OpenClinica

OpenClinica 3 Technical Documentation

Getting Started

This guide will help you interact with and extend OpenClinica 3 in a programmatic way. This guide is for developers who have a basic familiarity with HTTP and web services. HTML and javascript expertise helps too.

The Basics

You'll likely want to start with the REST API. This provides you with hooks to interact with an OpenClinica study in a lot of neat ways.

Access the OpenClinica REST API

REST API is a fancy way of saying you can access a URL that gives you back XML or JSON data in a consistent format. REST APIs are becoming the lingua franca of the web, and are widely used in mobile apps, social networking Web sites, mashup tools, and automated business processes. They are easy to leverage by most tools, including many that are free. Best of all, the learning curve is not too steep, especially if you're already familiar with the basics of how the web works.

We also have a SOAP API. If you're not sure what this is, use REST. We're no longer adding new functionality to SOAP. API references in this guide refer to the REST API unless explicitly mentioned otherwise.

Set Up Your Development Environment

You'll need an OpenClinica environment to develop and test your application or module against. If you're an OpenClinica Enterprise user, you may be able to use your test environment. Find out the URL of the test environment and make sure you have a test study and a user account on the system. Otherwise, you can download and install the latest version of OpenClinica Community Edition.

Security and Permissions

Different parts of the API use different security mechanisms. We're working on unifying these but for now you'll need to understand the differences, and take note of which API methods support which types of authentication.

- Session: You need to programmatically login to /pages/login/login and subsequently call the API.
- API Key: Authentication to the API is performed via HTTP Basic Auth. Provide your API key as the basic auth username value. You do not need to provide a password. You can obtain your api key by calling this API: https://dev.openclinica.com/apidoc/#api-User_Account-getAccountByUserName
- OAuth: All developers need to register their application before getting started. A registered

OAuth application is assigned a unique Client ID and Client Secret. The Client Secret should not be shared.
- Network/firewall dependent: Network/firewall constraints need to be put in place to secure these APIs.

Each API method is labled with the authentication method(s) it supports.

Getting under the hood

Parts of this guide contain specifications on the OpenClinica eCRF meta model, how we handle dates and datatypes, and how we interpret and implement the CDISC Operational Data Model.

- CDISC ODM
- CRF Template Elements
- Dates
- Item Data
- Internationalization and Localization
- Database Schema

Building Modules

Follow this guide to build new data extract/reporting formats.

OpenClinica will soon include a more comprehensive developer toolkit so you can easily develop your own custom modules, including dashboards, workflows, and visualizations. *** UNDER CONSTRUCTION, check back frequently! ***

Developing the Core

If you can't accomplish what you need via a module, and/or are itching to fix bugs, expand the API, and otherwise make OpenClinica better, you can help improve OpenClinica core.

This guide will help you build from the source code and set up a development environment.

Sharing

Sharing your work helps make OpenClinica more useful for everyone and gives you help improving and maintaining it. Submit code as a pull request to github. Share your ideas/questions/testing/writing on the OpenClinica forums (you'll have to register first). Someone out there might be thinking along the same lines and want to help you out!

1 REST Web Services

Overview

We are constantly looking at ways to make it possible (not to mention reliable and easy!) for users and developers to interact with and extend OpenClinica in a programmatic way. This can mean anything from data loading to more meaningful integrations of applications common to the clinical research environment.

As proponents of open, standards-based interoperability here at OpenClinica, our starting point is to develop interfaces using the protocols that power the World Wide Web (such as HTTP, SSL, XML, OAuth 2.0). They are relatively simple, extensively documented, widely understood, and well-supported out of the box. On top of this foundation, we rely heavily on the wonderful work of CDISC and the CDISC ODM to model and represent the clinical research protocol and clinical data.

This chapter describes a way to interact with OpenClinica using RESTful APIs and OAuth. The REST web services API relies on HTTP, SSL, XML, OAuth 2.0. This architecture makes the ODM study protocol representation for an OpenClinica study available and supports other interactions for study design.

Access the OpenClinica REST API

Why REST?

The OpenClinica RESTful architecture was developed to (initially) support one particular use case, but with the intention of becoming more broadly applicable over time. This use case is based on a frequent request of end users: for OpenClinica to support an visual method for designing, editing, and testing rules which define edit checks, email notifications, skip pattern definitions, and the like to be used in OpenClinica CRFs. Users have had to learn how to write rules in XML, which can be confusing and have a big learning curve for non-technical individuals. The OpenClinica Rule Designer is an application that allows end users to build cross field edit checks and dynamics within a GUI based application. It is a centrally hosted Software as a Service (SaaS) based application available for OpenClinica Enterprise customers at https://designer.openclinica.com.

To support interaction of the centrally hosted rule designer with any instance of OpenClinica Enterprise installed anywhere in the world, we needed to implement a secure protocol and set of API methods to allow exchange of study information between the two systems, and do so in a way where the user experience was as integrated as if these applications were part of the same integrated code base. In doing so, and by adopting the aforementioned web and clinical standards to achieve this, we have built an architecture that can be extended and adapted for a much more diverse set of uses.

This chapter specifies how 3rd party applications can interact with an OpenClinica instance via the REST API and OAuth security, and details the currently supported REST API methods. The currently supported API methods are not comprehensive, and you may get better coverage from our SOAP API. However the OpenClinica team is continuing to expand this API and since it is open source anyone may extend it to add new methods to meet their own purposes. If you do use the API in a meaningful way or if you extend the API with new methods, please let others know on the OpenClinica Forums, and submit your contributions for inclusion back into the codebase - youll get better support, increased QA, and compatibility with future OpenClinica releases.

RESTful Representation, based on ODM

REST, an acronym for **RE**presentational **S**tate **T**ransfer, describes an architectural style that allows definition and addressing of resources in a stateless manner, primarily through the use of Uniform Resource Identifiers (URIs) and HTTP. A RESTful web service (also called a RESTful web API) is a simple web service implemented using HTTP and the principles of REST. It is a collection of resources, with three defined aspects:

- the base URI for the web service, such as http://example.com/resources/
- the Internet media type of the data supported by the web service. This is often JSON, XML or YAML but can be any other valid Internet media type.
- the set of operations supported by the web service using HTTP methods (e.g., POST, GET, PUT or DELETE).

In the context of REST for clinical research using OpenClinica, we can conceptually think of an electronic case report form (CRF) as a **resource** that is essentially a bunch of metadata modeled in CDISC ODM with OpenClinica extensions. An OpenClinica Event CRF is that same bunch of metadata with the corresponding item data, plus references to the study subject, event definition, CRF version, event ordinal, etc that it pertains to.

- The notion of a CRF version pertains to the representation of the CRF. It is not intrinsic to the event CRF (this is debatable but it is how OpenClinica models CRFs). Theoretically you should be able to address and view any Event CRF in any available version of the CRF (ie http://oc/RESTpath/StudyA/Subj1234/VisitA/FormB/v1/edit and http://oc/RESTpath/StudyA/Subj1234/VisitA/FormB/v2/edit both show you the same data represented in different versions of the CRF). Of course the audit history needs to clearly show which version/representation of the CRF was used for key events such as data capture, signature, etc.
- Rules are also part of the representation metadata as opposed to intrinsic metadata, even though you dont need to specify them on a version-by-version basis.
- Anything attached to the actual event CRF object or its item data discrepancy notes, audit trails, signatures, SDV performance, etc is part of that event data and should be addressable in the same manner (eg http://oc/RESTpath/StudyA/Subj1234/VisitA/FormB/v1/GROUPOID/ORDINAL/ITEMOID/DN/1/view)

In this conceptual view of the world, CRFs (as well as CRF items, studies, study events, etc.) are RESTful resources with core, intrinsic properties and then some other metadata that has to do with how they are presented in a particular representation. We now have a model that allows us a great deal of flexibility and adaptability. We can support multiple modalities, with different representation metadata for rendering the same form, or perhaps the shared representation metadata but applied in a different way. We can address any part of the CRF in an atomic manner. This approach has been successfully applied in the Rule Designer, which takes the ODM study metadata and allows browse of the study CRFs and items, with the ability to drag and drop those resources into rule expressions. Some examples of additional future capabilities that could be easily realized on top of this architecture:

- Multiple data entry modalities a user may need to deploy patient based data entry via web, a tablet, a thick client, or even paper/OCR, each with a very different presentation. Each of these may be part of OpenClinica-web or a separate application altogether, but all will rely on the same resource metadata to represent the CRF (according to the UI + logic appropriate for that modality), and use the same REST-based URL and method for submitting/validating the data.

- Apply a custom view (an XSL or HTML/CSS) to a patient event CRF or full casebook some uses of this could be to represent as a PDF casebook, show with all audit trails/DNs embedded in line with the CRF data, show a listing of data for that subject, provide (via an XSL mapping) as an XForm or HL7 CCD document for use by another application) - http://oc/RESTpath/StudyA/Subj1234/VisitA/FormB/v1/view?renderer=somemapping.xsl
- The same path used in the URLs, eg http://oc/RESTpath/StudyA/Subj1234/VisitA/FormB/v1/GROUPOID/ORDINAL/ITEMOID could be used as the basis for XPath expressions operating on ODM XML representations of CRFs and of event crf data
- Internationalization OpenClinica ought to allow our CRF representation metadata to have an additional sub-layer to render the form in different languages, and then automatically show the appropriate language based on client/server HTTP negotiation (like we do with the rest of the app). Currently internationalization of CRFs requires versioning the CRF.
- View CRF & Print CRF use the same representation metadata (form metadata) but apply slightly different rules on how the presentation works (text values instead of form fields, no buttons, turn drop down lists into text values)
- Discrepancy manager popup one requested use case would allow a user to update a single event CRF item data value directly from the discrepancy note UI point of view. In this case you could think of just updating that one item as addressing the resource http://oc/RESTpath/StudyA/Subj1234/VisitA/FormB/v1/GROUPOID/ORDINAL/ITEMOID/edit?mode=DN. In this model, whatever rules and presentation metadata need to get applied at presentation and save time happen automatically.
- Import of CDISC ODM XML files imported data would be processed through the same model, but only use the metadata thats relevant to the data import modality. Same for data coming in as raw ODM XML via a REST web service. A lot of times the import only populates one part of a CRF and the other parts are expected to be finished via data entry. This model would help us manage that process better that the current implementation of ODM data import.

There are many considerations related to user roles and permissions, workflows, and event CRF/item data status attributes that need to be overlaid on top of this REST model, but the model itself is conceptually a most useful way to think about clinical trials and the information represented therein. When implemented using CDISC ODM XML syntax it becomes even more powerful. As widespread support for ODM becomes the norm, the barriers to true interoperability - shared, machine readable study protocol definitions, and robust, real-time, ALCOA-compliant exchange of clinical data and metadata that aligns with users business processes - get eviscerated.

* This chapter frequently refers to ODM-based representations of study metadata and clinical data in OpenClinica. We strive as much as possible to implement ODM-based representations of OpenClinica metadata and data according to the generic ODM specifications (currently using ODM version 1.3). However, to ensure our representations support the full richness of information used in OpenClinica we often have to rely on ODMs vendor extensions capability. I have not always made distinctions in this chapter as to where we are using generic ODM versus OpenClinica extensions, but that is documented here. It is our goal as ODM matures and supports richer representations of study information to migrate our extensions back into the generic ODM formats.

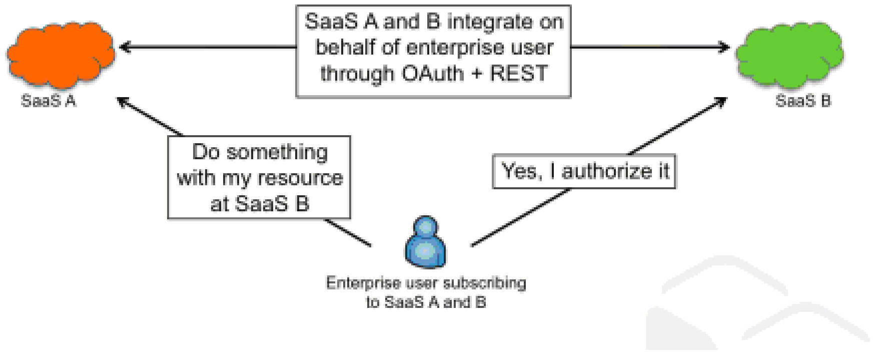
** Also note the RESTful URL patterns referred to above are conceptual. Refer to the technical subchapters of this REST API specification for the actual URLs.

Approved for publication by Cal Collins. Signed on 2016-03-11 3:42PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

1.1 OAuth and OpenClinica

Use of the REST API is dependent on authentication via the OAuth protocol (version 2). OAuth is a security protocol that enables users to grant third-party access to their web resources without sharing their passwords. See http://oauth.net/2/ and http://hueniverse.com/2010/05/introducing-oauth-2-0/ for more information on OAuth 2.0.


SaaS A and B integrate on behalf of enterprise user through OAuth + REST
SaaS A
SaaS B
Do something with my resource at SaaS B
Yes, I authorize it
Enterprise user subscribing to SaaS A and B

(Source: http://cloud.dzone.com/news/enterprise-saas-integration)

The communication/authentication steps between the OAuth client and server are described below.

(D) = Designer - OAuth Client

(OC) = Authorized OpenClinica Instance OAuth service provider

1. (D) /access.....

2. (D) hits restTemplate call to OC

3. (D) throws exception - No OAuth 2 security context has been established.

Unable to access resource 'ocInstance'.

