order to correctly identify which folder or subfolder is being addressed by a clinical data POST (create or update) request, Rave considers that each folder and sub-folder is numbered sequentially at each level as shown in this illustration:



RWS identifies a nested folder by specifying a value for the StudyEventRepeatKey attribute of the StudyEventData element. This is used in addition to the StudyEventOID attribute. For all folders, you can use either numeric values or a Rave folder path as the value for the StudyEventRepeatKey attribute.

For example, the second occurrence of FolderA, a top-level folder as shown in the illustration, can be described in the ODM XML as one of the following:

```
<StudyEventData StudyEventOID="FolderA" StudyEventRepeatKey="2">
```

or:

```
<StudyEventData StudyEventOID="FolderA" StudyEventRepeatKey="FolderA[2]">
```

In contrast, FolderC, which follows the second occurrence of FolderA in the illustration, is not a top-level folder, so it can be only described in the ODM XML as one of the following:

```
<StudyEventData StudyEventOID="FolderC" StudyEventRepeatKey="1">
```

or:

```
<StudyEventData StudyEventOID="FolderC" StudyEventRepeatKey="FolderA[2]/FolderB[1]/  
FolderC[1]">
```

In general, the Rave folder path can be expressed as the following:

```
FolderX[m] / ... / FolderY[n] (in the order of the folder tree structure)
```

In the above, X and Y refer to Folder OIDs and m and n refer to the relative ordinal number with respect to its parent folder.

If RWS receives a request to create more than one repetition of a folder (instance) at the same level, where the folder is marked as reusable, the RWS returns the error response message:

"Cannot create more than one reusable folder at the same level (RWS00053)".

[Back to top](#)

1.7.7 Definitions

Below is a list of the main definitions used in RWS.

[Measurement unit definitions](#)

[Folder definitions](#)

[Form definitions](#)

[Log line definitions](#)

[Field definitions](#)

[Field Edit Check definitions](#)

[Medidata Coder definitions](#)

[Code list definitions](#)

[Confirmation Message definitions](#)

Label definitions

1.7.7.1 Code List Definitions

Use Case

I am *updating* or *retrieving* my study metadata and...

I am updating or retrieving my Code list definitions.

My ODM snippet

```
...
<CodeList OID="{oid}" Name="{name}" DataType="{data type}">
  <CodeListItem CodedValue="{coded value}" mdsol:OrderNumber="{number}">
    <Decode>
      <TranslatedText xml:lang="{lang}">{decoded value}</TranslatedText>
    </Decode>
  </CodeListItem>
  ...
</CodeList>
...
```

CodeList Attributes:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Value	Notes	Rave mapping
OID	OID (255 characters max)	Required	Maps to Data Dictionary Name in Rave.
Name	name	Required	N/A
DataType	{integer float text string }	Required	N/A
SASFormatName	sasFormat	Optional	N/A

CodeListItem attributes:

Attribute	Value	Notes	Rave mapping
CodedValue	value	Required	Maps to the Coded Data value of the Data Dictionary entry in Rave.
mdsol:OrderNumber	integer	Optional	Maps to the position of the Data Dictionary entry within the list, that is, Data Dictionary ordinal positioning in Rave. See Note 29 .
mdsol:Specify	{Yes No (default) }	Optional	Maps to the Data Dictionary Specify field in Rave. Yes = checked No = unchecked.

Example:

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<CodeList OID="DAYSWEEKSMONTHSYEARS" Name="DAYSWEEKSMONTHSYEARS" DataType="text">
    <CodeListItem CodedValue="Y" mdsol:OrderNumber="1">
        <Decode>
            <TranslatedText xml:lang="en">Years</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="M" mdsol:OrderNumber="2">
        <Decode>
            <TranslatedText xml:lang="en">Months</TranslatedText>
        </Decode>
    </CodeListItem>
    ...
</CodeList>
...
```

[Back to top](#)

1.7.7.2 Confirmation Message Definitions

Use Case

I am updating study metadata and ...

I am updating or retrieving my Confirmation Message definitions.



Note: The mdsol:ConfirmationMessage element maps to the Draft Message, and displays as the Confirmation Message in the CRF Draft Settings section. This element is optional.

My ODM snippet

```
...
<mdsol:ConfirmationMessage xml:lang="{lang}">{message}</mdsol:ConfirmationMessage>
...
```

ODM Attributes:

Attribute	Value	Notes	Rave mapping
xml:lang	languageTag	Optional	N/A

Example:

```
...  
<mdsol:ConfirmationMessage xml:lang="en">Data Saved</mdsol:ConfirmationMessage>
```

[Back to top](#)

1.7.7.3 Field Definitions

Use Case

I am *updating* or *retrieving* my study metadata and...

I am updating or retrieving my Field definitions.

My ODM snippet

```
...  
<ItemDef  
OID="{oid}"  
Name="{name}"  
DataType="{type}"  
mdsol:VariableOID="{v-oid}"  
Length="1" mdsol:Active="{Yes|No}"  
mdsol:ControlType="{type}"  
mdsol:Visible="{Yes|No}">  
...  
</ItemDef>  
...
```

ODM Attributes:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Value	Notes	Rave mapping
OID	oid (This is limited to a maximum of 50 characters)	Required	Maps to the Fields Field OID & VarOID in Rave. See Note 16 .
Name	Name (This is limited to OID + (Name) 254 characters maximum length)	Required	Maps to Fields Field Name in Rave. See Note 16 and Note 17 .
DataType	{text integer float date datetime time }	Required	Maps to Fields Format and Control Type in Rave. text = \$ + Length text (for length > 50) = \$+Length integer = Length date = dd MMM yyyy time = HH:nn:ss.
Length	integer	Required by Rave for data types text, integer and float (see DataType, above); Optional for other data types	Maps to Fields Format in Rave (See DataType above).
SignificantDigits	integer	Optional	Maps to Fields Format Significant Digits (See DataType above).
SASFieldName	sasName	Optional	N/A
SDSVarName	sasName	Optional	N/A
Origin	text (255 characters max)	Optional	Not mapped in Rave.

Attribute	Value	Notes	Rave mapping
Comment	text (255 characters max)	Optional	N/A
mdsol:Active	{Yes (default) No }	Optional	Maps to Fields Active checked in Rave.
mdsol:ControlType	{CheckBox Text DateTime DropDownList SearchList RadioButton RadioButton (vertical) File Upload LongText Signature page Signature folder Signature subject Dynamic SearchList}	Optional	Maps to Fields Control Type in Rave. See Note 18 .
mdsol:AcceptableFileExtensions	text (255 characters max. (If mdsol:ControlType = File Upload, should contain file types delimited by commas, for example: ,doc,xml,jpeg)	Optional	Maps to Fields Accept files with extensions in Rave.
mdsol:IndentLevel	{0 (default) 1 2}	Optional	Maps to Fields Indent level in Rave.
mdsol:SourceDocument	{Yes No (default) }	Optional	Maps to Fields Requires Verification (under Field Verification and Reviews). Yes = checked No = unchecked
mdsol:DefaultValue	text (255 characters max)	Optional	Maps to Fields Default Value in Rave.

Attribute	Value	Notes	Rave mapping
mdsol:SASFormat	text (255 characters max)	Optional	Maps to Fields SAS Format in Rave.
mdsol:SASLabel	text (255 characters max)	Optional	Maps to Fields SAS Label in Rave.
mdsol:QueryFutureDate	{Yes No (default) }	Optional	Maps to Fields Auto-Query for future Date/time in Rave. Yes = checked No = unchecked
mdsol:Visible	{Yes(default) No }	Optional	Maps to Fields Is visible field in Rave. Yes = checked No = unchecked
mdsol:TranslationRequired	{Yes No (default) }	Optional	Maps to Fields Requires Translation field in Rave. Yes = checked No = unchecked See Note 19 .
mdsol:QueryNonConformance	{Yes No (default) }	Optional	Maps to Fields Auto-Query for non-conformant data. Yes = checked No = unchecked
mdsol:OtherVisits	{Yes No (default) }	Optional	Maps to Fields Show Previous Visit Values in Rave. Yes = checked No = unchecked
mdsol:CanSetItemGroupDate	{Yes No (default) }	Optional	Maps to Fields Can Set Record Date in Rave. Yes = checked No = unchecked

Attribute	Value	Notes	Rave mapping
mdsol:CanSetFormDate	{Yes No (default) }	Optional	Maps to Fields Can Set DataPage Date in Rave. Yes = checked No = unchecked
mdsol:CanSetStudyEventDate	{Yes No (default) }	Optional	Maps to Fields Can Set Instance Date in Rave. Yes = checked No = unchecked
mdsol:CanSetSubjectDate	{Yes No (default) }	Optional	Maps to Fields Can Set Subject Date in Rave. Yes = checked No = unchecked
mdsol:VisualVerify	{Yes No (default) }	Optional	Maps to Fields Visual Verify in Second Pass in Rave. Yes = checked No = unchecked Only appears on fields of DDE forms.
mdsol:DoesNotBreakSignature	{Yes No (default) }	Optional	Maps to Fields Does not participate in Signature in Rave. Yes = checked No = unchecked
mdsol:DateTimeFormat	Any valid Rave date-time variable format	Optional	Maps to field's variable format. See Note 20.

Attribute	Value	Notes	Rave mapping
mdsol:FieldNumber	text (maximum 50 characters)	Optional	Maps to Field Num field in Rave. See Note 21.
mdsol:VariableOID	oid (limited to 50 characters)	Optional	Maps to Variable OID in Rave. If this attribute is not present, the Field OID maps to the Variable OID in Rave.

Example:

i Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<ItemDef OID="DM.AGEU" Name="AGEU" DataType="text" mdsol:VariableOID="AGEU"
Length="1" mdsol:Active="Yes"
mdsol:ControlType="DropDownList" mdsol:Visible="Yes">
<Question>
  <TranslatedText xml:lang="en">Age Units</TranslatedText>
</Question>
<CodeListRef CodeListOID="DAYSWEEKSMONTHSYEARS" />
<mdsol:HeaderText xml:lang="en">□□</mdsol:HeaderText>
<mdsol:HelpText xml:lang="en">The trade or proprietary name of the drug is
recommended; otherwise the generic
name may be recorded.</mdsol:HelpText>
</ItemDef>
...

```

[Back to top](#)

1.7.7.4 Field Edit Check Definitions

Use Case

I am *updating* or *retrieving* my study metadata and...

I am updating or retrieving my Field Edit Check definitions.

i Note: There are five different Field Edit Checks supported by Rave. They are applied using the following:

1. Auto-Query for required data entry - use ItemRef element with Mandatory='Yes'.

2. Auto-Query for non-conformant data - use ItemDef element with mdsol:QueryNonConformance='Yes'.
3. Auto-Query for future date/time - use ItemDef element with mdsol:QueryFutureDate='Yes'.
4. Auto-Query for data out of range - use RangeCheck element with SoftHard attribute set to "Soft".
5. Mark non-conformant data out of range - use RangeCheck element with SoftHard attribute set to "Hard".

My ODM snippet

```
...
<ItemDef OID="{oid}" Name="{name}" DataType="{data type}" Length="{length}"  
mdsol:QueryNonConformance="Yes">  
  <Question>  
    <TranslatedText xml:lang="{lang}">{question}</TranslatedText>  
  </Question>  
  <RangeCheck Comparator="{comparator}" SoftHard="{Soft|Hard}">  
    <CheckValue>{check value}</CheckValue>  
  </RangeCheck>  
  <RangeCheck Comparator="{comparator}" SoftHard="{Soft|Hard}">  
    <CheckValue>{check value}</CheckValue>  
  </RangeCheck>  
</ItemDef>  
...
```

ItemDef Attribute:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Value	Notes	Rave mapping
mdsol:QueryNonConformance	{Yes No (default) }	Optional	Maps to Fields Auto-Query for non-conformant data. Yes = checked No = unchecked

RangeCheck Attributes:

Attribute	Value	Notes	Rave mapping
Comparator	{LT LE GT GE EQ NE IN NOTIN}	Required	RWS only supports LE and GE. This together with SoftHard maps to the low/high range values for Auto-Query for data out of range or Mark non-conformant data out of range in Rave See Note 23 .
SoftHard	{Soft Hard}	Required	This together with Comparator maps to the low/high range values for Auto-Query for data out of range or Mark non-conformant data out of range in Rave.