4. (D) As part of exception bubbling up OAuth2ClientContextFilter line 77 triggered results in

    response.sendRedirect ::

http://localhost:8080/OpenClinica-web-SNAPSHOT/oauth/user/authorize?client_id=designer&redirect_uri=http%3A%2F%2Flocalhost%3A8080%2FDesigner-0.1.0.BUILD-SNAPSHOT%2Faccess%3Fhost%3Dhttp%3A%2F%2Flocalhost%3A8080%26app%3DOpenClinica-web-SNAPSHOT%26study_oid%3DS_DEFAULTS1%26provider_user%3Droot&response_type=code

5. (OC) redirect hits OC

6. (OC) initiates http://localhost:8080/OpenClinica-web-SNAPSHOT/oauth/confirm_access

7. (OC) the above url initiates a maybe just a normal request or REST call back to (D)

8. (D) in OAuth2ClientContextFilter the request URL looks like http://localhost:8080/Designer-0.1.0.BUILD-SNAPSHOT/access?host=http://localhost:8080&app=OpenClinica-web-SNAPSHOT&study_oid=S_DEFAULTS1&provider_user=root&code=2zLl3b

9. (D) hits /access .. controller code again

10. (D) hits restTemplate call to OC

11. (D) throws

org.springframework.security.oauth2.consumer.OAuth2AccessTokenRequiredException:

No OAuth 2 security context has been established. Unable to access resource 'ocInstance'.

12. (D) As part of exception bubbling up OAuth2ClientContextFilter line 77 triggered results in

13. (D) line 83 in OAuth2ClientContextFilter will trigger REST Call to

    http://localhost:8080/OpenClinica-web-SNAPSHOT/oauth/authorize

14. (D) response from above call produces an accessToken

15. (D) returns back to /access ... controller code execution

16. (D) hits restTemplate call to OC

17. (D) call succeeds

Every time /access..... is invoked, Designer will receive the request and ask OpenClinica.com to check if the URL is an authorized instance (ie, an Enterprise instance or a Community instance that has registered here). If that check fails the user will be automatically directed to an error page. The above check will be continuously performed through out the OAuth handshake to check the validity of the URL.

1.2 RESTful URLs

In the context of REST for clinical research using OpenClinica, we can conceptually think of an electronic case report form (CRF) as a resource that is essentially a bunch of metadata modeled in CDISC ODM with OpenClinica extensions. Other OpenClinica objects (such as study definitions) can also be resources exposed in a RESTful manner.

About REST and Clinical Data Keys

In REST, resources are identified by logical URLs and are the key element of RESTful design. Interaction is stateless.

ClinicalData Objects (i.e. Study Subjects and their CRF data) in OpenClinica can be addressed by using ODM-based Clinical Data Keys (https://docs.openclinica.com/3.1/technical-documents/openclinica-and-cdisc-odm-specifications/cdisc-odm-representation-openclin-6#content-title-4523) as part of a URL path appended to /ClinicalData/, as follows:

```
GET /OpenClinica/ClinicalData/{format}/{mode}/ODM_XML_PATH?OPTIONS
```

Implementation in 3.1.3

The initial implementation of RESTful URLs in OpenClinica 3.1.3 supports a URL to return a read-only EventCRF with its associated FormData in HTML format. Only HTML format and view mode is implemented in OpenClinica 3.1.3. The general format of the URL is:

```
/OpenClinica/ClinicalData/html/view/{StudyOID}/{StudySubjectKey}/{StudyEventDefOID}[{StudyEventRepeatKey}]/{FormDefOID}?tabId={sectionNum}
```

The URL query string options supported for this URL are tabId & exitTo:

- tabId the CRF section number to show (HTML format only).
- exitTo URL for where the 'exit' button should take the user (optional, HTML format only, relative to the OpenClinica root URL).

An example would be:

```
GET
/OpenClinica/ClinicalData/html/view/S_CPCS/320999/SE_CPCS[1]/F_CPCS_1?tabId=1&exitTo=ViewStudySubject?id=1
```

These URLs do not support OAuth security authentication.

Future Implementation Plans

The {format} component of the URL may be html or xml. XML provides CDISC ODM XML. If omitted use CDISC ODM XML

The {mode} component may be view, edit, or print (html format only).

The path builds an address to the resource. In the example above, the path ends at the Form OID level so we can determine the resource were looking at an EventCRF. It could go further down to the ItemGroup or Item level. The format of whats returned and the available options may vary based on the level of the resource addressing an EventCRF may be consumed differently than addressing an Item.

The full form of the URL would be:

```
GET
/OpenClinica/ClinicalData/{format}/{mode}/{StudyOID}/{StudySubjectKey}/{Study
EventDefOID}[{StudyEventRepeatKey}]/{FormDefOID}?tabId={sectionNum}&exitTo={e
xitURL}
```

This page is not approved for publication.

1.3 RestFul URL access to OpenClinica metadata and print Resources.

Starting OpenClinica 3.1.4, there will be restful access to OpenClinica's ODM metadata and print CRFs. The metadata can be obtained in json/xml formats, once the user is logged into the system.The printing of a CRF without any data will also be accessible via restful url.

These rest based urls do not yet support OAUTH yet and the user needs to be logged into OpenClinica.

The urls will be of the following format:

/host/{WEB-APP_CONTEXT}/rest/metadata/{format}/{mode}/ODM_XML_PATH.

As of 3.1.4, OpenClinica supports formats:html/xml/json

mode:view/print

The mode should be 'print' if the user intends to view a printed CRF,

ODM_XML_PATH: Consists of typically 3 variables.
{STUDY_OID}/{STUDY_EVENT_OID}/{FORM_VERSION_OID}

All the 3 variables are study OID,studyevent oid and form oid which are generated by OpenClinica and are unique across the system. They can be substituted by a * in order to mention a generic or include all as explained below.

For printing CRFs that do not belong to any study and/or are used in multiple events:

ODM_XML_PATH: */*/FORM_VERSION_OID

For printing crfs/viewing the json or xml of all of the study(The ODM in json and xml formats brings the whole ODM metadata even though we put a form_version_OID. This filtering at XML level will be worked on in near future.):

ODM_XML_PATH:STUDY_OID/*/FORM_VERSION_OID

Examples: To obtain the ODM of the entire study the following would be the restful path:

/host/{WEB_APP}/rest/metadata/xml/view/STUDYOID/*/*

To obtain the printed view of a CRF that belongs to a particular event in a study:

/host/{WEB_APP}/rest/metadata/html/print/STUDYOID/STUDYEVENTOID/FORMVERSIONOID

Starting OpenClinica 3.2, there will be restful access to OpenClinica's ODM clinical Data for populated print CRFs. The Clinical Data can be obtained in json/xml formats, once the user is logged into the system.The printing of a CRF with populated data will also be accessible via restful url.

The urls will be of the following format:

/host/{WEB-APP_CONTEXT}/rest/clinicaldata/{format}/{mode}/ODM_XML_PATH.

formats:html/xml/json

mode:view/print   The mode should be 'print' and the format should be 'html' if the user intends to view a printed CRF,

For Clinical Data the ODM_XML_PATH: Consists of typically 4 variables. {STUDY_OID}/{Study_Subject_OID}/{STUDY_EVENT_OID}/{FORM_VERSION_OID}   plus parameters for additional options.

All the 4 variables are STUDY_OID , Study_Subject_OID , STUDY_EVENT_OID , FORM_VERSION_OID are generated by OpenClinica and are unique across the system. They can be substituted by an * in order to mention a generic or include all as explained below.

For printing CRFs for All Subjects per Study :   ODM_XML_PATH: /Study_oid/*/*/*

For printing CRFs for One Subject per Study :   ODM_XML_PATH: /Study_oid/Study_Subject_OID/*/*

For printing CRFs for One Subject and One event: ODM_XML_PATH: /Study_oid/Study_Subject_OID/Study_Event_OID/*

For printing an Event CRF : ODM_XML_PATH: /Study_OID/Study_Subject_OID/Study_Event_OID**%5B**1**%5D**/Form_Version_OID

%5B**1**%5E : Represents the Study Event Repeat #in brackets. [1]

%5B is the URL Escape code for Left Square Brackets and **%5D** is the URL Escape code for Right Square Brackets

Additional parameters will be needed to include Audit Logs and Discrepancy Notes and to filter Item or event status.

Add the following parameter '**?includeAudits=y**' (case sensitive) to the end of your URL to include all the Audit Log for SubjectData, StudyEventData, FormData and ItemData attributes that exist (not null)

Add the following parameter '**?includeDNs=y**' (case sensitive) to the end of your URL to include all the Discrepancy Note for SubjectData, StudyEventData, FormData and ItemData attributes that exist (not null)

The OC extension attributes and ItemData values will be always be incuded in the output whether parameters are added to the end of the URL or not. The OC extension attributes include for SubjectData, StudyEventData, FormData entities.

This page is not approved for publication.

1.4 Read OpenClinica ODM Metadata REST Service

The service will get ODM metadata about a single study in the OpenClinica. Standard authentication applies, and the user must have read privileges for the associated study.

This page is not approved for publication.

1.4.1 Calling Methods and Arguments

Reads is invoked as an HTTP GET method on a specific instance of a resource, qualified with a STUDY OID value.

GET pages/rule/studies/{study}/metadata

1.4.2 Responses

On success, a response with a 200 OK HTTP status code and a representation of the requested ODM object is returned. The response will be presented in the following schema.

- OpenClinica-ODM1-3-0-OC2-0-foundation.xsd
- OpenClinica-ODM1-3-0-OC2-0.xsd
- OpenClinica-ToODM1-3-0-OC2-0.xsd

1.5 Validate Rule REST Service

The service gets the Validity of a rule posted to it. Standard authentication applies, and the user must have privileges to conduct such an operation.

This page is not approved for publication.

1.5.1 Calling Methods and Arguments

This operation is invoked as an HTTP POST method on a specific instance of a resource qualified with a STUDY OID value.

POST pages/rule/studies/{study}/validateRule

1.5.2 Responses

On success, a response with a 200 OK HTTP status code and a representation of the requested validation result is returned. The response will be presented in the following schema.

- response.xsd

1.6 Test Rule REST Service

Gets the Validity of a rule and test keys/values posted to this service. Standard authentication applies, and user must have privileges to conduct such an operation.

This page is not approved for publication.

1.6.1 Calling Methods and Arguments

This operation is invoked as an HTTP POST method on a specific instance of a resource qualified with a STUDY OID value.

POST pages/rule/studies/{study}/validateAndSaveRule

1.6.2 Responses

On success, a response with a 200 OK HTTP status code and a representation of the result of saving the rule is returned. The response will be presented in the following schema.

- response.xsd

1.7 Save Rule REST Service

Saves the rule and returns a message. Standard authentication applies, and user must have privileges to conduct such an operation.

This page is not approved for publication.

1.7.1 Calling Methods and Arguments

This operation is invoked as an HTTP POST method on a specific instance of a resource qualified with a STUDY OID value.

POST pages/rule/studies/{study}/validateAndSaveRule

1.7.2 Responses

On success, a response with a 200 OK HTTP status code and a representation of the result of saving the rule is returned. The response will be presented in the following schema.

- response.xsd

2 SOAP Web Services

Introduction

Welcome to the OpenClinica SOAP Web Services Guide. Newcomers to OpenClinica SOAP Web Services should read the overview provided below, and then refer to the documentation for the specific web service(s) of interest.

The OpenClinica SOAP API is part of the OpenClinica-ws package and require installation separate from the standard OpenClinica-web deployment (unlike REST web services which are part of OpenClinica-web). See 'Using OpenClinica Web Services' for more info.

Approved for publication by Ben Baumann. Signed on 2014-07-18 9:04AM

Not valid unless obtained from the OpenClinica document management system on the day of use.

2.1 Using OpenClinica Web Services

Installation

To use OpenClinica Web Services, they must first be installed. Follow the instructions in the installation guide in your OpenClinica download, or contact your OpenClinica Enterprise support representative.

The OpenClinica Web Services module is deployed on Apache Tomcat as a separate "war" (web application archive) from the main OpenClinica application. A typical OpenClinica Web Services installation is as follows: the 2 directories ("OpenClinica" for 3.1.x-web and "OpenClinica-ws" for

3.1.x-ws) are present in the tomcat/webapps with the same db connection parameters (db=openclinica and clinica user).

In this configuration, you can access the usual web interface through https://yourDomain.com/OpenClinica, and access the web services by SOAP requests using https://yourDomain.com/OpenClinica-ws. When you verify (as it is explained in the documentation) your -ws installation by loading https://yourDomain.com/OpenClinica-ws in your browser, it is only for confirming the application is deployed and can connect to the database. It will return a login page with an empty rss feed, but you cannot access the web application through this login page.

You can then begin to work on SOAP requests, authenticating with the login and the password of a user with the option "Authorize SOAP web services in this account" ticked.

Dates

All date values in OpenClinica Web Services should use the (ISO 8601) YYYY-MM-DD format.

OpenClinica SOAP Web Services Security

OpenClinica Web services use the same security infrastructure as the OpenClinica web application. A valid username and password are required, and the roles/permissions for that account will apply. Passwords should not be added in plain-text, they must be hashed using SHA-1 before being

To authenticate with OpenClinica SOAP Web services:

- Make sure the user account is authorized to use web services (this authorization is granted in User Account setup).
- Hash the password using the SHA-1 algorithm (google for a 'SHA-1 hash generator' if you don't know what this is).
- Modify the "<soapenv:Header/>" line in the SOAP XML request with the following. Provide the user name in clear text and the hashed password string in the appropriate fields.

```
<soapenv:Header>
<wsse:Security soapenv:mustUnderstand="1"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
<wsse:UsernameToken wsu:Id="UsernameToken-27777511"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<wsse:Username>username</wsse:Username>
<wsse:Password
type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">
SHA1-password</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
</soapenv:Header>
```

Spring XwsSecurityInterceptor is used to secure OpenClinica Web Services. For more information on Spring XwsSecurity go to:

http://static.springsource.org/spring-ws/sites/1.5/reference/html/security.html

Approved for publication by Cal Collins. Signed on 2016-05-09 4:55PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

2.2 StudySubject Web Service

This service provides API methods for manipulating subjects in OpenClinica. The WSDL file associated with this service can be found here: http://${your instance}/OpenClinica-ws/ws/studySubject/v1/studySubjectWsdl.wsdl.

Approved for publication by Cal Collins. Signed on 2014-04-02 3:55PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

2.2.1 create (Creates a new studySubject)

Create Method generates a new Study Subject.

Usage

Creates a study subjects in a target study/site. Required and optional fields will be determined by the configuration of the study and site the subject is being created in.

SOAP XML Request and Response Template

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://openclinica.org/ws/studySubject/v1"
xmlns:bean="http://openclinica.org/ws/beans">
${insert header}
<soapenv:Body>
<v1:createRequest>
<!--1 or more repetitions:-->
<v1:studySubject>
<bean:label>?</bean:label>
<!--Optional:-->
<bean:secondaryLabel>?</bean:secondaryLabel>
```

```
<bean:enrollmentDate>?</bean:enrollmentDate>
<bean:subject>
<!--Optional:-->
<bean:uniqueIdentifier>?</bean:uniqueIdentifier>
<!--Optional:-->
<bean:gender>?</bean:gender>
<!--You have a CHOICE of the next 2 items at this level-->
<bean:dateOfBirth>?</bean:dateOfBirth>
<bean:yearOfBirth>?</bean:yearOfBirth>
</bean:subject>
<bean:studyRef>
<bean:identifier>?</bean:identifier>
<!--Optional:-->
<bean:siteRef>
<bean:identifier>?</bean:identifier>
</bean:siteRef>
</bean:studyRef>
</v1:studySubject>
</v1:createRequest>
</soapenv:Body>
</soapenv:Envelope>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<createResponse>
<result>success</result>
<label>?</label>
</createResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Parameters

Request Parameters

Parameter	UI	Domain	DB	Validations	Datatypes	Examples	Required
label	Study Subject ID	StudySubjectBean:label	study_subject:label	- cannot be longer than 30 characters	String	<label>SSID001</label> Note: The OID of the study subject is truncated if the label contains 9 or more characters. For example, if you create a subject with a Study Subject ID # ALC5000001 --> the Study Subject OID saved in the database is: SS_ALC50000 instead of SS_ALC5000001 If there is another study subject containing the same first 8 characters as the first study subject,the second study subject will have the same OID as the first study subject with an appended 4 randomly assigned digits to ensure uniqueness. For example, if after you submitted a request to create a Study subject of ALC5000001, you submit a request to create a Study Subject ID # ALC5000002, the Study Subject OID saved in the database is: SS_ALC50000_5363 instead of SS_ALC5000002	Y - if "Study Parameter Configuration - Generate the Study Subject ID" is set to *Manual Entry*. N - if "Study Parameter Configuration - Generate the Study Subject ID" is set to *Auto-generated and Editable* or *Auto-generated and Non-editable*.
secondaryLabel	Secondary ID	StudySubjectBean:secondaryLabel	study_subject:secondary_label	- cannot be longer than 30 characters	String	<secondaryLabel>?</secondaryLabel>	N
enrollmentDate	Date of Enrollment	StudySubjectBean:enrollmentDate	study_subject:enrollment_date	- should be a valid date in the ISO 8601 format - date should be in the past	Date	<enrollmentDate>2008-12-12</enrollmentDate>	Y
subject:uniqueIdentifier	Person ID	SubjectBean:uniqueIdentifier	subject:unique_identifier	- cannot be longer than 255 characters	String	<uniqueIdentifier>PID001</uniqueIdentifier>	Y - if "Study Parameter Configuration - Person ID Required" is set to *Required*. N - if "Study Parameter Configuration - Person ID Required" is set to *Optional* or *Not Used*.
subject:gender	Sex	SubjectBean:gender	subject:gender	- should be 'm' or 'f'	char	<gender>m<gender>	Y - if "Study Parameter Configuration - Sex Required" is set to *Yes*. N - if "Study Parameter Configuration - Sex Required" is set to *No*.
subject:dateOfBirth	Date of Birth	SubjectBean:dateOfBirth	subject:date_of_birth	- should be a valid date in the ISO 8601 format	Date	<dateOfBirth>2008-12-12</dateOfBirth>	Y - if "Study Parameter Configuration - Collect Subject Date of Birth" is set to *Yes*. N - if "Study Parameter Configuration - Collect Subject Date of Birth" is set to *Not Used or Only Year of Birth*.
subject:yearOfBirth	Date of Birth	SubjectBean:dateOfBirth	subject:date_of_birth	- should be a valid year	Date	<yearOfBirth>2008</yearOfBirth>	Y - if "Study Parameter Configuration - Collect Subject Date of Birth" is set to *Only Year of Birth*. N - if "Study Parameter Configuration - Collect Subject Date of Birth" is set to *Not Used* or *Yes*.
studyRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	- should be a valid study identifier	String	<identifier>StudyA</identifier>	Y
siteRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	- should be a valid site identifier	String	<identifier>SiteA</identifier>	N

Response Parameters

Parameter	UI	Domain	DB	Result on success	Result on fail
label	Study Subject ID	StudySubjectBean:label	study_subject:label	<label>SSID001</label>	<label/>
result	n/a	n/a	n/a	<result>Success</result>	<result>Fail</result>
warning	n/a	n/a	n/a	n/a	n/a
error	n/a	n/a	n/a	n/a	<error>An error message indicating reason of failure</error>

2.2.2 listAllByStudy (List studySubjects in a study)

Lists studySubjects in a study regardless of its state (i.e It also includes a removed studySubject in a study).

Usage

Lists study subjects that are a part of an OpenClinica study or site.

SOAP XML Request and Response Template

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://openclinica.org/ws/studySubject/v1"
xmlns:bean="http://openclinica.org/ws/beans">
${insert header}
<soapenv:Body>
<v1:listAllByStudyRequest>
<bean:studyRef>
<bean:identifier>?</bean:identifier>
<!--Optional:
<bean:siteRef>
<bean:identifier>?</bean:identifier>
</bean:siteRef>--!>
</bean:studyRef>
</v1:listAllByStudyRequest>
</soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<ns4:listAllByStudyResponse
xmlns:ns4="http://openclinica.org/ws/studySubject/v1"
```

```
xmlns:ns2="http://openclinica.org/ws/beans"
xmlns:ns3="http://openclinica.org/ws/crf/v1">
<ns4:result>Success</ns4:result>
<ns4:studySubjects>
<ns2:studySubject>
<ns2:label>label</ns2:label>
<ns2:secondaryLabel/>
<ns2:enrollmentDate>2010-03-31</ns2:enrollmentDate>
<ns2:subject>
<ns2:uniqueIdentifier/>
<ns2:gender>m</ns2:gender>
</ns2:subject>
<ns2:events>
<ns2:event>
<ns2:location>Boston</ns2:location>
<ns2:startDate>2010-03-31</ns2:startDate>
<ns2:startTime>00:00:00</ns2:startTime>
</ns2:event>
</ns2:events>
</ns2:studySubject>
</ns4:studySubjects>
</ns4:listAllByStudyResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Parameters

Request Parameters

Parameter	UI	Domain	DB	Validations	Datatypes	Examples	Required
studyRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	N/A	String	<identifier>study identifier</identifier>	Y
siteRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	N/A	String	<identifier>site identifier</identifier>	N

Response Parameters

Parameter	UI	Domain	DB	Result on success	Result on fail
result	n/a	n/a	n/a	<result>Success</result>	<result>Fail</result>
warning	n/a	n/a	n/a	<warning>A warning message</warning>	n/a
error	n/a	n/a	n/a	n/a	<error>An error message indicating reason of failure</error>
studySubject:label	StudySubject ID	StudySubjectBean:label	study_subject:label	<label>label</label>	n/a
studySubject:secondaryLabel	Secondary ID	StudySubjectBean:secondaryLabel	study_subject:secondary_label	<secondaryLabel/> or <secondaryLabel>label</secondaryLabel>	n/a
subject:uniqueIdentifier	Person ID	SubjectBean:uniqueIdentifier	subject:unique_identifier	<uniqueIdentifier/> or <uniqueIdentifier>uId</uniqueIdentifier>	n/a
subject:gender	Sex	SubjectBean:gender	subject:gender	<gender>m</gender> or <gender>f</gender>	n/a
subject:dateOfBirth	Date of Birth	SubjectBean:dateOfBirth	subject:date_of_birth	<dateOfBirth>2008-12-12</dateOfBirth>	n/a
subject:yearOfBirth	Date of Birth	SubjectBean:dateOfBirth	subject:date_of_birth	<yearOfBirth>2008</yearOfBirth>	n/a

Parameter	UI	Domain	DB	Result on success	Result on fail
event:location	Location	StudyEvent:location	study_event:location	<location>Boston</location>	n/a
event:startDate	Start Date/Time	StudyEvent:dateStarted	study_event:date_start	<startDate>2010-08-16</startDate>	n/a
event:startTime	Start Date/Time	StudyEvent:dateStarted	study_event:date_start	<startTime>12:00:00</startTime>	n/a
event:endDate	End Date/Time	StudyEvent:dateEnded	study_event:date_end	<endDate>2010-08-16</endDate>	n/a
event:endTime	End Date/Time	StudyEvent:dateEnded	study_event:date_ended	<endTime>15:00:00</endTime>	n/a

2.2.3 isStudySubject (Queries OpenClinica if studySubject exists in a Study)

Queries OpenClinica if Study Subject exists.

Usage

Queries OpenClinica if a study subjects is assigned to a target study/site. Study Subject is looked up by Label (aka Study Subject ID) only.
If only the Study ID is provided (and not the Site ID), the web service will check for the Study Subject across all sites in the study and, if found, return a result of 'Success'. If a Site Identifier is also provided the scope of the query will be limited to subjects in that study and site.

SOAP XML Request and Response Template

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://openclinica.org/ws/studySubject/v1"
xmlns:bean="http://openclinica.org/ws/beans">
${insert header}
 <soapenv:Body>
<v1:isStudySubjectRequest>
<v1:studySubject>
<bean:label>?</bean:label>
<!--Lines selected in grey can be deleted from envelope. This
information is NOT used by WS-->
 <!--Optional:-->
 <bean:secondaryLabel>?</bean:secondaryLabel>
 <bean:enrollmentDate>?</bean:enrollmentDate>
 <bean:subject>
 <!--Optional:-->
 <bean:uniqueIdentifier>?</bean:uniqueIdentifier>
 <!--Optional:-->
 <bean:gender>?</bean:gender>
 <!--You have a CHOICE of the next 2 items at this level-->
 <bean:dateOfBirth>?</bean:dateOfBirth>
 <bean:yearOfBirth>?</bean:yearOfBirth>
 </bean:subject>
<bean:studyRef>
<bean:identifier>?</bean:identifier>
```

```
<!--Optional:-->
<bean:siteRef>
<bean:identifier>?</bean:identifier>
</bean:siteRef>
</bean:studyRef>
</v1:studySubject>
</v1:isStudySubjectRequest>
</soapenv:Body>
</soapenv:Envelope>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<createResponse>
<result>success</result>
<label>?</label>
</createResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Parameters

Request Parameters

Parameter	UI	Domain	DB	Validations	Datatypes	Examples	Required
label	Study Subject ID	StudySubjectBean:label	study_subject:label	- cannot be longer than 30 characters	String	<label>Sub A 101</label>	Y
studyRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	- should be a valid study identifier	String	<identifier>StudyA</identifier>	Y
siteRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	- should be a valid site identifier	String	<identifier>SiteA</identifier>	N

Response Parameters

Parameter	UI	Domain	DB	Result on success	Result on fail
studySubject:oc_oid	Study Subject OID	studySubjectOID	study_subject:oc_oid	<label>SSID001</label>	<label/>
result	n/a	n/a	n/a	<result>Success</result>	<result>Fail</result>

Parameter	UI	Domain	DB	Result on success	Result on fail
warning	n/a	n/a	n/a	n/a	n/a
error	n/a	n/a	n/a	n/a	<error>An error message indicating reason of failure</error>

2.3 Event Web Service

This service provides API methods for manipulating study events in OpenClinica. The WSDL file associated with this service could be found here: http://${your instance}/OpenClinica-ws/ws/event/v1/eventWsdl.wsdl.

2.3.1 schedule (Schedules an event)

Schedule an event

Usage

Schedules a single study subject event (visit) in an OpenClinica Study.

NOTE: the scheduleEvent WSDL states that more than one event can be scheduled at a time, however the application will only schedule one event per request. See 13008 for further information.