Example:

i Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<ItemDef Name="Value" DataType="integer" Length="3" mdsol:QueryNonConformance="Yes"
OID="AE.VALUE">
  <Question>
    <TranslatedText xml:lang="en">Systolic Blood Pressure:</TranslatedText>
  </Question>
  <RangeCheck Comparator="GE" SoftHard="Soft">
    <CheckValue>20</CheckValue>
  </RangeCheck>
  <RangeCheck Comparator="LE" SoftHard="Soft">
    <CheckValue>100</CheckValue>
  </RangeCheck>
</ItemDef>
...
```

Scenarios**Background:**

Given I am a user

Scenario Outline: Apply non-conformance or future-date field edit check to a field
 Given I have the following snapshot ODM with "<attribute>" attribute set to "Yes":
 """

```
<ItemDef OID='AE.STARTDATE' Name='STARTDATE' <attribute>='Yes'
mdsol:ControlType='DateTime' DataType='date' />
"""
```

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"
 Then I should receive a successful XML response
 And I should see "<field edit check>" field edit check is "enabled" for field
 "AE.STARTDATE" in Rave

Examples:

field edit check	attribute	
Auto-query for non-conformant data	mdsol:QueryNonConformance	
Auto-query for future date/time	mdsol:QueryFutureDate	

Scenario: Apply data entry required field edit check to a field

Given I have the following snapshot ODM with mandatory attribute set to "Yes":

```
"""
<ItemRef ItemOID='AE.STARTDATE' Mandatory='Yes' />
"""

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"
Then I should receive a successful XML response
And I should see "Auto-Query for required data entry" field edit check is
"enabled" for field "AE.STARTDATE" in Rave
```

Scenario Outline: Data out of range edit checks for a field

Given I have the following snapshot ODM to set the "<soft_hard>" range boundaries to "<low value>" and "<high value>":

```
"""
<RangeCheck Comparator='GE' SoftHard=<soft_hard>>
  <CheckValue><low value></CheckValue>
</RangeCheck>
<RangeCheck Comparator='LE' SoftHard=<soft_hard>>
  <CheckValue><high value></CheckValue>
</RangeCheck>
"""

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"
Then I should receive a successful XML response
And I should see "<field edit check>" field edit check has range boundaries of
"<low value>" and "<high value>" in Rave
```

Examples:

	field edit check	soft_hard	low value	high value
	Auto-query for data out of range	Soft	80	210
	Mark non-conformant data out of range	Hard	10	15

Scenario Outline: Set either upper or lower limit in data out of range edit check for a field

Given I have the following snapshot ODM to set the "<range>" range boundary to "<value>":

```
"""
<RangeCheck Comparator='GE' SoftHard='Soft'>
  <CheckValue>low value</CheckValue>
</RangeCheck>
<RangeCheck Comparator='LE' SoftHard='Soft'>
  <CheckValue>high value</CheckValue>
</RangeCheck>
"""

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"
Then I should receive a successful XML response
And I should see "Auto-query for data out of range" field edit check has only
"<range>" range boundary of "<value>"
```

Examples:

	range	value
	lower	80

upper	210	
-------	-----	--

Scenario Outline: Remove non-conformance or future-date field edit check from a field

Given I have "<field edit check>" field edit check enabled in the study design for a field

And I have the following ODM with "<attribute>" attribute set to "No":

"""

```
<ItemDef OID='AE.STARTDATE' ... <attribute> ="No" >
</ItemDef>
```

"""

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"

Then I should receive a successful XML response

And I should see "<field edit check>" field edit check is "disabled" for field "AE.STARTDATE" in Rave

Examples:

field edit check		attribute	
Auto-query for non-conformant data		mdsol:QueryNonConformance	
Auto-query for future date/time		mdsol:QueryFutureDate	

Scenario: Modify range boundaries of field edit check

Given I have the following range field edit checks in study design:

field edit check	low	high	
Auto-query for data out of range	80	210	

And I have an ODM to set the range boundaries to "95" and "175"

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"

Then I should receive a successful XML response

And I should see "Auto-query for data out of range" field edit check has range boundaries of "95" and "175" in Rave

Scenario Outline: For an invalid range data out of range field edit check should fail

Given I have an ODM to set the range boundaries to "<low value>" and "<high value>"

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"

Then I should receive an HTTP response with status code "Bad Request"

And reason code "RWS00091"

And the response message contains " Auto query edit check ranges are not valid"

Examples:

low value	high value	
210	80	
A	Z	
0	-20	

[Back to top](#)

1.7.7.5 Folder Definitions

Use Case

I am *updating* or *retrieving* my study metadata and...

I am updating or retrieving my folder definitions.

My ODM snippet

```
...
<StudyEventDef mdsol:AccessDays="{number}"
  mdsol:StartWinDays="{number}"
  mdsol:TargetDays="{number}"
  mdsol:OverDueDays="{number}"
  mdsol:CloseDays="{number}"
  OID="{oid}"
  Name="{name}"
  Repeating="{Yes | No}"
  Type="{type}"
  Category="{category}">
  <FormRef />
...
</StudyEventDef>
...
```

ODM Attributes:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Value	Notes	Rave mapping
OID	oid (limited to a maximum of 50 characters)	Required	Maps to Folder OID in Rave. See Note 8.
Name	name	Required	Maps to Folder Name in Rave. See Note 9.
Repeating	{Yes No(default) }	Required	Maps to Folder Reusable in Rave. Yes = Reusable option is unchecked in Rave. No = Reusable option is checked in Rave.
Type	{Scheduled Unscheduled Common}	Required	N/A
Category	text (255 characters max)	Optional	Not mapped in Rave.
mdsol:AccessDays	integer	Optional	Maps to Folder Access field in Rave.
mdsol:StartWinDays	integer	Optional	Maps to Folder Start field in Rave.
mdsol:TargetDays	integer	Optional	Maps to Folder Target field in Rave.
mdsol:EndWinDays	integer	Optional	Maps to Folder End field in Rave.
mdsol:OverDueDays	integer	Optional	Maps to Folder Overdue field in Rave.

Attribute	Value	Notes	Rave mapping
mdsol:CloseDays	integer	Optional	Maps to Folder Close field in Rave.

Example:

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<MetaDataVersion OID="1" Name="Draft Number 1" mdsol:PrimaryFormOID="EN">
  <Protocol>
    <StudyEventRef StudyEventOID="SE.SCR" OrderNumber="1" Mandatory="Yes"/>
  </Protocol>
  <StudyEventDef mdsol:AccessDays="20" mdsol:StartWinDays="10" mdsol:TargetDays="5"
    mdsol:OverDueDays="10" mdsol:CloseDays="20" OID="SE.SCR" Name="Screening
Visit"
    Repeating="No" Type="Scheduled" Category="Screening">
      <FormRef FormOID="DM" OrderNumber="1" Mandatory="No"/>
    </StudyEventDef>
  ...
</MetaDataVersion>
...
```

[Back to top](#)

1.7.7.6 Form Definitions

Scenario

I am *updating* or *retrieving* my study metadata and...

I am updating or retrieving my form definitions.

My ODM snippet

```
...
<FormDef
  OID="{oid}"
  Name="{name}"
  Repeating="{Yes|No}"
  mdsol:OrderNumber="{number}"
  mdsol:Active="{Yes|No}"
  mdsol:Template="{Yes|No}"
```

```
mdsol:SignatureRequired="{Yes|No}"  
mdsol:LogDirection="{text}"  
mdsol:DoubleDataEntry="{dde}"  
mdsol:ConfirmationStyle="{style}"  
mdsol:LinkStudyEventOID="{oid}"  
mdsol:LinkFormOID="{oid}">  
<ItemGroupRef />  
</FormDef>  
...
```

ODM Attributes:

Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Value	Notes	Rave mapping
OID	oid (This is limited to a maximum of 50 characters)	Required	Maps to Form OID in Rave. This may be used as a PrimaryFormOID (see MetaDataVersion for more information).
Name	name (limited to OID + (Name) 254 characters maximum length)	Required	Maps to Form Name in Rave. See Note 10.
Repeating	{ Yes No }	Optional	FormRef can have the Mandatory attribute, although it is not imported to Rave.
mdsol:OrderNumber	integer	Optional	Maps to Form ordinal positioning in Rave. See Note 11.
mdsol:Active	{Yes(default) No }	Optional	Yes = Active option is checked for this Form. No = Active option is unchecked for this Form.
mdsol:Template	{Yes No(default) }	Optional	Yes = Template option is checked (this is a template form). No = Template option is unchecked (this is a standard form)

Attribute	Value	Notes	Rave mapping
mdsol:SignatureRequired	{Yes No(default) }	Optional	Yes = Signature Required option is checked for this form. No = Signature Required option is unchecked for this form.
mdsol:LogDirection	{Portrait(default) Landscape}	Optional	Portrait = Log Direction will be Portrait for this form. Landscape = Log Direction will be Landscape for this form. See Note 12 .
mdsol:DoubleDataEntry	{MustNotDDE (default MayDDE MustDDE}	Optional	Maps to the Forms DDE field in Rave. MustNotDDE = Never, MayDDE = As per Site and MustDDE = Always.
mdsol:ConfirmationStyle	{LinkNext NoLink (default) LinkCustom None}	Optional	Maps to Forms Redirect in Rave. See Note 13 .
mdsol:LinkStudyEventOID	oid	Optional	Maps to the Folder dropdown when Redirect = Link Custom in Rave.
mdsol:LinkFormOID	oid	Optional	Maps to the Form dropdown when Redirect = Link Custom in Rave.

Example:

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<FormDef OID="F.CONSENT" Name="Informed Consent" Repeating="No" mdsol:OrderNumber="1"
  mdsol:Active="Yes" mdsol:Template="Yes" mdsol:SignatureRequired="Yes"
  mdsol:LogDirection="Portrait"
  mdsol:DoubleDataEntry="MustDDE" mdsol:ConfirmationStyle="NoLink"
  mdsol:LinkStudyEventOID="SE.2"
  mdsol:LinkFormOID="F.AE">
  <ItemGroupRef ItemGroupOID="ITG00001" Mandatory="Yes" />
</FormDef>
...
```

[Back to top](#)

1.7.7.7 Label Definitions

Use Case

I am *updating* or *retrieving* my study metadata and...

I am updating or retrieving my Label definitions

My ODM snippet

```
...
<ItemGroupDef OID="FRM" Name="FRM" Repeating="No">
  <ItemRef ItemOID="FRM.FLD" OrderNumber="4" Mandatory="No" />
  <mdsol:LabelRef LabelOID="FRM.LBL1" OrderNumber="1" />
  <mdsol:LabelRef LabelOID="FRM.LBL2" OrderNumber="2" />
  <mdsol:LabelRef LabelOID="FRM.LBL3" OrderNumber="3" />
  <mdsol:LabelRef LabelOID="FRM.LBL4" OrderNumber="5" />
  <mdsol:LabelRef LabelOID="FRM.LBL5" OrderNumber="6" />
  <mdsol:LabelRef LabelOID="FRM.LBL6" OrderNumber="7" />
</ItemGroupDef>
<ItemDef OID="FRM.FLD" Name="FLD" DataType="text" Length="10"/>
<mdsol:LabelDef OID="FRM.LBL1" Name="LBL1" />
```

```
<mdsol:LabelDef OID="FRM.LBL2" Name="LBL2" />
<mdsol:LabelDef OID="FRM.LBL3" Name="LBL3" />
<mdsol:LabelDef OID="FRM.LBL4" Name="LBL4" />
<mdsol:LabelDef OID="FRM.LBL5" Name="LBL5" />
<mdsol:LabelDef OID="FRM.LBL6" Name="LBL6" />
...
...
```

mdsol:LabelDef Attributes:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
odm:TranslatedText	odm:TranslatedText	No	
mdsol:ViewRestriction	mdsol:ViewRestriction	No	
mdsol:HelpText	mdsol:HelpText	No	
OID	OID	Yes	
Name	Name	Yes	
Active	Active	No	
Visible	Visible	No	
IndentLevel	IndentLevel	No	
FieldNumber	FieldNumber	No	

Example: mdsol:LabelDef

i Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<ItemGroupDef OID="FRM" Name="FRM" Repeating="No">
    <ItemRef ItemOID="FRM.FLD" OrderNumber="4" Mandatory="No" />
    <mdsol:LabelRef LabelOID="FRM.LBL1" OrderNumber="1" />
    <mdsol:LabelRef LabelOID="FRM.LBL2" OrderNumber="2" />
    <mdsol:LabelRef LabelOID="FRM.LBL3" OrderNumber="3" />
    <mdsol:LabelRef LabelOID="FRM.LBL4" OrderNumber="5" />
    <mdsol:LabelRef LabelOID="FRM.LBL5" OrderNumber="6" />
    <mdsol:LabelRef LabelOID="FRM.LBL6" OrderNumber="7" />
</ItemGroupDef>
<ItemDef OID="FRM.FLD" Name="FLD" DataType="text" Length="10"/>
<mdsol:LabelDef OID="FRM.LBL1" Name="LBL1" />
<mdsol:LabelDef OID="FRM.LBL2" Name="LBL2" />
<mdsol:LabelDef OID="FRM.LBL3" Name="LBL3" />
<mdsol:LabelDef OID="FRM.LBL4" Name="LBL4" />
<mdsol:LabelDef OID="FRM.LBL5" Name="LBL5" />
<mdsol:LabelDef OID="FRM.LBL6" Name="LBL6" />
...
...
```