SOAP XML Request and Response Template

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://openclinica.org/ws/event/v1"
xmlns:bean="http://openclinica.org/ws/beans">
${insert header}
<soapenv:Body>
<v1:scheduleRequest>

<v1:event>
<bean:studySubjectRef>
<bean:label>?</bean:label>
</bean:studySubjectRef>
<bean:studyRef>
```

```
<bean:identifier>?</bean:identifier>
<!--Optional:-->
<bean:siteRef>
<bean:identifier>?</bean:identifier>
</bean:siteRef>
</bean:studyRef>
<bean:eventDefinitionOID>?</bean:eventDefinitionOID>
<bean:location>?</bean:location>
<bean:startDate>?</bean:startDate>
<!--Optional:-->
<bean:startTime>?</bean:startTime>
<!--Optional:-->
<bean:endDate>?</bean:endDate>
<!--Optional:-->
<bean:endTime>?</bean:endTime>
</v1:event>
</v1:scheduleRequest>
</soapenv:Body>
</soapenv:Envelope>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<scheduleResponse>
<result>success</result>
<eventDefinitionOID>SE_TEST</eventDefinitionOID>
<studySubjectOID>SS_PERSON</studySubjectOID>
<studyEventOrdinal>1</studyEventOrdinal>
</scheduleResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Parameters

Request Parameters

Element	UI	Domain	DB	Validations	Datatypes	Examples	Required
studySubjectRef:label	Study Subject ID	StudySubjectBean:label	study_subject:label	- should be a valid study subject ID	String	<label>SSID001</label>	Y
eventDefinitionOID	OID (Manage All Event Definitions Screen)	StudyEventDefinitionBean:oid	study_event_definition:oc_oid	- should be a valid event definition OID	String	<eventDefinitionOID>OC_OID</eventDefinitionOID>	Y
location	Location (Schedule Study Event Screen)	StudyEventBean:location	study_event:location	n/a	String	<location>Waltham</location>	Y

Element	UI	Domain	DB	Validations	Datatypes	Examples	Required
startDate	Start Date/Time (Schedule Study Event Screen)	StudyEventBean:dateStarted	study_event:date_start	- should be a valid date in the ISO 8601 format	Date	<startDate>2008-12-12</startDate>	Y
startTime	Start Date/Time (Schedule Study Event Screen)	StudyEventBean:dateStarted	study_event:date_start	- should be a valid time in the ISO 8601 format	Date	<startTime>12:00</startTime>	N
endDate	End Date/Time (Schedule Study Event Screen)	StudyEventBean:dateEnded	study_event:date_end	- should be a valid date in the ISO 8601 format	Date	<endDate>2008-12-12</endDate>	N
endTime	End Date/Time (Schedule Study Event Screen)	StudyEventBean:dateEnded	study_event:date_end	- should be a valid time in the ISO 8601 format	Date	<endTime>15:00</endTime>	N
studyRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	- should be a valid study identifier	String	<identifier>StudyA</identifier>	Y
siteRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	- should be a valid site identifier	String	<identifier>SiteA</identifier>	N

Response Parameters

Parameter	UI	Domain	DB	Result on success	Result on fail
eventDefinitionOID	OID (Manage All Event Definitions Screen)	StudyEventDefinitionBean:oid	study_event_definition: oc_oid	<eventDefinitionOID>SE_TEST</eventDefinitionOID>	n/a
studySubjectOID	OID (Subject Matrix)	StudySubjectBean:oid	study_subject:oc_oid	<studySubjectOID>SS_OID</studySubjectOID>	n/a
studyEventOrdinal	Event (Occurrence Number) - (View Subject Screen	StudyEventBean:ordinal	study_event:sample_ordinal	<studyEventOrdinal>1</studyEventOrdinal>	n/a
result	n/a	n/a	n/a	<result>Success</result>	<result>Fail</result>
warning	n/a	n/a	n/a		n/a
error	n/a	n/a	n/a	n/a	<error>An error message indicating reason of failure</error>

2.4 Data Web Service

This service provides API methods for manipulating CRF item data in OpenClinica. The WSDL file associated with this service could be found here: http://host:8080/OpenClinica/ws/data/v1/dataWsdl.wsdl .

2.4.1 Import

Usage

This web service inserts CRF item data. The data supplied to the method should be in CDISC ODM XML format.

Starting with OpenClinica release 3.6, the **UpsertOn** tag and **OpenClinica:Status** clause are available. Details are provided in CDISC ODM XML format

Starting with OpenClinica release 3.1.3 if you have Rules attached to your CRF, you can run them when uploading data from web services by turning on ImportDataEntry flag in rules definition. For example, in order to enable a discrepancy note rule action executed on import the rule should have ImportDataEntry flag set to true:

```
<?xml version="1.0" encoding="UTF-8"?>

<RuleImport>

   <RuleAssignment>

      <Target>I_RULES_WEIGHT</Target>

      <RuleRef OID="RU2">

         <DiscrepancyNoteAction IfExpressionEvaluates="false">

            <Run AdministrativeDataEntry="true"

               InitialDataEntry="true" DoubleDataEntry="true"

               ImportDataEntry="true" Batch="false"/>

            <Message>This is a discrepancy note test</Message>

         </DiscrepancyNoteAction>

      </RuleRef>

   </RuleAssignment>

   <RuleDef OID="RU2" Name="Blank exam date">

      <Description>CRF can not be blank.</Description>

      <Expression>I_RULES_HEIGHT eq "100"</Expression>

   </RuleDef>

</RuleImport>
```

SOAP XML Request and Response Template

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://openclinica.org/ws/data/v1"
xmlns:OpenClinica="http://www.openclinica.org/ns/odm_ext_v130/v3.1"
>
<soapenv:Header>
<wsse:Security soapenv:mustUnderstand="1"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss
-wssecurity-secext-1.0.xsd">
<wsse:UsernameToken wsu:Id="UsernameToken-27777511"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
 <wsse:Username>user</wsse:Username>
 <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-usern
ame-token-profile-1.0#PasswordText">password</wsse:Password>
 </wsse:UsernameToken>
 </wsse:Security>
</soapenv:Header>
 <soapenv:Body>
 <v1:importRequest>
<odm><![CDATA[
<ODM ....
 .....

 </ODM>]]>
 </odm>
 </v1:importRequest>
</soapenv:Body>
</soapenv:Envelope>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>

 .....
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Parameters

Request Parameters

Parameter	UI	Domain	DB	Validations	Datatypes	Examples	Required
studyRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	N/A	String	<identifier>study identifier</identifier>	Y
siteRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	N/A	String	<identifier>site identifier</identifier>	N

Response Parameters

Parameter	UI	Domain	DB	Result on success	Result on fail
result	n/a	n/a	n/a	<result>Success</result>	<result>Fail</result>
warning	n/a	n/a	n/a	<warning>A warning message</warning>	n/a
error	n/a	n/a	n/a	n/a	<error>An error message indicating reason of failure</error>

2.5 Study Web Service

This service provides API methods for manipulating studies in OpenClinica. The WSDL file associated with this service could be found here: http://host:8080/OpenClinica/ws/study/v1/studyWsdl.wsdl

Approved for publication by Cal Collins. Signed on 2014-04-02 3:51PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

2.5.1 getMetadata (Returns study metadata)

Returns the study metadata. The roles associated with the user running this webservice will determine wether the study metadata is displayed.

Usage

Returns the study metadata for the requested study in OpenClinica. It will contain information like the Study Name, Unique Protocol ID, and Study OID.

SOAP XML Request and Response Template

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://openclinica.org/ws/study/v1"
xmlns:bean="http://openclinica.org/ws/beans">
```

```
<soapenv:Header>
<wsse:Security soapenv:mustUnderstand="1"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss
-wssecurity-secext-1.0.xsd">
<wsse:UsernameToken wsu:Id="UsernameToken-27777511"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
<wsse:Username>user</wsse:Username>
<wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-usern
ame-token-profile-1.0#PasswordText">password</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
</soapenv:Header>
<soapenv:Body>
<v1:getMetadataRequest>
<v1:studyMetadata>
<bean:studyRef>
<bean:identifier>identifier</bean:identifier>
</bean:studyRef>
</v1:studyMetadata>
</v1:getMetadataRequest>
</soapenv:Body>
</soapenv:Envelope>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<createResponse xmlns="http://openclinica.org/ws/study/v1">
<result>Success</result>
<odm><![CDATA[<Study OID="S_DEFAULTS1">
<GlobalVariables>
<StudyName>Default Study</StudyName>
<StudyDescription>

</StudyDescription>
<ProtocolName>default-study</ProtocolName>
</GlobalVariables>
<MetaDataVersion>
<Protocol>
</Protocol>
</MetaDataVersion>
</Study>]]></odm>
</createResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Parameters

Request Parameters

Parameter	UI	Domain	DB	Validations	Datatypes	Examples	Required
studyRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	N/A	String	<identifier>study identifier</identifier>	Y
siteRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	N/A	String	<identifier>site identifier</identifier>	N

Response Parameters

Parameter	UI	Domain	DB	Result on success	Result on fail
result	n/a	n/a	n/a	<result>Success</result>	<result>Fail</result>
warning	n/a	n/a	n/a	<warning>A warning message</warning>	n/a
error	n/a	n/a	n/a	n/a	<error>An error message indicating reason of failure</error>

2.5.2 listAll (Lists studies)

Returns a list of studies/sites. The roles associated with the user running this webservice will determine which studies/sites are listed.

Usage

Returns a list of all the studies and sites that are part of your OpenClinica instance and accessible by the web services user. The system will not list studies and sites the user does not have access to.

SOAP XML Request and Response Template

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://openclinica.org/ws/study/v1">
<soapenv:Header>
<wsse:Security soapenv:mustUnderstand="1"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss
-wssecurity-secext-1.0.xsd">
<wsse:UsernameToken wsu:Id="UsernameToken-27777511"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
```

```
<wsse:Username>user</wsse:Username>
<wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">password</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
</soapenv:Header>
<soapenv:Body>
<v1:listAllRequest/>
</soapenv:Body>
</soapenv:Envelope>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<listAllResponse xmlns="http://openclinica.org/ws/study/v1">
<result>Success</result>
<studies>
<study>
<identifier>default-study</identifier>
<oid>S_DEFAULTS1</oid>
<name>Default Study</name>
</study>
<study>
<identifier>Study A</identifier>
<oid>S_STUDYA</oid>
<name>A new study</name>
<sites>
<site>
<identifier>Site A</identifier>
<oid>S_SITEA</oid>
<name>1 Site</name>
</site>
</sites>
</study>
</studies>
</listAllResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Parameters

Request Parameters

Not Available

Response Parameters

Parameter	UI	Domain	DB	Result on success	Result on fail
result	n/a	n/a	n/a	<result>Success</result>	<result>Fail</result>
warning	n/a	n/a	n/a	<warning>A warning message</warning>	n/a
error	n/a	n/a	n/a	n/a	<error>An error message indicating reason of failure</error>
study:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	<identifier>default-study</identifier>	n/a
study:oid	OID	StudyBean:oid	study:oc_oid	<oid>S_STUDY</oid>	n/a
study:name	Name	StudyBean:name	study:name	<name>Default Study</name>	n/a
site:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	<identifier>default-site</identifier>	n/a
site:oid	OID	StudyBean:name	study:oc_oid	<oid>S_SITE</oid>	n/a
site:name	Name	StudyBean:oid	study:name	<name>Default Site</name>	n/a

2.6 StudyEventDefinition Web Service

This service provides API methods for retrieving study event definitions for a study in OpenClinica. The WSDL file associated with this service can be found here: ws/studyEventDefinition/v1/studyEventDefinitionWsdl.wsdl.

2.6.1 listAll (Lists study event definitions in study)

Returns a list of study event definitions.

Usage

Returns a list of study event definitions and their properties for a single study in OpenClinica.

SOAP XML based Request and Response Template

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns:v1="http://openclinica.org/ws/studyEventDefinition/v1"
xmlns:bean="http://openclinica.org/ws/beans">
<soapenv:Header>
<wsse:Security soapenv:mustUnderstand="1"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss
-wssecurity-secext-1.0.xsd">
<wsse:UsernameToken wsu:Id="UsernameToken-27777511"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
<wsse:Username>user</wsse:Username>
<wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-usern
ame-token-profile-1.0#PasswordText">password</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
</soapenv:Header>
<soapenv:Body>
<v1:listAllRequest>
<v1:studyEventDefinitionListAll>
<bean:studyRef>
<bean:identifier>PG33xyz</bean:identifier>
</bean:studyRef>
</v1:studyEventDefinitionListAll>
</v1:listAllRequest>
</soapenv:Body>
</soapenv:Envelope>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<listAllResponse
xmlns="http://openclinica.org/ws/studyEventDefinition/v1">
<result>Success</result>
<studyEventDefinitions>
<studyEventDefinition>
<oid>SE_INITIALV</oid>
<name>Initial Visit</name>
<eventDefinitionCrfs>
<eventDefinitionCrf>
<required>true</required>
<doubleDataEntry>false</doubleDataEntry>
<passwordRequired>false</passwordRequired>
<hideCrf>false</hideCrf>
<sourceDataVerificaiton>100% Required</sourceDataVerificaiton>
<crf>
<oid>F_DEMO</oid>
```

```
<name>Demographics-Dynamics</name>
</crf>
<defaultCrfVersion>
<oid>F_DEMO_1</oid>
<name>1</name>
</defaultCrfVersion>
</eventDefinitionCrf>
<eventDefinitionCrf>
<required>true</required>
<doubleDataEntry>false</doubleDataEntry>
<passwordRequired>false</passwordRequired>
<hideCrf>false</hideCrf>
<sourceDataVerificaiton>Not Required</sourceDataVerificaiton>
<crf>
<oid>F_GROU</oid>
<name>Groups_Adverse_Events</name>
</crf>
<defaultCrfVersion>
<oid>F_GROU_NONC</oid>
<name>Non-Cascade</name>
</defaultCrfVersion>
</eventDefinitionCrf>
</eventDefinitionCrfs>
</studyEventDefinition>
</studyEventDefinitions>
</listAllResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Parameters

Request Parameters

Parameter	UI	Domain	DB	Validations	Datatypes	Examples	Required
studyRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	N/A	String	<identifier>study identifier</identifier>	Y
siteRef:identifier	Unique Protocol ID	StudyBean:identifier	study:unique_identifier	N/A	String	<identifier>site identifier</identifier>	N

Response Parameters

Parameter	UI	Domain	DB	Result on success	Result on fail
result	n/a	n/a	n/a	<result>Success</result>	<result>Fail</result>
warning	n/a	n/a	n/a	<warning>A warning message</warning>	n/a
error	n/a	n/a	n/a	n/a	<error>An error message indicating reason of failure</error>
studyEventDefinition:oid	OID	StudyEventDefinitionBean:oid	study_event_definition:oc_oid	<oid>SE_INITIALV</oid>	n/a
studyEventDefinition:name	Name	StudyEventDefinitionBean:name	study_event_definition:name	<name>S_STUDY</name>	n/a
eventDefinitionCrf:required	Required	EventDefinitionCRFBean:requiredCRF	event_definition_crf:required_crf	<required>true</required>	n/a
eventDefinitionCrf:doubleDataEntry	Double Data Entry	EventDefinitionCRFBean:doubleEntry	event_definition_crf:double_entry	<doubleDataEntry>false</doubleDataEntry>	n/a
eventDefinitionCrf:passwordRequired	Password Required	EventDefinitionCRFBean:electronicSignature	event_definition_crf:electronic_signature	<passwordRequired>false</passwordRequired>	n/a
eventDefinitionCrf:hideCrf	Hide CRF	EventDefinitionCRFBean:hideCrf	event_definition_crf:hide_crf	<hideCrf>false</hideCrf>	n/a

Parameter	UI	Domain	DB	Result on success	Result on fail
eventDefinitionCrf:sourceDataVerificaiton	Source Data Verification	EventDefinitionCRFBean:sourceDataVerification	event_definition_crf:source_data_verification_code	<sourceDataVerificaiton>100% Required</sourceDataVerificaiton>	n/a
crf:oid	OID	CRFBean:name	crf:name	<name>Demographics-Dynamics</name>	n/a
crf:name	Name	CRFBean:oid	crf:oc_oid	<oid>F_DEMO</oid>	n/a
defaultCrfVersion:oid	n/a	CRFVersionBean:oid	crf_version:oc_oid	<oid>F_DEMO</oid>	n/a
defaultCrfVersion:name	Default Version	CRFVersionBean:name	crf_version:name	<name>1</name>	n/a

2.7 Screencasts: Intro, Installation, and Usage Examples

About Web Service, and Installation

)

Usage Example: Create a Study Subject

)

Usage Example: List Study Subjects, If a Study Subject Exists, and Schedule Event

)

Approved for publication by Cal Collins. Signed on 2014-04-02 3:50PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

3 CRF Template Elements

Definitions, restrictions and when to use the OpenClinica CRF Template Elements

- CRF Worksheet
- Sections Worksheet
- Groups Worksheet
- Items Worksheet

OpenClinica CRF Spreadsheet Field	**Description**	**Can vary by CRF Version**	**Restrictions**	**When to use**
CRF Worksheet				

OpenClinica CRF Spreadsheet Field	**Description**	**Can vary by CRF Version**	**Restrictions**	**When to use**
CRF_NAME	Defines the name of the CRF as it will be displayed in the OpenClinica user interface. When a user is assigning CRFs to an event definition, they will be viewing this name. A user performing data entry will identify the form by this name.	No	Required Alpha-numeric characters are allowed 255 characters max UNIQUIE	If the field is blank, the CRF will be rejected at upload time.
VERSION	Defines the version of the CRF as it will be displayed in the OpenClinica user interface. You can not provide a value that has already been used in the OpenClinica instance unless it has not been assigned to an event definition yet. If a particular CRF version has not been used in an event definition, you may overwrite it.	Yes	Required Alpha-numeric characters are allowed 255 characters max UNIQUE	If this is a new version of a CRF that already exists, the CRF_NAME field must match the value of the form already in OpenClinica. A new version of a CRF would be needed due to a protocol change, adding or removing an item from a CRF, or changing some of the questions.
VERSION_DESCRIPTION	This field is used for informational purposes to keep track of what this version of the CRF was created for.	Yes	Required Alpha-numeric characters are allowed 4000 characters max NOT UNIQUE	This information appears as part of the CRF Metadata when the user clicks on View (original). This information is not displayed during data entry.
REVISION_NOTES	This field is used to keep track of the revisions you made to this particular CRF.	Yes	Required Alpha-numeric characters are allowed 255 characters max	This information appears as part of the CRF Metadata when the user clicks on View (original). This information is not displayed during data entry. If this is the first version of the CRF, you can simply state this is a brand new form. Going forward, as you make changes and update the versions you can provide information on what is different between the first version and each subsequent version.
Sections Worksheet				
SECTION_LABEL	Must be unique in the worksheet and contain no spaces. The SECTION_LABEL is referenced in the Items Worksheet to associate items with the appropriate section of the CRF. When the CRF is accessed for data entry, each section will be a page. The items defined with the corresponding SECTION_LABEL will be shown on that single page. This field is not displayed as part of the CRF but can be seen on the CRF Metadata page.	No (added or removed only)	Required Alpha-numeric No spaces 255 characters max UNIQUE	This value will be used in the Items Worksheet to define where the items will appear during data entry. A CRF must have at least one section.
SECTION_TITLE	The value in this field will be displayed at the top of each page when a user is performing data entry, as well as in the tabs and drop down list used to navigate between sections in a CRF. It does not have to be unique but should be a readable value that makes sense to people entering data. An example would be Inclusion Criteria	Yes	Required All characters allowed 2000 characters max	This field must be used at all times. If a CRF does not have a value for SECTION_TITLE the form will be rejected at upload time. Long section titles may not display well.
SUBTITLE	A sub-title shown under the section title.	Yes	All characters allowed 2000 characters max	HTML elements are supported for this field.
INSTRUCTIONS	Instructions at the top of the section (under the subtitle) that explains to the data entry person what to do on this section of the form.	Yes	All characters allowed 2000 characters max	HTML elements are supported for this field. This field should be used if there are particular data entry instructions that should be conveyed or followed to users.
PAGE_NUMBER	The page number on which the section begins. If using paper source documents and have a multi-page CRF, put in the printed page number.	Yes	All characters allowed 5 characters max	For the most part, this field is only used in studies collecting data on multi-page paper forms and then having the data keyed in at a central location performing double data entry.
PARENT_SECTION	Sections can be sub-sections of ones that have been previously defined. Provide the parent section for this sub-section in this field.		255 characters max Must equal a valid SECTION_TITLE	This field does not have any affect in the User Interface. It does not affect data being exported. This will be deprecated after version 3.1.3.
Groups Worksheet				
GROUP_LABEL	Must be unique in the worksheet and contain no spaces. GROUP_LABEL is referenced in the Items Worksheet to associate items with the appropriate item group in the CRF. The GROUP_LABEL value will be used to generate the GROUP OID to be used in Rules, Data Import, and Export. The field is not displayed as part of the CRF but can be seen on the CRF Metadata page.	No (added or removed only)	Alpha-numeric No spaces 255 characters max UNIQUE	This field should be used in order to logically group items together. Item groups may be used to define repeating items in the CRF, or for logical grouping of non-repeating items for easier data analysis once data has been exported from OpenClinica. The entire GROUPS worksheet can be left blank if desired and all the items in the CRF can be part of a single group called UNGROUPED. We suggest to provide records in this worksheet only if you are going to use Groups for Items grouping on Items worksheet.
GROUP_LAYOUT	Defines the type of item group (GRID or NON-REPEATING, Blank). A layout of GRID means the items defined as part of the item group can be dynamically repeated during the data entry process, and will be displayed in a horizontal grid where each item is represented as a column and users can add rows of data to the table during data entry. If NON-REPEATING is chosen, the fields will be logically grouped together in the CRF metadata, but cannot repeat and are displayed using the standard OpenClinica CRF layout. Logically associating items as a non-repeating group can be used with rules to implement (dynamic) show and hide logic. If left blank the item group will be treated as if it is set to NON-REPEATING.	Yes	Blank, GRID, NON-REPEATING	This field should be used to define the item groups as repeating (GRID) or non-repeating. Non-repeating item groups can include items from across multiple sections of the CRF. Items that are members of a repeating (GRID) item group must appear in the same section and must be placed together in the Items Worksheet. Definition of a repeating item group activates the use of the attributes GROUP_HEADER, GROUP_REPEAT_NUMBER and GROUP_REPEAT_MAX.
GROUP_HEADER	The value is displayed above the GRID when a user is performing data entry. Used only when the GROUP_LAYOUT is equal to GRID.	Yes	All characters allowed 255 characters max	Only use when the layout is set to GRID. This value is like a title for the group. An example of a GROUP_HEADER would be Medications Log. The field can be left blank if you do not want a title or header. If the Layout is set to NON-REPEATING, the value will be ignored and not displayed during data entry.
GROUP_REPEAT_NUMBER	The default (initial) number of repeats on the form should be provided here. If left blank, only one row of information will be displayed by default. Used only when the GROUP_LAYOUT is equal to GRID.	Yes	Integer Should be left blank for non GRID layout	This field should be used to specify how many rows of data should exist for the item group upon initiation of data entry, or in printing of a blank CRF from OpenClinica. If three rows of information, specify the number 3 in the field. When a user accesses the CRF, the row will be repeated 3 times by default. A user will be allowed to add more rows up to the GROUP_REPEAT_MAX and even remove some of the rows displayed by default. If the Layout is set to NON-REPEATING, the value will be ignored and not displayed during data entry.
GROUP_REPEAT_MAX	The total number of rows a user will be allowed to repeat in the item group. When left blank, the default number of repeats is 40. Used only when the GROUP_LAYOUT is equal to GRID.	Yes	Integer There is no max value Should be left blank for non GRID layout	This field should be used to restrict users to a certain number of repeats for the GRID. However, this restriction works only if data are entered through OpenClinica Web UI. If data are imported using Task-> Import Data option or through web services, all rows of data in the import file will be allowed to import, even if the rows of data in the import exceed the GROUP_REPEAT_MAX. If GROUP_REPEAT_MAX is less than GROUP_REPEAT_NUMBER group will have GROUP_REPEAT_MAX number of rows on initial data entry displayed and no additional rows can be added.
GROUP_DISPLAY_STATUS	Used in conjunction with Dynamics in Rules. If set to HIDE, the item group will not appear in the CRF when a user is entering data until certain criteria are met. This criteria is specified with a Rule and using the ShowAction. If left blank, the value defaults to SHOW.	Yes	Blank, SHOW, HIDE	If you would like to design skip patterns and dynamic logic for the item groups, set the display status to HIDE. When the form is accessed for data entry, the group of items will be hidden from view from the user. With Rules, entry of a desired value elsewhere in data entry can trigger the item group to then be shown instead of hidden (this applies to GRID and NON-REPEATING). The opposite is true as well. Groups that are set to HIDE which have been shown as a result of a rule can also be hidden based on certain criteria being met. Note that the GROUP is the container for items and that items within the group can have nested SHOW/HIDE logic as well. If a hidden group is shown but some of the items within that group have been set to HIDE within the ITEMS worksheet, additional logic will have to be built for those individual items. If all items within the group are set to SHOW within the items worksheet, then showing the time group will show all items.
Items Worksheet				