Scenarios

Scenario: Single label with ordering

Given I have the following ODM that represents a draft in a study:

```
"""
<MetaDataVersion OID="1" Name="Draft 1">
    <FormDef OID="VS" Name="Vital Signs" Repeating="Yes" >
        <ItemGroupRef ItemGroupOID="VS" Mandatory="Yes" />
    </FormDef>
    <ItemGroupDef OID="VS" Name="VS" Repeating="No">
        <ItemRef ItemOID="VS.HEART" OrderNumber="2" Mandatory="No" />
        <mdsol:LabelRef LabelOID="VS.LBL1" OrderNumber="1" />
    </ItemGroupDef>
    <ItemDef OID="VS.HEART" Name="HeartRate" DataType="text" Length="10">
        <Question>
            <TranslatedText><![CDATA[<b>Heart</b> rate]]></TranslatedText>
        </Question>
    </ItemDef>
```

```

<mdsol:LabelDef OID="VS.LBL1" Name="HeartRateLabel">
    <TranslatedText>Please take resting heart rate</TranslatedText>
</mdsol:LabelDef>
</MetaDataVersion>
"""

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"
Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see the following structure for the draft "Draft 1"
| field_sequence_within_form | form | field | field_name      |
field_label           |                               |
|           1             | VS   | LBL1  | HeartRateLabel | Please take
resting heart rate |                               |
|           2             | VS   | HEART | HeartRate       | <b>Heart</b>
rate               |                               |

```

Scenario: Multiple labels at top and bottom of form

Given I have the following ODM that represents a draft in a study:

```

"""

<MetaDataVersion OID="1" Name="Draft 1">
    <FormDef OID="FRM" Name="Example Form" Repeating="Yes" >
        <ItemGroupRef ItemGroupOID="FRM" Mandatory="Yes" />
    </FormDef>
    <ItemGroupDef OID="FRM" Name="FRM" Repeating="No">
        <ItemRef ItemOID="FRM.FLD" OrderNumber="4" Mandatory="No" />
        <mdsol:LabelRef LabelOID="FRM.LBL1" OrderNumber="1" />
        <mdsol:LabelRef LabelOID="FRM.LBL2" OrderNumber="2" />
        <mdsol:LabelRef LabelOID="FRM.LBL3" OrderNumber="3" />
        <mdsol:LabelRef LabelOID="FRM.LBL4" OrderNumber="5" />
        <mdsol:LabelRef LabelOID="FRM.LBL5" OrderNumber="6" />
        <mdsol:LabelRef LabelOID="FRM.LBL6" OrderNumber="7" />
    </ItemGroupDef>
    <ItemDef OID="FRM.FLD" Name="FLD" DataType="text" Length="10"/>
    <mdsol:LabelDef OID="FRM.LBL1" Name="LBL1" />
    <mdsol:LabelDef OID="FRM.LBL2" Name="LBL2" />
    <mdsol:LabelDef OID="FRM.LBL3" Name="LBL3" />
    <mdsol:LabelDef OID="FRM.LBL4" Name="LBL4" />
    <mdsol:LabelDef OID="FRM.LBL5" Name="LBL5" />
    <mdsol:LabelDef OID="FRM.LBL6" Name="LBL6" />
</MetaDataVersion>
"""


```

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see the following structure for the draft "Draft 1"

	field_sequence_within_form	form	field	
	1	FRM	LBL1	
	2	FRM	LBL2	
	3	FRM	LBL3	

	4		FRM		FLD	
	5		FRM		LBL4	
	6		FRM		LBL5	
	7		FRM		LBL6	

Scenario: Labels on log lines

Given I have the following ODM that represents a draft in a study:

```
"""
<MetaDataVersion OID="1" Name="Draft 1">
  <FormDef OID="AE" Name="Adverse Events" Repeating="Yes">
    <ItemGroupRef ItemGroupOID="AE" Mandatory="Yes" />
    <ItemGroupRef ItemGroupOID="AE_LOG_LINE" Mandatory="Yes" />
  </FormDef>
  <ItemGroupDef OID="AE" Name="AE" Repeating="No">
    <ItemRef ItemOID="AE.YN" OrderNumber="1" Mandatory="No" />
    <mdsol:LabelRef LabelOID="AE.LBL1" OrderNumber="2" />
  </ItemGroupDef>
  <ItemGroupDef OID="AE_LOG_LINE" Name="AE" Repeating="Yes">
    <ItemRef ItemOID="AE.TERM" OrderNumber="3" Mandatory="No" />
    <mdsol:LabelRef LabelOID="AE.LBL2" OrderNumber="4" />
  </ItemGroupDef>
  <ItemDef OID="AE.YN" Name="YN" DataType="text" Length="10" />
  <ItemDef OID="AE.TERM" Name="TERM" DataType="text" Length="10" />
  <mdsol:LabelDef OID="AE.LBL1" Name="LBL1" />
  <mdsol:LabelDef OID="AE.LBL2" Name="LBL2" />
</MetaDataVersion>
"""

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"
Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see the following structure for the draft "Draft 1"
```

field_sequence_within_form	form	field	log_line	
1	AE	YN	No	
2	AE	LBL1	No	
3	AE	TERM	Yes	
4	AE	LBL2	Yes	

[Back to top](#)

1.7.7.8 Log Line Definitions

Use Case

I am *updating* or *retrieving* my study metadata and...

I am updating or retrieving my Log Line definitions.

My ODM snippet

```
...  
  <ItemGroupDef OID="{oid}" Name="{name}" Repeating="{Yes|No}">  
    <ItemRef />  
    ...  
  </ItemGroupDef>  
...
```

ODM Attributes:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Value	Notes	Rave mapping
OID	oid	Required	Defines the ItemRef OID referred to by ItemGroupRef elements. Not imported into Rave.
Name	name	Required	N/A
Repeating	{Yes No(default) }	Required	Yes = all fields within the ItemGroupDef will be log line fields in Rave. No = all fields within the ItemGroupDef will be standard fields (that is, non-log line fields).
IsReferenceData	{ Yes No }	Optional	N/A
SASDatasetName	sasName	Optional	N/A
Domain	text (255 characters max)	Optional	N/A
Origin	text (255 characters max)	Optional	N/A
Role	name	Optional	N/A
Purpose	text (255 characters max)	Optional	N/A
Comment	text	Optional	N/A

Example:

 **Note:** Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<ItemGroupDef OID="ITG00002" Domain="PA" Name="Chest X-Ray" Repeating="No">
    <ItemRef ItemOID="ITD00006" Mandatory="Yes" />
    <ItemRef ItemOID="ITD00007" Mandatory="Yes" />
</ItemGroupDef>
...
```

[Back to top](#)

1.7.7.9 Measurement Unit Definitions

Use Case

I am *updating* or *retrieving* my study metadata and...

I am updating or retrieving my Measurement Unit definitions.

My ODM snippet

```
...
<BasicDefinitions>
    <MeasurementUnit OID="{oid}" Name="{name}">
        <Symbol>
            <TranslatedText xml:lang="{lang}">{text}</TranslatedText>
            ...
        </Symbol>
    </MeasurementUnit>
    ...
</BasicDefinitions>
...
```

ODM Attributes:



Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

Attribute	Description	Mandatory?	Notes
MeasurementUnit OID	MeasurementUnit OID	Yes	for example, Kilo
MeasurementUnit Name	MeasurementUnit Name	Yes	for example, Kilo
TranslatedText xml:lang	The translated language	Yes	for example, en

Example:

i Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<BasicDefinitions>
  <MeasurementUnit OID="Kilos" Name="Kilos">
    <Symbol>
      <TranslatedText xml:lang="en">Kilos</TranslatedText>
      <TranslatedText xml:lang="ja">Kilos</TranslatedText>
    </Symbol>
  </MeasurementUnit>
</BasicDefinitions>
...
```

[Back to top](#)

1.7.7.10 Medidata Coder Definitions

Use Case

I am *updating* or *retrieving* my study metadata and...

I am updating or retrieving my Field definitions that use [Rave Coder](#).

My ODM snippet

```
...
<ItemDef >
  <mdsol:CodingRequestDef
    CodingDictionaryOID="{coding-dictionary-oid}"
    CodingWorkflowOID="{workflow-oid}"
    CodingDictionaryLevelOID="{coding-level}"
    CodingModeType="{coding-mode}"
    CodingPriority="{coding-priority}"
    CodingLocale="{locale}"
```

```
CodingApprovalType="{approval-type}" >  
  
<mdsol:DictionaryLevelComponentRef  
    ItemRef="{component-oid}"  
    DictionaryLevelComponentOID="{component-level}" />  
  
<mdsol:SupplementalRef  
    ItemRef="{supplemental-oid}" />  
  
</mdsol:CodingRequestDef>  
</ItemDef>  
...
```

CodingRequestDef Attributes:

Attribute	Value	Notes	Rave mapping
CodingDictionaryOID	Coding dictionary oid (50 characters max)	Required	Coding Dictionary
CodingWorkflowOID	text (255 characters max)	Required	N/A
CodingDictionaryLevelOID	text (255 characters max)	Required	Coding level
CodingPriority	integer	Required	Priority
CodingLocale	text (255 characters max)	Required	N/A
CodingApprovalType	{AutoApproval ManualApproval None}	Required	Workflow approval variables
CodingModeType	{Coder Classic}	Required	N/A

DictionaryLevelComponentRef Attributes:

Attribute	Value	Notes	Rave mapping
ItemRef	text (255 characters max)	Required	Component Terms Name
DictionaryLevelComponentOID	text (255 characters max)	Required	Component Terms Component Name

SupplementalRef Attribute:

Attribute	Value	Notes	Rave mapping
ItemRef	text (255 characters max)	Required	Supplemental Terms Name

Example:

i Note: Some attributes are omitted from the ODM document for clarity. See [ODM Schema](#) for more details.

```
...
<ItemDef OID="CM.MEDICINE" Name="Medicine" DataType="text" Length="100">
  <mdsol:CodingRequestDef CodingDictionaryOID="WHODrugB2"
    CodingWorkflowOID="DefaultWorkflow"
      CodingDictionaryLevelOID="PRODUCT"
      CodingModeType="Coder" CodingPriority="1" CodingLocale="eng"
      CodingApprovalType="None">
    <mdsol:DictionaryLevelComponentRef ItemRef="CM.COMPONENT"
      DictionaryLevelComponentOID="DRUGRECORDNUMBER" />
    <mdsol:SupplementalRef ItemRef="CM.SUPPLEMENTAL" />
  </mdsol:CodingRequestDef>
</ItemDef>
...