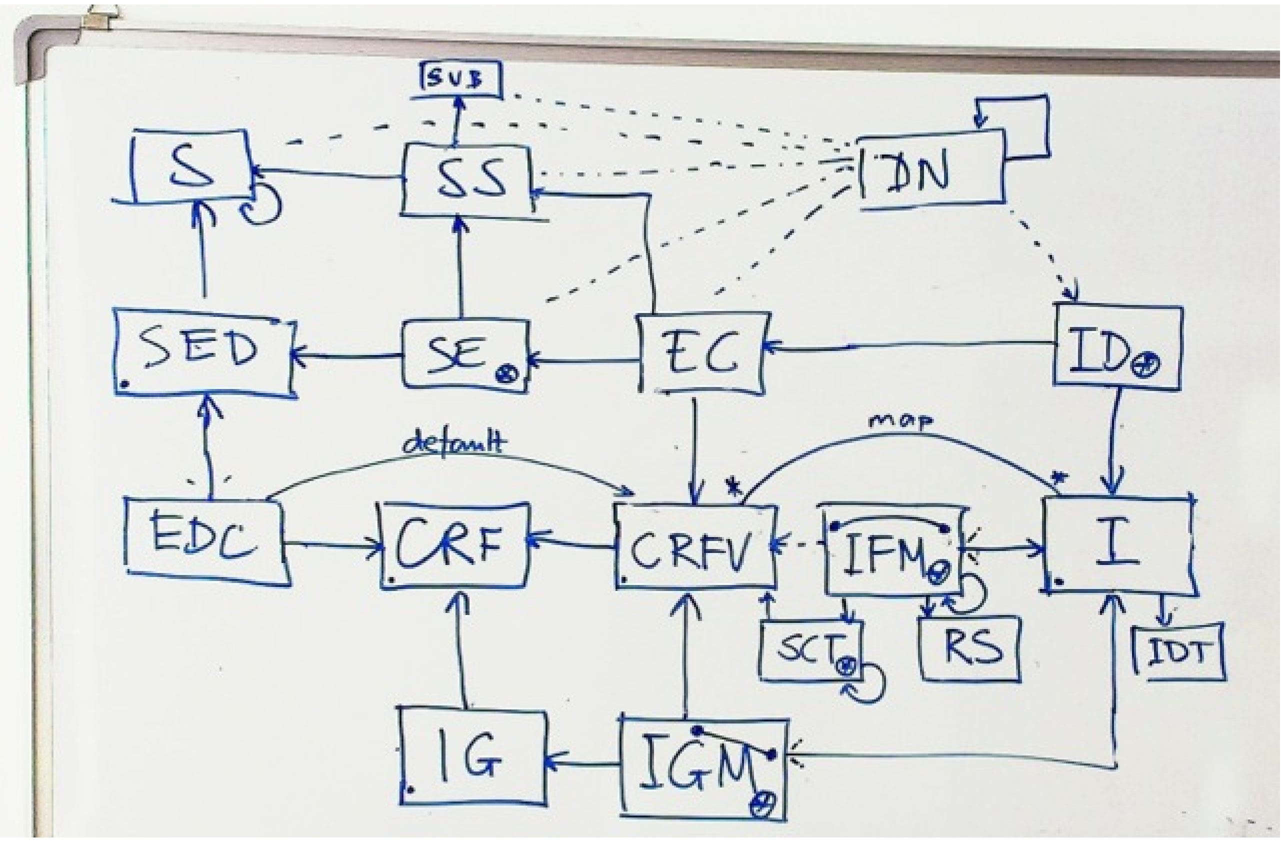
OpenClinica CRF Spreadsheet Field	**Description**	**Can vary by CRF Version**	**Restrictions**	**When to use**
ITEM_NAME	The unique label or variable name for the data element. The field is not displayed as part of the CRF but can be viewed as part of the CRF and Item Metadata, and is shown in the Discrepancy Notes and Rules modules. This field is case-sensitive. Items with names item1 and Item1 will be treated as different items. This can cause issues with many data analysis tools and should be avoided in most cases. **For Enterprise customers interested in using Datamart:** Please note that Datamart treat items in case-insensitive manner. Please treat all ITEM_NAMES as case-insensitive. Also for use with Datamart, Postgres has a list of reserved words and special characters which should not be used as item names.	No (added or removed only)	Required Only ASCII characters from "a" to "z", "A", to "Z", "0" to "9", and "_" are supported. No spaces. 255 characters max Must be UNIQUE within the CRF version	This field should be used at all times. ITEM_NAME will be used to form the OID and the variable name when exporting data from OpenClinica. Brevity is recommended for the value as it will be used to generate the unique OC_OID. Re-use of the same ITEM_NAME across CRF Versions indicates the variable is the same item. Once created, an item name cannot be modified within the CRF. See CRF Versioning and Scope of CRFs and Items in this document for more detail.
DESCRIPTION_LABEL	The description, or definition of the item. Should give an explanation of the data element and the value(s) it captures. It is not shown on the CRF but is in the data dictionary.	No	Required All characters allowed 4000 characters max	This field must be used at all times. Provide a description that will help explain what the variable means and what values it is collecting. For example, if the variable is looking to collect HEIGHT, the DESCRIPTION_LABEL would be This variable collects the height of the subject. It captures the value in inches. This field can not be changed in any subsequent versions of the CRF. If you do change it, it will be ignored on upload and the DESCRIPTION_LABEL provided with the first version of the CRF will be used.
LEFT_ITEM_TEXT	Descriptive text that appears to the left of the input on the CRF. Often phrased in the form of a question, or descriptive label for the form field input.	Yes	All characters allowed 2000 characters max	This field should be used as a way of describing the expected input to users entering or reviewing CRF data. The value of LEFT_ITEM_TEXT is displayed to the left of the form input. The text wraps after the first 20 characters. An example question would be What is the subjects height? Or, a simple one word Height suffices as well. If the item is part of a repeating group (GRID), the LEFT_ITEM_TEXT is displayed as a column header above the field and not be displayed to the left of the item. HTML elements are allowed; however, only limited subsets of tags are officially supported (bold <b>, italics <i>, underline <u>, superscript <sup>, subscript <sub>, line break <br/>, link <a href="">, image <img src="">). Tokens can also be used to substitute Study Objects. These can be used to display the Study Object value for use in a URL or for use in other scripts (e.g. jquery). The supported tokens are: $ {studySubject} $ {studyName} $ {eventName} $ {eventOrdinal} $ {crfName} $ {crfVersion} $ {item[item_name]} For more information please see: https://docs.openclinica.com/3.1/study-setup/build-study/create-case-report-forms-crfs#content-title-5514
UNITS	Used to define the type of values being collected. It appears to the right of the input field on the CRF.	No	64 characters max	If you are collecting data in Inches, this field can specify your units as Inches, IN, or in. This field can not be changed in any subsequent versions of the CRF. If you do change it, it will be ignored on upload and the UNITS provided with the first version of the CRF will be used. There are no edit checks associated specifically with units. This will appear as text to right of the input field and will be displayed between parenthesis. If you are exporting to CDISC ODM XML format, this will appear in the metadata as measurement units.
RIGHT_ITEM_TEXT	Descriptive text that appears to the right of the form input on the CRF, and to the right of any UNITS that are specified too. Often phrased in the form of a question, or supporting instructions for the form field input.	Yes	All characters allowed 2000 characters max	This field can be be used as a way of describing the expected input to users entering or for field-specific instructions. The value of RIGHT_ITEM_TEXT is displayed to the right of the form input. The text wraps after the first 20 characters. An example of use of right item text is If other, please specify. If the item is part of a repeating group (GRID), the RIGHT_ITEM_TEXT will be ignored and never displayed. HTML elements are allowed; however, only limited subsets of tags are officially supported (bold <b>, italics <i>, underline <u>, superscript <sup>, subscript <sub>, line break <br/>, link <a href="">, image <img src="">). Tokens can also be used to substitute Study Objects. These can be used to display the Study Object value for use in a URL or for use in other scripts (e.g. jquery). The supported tokens are: $ {studySubject} $ {studyName} $ {eventName} $ {eventOrdinal} $ {crfName} $ {crfVersion} $ {item[item_name]} For more information please see: https://docs.openclinica.com/3.1/study-setup/build-study/create-case-report-forms-crfs#content-title-5514
SECTION_LABEL	Logically organizes the items that should be together on a section. The items in a given section are displayed on a single web page when the user is entering data, and appear in the order they are entered in the Template. Every item in the worksheet must be assigned to a section of the CRF.	Yes	Required Value must exist on the Sections Worksheet	For example, all of the information collected as part of a physical exam like Height, Weight, Blood Pressure, and Heart Rate should be on the same section.
GROUP_LABEL	Assigns items to an item group. If the group is repeating, the items need to have the same SECTION_LABEL as all other items in the group and must be consecutively defined in the ITEMS worksheet. Repeating items are displayed on a single row with the LEFT_ITEM_TEXT (if any exists) as a column header.	Yes	Value must exist on the Groups Worksheet	This field should be used to identify whether an item belongs to an item group defined in the GROUPS worksheet. If the group is a repeating group (GRID layout), each item in the group is displayed as a column in the grid. Too many items in a group, or use of long LEFT_ITEM_TEXT values, will make the grid extremely wide and force the data entry user to scroll the page to the right to complete data entry. For non-repeating items, specify a group label to be used to logically assemble related items together for easier data analysis. OpenClinica 3.1.2 and previous versions allowed items to be moved from one item group to another between versions (i.e. UNGROUPED items could later be grouped). While OpenClinica allowed this functionality, ODM does not support this type of structure change between different CRF versions. As a result, these types of structural changes could break extracts which contain the affected CRF. A new table has been introduced to View CRF page to allow a user to verify CRF integrity. The new table, called Items gives a list of items in a CRF where the last two columns (Version(s) and Integrity Check) provide information about which version(s) the item belongs to and if the item passes the integrity check (verifying that the item has not been assigned to more than one item group). OpenClinica 3.1.3 and future versions will not allow items to be assigned to different item groups between versions.
HEADER	Contains text that used as a header for a particular item. Using this field will break up the items with a distinct line between the header information and the next set of items. The text is bolded to call greater attention to it.	Yes	All characters allowed 2000 characters max	This field can be used as a replacement for left and right item text or as a replacement for instructions. It allows a greater number of characters, along with bolding the text, to get the data entry persons attention. HTML elements are allowed; however, only limited subsets of tags are officially supported (bold <b>, italics <i>, underline <u>, superscript <sup>, subscript <sub>, line break <br/>, link <a href="">, image <img src="">). Tokens can also be used to substitute Study Objects. These can be used to display the Study Object value for use in a URL or for use in other scripts (e.g. jquery). The supported tokens are: $ {studySubject} $ {studyName} $ {eventName} $ {eventOrdinal} $ {crfName} $ {crfVersion} $ {item[item_name]} For more information please see: https://docs.openclinica.com/3.1/study-setup/build-study/create-case-report-forms-crfs#content-title-5514
SUBHEADER	This field can contain text that will be used underneath the HEADER, or independently of a HEADER being provided. The text will be separated by a line and have a grey background.	Yes	All characters allowed 240 characters max	This field can be used as a replacement or augmentation for left and right item text or as a replacement/augmentation for section/group instructions. It allows a greater number of characters, along with providing a grey background to the text in order to get the data entry users attention. HTML elements are allowed; however, only limited subsets of tags are officially supported (bold <b>, italics <i>, underline <u>, superscript <sup>, subscript <sub>, line break <br/>, link <a href="">, image <img src="">). Tokens can also be used to substitute Study Objects. These can be used to display the Study Object value for use in a URL or for use in other scripts (e.g. jquery). The supported tokens are: $ {studySubject} $ {studyName} $ {eventName} $ {eventOrdinal} $ {crfName} $ {crfVersion} $ {item[item_name]} For more information please see: https://docs.openclinica.com/3.1/study-setup/build-study/create-case-report-forms-crfs#content-title-5514
PARENT_ITEM	This field can contain an ITEM_NAME that immediately precedes this item, and is in the same section. This will cause the child item to be indented underneath the ITEM_NAME specified	Yes?	Value for ITEM_NAME must already exist in the CRF and immediately precede the current ITEM Only one level of indention is allowed	This can only be used with non-repeating items and must be a valid ITEM_NAME. It is used strictly for visual purposes in the user interface when people are entering data.
COLUMN_NUMBER	Data entry screens are set up by columns. By default a blank value will put the item in column 1. To have non-repeating items show up on a horizontal plane next to each other, specify column numbers greater than 1.	Yes	Integer	This is to be used with only non-repeating items and controls display of multiple items on a single row. If you set the column to 3 for an item, the previous two items in the worksheet should have COLUMN_NUMBERS of 1 and 2. Otherwise, it will just be applied to the first column. Use of COLUMN_NUMBERS greater than 3 is not recommended due to page width limitations.
PAGE_NUMBER	Page number where the item would appear. If using paper source documents and have a multi-page CRF, put in the printed page number.	Yes	Alpha-numeric 5 characters max	For the most part, this field is only used in studies collecting data on multi-page paper forms and then having the data keyed in at a central location performing double data entry.
QUESTION_NUMBER	This field is used to specify an identifier for each item or question in the Items worksheet. It appears to the left of the LEFT_ITEM_TEXT field, or if that field was left blank, to the left of the form input.	Yes	Alpha-numeric 20 characters max	This field allows you to specify questions as 1, 2, 2a etc. in a field.

OpenClinica CRF Spreadsheet Field	Description	Can vary by CRF Version	Restrictions	When to use
RESPONSE_TYPE	The types of responses are based on standard HTML elements web browsers can display in a form. Allowed use of the available RESPONSE_TYPEs depends on the item DATA_TYPE and use of Response Sets.	Yes	Required Option must be selected from the drop down box: -text -textarea -single-select -radio -multi-select -checkbox -calculation -group-calculation -file -instant-calculation (OpenClinica 3.1.3 and up)	Text is a rectangular box to enter information. Textarea is a larger box where more information is visible to the person viewing the form with data. Radio and Single-Select only allow one option to be chosen for an item. Radio buttons can not be deselected in the user interface once an option has been chosen. Multi-Select and Checkbox allow multiple options to be selected at once. Calculation and Group-Calculation are used to derive values. Calculations allow for the execution of arithmetic expressions and support some basic functions. The calculations use values from previous fields within the same CRF as variables. The calculated field can not be directly edited by the data entry person and will appear grayed out. Group-calculation allows the user to find a value based off of the column in a grid (e.g. sum). The group-calculation should not be contained in a repeating group, rather the variable that is being used in the calculation should be in a grid while the calculated field itself is non-repeating. The values of calculated fields are generated when the user saves the section of the form. Forced reason for change (when turned on) is not enforced for calculated fields. File allows a file to be uploaded and attached to the CRF by the data entry person. Instant-calculation (introduced in 3.1.3) is used to populate a destination field with date/time information when content of a parent field is changed. The trigger field must precede the intstant-calculation field. This is a client side action; it is executed by triggering onchange function defined by the CRF designer in the RESPONSE_VALUES_OR CALCULATIONS field in the CRF template. The onchange function takes arguments of item name (parent item) and value. The item name indicates the name of parent field. The value indicates what date format should be used (_CURRENT_DATE - the current server date, _CURRENT_DATE_TIME - current server date and time). Please note that instant-calculation field can be used in a repeating group only if its trigger field is on the same row of a repeating group. The trigger field for an instant-calculation field can be of any response type.
RESPONSE_LABEL	Create a custom label associated with a response set. This label must be defined once and may be reused by other items with the same responses (eg Yes, No) and values.	No	Required 80 characters max Must be alphanumeric	In order to facilitate the creation of a CRF, unnecessary duplication of RESPONSE_OPTIONS_TEXT and RESPONSE_VALUES_OR_CALCULATIONS values can be mitigated by the RESPONSE_LABEL. If the same options and values are going to be used for multiple items like Yes,No and 1,2, provide the information once and enter a unique response label. This label can be used throughout the rest of the Items worksheet so other items will use the exact same options and values. If a RESPONSE_LABEL is reused within a CRF, the RESPONSE_OPTIONS_TEXT and RESPONSE_VALUES_OR_CALCULATIONS must be left blank or exactly match the values of the original RESPONSE_LABEL in the CRF.
RESPONSE_OPTIONS_TEXT	A comma delimited string of values that will be used as the options to be chosen by a data entry person when they are entering data in a CRF.	Yes (with limitations)	Required for each unique RESPONSE_LABEL 4000 characters max	This field is only used for checkbox, multi-select, radio and single-select fields. This will be the text displayed to the data entry person which they will choose for each item. If the options themselves contain commas (,) you must escape the commas with a /
RESPONSE_VALUES_OR_CALCULATIONS	If the field is not a calculation or group-calculation, this will be a comma delimited string of values that will be used as the values saved to the database when a user chooses the corresponding RESPONSE_OPTIONS_TEXT. If this is a calculation or group-calculation field, it will be an expression that takes the inputs of other items in the Items worksheet that are of INT or REAL data type to calculate a value.	Yes (with limitations)	Required for each unique RESPONSE_LABEL 4000 characters max	For checkbox, multi-select, radio and single-select fields, this will be the values that correspond to a RESPONSE_OPTIONS_TEXT. The number of options and values must match exactly or the CRF will be rejected when it is uploaded into OpenClinica. Items with a RESPONSE_TYPE of "calculation" support arithmetic operators to populate the field with a derived value upon CRF save. ITEM_NAMEs of other INT or REAL items in the same CRF can be used as variables to compute the derived value. For example, the calculation for a line item total price in an invoice would be "func: (ITEM_PRICE * QTY)". The following functions are also allowed in the calculation expression: sum(), avg(), min(), max(), median(), stdev(), pow(), and decode(). Cumulative calculations on a group of repeating items must be of RESPONSE_TYPE group-calculation. Only cumulative calculations on the entire set of repeating items are allowed. The allowed functions are sum(), avg(), min(), max(), median(), and stdev(). For example, in an invoice with a repeating group of line items, the calculation for a total price would be the group-calculation func: (sum (LINE_ITEM_PRICE)). Cumulative calculations using the group-calculation input type cannot be in the same group with fields of a calculation input type. Also a group calculation field cannot use a calculation field as an input to its calculation.
RESPONSE_LAYOUT	The layout of the options for radio and checkbox fields. The options can be left to right, or top to bottom depending on the value specified in the Items worksheet	Yes	Blank, Horizontal, Vertical	Leaving the field blank and selecting Vertical display the items in a single column from top to bottom. Choosing Horizontal will put the items in a single row, left to right.
DEFAULT_VALUE	Default text for RESPONSE_OPTIONS_TEXT	Yes	4000 characters max	This field allows the user to specify a default value that will appear in the CRF section the first time the user accesses the form. For single-select default value does not have to be part of the response set and can be instructive text if need be. It will be interpreted as a blank value if the user does not choose anything. Default values can be used for the following RESPONSE_TYPEs: • TEXT • TEXTAREA • SINGLE-SELCT • MULTI-SELECT • CHECKBOX Default values can not be used for the following RESPONSE_TYPEs (CRF will be rejected on upload): • CALCULATION • GROUP_CALCULATION • FILE • INSTANT_CALCULATION • RADIO Be careful in using this field because if the default value corresponds to an option in the response set, it will be saved to the database even if the user does not select it.
DATA_TYPE	The data type is the format the value should be supplied in.	No	Required Option must be selected from the drop down box: -ST -INT -REAL -DATE -PDATE -FILE	ST = String. Any characters can be provided for this data type. INT = Integer. Only numbers with no decimal places are allowed for this data type. REAL = Numbers with decimal places are allowed for this data type. DATE = Only full dates are allowed for this data type. The default date format the user must provide the value in is DD-MMM-YYYY. PDATE = Partial dates are allowed for this data type. The default date format is DD-MMM-YYYY so users can provide either MMM-YYYY or YYYY values. FILE = This data type allows files to be attached to the item. It must be used in conjunction with a RESPONSE_TYPE of file. The attached file is saved to the server and a URL is displayed to the user viewing the form.
WIDTH_DECIMAL	Specify the width (the length of the field) and the number of decimal places to use for the field	Yes	Not required. If provided must be in the form w(d) as follows: w integer from 1 to 32, or literal w if INT or REAL. If ST, from 1 to 255 is allowed. d literal d. if the item has DATA_TYPE of REAL, may also be an integer from 1 to 20.	Defines the width (the maximum allowed length of the field) and the number of decimal places to use for the field in the form w(d). The first input defines the width of the field. The second input specifies the number of decimal places for the field, if the item has a DATA_TYPE of REAL. The WIDTH_DECIMAL attribute should only be used for items with the ST, INT or REAL data types. The width attribute specifies the length of the field treated as a string, so even if the item is of type INT or REAL, leading/trailing zeroes and decimal points count towards the width. For items of type REAL, evaluation of the width occurs prior to evalution of the decimal, so values exceeding the specified or system default width will be rejected even if they could be rounded to a length within the width limit. Examples.: DATA_TYPE REAL, WIDTH_DECIMAL 5(1) Allows a maximum of 5 characters with only 1 decimal place. OpenClinica will accept 12345 and 1234., 123.4, or 12.30 but will not accept 012345 or 123456. Inputs such as 12.345 or 1234.5678 or 012345 or 12.300 will be allowed and rounded. REAL w(4) Allows up to OpenClinicas maximum length for an item of ST, INT, or REAL (32 characters), with any decimal in excess of 1/10000th rounded to the 4th decimal place. REAL 20(d) Allows a maximum length of 20 and ahydrocephalusdecimal length of 4 (the default in OpenClinica). . ST 20(d) or INT 20(d) Allows a maximum length of 20 characters. If the DATA_TYPE of the item is DATE, PDATE, or FILE, the WIDTH_DECIMAL attribute should be left blank.
VALIDATION	Specify a validation expression to run an edit check on this item at the point of data entry.	Yes	1000 characters max	The validation will run when the user hits 'save'. If the user has entered data which satisfy the validation expression, data will save normally. If the value entered does not meet the requirements of the validation, an error message will appear (i.e., the VALIDATION_ERROR_MESSAGE) and the user will need to correct the value or enter a discrepancy note to continue. The validation should be of the format "expressionType: expression". Must be between 1 and 1000 characters and is an optional field. regexp: This Supports Java-style regular expressions (similar to Perl). For more information, see http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html Examples: This example requires a three-letter string (all uppercase) regexp: /regular expression/ = regexp: /[A-Z]{3}/ func: Supports built-in library of basic integer functions. Currently supported functions include: (1) greater than - gt(int) or gt(real) (2) less than - lt(int) or lt(real) (3) range - range(int1, int2) or range(real1, real2) (4) gte(int) or gte(real) (5) lte(int) or lte(real) (6) ne(int) or ne(real) (7) eq(int) or eq(real) Examples: This example requires a number between 1 and 10 func: func(args) = func: range(1, 10)
VALIDATION_ERROR_MESSAGE	Defines an error message provided to on the data entry screen when a user enters data that does not meet the VALIDATION.	Yes	255 characters max Required if VALIDATION is provided	Must be used when a VALIDATION is specified and it should be clear to the data entry person what the problem is. If there is a VALIDATION stating the number must be between 1-10, write that message in this field for the user to see if they enter 11 or 0.
PHI	Signifies whether this item would be considered Protected Health Information	No	Blank, 1, 0	Leaving the field blank or selecting 0 mean the item would not be considered Protected Health Information. This flag does not do anything to mask the data or prevent people from seeing it. The field is used more as a label. When creating a data set, this label will show in the metadata and the user could choose to include this item in the dataset (create dataset step) or not based on this label.
REQUIRED	This field determines whether the user must provide a value for it before saving the section the item appears in.	Yes	Blank, 1, 0	Leaving the field blank or selecting 0 mean the item would be optional so the data entry person does not have to provide a value for it. If 1 is selected, the data entry person must provide a value, or enter a discrepancy note explaining why the field is left blank. This can be used for any RESPONSE_TYPE

OpenClinica CRF Spreadsheet Field	Description	Can vary by CRF Version	Restrictions	When to use
ITEM_DISPLAY_STATUS	Used in conjunction with Dynamics in Rules or SIMPLE_CONDITIONAL_DISPLAY. If set to HIDE, the item will not appear in the CRF when a user is entering data unless certain conditions are met. The conditions for display are specified with a Rule using the ShowAction, or via SIMPLE_CONDITIONAL_DISPLAY. If left blank, the value defaults to SHOW.	Yes	Blank, SHOW, HIDE	If you would like to design skip patterns and dynamic logic for a single item, set the display status to HIDE. When the form is accessed for data entry, the item will be hidden from view from the user. With Rules, another value can trigger the group of items to be shown instead of hidden. Instead of Rules, you can use the SIMPLE_CONDITIONAL_DISPLAY field to decide when this item should be shown. SIMPLE_CONDITIONAL_DISPLAY only works with items set to HIDE.
SIMPLE_CONDITIONAL_DISPLAY	Contains 3 parts, all separated by a comma: ITEM_NAME, RESPONSE_VALUE, Message. ITEM_NAME - The item name of the field determining whether this hidden item becomes shown. RESPONSE_VALUE - The value of the ITEM_NAME that will trigger this hidden item to be shown Message - A validation message that will be displayed if this item has a value but should not be shown anymore.	Yes	Comma separated list	Simple Conditional Display works with items that have a defined response set (radio, checkbox, multi-select and single-select fields). The hidden item can be of any response type. Many levels of hierarchy are supported. Several levels of hierarchy of Simple Conditional fields work similar to a single parent-child simple conditional relationship. SIMPLE_CONDITIONAL_DISPLAY (SCD) has an effect only when ITEM_DISPLAY_STATUS (IDS) of the item is set to HIDE. For example, assume there is a SEX item with response options of Male, Female, and response values of 1,2. If the user chooses Female option, additional questions about pregnancy are asked. If Male is chosen, these questions are hidden. However, if the user chooses Female, fills in pregnancy data and after that gets back to the SEX item and switches the answer to Male, the items about pregnancy will remain on the screen (not hidden). The user can delete pregnancy answers and in that case the UI items will get hidden. Note that the database gets updated only on SAVE. In the above example the system will allow saving inconsistent data (SEX = Male, but pregnancy items filled), but it is up to a user to create discrepancy fields for them explaining the situation. However, radio button control cannot be deselected (meaning there is no way to delete its value once it was selected).

4 Database Schema

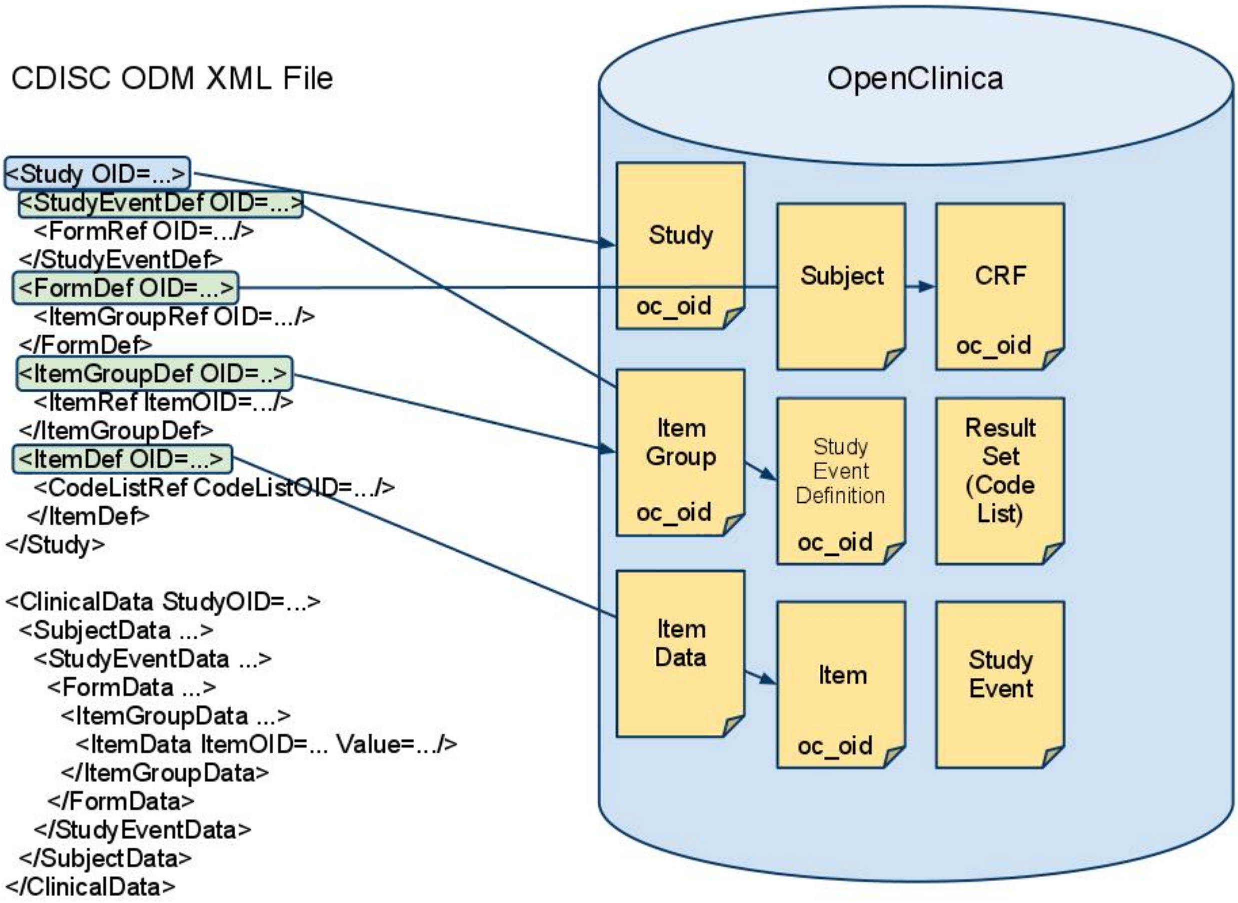
The OpenClinica data model is designed to mirror the structure and nomenclature of the CDISC ODM standard as closely as possible. Key tables in the physical schema represent studies, study subjects, CRFs, items, item data, and other objects, with the relationships between them modelled as foreign keys. The data model follows an 'Entity Attribute Value', or EAV, approach, where data values are saved as individual records in a 'long & skinny' table (item_data) with the entity name and attributes (metadata and other properties) related to the value through foreign key relationships [1].