```

Scenarios

Background:

Given I am a user authorised to perform coding

And there is a coding dictionary "WHODrugB2" for Coder with the following terms:

Level	Term	Code	IsDefault
ATC	Local and oral treatment	A02AA	false
PRODUCT	Alevdon	00020001004	true
INGREDIENT	Paracetamol	03454301068	false

And the coding dictionary "WHODrugB2" has the following components:

Level	Component
PRODUCT	DRUGRECORDNUMBER
PRODUCT	SEQUENCENUMBER

And there is a study that is registered with iMedidata

And the coding dictionary "WHODrugB2" is registered for this study with workflow "DefaultWorkflow"

Scenario: Configure one field for Medidata Coder

Given I have the following ODM that represents a draft in a study:

"""

```
<MetaDataVersion OID="1" Name="Draft Number 1">
  <FormDef OID="CM" Name="Concomitant Medications" Repeating="Yes">
    <ItemGroupRef ItemGroupOID="CM" Mandatory="Yes" />
```

```

</FormDef>
<ItemGroupDef OID="CM" Name="CM" Repeating="No">
    <ItemRef ItemOID="CM.MEDICINE" OrderNumber="1" Mandatory="No" />
    <ItemRef ItemOID="CM.STARTDATE" OrderNumber="2" Mandatory="No" />
</ItemGroupDef>
<ItemDef OID="CM.MEDICINE" Name="Medicine" DataType="text" Length="100"
mdsol:ControlType="Text"
    mdsol:Active="Yes" mdsol:Visible="Yes">
    <mdsol:CodingRequestDef CodingDictionaryOID="WHODrugB2"
CodingWorkflowOID="DefaultWorkflow"
        CodingDictionaryLevelOID="INGREDIENT"
        CodingModeType="Coder" CodingPriority="1" CodingLocale="eng"
CodingApprovalType="None"/>
</ItemDef>
<ItemDef OID="CM.STARTDATE" Name="Start Date" DataType="date"
mdsol:DateTimeFormat="dd MMM yyyy"
    mdsol:ControlType="DateTime" mdsol:Active="Yes" mdsol:Visible="Yes" />
</MetaDataVersion>
"""

```

When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"

Then I should receive a successful response

And the body of the response should be XML indicating success

And I should see the following structure for the draft "Draft Number 1"

form	field	field name	
CM	MEDICINE	Medicine	
CM	STARTDATE	Start Date	

Scenario: Configure one field for Medidata Coder that uses one Supplemental field

Given I have the following ODM that represents a draft in a study:

```

"""
<MetaDataVersion OID="1" Name="Draft Number 1">
    <FormDef OID="CM" Name="Concomitant Medications" Repeating="Yes">
        <ItemGroupRef ItemGroupOID="CM" Mandatory="Yes" />
    </FormDef>
    <ItemGroupDef OID="CM" Name="CM" Repeating="No">
        <ItemRef ItemOID="CM.MEDICINE" OrderNumber="1" Mandatory="No" />
        <ItemRef ItemOID="CM.STARTDATE" OrderNumber="2" Mandatory="No" />
        <ItemRef ItemOID="CM.SUPPLEMENTAL" OrderNumber="3" Mandatory="No" />
    </ItemGroupDef>
    <ItemDef OID="CM.MEDICINE" Name="Medicine" DataType="text" Length="100"
mdsol:ControlType="Text"
        mdsol:Active="Yes" mdsol:Visible="Yes">
        <mdsol:CodingRequestDef CodingDictionaryOID="WHODrugB2"
CodingWorkflowOID="DefaultWorkflow"
            CodingDictionaryLevelOID="INGREDIENT"
            CodingModeType="Coder" CodingPriority="1" CodingLocale="eng"
CodingApprovalType="None">
            <mdsol:SupplementalRef ItemRef="CM.SUPPLEMENTAL" />

```

```

        </mdsol:CodingRequestDef>
    </ItemDef>
    <ItemDef OID="CM.STARTDATE" Name="Start Date" DataType="date"
mdsol:DateTimeFormat="dd MMM yyyy"
        mdsol:ControlType="DateTime" mdsol:Active="Yes" mdsol:Visible="Yes" />
    <ItemDef OID="CM.SUPPLEMENTAL" Name="Supplemental" DataType="text" Length="100"
mdsol:ControlType="Text"
        mdsol:Active="Yes" mdsol:Visible="Yes" />
</MetaDataVersion>
"""
When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"
Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see the following structure for the draft "Draft Number 1"
| form | field           | field_name   |
| CM   | MEDICINE         | Medicine     |
| CM   | STARTDATE        | Start Date   |
| CM   | SUPPLEMENTAL     | Supplemental |

```

Scenario: Configure one field for Medidata Coder that uses one Component field
Given I have the following ODM that represents a draft in a study:

```

<MetaDataVersion OID="1" Name="Draft Number 1">
    <FormDef OID="CM" Name="Concomitant Medications" Repeating="Yes">
        <ItemGroupRef ItemGroupOID="CM" Mandatory="Yes" />
    </FormDef>
    <ItemGroupDef OID="CM" Name="CM" Repeating="No">
        <ItemRef ItemOID="CM.MEDICINE" OrderNumber="1" Mandatory="No" />
        <ItemRef ItemOID="CM.STARTDATE" OrderNumber="2" Mandatory="No" />
        <ItemRef ItemOID="CM.COMPONENT" OrderNumber="3" Mandatory="No" />
    </ItemGroupDef>
    <ItemDef OID="CM.MEDICINE" Name="Medicine" DataType="text" Length="100"
mdsol:ControlType="Text"
        mdsol:Active="Yes" mdsol:Visible="Yes">
        <mdsol:CodingRequestDef CodingDictionaryOID="WHODrugB2"
CodingWorkflowOID="DefaultWorkflow"
            CodingDictionaryLevelOID="PRODUCT"
            CodingModeType="Coder" CodingPriority="1" CodingLocale="eng"
CodingApprovalType="None">

            <mdsol:DictionaryLevelComponentRef ItemRef="CM.COMPONENT"
DictionaryLevelComponentOID="DRUGRECORDNUMBER" />

        </mdsol:CodingRequestDef>
    </ItemDef>
    <ItemDef OID="CM.STARTDATE" Name="Start Date" DataType="date"
mdsol:DateTimeFormat="dd MMM yyyy"
        mdsol:ControlType="DateTime" mdsol:Active="Yes" mdsol:Visible="Yes" />
    <ItemDef OID="CM.COMPONENT" Name="Component" DataType="text" Length="100"
mdsol:ControlType="Text"

```

```
    mdsol:Active="Yes" mdsol:Visible="Yes" />
</MetaDataVersion>
"""
When I post the ODM to the URL "/metadata/studies/{study oid}/drafts"
Then I should receive a successful response
And the body of the response should be XML indicating success
And I should see the following structure for the draft "Draft Number 1"
| form | field      | field_name |
| CM   | MEDICINE   | Medicine   |
| CM   | STARTDATE  | Start Date |
| CM   | COMPONENT  | Component |
```

1.7.8 Error Responses - Complete List

This page contains details of all possible outcomes from the RWS request services. The response format varies between the request services, and may not contain all of this information.

**Note:** See also

- [Troubleshooting guide](#)
- [The pre-installed Log Messages Custom Dataset](#)

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
N/A	200	signifies a successful import	N/A
RWS00005	401	unauthorised	User is locked out.
RWS00006	500	Internal Server error	Internal Error - Web Service currently unavailable.
RWS00008	401	unauthorised	Incorrect login and password combination.
RWS00008	401	unauthorised	Incorrect login and password combination.
RWS00009	403	forbidden	Insecure HTTP requests are forbidden by configuration. Only secure HTTPS requests are allowed.
RWS00010	400	bad request	This is not a valid XML document.
RWS00011	400	bad request	This is not a valid ODM 1.2 standard document. {0}
RWS00012	400	bad request	Filetype '{0}' not supported.
RWS00013	400	bad request	Data not imported as more than one study was found in the document.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00014	404	not found	Study does not exist.
RWS00015	409	conflict	Site has no CRF version assigned.
RWS00016	409	conflict	Site not active.
RWS00017	404	not found	Site does not exist.
RWS00018	404	not found	Subject not authorised.
RWS00019	409	conflict	Subject not active.
RWS00020	400	bad request	Data for more than one subject was found in the document.
RWS00021	409	conflict	Transaction not processed as more than one subject has the same identifier at the given site.
RWS00022	409	conflict	Transaction not processed as more then one subject (without site details) identified in study.
RWS00023	404	not found	Subject does not exist.
RWS00024	409	conflict	Subject already exists.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00025	400	bad request	Subjects must have site details to be created.
RWS00026	404	not found	Folder not authorised.
RWS00027	409	conflict	Folder does not exist.
RWS00028	404	not found	Folder not found.
RWS00029	409	conflict	Folder already exists.
RWS00030	400	bad request	Non sequential or non consecutive folders are not permitted.
RWS00031	404	not found	Form not authorised.
RWS00032	409	conflict	Form does not exist.
RWS00033	404	not found	Form does not exist.
RWS00034	409	conflict	Form already exists.
RWS00035	400	bad request	Non sequential or non consecutive forms are not permitted.
RWS00036	409	conflict	Form locked.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00037	404	not found	Record does not exist.
RWS00038	409	conflict	Record already exists.
RWS00039	400	bad request	Non sequential or non consecutive records are not permitted.
RWS00040	409	conflict	Record locked.
RWS00041	404	not found	Field does not exist.
RWS00042	404	not found	Field not authorised.
RWS00043	409	conflict	Field already exists.
RWS00044	409	conflict	Field locked.
RWS00045	409	conflict	Hidden field/ field not visible.
RWS00046	400	bad request	Picture/file upload field cannot be updated.
RWS00047	400	bad request	Data not in dictionary.
RWS00048	409	conflict	Transaction on frozen field is not permitted.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00049	409	conflict	Transaction on derived field is not permitted.
RWS00050	404	not found	Unit Dictionary does not exist.
RWS00051	404	not found	Unit Dictionary entry does not exist.
RWS00052	404	not found	Requested web service does not exist.
RWS00053	404	not found	Cannot create more than one reusable folder at the same level.
RWS00054	403	forbidden	Removing a subject is not permitted.
RWS00055	404	not found	RWS URL does not exist.
RWS00056	400	bad request	This is not a valid ODM 1.3 standard document. {0}
RWS00057	403	forbidden	Lock not authorized.
RWS00058	403	forbidden	Unlock not authorized.
RWS00059	403	forbidden	Freeze not authorized.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00060	403	forbidden	Unfreeze not authorized.
RWS00061	403	forbidden	Review not authorized.
RWS00062	403	forbidden	UnReview not authorized.
RWS00063	404	not found	Review group name not found.
RWS00064	409	conflict	Field not associated with the Review Group.
RWS00065	403	forbidden	Verify not authorized.
RWS00066	403	forbidden	UnVerify not authorized.
RWS00067	403	forbidden	Duplicate ReviewGroup name.
RWS00068	403	forbidden	Open query not authorized.
RWS00069	403	forbidden	Cancel query not authorized.
RWS00070	403	forbidden	Close query not authorized.
RWS00071	404	not found	Query marking group name not found.
RWS00072	404	not found	Query repeat key not found.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00073	403	forbidden	Answer query not authorized.
RWS00074	403	forbidden	Query not does not require response.
RWS00075	403	forbidden	Forward query not authorized.
RWS00076	403	forbidden	Place sticky not authorized.
RWS00077	404	not found	Sticky marking group name not found.
RWS00078	403	forbidden	Acknowledge sticky not authorized.
RWS00079	404	not found	Sticky repeat key not found.
RWS00080	403	forbidden	Add protocol deviation not authorized.
RWS00081	403	forbidden	Edit protocol deviation not authorized.
RWS00082	403	forbidden	Remove protocol deviation not authorized.
RWS00083	404	not found	Protocol deviation repeat key not found.
RWS00084	404	not found	Protocol deviation class not found.
RWS00085	404	not found	Protocol deviation code not found.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00086	403	forbidden	Translation not authorized.
RWS00087	403	forbidden	Item does not require translation.
RWS00088	409	conflict	Record restricted by max limit.
RWS00089	403	forbidden	Record inactivation not authorised.
RWS00090	403	forbidden	Field requiring translation, update in default locale not authorised.
RWS00091	400	bad request	Architect loader error '{0}'
RWS00092	404	not found	CRF version not found.
RWS00093	403	forbidden	Data posted to incorrect study URL .
RWS00094	403	forbidden	Query is already closed or cancelled.
RWS00095	400	bad request	Updating a field associated to a dynamic search list is not supported.
RWS00096	403	forbidden	Field not enabled for coding.
RWS00097	403	forbidden	User not authorised for coding.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00098	409	conflict	Coding dictionary not associated with the field.
RWS00099	404	not found	Coding dictionary level/column does not exist.
RWS00100	409	conflict	Datapoint context value does not match current value.
RWS00101	403	forbidden	Non log line form, cannot insert multiple records.
RWS00102	403	forbidden	Data posted to incorrect draft URL.
RWS00103	409	conflict	Vendor extension mdsol:Attribute namespace-name pair already exists.
RWS00104	404	not found	Vendor extension mdsol:Attribute namespace-name pair does not exist.
RWS00105	404	not found	Vendor extension mdsol:Attribute namespace does not exist.
RWS00106	404	not found	Draft not found.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00107	409	conflict	Transaction type not supported for this element.
RWS00108	403	forbidden	Identifier too long.
RWS00109	409	conflict	Duplicate study identified.
RWS00110	409	conflict	Can not accept more than one element for Coding Data.
RWS00112	403	forbidden	Dictionary item does not require specify value.
RWS00113	403	forbidden	Simultaneous coding and translating not supported.
RWS00114	409	conflict	Coding value already exists.
RWS00115	409	conflict	Can not accept 2 elements for Translated Data.
RWS00116	403	forbidden	Query cannot be closed without answer.
RWS00117	403	forbidden	Preceding query does not exist.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00118	403	forbidden	Cannot re-query an unanswered query.
RWS00119	403	forbidden	Cannot Forward Query to the same Marking Group.
RWS00120	400	bad request	Query cannot be opened with query repeat key.
RWS00121	403	forbidden	Query already answered.
RWS00122	403	forbidden	Query can only be forwarded or cancelled.
RWS00123	403	forbidden	Coding integration not available.
RWS00124	404	not found	Segment does not exist.
RWS00125	404	not found	Coding dictionary does not exist.
RWS00126	404	not found	Coding dictionary version does not exist.
RWS00127	403	forbidden	Coding dictionary version not associated with the segment.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00128	403	forbidden	Coding dictionary version not used in the study.
RWS00129	403	forbidden	Query requires response.
RWS00130	415	unsupported media type	Request format not found.
RWS00131	406	not acceptable	Response format not found.
RWS00132	403	forbidden	Formats supplied in header and query string do not match.
RWS00133	500	Internal Server error	Failure during format transformation.
RWS00134	503	service unavailable	Dataset does not exist.
RWS00134	404	not found	Dataset does not exist.
RWS00135	403	forbidden	Dataset is not configured for this request.
RWS00136	403	forbidden	User does not have permission for this action.



Note: Log lines are not processed if entry or view

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
			restrictions are set on all the log fields for the role that executes RWS (Rave 2017.2.3 or later).
RWS00137	404	not found	Format not found.
RWS00138	400	bad request	Not a valid xslt document.
RWS00139	400	bad request	Invalid parameter supplied.
RWS00140	501	Not Implemented	This service is not implemented.
RWS00141	403	forbidden	Cannot set the lab for this particular form.
RWS00142	404	not found	Lab does not exist.
RWS00143	404	not found	Log line not found.
RWS00144	404	not found	Unique log line not found.
RWS00145	409	conflict	CrfVersion already exists.
RWS00146	404	not found	User does not exists.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00147	409	conflict	Study not assigned to user.
RWS00148	404	not found	Dataset does not exist.
RWS00149	500	Internal Server error	Invalid object name.
RWS00150	404	not found	Dataset format not found.
RWS00151	400	bad request	Invalid column name.
RWS00152	400	bad request	Invalid data type of filter value.
RWS00153	409	conflict	Unique form not found.
RWS00154	404	not found	No matching form found.
RWS00155	409	conflict	No context provided.
RWS00156	400	bad request	Multiple folders found for the tag.
RWS00157	400	bad request	Multiple forms found for the tag.
RWS00158	400	bad request	Multiple fields found for the tag.
RWS00159	409	conflict	Unique folder not found.
RWS00160	404	not found	No matching folder found.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00161	500	Internal Server error	Invalid template.
RWS00162	200	Success	Queue item not found.
RWS00163	404	not found	Tag does not exist.
RWS00164	400	bad request	This is not a valid ODM 1.3 standard document.
RWS00165	401	unauthorised	User is not authorised to delete items from the queue.
RWS00166	400	bad request	Coding can only be applied to the items with transaction type of Context.
RWS00167	404	not found	iMedidata authentication exception. '{0}'
RWS00168	401	unauthorised	MAuth authentication exception. '{0}'
RWS00169	503	service unavailable	MAuth server is unavailable. '{0}'
RWS00170	501	Not Implemented	Authentication type not supported.
RWS00171	403	forbidden	Using the Subject UUID is not authorized for this action.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00172	400	bad request	Request file size exceeds the allowed limit.
RWS00173	400	bad request	Unexpected encoding.
RWS00174	400	bad request	Attribute '{0}' is incompatible with the field's control type.
RWS00175	400	bad request	Request size exceeds the allowed limit: '{0}'
RWS00176	400	bad request	A potentially dangerous value was detected from the client request.
RWS00177	500	Internal Server error	Error while logging the request.
RWS00178	403	forbidden	User does not have the permission to use this service.
RWS00179	404	not found	Workflow with OID '{0}' does not exist.
RWS00180	404	not found	Coding dictionary with OID '{0}' does not exist.
RWS00181	400	bad request	DictionaryLevelData element required.

RWS Reason code	HTTP return code	HTTP Code Description	Response Message
RWS00182	400	bad request	You must use the iMedidata StudyOIDType.
RWS00183	400	bad request	Study registration requires at least one dictionary and workflow.
RWS00184	403	forbidden	User not authorised to reactivate the logline.
RWS00185	404	not found	Logline repeat key not found.
RWS00204	404	not found	Form is inactivated and could not be updated.
RWS00205	404	not found	Form does not exist in the designated folder.
RWS00206	403	forbidden	Exception occurred during applying changes. '<Detail information here>'.
RWS00209	500	Internal Server error	Your changes cannot be saved because a subject has been updated by another user.
UnknownError	500	Internal Server error	Internal Error - Web service currently unavailable.

1.7.9 Specify a Study OID

A clinical study in Rave is specified as a combination of the study name and the environment name. This applies throughout this guide where “{study-oid}” is specified. The clinical study is represented with three possible formats:

```
{study-name} ({environment-name})  
{study-name} ({environment-name})  
{study-name}
```

In the 3rd example, where the environment name is not specified, the production environment is the default environment.

Therefore, for studies in the production environment, any of the three formats are valid. For auxiliary studies, the environment must be specified. Consequently only the first two are valid.

Edge case scenario

Rave functionality permits, although it is not advisable, to include parentheses in the study name. Therefore, it is theoretically possible to have ambiguity, but in such cases the production environment will take precedence.

This precedence is described in the following scenario:

```
Scenario: A study name that is indistinguishable from a study-oid  
Given there are two studies having the names "abc" and "abc(TEST)"  
And both studies exist in a "TEST" auxiliary environment  
And both studies exist in a production environment  
When I specify the "{study-oid}" as "abc(TEST)"  
Then the operation will apply to the "abc(TEST)" study in the production environment  
And the operation will not apply to the "abc" study in the "TEST" environment
```

1.7.10 URL Special Character Escaping

This functionality has been implemented to test the use of special characters in a study name.

Use Case

As an integrating system

I want to request any resource in Rave

so that I can perform any action I need.

i **Important:** ‘?’ is a special character and can **not** be used in the URL path. This is due to the fact that all characters placed after the question mark are considered to be part of the query string.

Assumptions

1. Include forward slash and back slash.
2. In the url requests, the study name must be url encoded.
3. In the post request xml body, the study name must be xml encoded.

Scenarios

Background:

Given I am a Rave user

Scenario Outline: Allow special characters in study name in the metadata urls

 Given the background

 And there is a <project> with a name containing "<special_char>" with THE following structure

	folder	form	field	
	SCREEN	DM	BIRTHDAY	

 When I make an HTTP GET request to the URL "/metadata/<projectsegment>/ {study oid}<requesttype>"

 Then I should receive a successful response

Examples:

special_char	project	projectsegment	requesttype
%2F	study	studies	drafts
%5C	study	studies	drafts
%2F	study	studies	versions
%5C	study	studies	versions
'	study	studies	versions
%27	study	studies	versions
;	study	studies	versions

%3B	study	studies	versions
(study	studies	versions
%28	study	studies	versions
)	study	studies	versions
%29	study	studies	versions
%23	study	studies	versions
%2F	library	libraries	versions
%5C	library	libraries	versions
'	library	libraries	versions
%27	library	libraries	versions
;	library	libraries	versions
%3B	library	libraries	versions
(library	libraries	versions
%28	library	libraries	versions
)	library	libraries	versions
%29	library	libraries	versions
%23	library	libraries	versions
%2F	library	libraries	drafts
%5C	library	libraries	drafts
'	library	libraries	drafts
%27	library	libraries	drafts
;	library	libraries	drafts
%3B	library	libraries	drafts
(library	libraries	drafts
%28	library	libraries	drafts
)	library	libraries	drafts
%29	library	libraries	drafts
%23	library	libraries	drafts

Scenario Outline: Allow special characters in GET service requests

Given the background

And there is a <project> with the name containing "<special_char>" with the following structure

folder	form	field	value
SCREEN	DM	BRTHDTC	12 Jan 2001

When I have made a RWS <service> request for a study with dynamic name

Then I should receive a successful response

And the body of the response should be an XML document

Examples:

service	special_char	project
GET Subjects in a Study	%2F	study
GET Study Draft list	%2F	study
GET Studies	%2F	study
GET Metadata Study Attribute	%2F	study
GET Study Crf Versions List	%2F	study
GET Study Metadata Version	%2F	study
GET Metadata Study Attribute	%2F	study
GET Library Metadata Version	%2F	library

GET Metadata Library Attribute	%2F	library	
GET Library Crf Versions List	%2F	library	
GET Metadata Library Attribute	%2F	library	
GET Library Metadata Version	%2F	library	
GET Subjects in a Study	%5C	study	
GET Study Draft list	%5C	study	
GET Studies	%5C	study	
GET Metadata Study Attribute	%5C	study	
GET Study Crf Versions List	%5C	study	
GET Study Metadata Version	%5C	study	
GET Metadata Study Attribute	%5C	study	
GET Library Metadata Version	%5C	library	
GET Metadata Library Attribute	%5C	library	
GET Library Crf Versions List	%5C	library	
GET Metadata Library Attribute	%5C	library	
GET Library Metadata Version	%5C	library	
GET Subjects in a Study	%23	study	
GET Study Draft list	%23	study	
GET Studies	%23	study	
GET Metadata Study Attribute	%23	study	
GET Study Crf Versions List	%23	study	
GET Study Metadata Version	%23	study	
GET Metadata Study Attribute	%23	study	
GET Library Metadata Version	%23	library	
GET Metadata Library Attribute	%23	library	
GET Library Crf Versions List	%23	library	
GET Metadata Library Attribute	%23	library	
GET Library Metadata Version	%23	library	
GET Subjects in a Study	%27	study	
GET Study Draft list	%27	study	
GET Studies	%27	study	
GET Metadata Study Attribute	%27	study	
GET Study Crf Versions List	%27	study	
GET Study Metadata Version	%27	study	
GET Metadata Study Attribute	%27	study	
GET Library Metadata Version	%27	library	
GET Metadata Library Attribute	%27	library	
GET Library Crf Versions List	%27	library	
GET Metadata Library Attribute	%27	library	
GET Library Metadata Version	%27	library	
GET Subjects in a Study	%3B	study	
GET Study Draft list	%3B	study	
GET Studies	%3B	study	
GET Metadata Study Attribute	%3B	study	
GET Study Crf Versions List	%3B	study	
GET Study Metadata Version	%3B	study	
GET Metadata Study Attribute	%3B	study	
GET Library Metadata Version	%3B	library	
GET Metadata Library Attribute	%3B	library	
GET Library Crf Versions List	%3B	library	

GET Metadata Library Attribute	%3B	library	
GET Library Metadata Version	%3B	library	
GET Subjects in a Study	%28	study	
GET Study Draft list	%28	study	
GET Studies	%28	study	
GET Metadata Study Attribute	%28	study	
GET Study Crf Versions List	%28	study	
GET Study Metadata Version	%28	study	
GET Metadata Study Attribute	%28	study	
GET Library Metadata Version	%28	library	
GET Metadata Library Attribute	%28	library	
GET Library Crf Versions List	%28	library	
GET Metadata Library Attribute	%28	library	
GET Library Metadata Version	%28	library	
GET Subjects in a Study	%29	study	
GET Study Draft list	%29	study	
GET Studies	%29	study	
GET Metadata Study Attribute	%29	study	
GET Study Crf Versions List	%29	study	
GET Study Metadata Version	%29	study	
GET Metadata Study Attribute	%29	study	
GET Library Metadata Version	%29	library	
GET Metadata Library Attribute	%29	library	
GET Library Crf Versions List	%29	library	
GET Metadata Library Attribute	%29	library	
GET Library Metadata Version	%29	library	

Scenario Outline: Get successful Response status on POST service requests for Post Metadata Draft

Given the background

And I have the following snapshot ODM that represents a study containing "<special_char>":

"""

```
<Study OID="{study oid}">
    <GlobalVariables>
        <StudyName>{study oid}</StudyName>
        <StudyDescription></StudyDescription>
        <ProtocolName>{study oid}</ProtocolName>
    </GlobalVariables>
    <MetaDataTable OID="1" Name="Draft Number 1" >
    </MetaDataTable>
</Study>
"""
```

When I post the ODM to the URL "/metadata/<url_segment>/{study oid}/drafts"
Then I should receive a successful response

Examples:

url_segment	special_char	
studies	%2F	
studies	%5C	

```

| studies      | %23
| studies      | ;
| studies      | '
| studies      | %3B
| studies      | (
| studies      | %28
| studies      | )
| studies      | %29
| libraries    | %2F
| libraries    | %5C
| libraries    | %23
| libraries    | ;
| libraries    | '
| libraries    | %3B
| libraries    | (
| libraries    | %28
| libraries    | )
| libraries    | %29

```

Scenario Outline: Get successful Response status on POST service requests for Post Metadata Draft Attributes

Given the background

And there is a <project> with the name containing "<special_char>" with the following structure

form	field	
VS	HEART	

When I update the <project> draft with the following ODM to the URL "/metadata/<url_segment>/<study oid>/drafts/<draft id>/attributes":

"""

```

<MetaDataVersion OID="{draft id}" Name="Draft 1">
    <FormDef OID="VS" Name="Vital Signs" Repeating="Yes" >
        <ItemGroupRef ItemGroupOID="VS" Mandatory="Yes" />
    </FormDef>
    <ItemGroupDef OID="VS" Name="VS" Repeating="No">
        <ItemRef ItemOID="VS.HEART" OrderNumber="1"
Mandatory="No">
            <mdsol:Attribute
Namespace="TestAttributesABC" Name="Ignore" Value="No" TransactionType="Upsert"/>
        </ItemRef>
    </ItemGroupDef>
    <ItemDef OID="VS.HEART" Name="HeartRate" DataType="text"
Length="200" mdsol:FieldNumber="1"/>
</MetaDataVersion>
"""

```

Then I should receive a successful response.

Examples:

project	url_segment	special_char	
study	studies	%2F	
library	libraries	%2F	
study	studies	%23	

library	libraries	%23	
study	studies	%27	
library	libraries	%27	
study	studies	'	
library	libraries	'	
study	studies	;	
library	libraries	;	
study	studies	%3B	
library	libraries	%3B	
study	studies	%28	
library	libraries	%28	
study	studies)	
library	libraries)	
study	studies	%29	
library	libraries	%29	
study	studies	(
library	libraries	(

Scenario Outline: Allowed special characters in the study name in subject urls
 Given the background

And there is a study with the name containing "<special_char>" with the following structure

	folder	form	field	value	
	SCREEN	DM	BIRTHDT	12 Jan 2001	

When I make an HTTP GET request to the URL "/studies/{study oid}/subjects"

Then I should receive a successful response

Examples:

special_char	
%2F	
%5C	
%23	
%27	
;	
'	
%3B	
(
%28	
)	
%29	

[Back to top](#)

1.7.11 Product Mapping Terminology



Note: In most cases of data transfer, the primary identifier of interest is the Study Environment UUID. As Study Environment is not an ODM concept, any reference to StudyUUID can be considered an alias of the Study Environment UUID.

If you are familiar with Classic Rave and Rave EDC terminology, the following table helps you learn the new terminology used in the Medidata Platform.

*

Oops, it seems that you need to place a table or a macro generating a table within the Table Filter macro.

The table is being loaded. Please wait for a bit ...

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>Customer — A company with a contract with Medidata to provide software. A customer owns a study group.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none">Provided when iMedidata administrators add study groups.	<p>Client Division (CD) — A company with a contract with Medidata to provide software. A client division comprises client division schemes, client division organizations, and client division people.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none">Added by Professional Services.	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>Study Group — A collection of studies configured to access certain applications. Study groups comprise studies, sites, apps, users, and eLearning. There is a one-to-one relationship between study groups and Classic Rave URLs.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> • When you integrate with Classic Rave, a study group is a collection of studies on a particular Classic Rave URL. 	<p>Client Division Scheme (CDS) — The association between a group of a customer's studies and a group of Medidata's applications.</p> <p>Client Division Scheme comprises study environments, study environment sites, roles, users, and eLearning,</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> • You can assign users to CDSes. Integration of studies with potential access to a number of apps. 	<p>Client Division Schemes</p>

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<ul style="list-style-type: none"> • You can assign users at the study group level. <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Manage Study Groups in iMedidata 	<u>For more information:</u> <ul style="list-style-type: none"> • Manage Client Divisions 	
Study — In iMedidata, a study is a collection of data in Classic Rave and other platform modules. A study in iMedidata is equivalent to a project in Classic Rave Architect, plus six or more study environments. A study belongs to a study group.	Study/Study Environments (SE) — In Cloud Administration, a study is a collection of data in Rave EDC and other platform modules. A study in Rave EDC is equivalent to a project in Rave Architect, plus six or more study environments. A study belongs to a client division.	Studies Study Environments

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> • For iMedidata, a study is a member of a study group. You can assign users at the study level. • You invite study owners to manage the study and invite site and study personnel. • When integrated with Classic Rave, a study in iMedidata follows the same definition as a Classic Rave 	<p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> • You can assign users to study environments. • When you create a study in Cloud Administration, it comes with six environments: Production, Development, Interactive Verification, Functional Testing, User Acceptance Testing, and Training. • The non-production environment 	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>project. You create one study in iMedidata for each project-environment combination.</p> <ul style="list-style-type: none"> Creating a study in iMedidata for Classic Rave relies on a naming convention. For example, if my project name is ABCD123 and I want to create a DEV environment, then I enter "ABCD123 (DEV)" in the Study 	<p>nts surface on the iMedidata home page with the environment name in parentheses; such as, "Study (DEV)." Client Division Admins or Study Environment Admins in prod can create additional environments, which appear as "Other (Envname)" in Cloud Administration and Cloud Navigation and "Study (ENVNAME)"</p>	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>Name field and ABCD123 in the Protocol Number field in iMedidata. This creates the ABCD123 Project in Rave Architect and also creates an auxiliary environment named DEV.</p> <ul style="list-style-type: none"> In iMedidata, you create a separate study for each auxiliary environment that you need; for example, ABCD123 (UAT), 	<p>E)" in iMedidata.</p> <p><u>For more information:</u></p> <ul style="list-style-type: none"> • About Study Environments in Cloud Administration • Create Studies 	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>ABCD123 (TRAIN), and so on.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • About iMedidata- Classic Rave Integration • Manage Studies in iMedidata 		
<p>Study Group Owner – Maintains the study group, creates studies, invites study owners, and manages eLearning mappings that gate access to Classic Rave studies and apps like Safety Gateway.</p> <p><u>For more Information:</u></p>	<p>Client Division Admin — Maintains studies and creates associated objects at the client division level.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Manage Client Divisions 	N/A

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<ul style="list-style-type: none"> • Study Group Owners 		
<p>Study Owner — A person with ultimate administrative rights to the Study. Creates study-sites, adds existing sites, invites study and site personnel, and in some cases may modify eLearning mappings.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Manage Studies in iMedidata 	<p>Study Environment Admin — Maintains Study Environments and creates Study Environment-associated objects.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Manage Studies 	N/A
<p>Role — A generic term that represents a collection of permissions granted to a user</p>	<p>Role (aka Platform Role or Client Role) — A combination of building blocks</p>	<p>Medidata Roles Configuration Type Roles</p>

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>to determine what tasks the user can perform within Classic and Rave EDC.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> When you integrate with iMedidata, you configure and associate Classic Rave roles with <ul style="list-style-type: none"> EDC, user group s, security roles, security group role assig nmen 	<p>that you can assign to users. A role comprises all things a person can do for across all products.</p> <p>However, a person can still be assigned multiple roles.</p> <p>For more information:</p> <ul style="list-style-type: none"> Manage Roles 	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>ts, and ◦ Safety Gateway.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Roles • About iMedidata Rave Architect and Architect Security 		
N/A	Building Block — A set of permissions representing a complete, supported workflow in one product. A set of building blocks may include permissions for any number of	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
	<p>apps and contains the app-level permissions in a given role assigned to a user.</p> <p>For more Information:</p> <ul style="list-style-type: none"> Cloud Administration Role Privileges Table 	
<p>Site — The place where clinical trials are performed. iMedidata has the same site object as Classic Rave.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> As in Classic Rave, you can create new sites or add existing 	<p>Client Division Organization (CDO) — The place where clinical trials are performed. Cloud Administration has the same site object as Rave EDC.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>Note:</p>  Client Division Organization </div>	<p>Client Division Organizations</p>

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>sites to studies.</p> <ul style="list-style-type: none"> When you transition sites to iMedidata from a Classic Rave study all the associated information in Rave also migrates to iMedidata. <p><u>For more Information:</u></p> <ul style="list-style-type: none"> Manage Sites in iMedidata 	 ons are mast er data recor ds for any type of physi cal organ izatio n such as a spon sor, site, healt h care facilit y, or depot . Orga nizati ons must have the	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
	<p>i Orga nizati on Type "Heal th Care Facili ty" which the app uses down strea m as a site in a Clou d Admi nistra tion study envir onme nt.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none">• You can use Client Division	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
	<p>Organizations as templates to create sites downstream in study environments if the "Health Care Facility" Organization Type is applied.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Manage Client Division Organizations 	
Site Number — A user-specified unique number you enter when you define the site. The site number links a site between	Client Division Organization ID — The unique number for an organization in a CD. This is passed to iMedidata as the	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>iMedidata and other Medidata applications, including Classic Rave.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> The site number matches in both applications. For example, if you create a site in iMedidata that already exists in Classic Rave, you must specify the site number exactly as it appears in Rave. <p><u>For more Information:</u></p>	<p>site number when the organization has an industry type of Health Care Facility.</p> <p>Organizations are associated to client divisions, not client division schemes, and they are shared with all of the customer's studies.</p>	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<ul style="list-style-type: none"> • Site Numbers 		
<p>Study-site – iMedidata has the same study-site object as in Classic Rave. In iMedidata you create study-sites while you are managing a study.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> • A study-site represents the same physical location of a clinical site but it can have attributes unique to a study such as principal investigator and study-site number. 	<p>(SE) Site — The combination of a health care facility and a principal investigator. A Health Care Facility is where a study may be conducted. It is an industry type that you select for organizations.</p> <p><u>Usage Notes:</u> The principal investigator may be added or changed at a later date, after a site is created.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Study Environment Site 	<p>Study Environment Sites</p>

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<ul style="list-style-type: none"> As study-sites are created for a study, a site object is also created at the same time. This facilitates the reuse of sites. In Classic Rave, a study-site is a site that has been mapped to a specific study environment. <p><u>For more information:</u></p> <ul style="list-style-type: none"> Study-Site 		
Study-Site Number — The site number given at the study level for	(SE) Site Number — The unique site number for a site in a SE. The app	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>some reports in Rave. For more Information:</p> <ul style="list-style-type: none"> • Study-Site Number 	<p>passes this to iMedidata as the study site number. For more Information:</p> <ul style="list-style-type: none"> • Study Environment Site Number 	
N/A	Client Division People — People associated with a single client division. An informational record of an individual who is associated with a particular client division but may or may not be a Cloud Administration user, depending on whether you assign them a client division role or not.	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
	<p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Manage Client Division People 	
<p>Role Action and Permissions — Grant users different permissions within Classic Rave EDC. Each EDC role is associated with a set of actions that a user can perform and permission to create subjects, modify the primary record, and see audits.</p> <p>Usage Notes:</p> <ul style="list-style-type: none"> • iMedidata uses the same generic concept as 	<p>Role Action and Permissions — Grant users different permissions within Rave EDC. Each EDC role is associated with a set of actions that a user can perform and permission to create subjects, modify the primary record, and see audits.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Role Action and Permissions 	N/A

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>Classic Rave but Rave EDC, user groups, security roles, and so on, are configured as apps and each app has its own collection of roles. These collections of roles are pulled from Rave, Safety Gateway, Rave RTSM, and other platform components.</p> <ul style="list-style-type: none">• When a study owner invites a site user to		

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>participate in a study, they select the EDC app and an appropriate role like clinical research coordinator</p> <p>.</p> <ul style="list-style-type: none">• iMedidata interacts with role actions and permissions indirectly when a study owner invites a user to a study with an EDC role. <p><u>For more Information:</u></p> <ul style="list-style-type: none">• Role Actions• About iMedidata Rave		

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>Architect and Architect Security</p>		
<p>Classic Rave User Groups — Configured to arrange Classic Rave modules into logical groups for user access. User groups control which Rave modules are presented to the user in the left menu on the user Rave home page.</p> <p><u>For more information:</u></p> <ul style="list-style-type: none"> • About iMedidata Rave Architect and Architect Security 	<p>Rave EDC User Groups — Configured to arrange Rave modules into logical groups for user access.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> • If the Rave URL has additional Rave modules enabled, you can grant permission and access to the user in the left menu on the user Rave home page. <p><u>For more information:</u></p>	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
	<ul style="list-style-type: none"> • Manage User Groups 	
<p>Rave Modules</p> <p>Roles — An iMedidata app that allows study group and study owners use to invite users to Rave user groups in addition to their iMedidata EDC role.</p>	N/A	
<p>Rave Security Roles — Configured to grant permission to Rave Architect and Architect Global Library.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • About iMedidata Rave Architect and 	<p>Rave Security Roles — Configured to grant permission to Rave Architect and Architect Global Library.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> • Assign a role with a building block that provides access to Rave 	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
<p>Architect Security</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Security Groups and Roles 	<p>Architect Security.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • Security Groups and Roles 	
<p>Rave Architect Security Groups — An iMedidata app that allows study group and study owners to invite users to Rave Architect projects and global library volumes.</p> <p><u>For more Information:</u></p> <ul style="list-style-type: none"> • About iMedidata Rave Architect and Architect Security 	<p>Rave Architect Security Group Role Assignments — Configured to grant access to Rave Architect. Makes use of Rave security roles.</p> <p><u>Usage Notes:</u></p> <ul style="list-style-type: none"> • Assign a role with a Rave Architect Security building block select the Rave Architect Security 	

Classic Rave Linked to iMedidata	Rave EDC Linked to Cloud Administration	Medidata Platform API Resources
	Groups dropdown.	
N/A	Team Memberships — Memberships of a person to a team at the study environment, study environment country, or study environment site level.	

1.8 Troubleshooting

This page contains details and advice for some common issues. When diagnosing an issue, use "Quick search" to research your error response code, for example "RWS00010".



Note: See also

- [Error Responses - the complete list](#)
- [The pre-installed Log Messages Custom Dataset](#)

[Working with Study Data](#)

You work with study data when you update or insert data into Rave. You can make the same changes in Rave as you can do with RWS. Therefore, we recommend you log into Rave and check that the same action can be

achieved. For example, if a datapoint is locked, then you will not be able to update the data neither through Rave nor through RWS.

The main things that can go wrong with posting clinical data into Rave are:

- Your Rave user account may not have sufficient permissions to apply a change.
- The state of the data in Rave may not permit the change, for example, updating something that does not exist, is deleted, or cannot be found or displayed.
- Some things cannot be done, such as simultaneously coding and translating a single datapoint or data item.

Errors and their causes

- 400 bad request RWS00010 - This is not a valid XML document. The xml is badly formed. Check that it opens correctly in a browser such as Internet Explorer.
- 400 bad request RWS00010 - This is not a valid XML document. The Request exceeds Max Allowable byte size for this operation (the request is too big).
- 400 bad request RWS00010 - This is not a valid XML document. The Request header Content-Type is not supported. Supported types are 'text/xml'; 'text/xml; charset=UTF-8'; 'application/soap+xml'.
- 400 bad request RWS00011 - This is not a valid ODM 1.2 standard document. Check that the ODM is well formed against the CDISC standard.
- 400 bad request RWS00164 - This is not a valid ODM 1.3 standard document. Check that the ODM is well formed against the CDISC standard.
- 400 bad request RWS00056 - This is not a valid ODM 1.3 standard document. The supplied data failed syntax validation, as detailed.
- 400 bad request RWS00012 - Filetype '{0}' not supported. The specified ODM Filetype value is not one that is supported by RWS.
- 403 forbidden RWS00093 - Data posted to incorrect study URL. The clinicalData element and the study element cannot both be present in the ODM document.
- 403 forbidden RWS00096 - Field not enabled for coding. The field is not associated with a coding dictionary.
- 409 conflict RWS00098 - Coding dictionary not associated with the field. The field is not associated with this coding dictionary.
- 409 conflict RWS00114 - Coding value already exists. The datapoint already has coding entries.
- 403 forbidden RWS00097 - User not authorised for coding. The Rave user account for this request does not have a role with the required permissions.
- 409 conflict RWS00107 - Transaction type not supported for this element. For example, 'Upsert' not supported or 'Context' must be used to ensure the term is correct for the coding to be applied.
- 404 not found RWS00099 - Coding dictionary level/column does not exist. There is no matching dictionary Column.
- 404 not found RWS00037 - Record does not exist. For example, a record "Zero" does not exist for a form.

- 400 bad request RWS00166 - Coding can only be applied to the items with transaction type of Context. 'Context' must be used to ensure the term is correct for the coding to be applied.
- 403 forbidden RWS00090 - Field requiring translation, update in default locale not authorised. The Rave user locale must be the same as the UntranslatedData Locale.
- 409 conflict RWS00043 - Field already exists. RWS receives clinical data to create a field that already exists. Use 'update' instead of 'insert'.
- 409 conflict RWS00100 - Datapoint context value does not match current value. Coded data, datapoint and 'specify dictionary' values must all match.
- 409 conflict RWS00115 - Cannot accept 2 elements for Translated Data. There must be only one translated data element supplied.
- 403 forbidden RWS00113 - Simultaneous coding and translating not supported. You cannot code and translate at the same time.
- 403 forbidden RWS00108 - Identifier too long. The field OID is too long. Note this may be erroneous and covering a permissions type error.
- 400 bad request RWS00047 - Data not in dictionary. The Value is Not in the Data Dictionary.
- 403 forbidden RWS00112 - Dictionary item does not require specify value. For example, there is not an option available to select 'other' and specify what 'other' is.
- 400 bad request RWS00046 - Picture/file upload field cannot be updated. This field control type does not permit update through the currently used service.
- 400 bad request RWS00095 - Updating a field associated to a dynamic search list is not supported. This field control type does not allow update through the currently used service.
- 404 not found RWS00041 - Field does not exist. The target field being updated is deleted, inactive, invisible, hidden, or the OID is not supplied.
- 409 conflict RWS00049 - Transaction on derived field is not permitted. Data for this field is derived from other fields and cannot be directly entered.
- 404 not found RWS00042 - Field not authorised. You are not authorised to update the field.
- 400 bad request RWS00158 - Multiple fields found for the tag. The field could not be uniquely identified using this tag.
- 404 not found RWS00050 - Unit Dictionary does not exist. A unit dictionary and value was expected, or the wrong dictionary name supplied.
- 404 not found RWS00051 - Unit Dictionary entry does not exist. You supplied a wrong value.
- 403 forbidden RWS00184 - User not authorised to reactivate the logline. You do not have permission for this operation.

Using Medidata Coder

You use Medidata Coder when updating or inserting data into Rave, as part of a Coder operation.

Errors and their causes

- 403 forbidden RWS00178 - You do not have the permission to use this service, that is, you do not have access to the installed module that is required for this operation. For example, Coder dictionary registration requires that you confirm on the Rave URL that the Coder Import Group User Group is created and has SeeAllModules permissions. Check the Coder Import Role and ensure that the role is assigned to the correct user group. Ensure the Coder Import Role is assigned to the correct role.
- 200 RWS00162 - Queue item not found. This error occurs while attempting to remove the event queue item.
- 404 RWS00163 - Tag does not exist. This error occurs while looking for tag by name.
- 400 RWS00164 - This is not a valid ODM 1.3 standard document. This error occurs while validating the ODM Version.
- 401 RWS00165 - User is not authorised to delete items from the queue. You do not have the required role.
- 404 RWS00167 - Apild not found. The iMedidata application not found.
- 404 RWS00168 - Error retrieving application information from iMedidata.
- 404 RWS00169 - Invalid security token received.
- 404 RWS00170 - Token has expired.
- 404 RWS00171 - Parameter api_id is required.
- 404 RWS00172 - No matching token found.
- 404 RWS00173 - Parameter request_token is required.
- 500 RWS00174 - Unhandled iMedidata API exception.
- 404 RWS00175 - Parameter study_id is required.
- 404 RWS00179 - Workflow with OID "{0}" does not exist.
- 404 RWS00180 - Coding dictionary with OID "{0}" does not exist.
- 400 RWS00181 - DictionaryLevelData element required. This element is missing in the supplied ODM data.
- 400 RWS00182 - You must use the iMedidata StudyOIDType. Missing attribute in the supplied ODM data.
- 400 RWS00183 - Study registration requires at least one dictionary and workflow. Registration or workflow is missing.

Working with Metadata

You work with metadata when updating or inserting a draft into Rave.

Errors and their causes

- 400 bad request RWS00010 - This is not a valid XML document. The XML is badly formed. Check that it opens correctly in a browser, such as Internet Explorer.
- 400 bad request RWS00010 - This is not a valid XML document. The Request exceeds Max Allowable byte size for this operation (the request is too long).
- 400 bad request RWS00010 - This is not a valid XML document. The Request header Content-Type is not supported. Supported types are 'text/xml', 'text/xml, charset=UTF-8', and 'application/soap+xml'.

- 400 bad request RWS00011 - This is not a valid ODM 1.2 standard document. Check that the ODM is well formed against the CDISC standard, for example, the ODM version 1.2 must match the service being used.
- 400 bad request RWS00164 - This is not a valid ODM 1.3 standard document. Check that the ODM is well formed against the CDISC standard.
- 400 bad request RWS00056 - This is not a valid ODM 1.3 standard document. The supplied data failed syntax validation, as detailed.
- 400 bad request RWS00012 - Filetype '{0}' not supported. The specified ODM Filetype value is not supported by RWS.
- 404 not found RWS00105 - Vendor extension mdsol:Attribute namespace does not exist.
- 403 forbidden RWS00102 - Data posted to incorrect draft URL. The DraftVersionOID and the CRF draft in the URL do not match.
- 400 bad request RWS00091 - Architect loader error. RWS passes the draft to Architect loader for loading into Rave.
- 403 forbidden RWS00093 - Data posted to incorrect study URL. Imported ODM contains both a Study and ClinicalData element.
- 403 forbidden RWS00093 - Data posted to incorrect study URL. <ODM/Study/@OID> does not equal study name (found from URL being split).

Retrieving Study Metadata

When you retrieve study metadata, you get a studies list, versions list and drafts from Rave.

No known errors.

Retrieving Clinical Data

When you retrieve clinical data, you get a list of studies, list of subjects, and deeplinking to forms.

Errors and their causes

- 400 bad request RWS00139 - Invalid parameter supplied, for example, error parsing the start date parameter. Check that the correct date format is used.
- 404 not found RWS00092 - CRF version not found. The number supplied is not a number or is missing, or a CrfVersion with this number does not exist.

Retrieving Datasets

When you retrieve datasets, you can do the following:

- Retrieve metadata or clinical data from Rave using clinical views,

- Retrieve data from Rave using configurable datasets, and
- Retrieve data from Rave using configurable datasets in CSV format.



Important: Due to the streaming nature of retrieving datasets requests, if an error occurs mid stream, you cannot correctly recover and return a useful error. For this reason, you must ensure that the receiving system checks for an 'end of file' marker to validate if the whole dataset is received.

Errors and their causes

- 400 bad request RWS00139 - Invalid parameter supplied. For example, error parsing the start date parameter - check that the correct date format was used.
- 500 Internal Server error RWS00161 - Invalid template. The complex template configured by MDSOL is broken (syntax failure).
- 400 bad request RWS00152 - Invalid data type of filter value. Conversion failed when converting filter value to correct datatype of column, for example, expecting a number, not a string.
- 500 Internal Server error RWS00149 - Invalid object name. The dataset configured by MDSOL is broken. The Stored Procedure or view does not exist.
- 400 bad request RWS00151 - Invalid column name. The view does not exist.
- 404 not found RWS00148 - Dataset does not exist. No configuration found for the Dataset - it is not installed or incorrectly spelt.
- 501 Not Implemented RWS00140 - This service is not implemented. No configuration found for the Dataset, or the Project is not configured to include inactive records. An upgrade to patch 229 (Rave 563) or higher may be required. In addition, if your Dataset relies on Clinical Views, check that the Clinical Views refresh job is set to something other than "Do Not Run" (for example, for Biostat Adapter).
- 404 not found RWS00150 - Dataset format not found. Either the format, for example, 'csv', does not exist or it is incorrectly spelt.
- 404 not found RWS00134 - Dataset does not exist. The Clinical View does not exist in Rave. Check that it is named correctly.
- 503 service unavailable RWS00134 - Dataset does not exist. No Clinical View exists for ProjectId, isProd, or ClinicalViewType.

When retrieving data from Rave using Configurable datasets in CSV format, you may receive an error that includes the HTTP status code "200/OK: XML Parsing Error: not well-formed". This means that the ContentType column of the WebServicesConfigurableDatasetFormats table is not set correctly, for example, to 'text/plain' or 'text/csv'.

Other causes of typical errors include:

- Incorrect URL, incorrect Rave user account name, or password.
- Configuration changes, for example, a change of user role.

1.9 FAQs

The following is a list of Frequently Asked Questions (FAQs) related to Rave Web Services.

ODM Adapter

Q: Can two ODM Adapter versions run side-by-side, that is, can we use both services?

A: Once you update to a new ODM Adapter version, the older version is no longer available.



Note: The API remains unchanged.

Q: Does the update to a new ODM Adapter version need to be applied to the URL where the study/database is?

A: The update needs to be applied to the URL that hosts the studies. Once the update is made, it will apply to all the studies hosted on that URL, and that URL only.

Q: For how long will you support an older ODM Adapter version after a new one is released?

A: Older ODM Adapter versions will be supported for the foreseeable future, until a new decision is made.

Q: If we update to a new ODM Adapter version and something goes wrong, can we go back to the old version?

A: Rollbacks are complex, hence the short answer is no. However, Medidata is committed to helping you resolve any issues. We recommend testing the update on a URL copy.

Q: Is there a service that tells us when refresh occurs?

A: There is no such service; you should continue your calls as scheduled and wait for data to be returned.

Q: Can we backup the current production databases onto another URL, for example, Training, and then upgrade the Training URL to a new ODM Adapter version?

A: You are advised to test the update to a new ODM Adapter version on a sandbox first, prior to updating your production URL.

Q: Why does some data entered in Rave not show properly in the ODM ClinicalAuditRecords Custom Set ?

A: Certain Audit Record Subcategories that exist in Rave are not supported currently by the ODM Adapter. This causes missing/inconsistent data upon retrieval of clinical data transactions with the ODM Adapter - see list of [ODM Adapter Included Audit Subcategories](#).

Q: Does the ODM Adapter support Unicode characters?

A:



Important:

- Prior to Classic Rave 2019.2.0, entering Unicode characters in Rave EDC was supported but retrieving the same Unicode characters in ODM extracts through RWS was not supported. The Unicode characters supported in RWS were equivalent to Latin ASCII characters and special characters like '?', '!', '/', and so on. Many Unicode characters were converted to question mark ?, as specified in the [ODM Unicode character conversion list](#).
- From Classic Rave 2019.2.0 onwards, the ODM Adapter V2 [Clinical Audit Records \(CAR\) service](#) has introduced an optional Boolean query string parameter, `unicode`, to support Unicode characters.

- Add `unicode=true` to the query string to return Unicode data in RWS, for example:

```
{url}/datasets/ClinicalAuditRecords.odm?studyoid=<Study  
OID>&startid=<Start Audit ID>&per_page=<Per Page>&unicode=true
```

- If you do not require Unicode data, either omit specifying the `unicode` query parameter or add `unicode=false` to the query string, for example:

```
{url}/datasets/ClinicalAuditRecords.odm?studyoid=<Study  
OID>&startid=<Start Audit ID>&per_page=<Per Page>&unicode=false
```

```
{ {url}}/datasets/ClinicalAuditRecords.odm?studyoid=<Study  
OID>&startid=<Start Audit ID>&per_page=<Per Page>
```

ODM output.

Q: What are the main differences between the Biostat Adapter and the ODM Adapter?

A: The Biostat Adapter is a visual tool that works from Clinical Views and the data you get is a readable snapshot of clinical data. You can think of the Biostat Adapter as a quick way to see the state of the data in Rave, or a SAS on Demand on the web. The data extracted from both the Biostat Adapter and SAS on Demand is the same, but the Biostat Adapter delivers it over web services, while SAS on Demand delivers it in file form to an FTP.

The ODM Adapter also provides operational data, such as query information and status updates. The ODM Adapter works from the Rave Audit Trail among other Rave tables and generates data as transactions, which need to be aggregated in order to produce an output that is legible. The ODM Adapter is a data transportation tool used to populate clinical data warehouses with data.

Security Protocols

Q: What is TLS 1.2?

A: The Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL) are cryptographic protocols that provide communications security over a computer network. TLS versions 1.0 and 1.1 will no longer be supported by the Medidata Clinical Cloud platforms as of June 2018. This affects all web services for Rave, RaveX, and iMedidata - including FTP over SSL and other add-ons.

- TLS 1.0 is no longer considered secure by industry standards and contains known vulnerabilities such as Beast and TLS Poodle. This protocol is scheduled for deprecation in 2018 by major vendors such as Microsoft, Google and Salesforce along with security frameworks such as PCI and NIST.
- TLS 1.1 is underutilized since the introduction of version 1.2, given there is little security difference and value between versions 1.0 and v1.1.

Q: What is an Entrust certificate?

A: Secure Sockets Layer (SSL) certificates are used to secure access to web services using TLS encryption protocols. Medidata is transitioning to Entrust Datacard as its primary certificate authority instead of GoDaddy as part of our certificate renewal. GlobalSign will continue to be our certificate authority for our European presence.

Browsers and integration Application Programming Interface (API) software rely on root stores for initializing secure communication with TLS (SSL). These root stores contain the root certificates of major certificate authorities such as Entrust and GoDaddy. If these root stores are not updated regularly, you will experience a connection error if the root certificate from Entrust is not loaded. This is normally updated automatically by the software vendors.

Q: What is changing in 2018?

A:

1. The supported TLS protocols will be restricted to version 1.2 by June 2, 2018.
2. The current SSL certificate for mdsol.com will expire on July 6, 2018. We will transition to Entrust Datacard as part of the certificate update on June 23, 2018.

Q: How can TLS and Entrust certificate updates affect me as a user?

A:

TLS 1.2 updates may impact:

- RWS calls to HDC/FRA
- Client browser access to Rave and add-ons
- API calls to iMedidata and AWS services

Entrust Certificate updates may impact:

- RWS calls to HDC/FRA
- Client browser access to Rave and add-ons

Q: How do I test for TLS 1.2 compatibility?

A: <https://tls-rave-test.mdsol.com> (secops-rave-test.mdsol.com)

- Supports the TLS 1.2 protocol only
- Current GoDaddy certificate loaded
- Logging of all TLS protocol versions is now enabled, so that we may confirm a successful TLS connection regardless of the RWS errors encountered by the client source.

For outbound web services to RWS and iMedidata, the application's web services software must have the TLS 1.2 protocol enabled for outbound API calls initiated by the host. TLS 1.2 support varies depending on the

software and software version used for initiating web services, such as Java, WebLogic, Ruby, Apache, Microsoft.Net, and so on. Some software has support for TLS 1.2, but it may not be enabled by default.

- All modern browsers with a release date of 2014 or later support TLS1.2 by default. Some older browsers might support TLS1.2, but it must be enabled first.
- Most current integration software supports TLS1.2, but it might have to be enabled. Please check with your software vendor.

Q: How do I test for Entrust root certificates?

A: <https://cert-rave-test.mdsol.com> (secops-rave-test2.mdsol.com)

- Supports all TLS versions: 1.0, 1.1, 1.2
- New Entrust certificate loaded
- Logging of all TLS protocol versions is now enabled so that we may confirm a successful TLS connection regardless of the RWS errors encountered by the client source.

The root stores may not be updated for a couple conditions:

- The browser clients are too old or not updated regularly. Entrust is supported by most major browser software including unsupported versions, so this should not be an issue.
- The integration software used for API calls to iMedidata is too old or not updated regularly. There is no list of supported software versions, nor can this be identified ahead of time by Medidata. The clients and integration partners would have to check their root stores on their integration software.
- The fix is to update the browser software or update the root store in the integration software. The Entrust certificate chain may be downloaded [here](#).

Other FAQs

Q: When does RWS create/insert a subject with no audits?

Prior to Rave 2018.1.0, RWS post requests supported subject creation with multiple <Subjectdata> elements in the same ODM post. However, from RaveX/Rave 2018.1.0 and onwards, an RWS post request will fail if you attempt to insert/create a subject and update a separate event using multiple <Subjectdata> elements, on the same subject in a single ODM post.

For example, if you attempt to create a subject and populate the Demography form in the same RWS post request using multiple <Subjectdata> elements, the post fails and results in an error indicating “Subject doesn’t exist”.

If the <SubjectData> and <SiteRef> are listed only once in the ODM post, then the RWS post executes successfully. For more details, please refer to the examples below.

Previously uploadable ODM (prior to 2018.1.X):

```
<?xml version="1.0" encoding="UTF-8"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/
v1.3" ODMVersion="1.3" FileType="Transactional" FileOID="11111111-1111-1111-1111-111111111111" CreationDateTime=-
<ClinicalData StudyOID="STUDYNAME" MetaDataVersionOID="1">
  <SubjectData SubjectKey="SUBJECTNAME1" TransactionType="Insert">
    <SiteRef LocationOID="SITEID" />
    <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
      <FormData FormOID="SUBJECT">
        <ItemGroupData ItemGroupOID="SUBJECT">
          <ItemData ItemOID="XXX" Value="XXXXXX" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
  <SubjectData SubjectKey="SUBJECTNAME1" TransactionType="Update">
    <SiteRef LocationOID="XXX" />
    <StudyEventData StudyEventOID="DM">
      <FormData FormOID="DM">
        <ItemGroupData ItemGroupOID="DM">
          <ItemData ItemOID="BRTHDAT" Value="XX XXX XXXX" />
          <ItemData ItemOID="SBJINIT" Value="XXX" />
          <ItemData ItemOID="GENDER" Value="X" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
</ODM>
```

Modifying the XML above by removing the three lines shown in pink and adding the text in green makes it work. This sends a single update as an RWS call instead of two SubjectData posts - see below:

```
<?xml version="1.0" encoding="UTF-8"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/
```

```
v1.3" ODMVersion="1.3" FileType="Transactional" FileOID="11111111-1111-1111-111111111111" CreationDateTime="2023-10-20T10:30:00Z"
<ClinicalData StudyOID="STUDYNAME" MetaDataVersionOID="1">
  <SubjectData SubjectKey="SUJECTNAME1" TransactionType="Insert">
    <SiteRef LocationOID="SITEID" />
    <StudyEventData StudyEventOID="SUBJECT" TransactionType="Update">
      <FormData FormOID="SUBJECT">
        <ItemGroupData ItemGroupOID="SUBJECT">
          <ItemData ItemOID="XXX" Value="XXXXXXXXXX" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
  <SubjectData SubjectKey="SUBJECTNAME1" TransactionType="Update">
    <SiteRef LocationOID="SITEID" />
    <StudyEventData StudyEventOID="DM" TransactionType="Update">
      <FormData FormOID="DM">
        <ItemGroupData ItemGroupOID="DM">
          <ItemData ItemOID="BIRTHDAT" Value="XX XXX XXXX" />
          <ItemData ItemOID="SBJINIT" Value="XXX" />
          <ItemData ItemOID="GENDER" Value="X" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
</ODM>
```

[Back to top](#)