The diagram above is a 'cheat sheet' version of the OpenClinica logical model [2], showing key tables in the schema and their relationships. Note that shorthand abbreviations are used rather than the full table names, so for example 'IG' is used in place of 'item_group'. The arrows represent foriegn keys, pointing toward the primary keys. The circled stars mark repeating objects (those with an 'ordinal' column). The lines through IGM and IFM indicate that they are ternary: each of their instances describes a 1:1 relation of a CRFV and an Item.

For a more comprehensive diagram of the current physical data model, see https://dev.openclinica.com/tools/db/relationships.html. From here you can also use the tabs at the top of the page to navigate to more detailed technical views of the database objects. Alternatively, a technical report on the OpenClinica 3.1 Database Model can be downloaded as a PDF here.

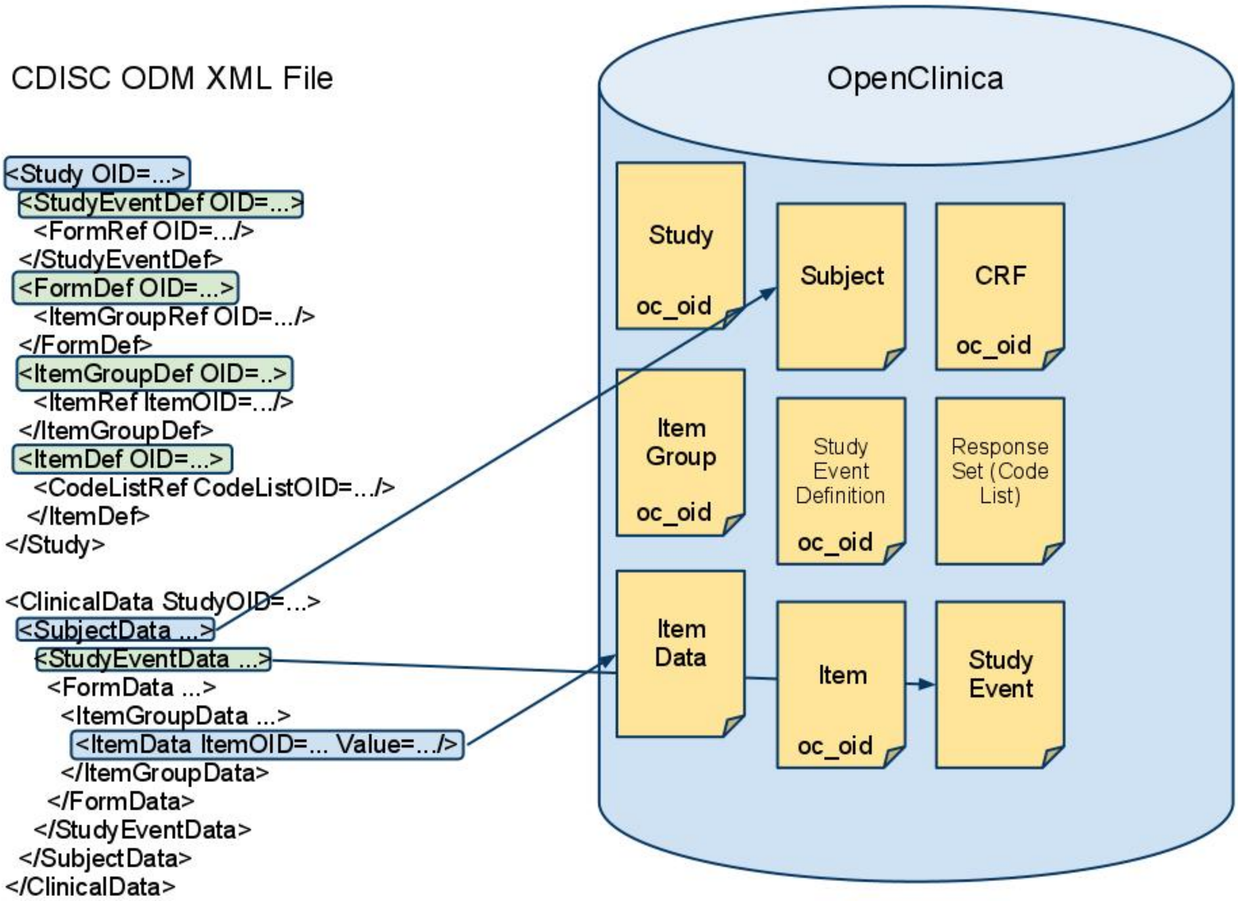
Here's a mapping[3] of how key tables in the data model map to CDISC ODM, first for study metadata:



```
<Study OID=...>
 <StudyEventDef OID=...>
  <FormRef OID=.../>
 </StudyEventDef>
 <FormDef OID=...>
  <ItemGroupRef OID=.../>
 </FormDef>
 <ItemGroupDef OID=..>
  <ItemRef ItemOID=.../>
 </ItemGroupDef>
 <ItemDef OID=...>
  <CodeListRef CodeListOID=.../>
  </ItemDef>
</Study>

<ClinicalData StudyOID=...>
 <SubjectData ...>
  <StudyEventData ...>
   <FormData ...>
    <ItemGroupData ...>
     <ItemData ItemOID=... Value=.../>
    </ItemGroupData>
   </FormData>
  </StudyEventData>
 </SubjectData>
</ClinicalData>
```

and next, for clinical data that is part of a study:



```
<Study OID=...>
 <StudyEventDef OID=...>
   <FormRef OID=.../>
 </StudyEventDef>
 <FormDef OID=...>
   <ItemGroupRef OID=.../>
 </FormDef>
 <ItemGroupDef OID=..>
   <ItemRef ItemOID=.../>
 </ItemGroupDef>
 <ItemDef OID=...>
   <CodeListRef CodeListOID=.../>
  </ItemDef>
</Study>

<ClinicalData StudyOID=...>
 <SubjectData ...>
  <StudyEventData ...>
   <FormData ...>
     <ItemGroupData ...>
      <ItemData ItemOID=... Value=.../>
     </ItemGroupData>
   </FormData>
  </StudyEventData>
 </SubjectData>
</ClinicalData>
```

In principle, the OpenClinica data model is designed to closely mirror the structure and nomenclature of CDISC ODM. In practice there are deviations, either where the logical design of OpenClinica is different from ODM or where the physical implementation of the data model is different than ODM's XML structure. While we try to avoid both types of deviations they are sometimes unavoidable. There are also some deviations that are simply legacy artifacts, without any really good reason for departing from the standard. Where these deviations do exist, we look for ways to refactor the database schema to ensure better harmonization with ODM, since we believe this makes OpenClinica more consistent, more easily understood, and easier to develop for.

[1] - For more on EAV data models, see Wikipedia and Nadkarni, et al, Organization of heterogeneous scientific data using the EAV/CR representation. Journal of the American Medical Informatics Association 1999 Nov-Dec;6(6):478-93. Abstract.

[2] - Many thanks to Marco van Zwetselaar of Kilimanjaro Clinical Research Institute for contributing this diagram on the OpenClinica users mailing list.

[3] - See the OpenClinica Blog for more detail on how the data model maps to ODM.

This page is not approved for publication.

5 Date Format Specifications

Starting with version 3.1, OpenClinica uses the ISO 8601 standard date format as its canonical representation for dates and partial dates. An OpenClinica DATE is a string with year, month, day separated by a -which is the ISO8601 standard date format and all dates are stored internally in this format.

The OpenClinica user interface supports localization of dates, with the default date format in the user interface being DD-MMM-YYYY, but others (such as MM/DD/YYYY or DD/MM/YYYY) may be used depending on the localization/translation packages installed on a given OpenClinica instance. All dates are translated by the application to the canonical format specified here prior to insert/update into the database.

In versions 3.0.x and earlier, the OpenClinica canonical date format for internal storage of CRF item data values is MM/DD/YYYY, while for other fields (such as date of birth or enrollment date) it was ISO 8601. This approach limited the consistency of dates and partial dates and is not a universally recognized format. In 3.1 the canonical ISO 8601 date format was established for internal storage of all dates, including CRF item data values for dates and partial dates.

This document helps readers understand the impact of changing the default date format in OpenClinica, as well as explains the transformation of existing date values in the item_data table that will occur when upgrading to OpenClinica version 3.1.

For CRF items, OpenClinica 3.0.x and prior stored dates and partial dates in either the U.S. date format, or as Strings dependent on the users localization settings. Because of this date values (especially partial dates) were sometimes ambiguous. It is possible for OpenClinica instances upgraded from 3.0.x and older versions that some CRF items of type DATE may have existing values that do not match the allowed string pattern, in this case they will not be converted to ISO 8601 upon upgrade. Further sections of this chapter specify the changes being made to the default date format stored in the item_data table of the OpenClinica database and also describe in detail the migration/transformation of existing date values in the item_data table that occur when upgrading from OpenClinica 3.0.x to version 3.1.

For a more formal specification of data typing for user-defined CRF Item Data values in OpenClinica version 3.1 (Amethyst), see OpenClinica Item Data Specifications.

This page is not approved for publication.

5.1 Overview with Definitions and Acronyms

The date format in the database was changed to ISO 8601 for the following reasons:

1. Allow partial dates to be saved in a standard, consistent format, rather than as a string based on the users locale
2. Eliminate confusion amongst people accessing the database directly as to what format the date is in.

3. Support future introduction of time zone and time properties in OpenClinica.

Definitions and Acronyms:

ISO 8601 is the standard date format to be supported for CRF items with a data type of DATE or PDATE.

ISO 8601 is a universally recognized standard date format. More information about the standard can be found at the following URLs:

http://www.iso.org/iso/date_and_time_format

http://en.wikipedia.org/wiki/ISO_8601

This page is not approved for publication.

5.2 DATE and PDATE Datatypes

When upgrading to OpenClinica version 3.1, the application will attempt to transform existing data in the item_data table of type DATE or PDATE to match the ISO 8061 format, as discussed below. Transformations will be applied at upgrade time by OpenClinicas Liquibase database change tracking library.

Date values stored in other parts of the database other than the item_data table will not be modified.

This page is not approved for publication.

5.2.1 DATE Datatype

Values collected in prior versions of OpenClinica should be stored in the MM/DD/YYYY format. If the data is stored in a different format, the values will not be updated and will remain as they are. A separate script will need to be created to identify and update these values.

For all dates stored in MM/DD/YYYY, a liquibase script will convert the values to conform to the ISO 8601 standard date format, YYYY-MM-DD.

5.2.2 PDATE Datatype (Complete Dates)

Values collected in prior versions of OpenClinica should be stored in the MM/DD/YYYY format. If the data is stored in a different format, the values will not be updated and will remain as they are. A separate script will need to be created to identify and update these values.

For all partial dates stored in MM/DD/YYYY, a liquibase script will convert the values to conform to the ISO 8601 standard date format, YYYY-MM-DD.

5.2.3 PDATE Datatype (Month and Year Provided)

Values collected in prior version of OpenClinica would be stored in the format specified by the users locale. The default format in the **user interface** for DATE and PDATE in OpenClinica is DD-MMM-YYYY. The default format in the user interface for DATE and PDATE in OpenClinica is DD-MMM-YYYY. Therefore, when only a month and year are provided in a PDATE field, the value in the database would be MMM-YYYY. This is the only format that will be updated automatically by a liquibase script when OpenClinica is upgraded to Amethyst. A separate script will need to be created to identify and update values in a different format.

For all partial dates stored in MMM-YYYY format, the liquibase script will convert the values YYYY-MM in order to conform to the ISO 8601 standard date format.

5.2.4 PDATE Datatype (Year Provided)

No changes will be made. The values saved in prior versions of OpenClinica (in the format YYYY) meet the ISO 8601 standard date format.

6 Item Data Specifications

OpenClinica supports user-defined CRF Item Data values according to Item Data properties defined during CRF and Study design. **OpenClinica enforces strict data typing** corresponding with the Item Datatype. Each data value must conform to the canonical format for its data type. If a value with an invalid data type is inserted or updated, it must be rejected and an exception thrown. This document describes data typing for user-defined CRF Item Data values in OpenClinica version 3.1.

This page is not approved for publication.

6.1 Item Properties

Each Item (specified in the item table) has a unique item_id and is associated with a single definition that does not change for the life of the Item* [[list exceptions to this such as ability to convert from date to pdate]]. The definition includes declaration of a dataype (item_data_type) as well as name, description and units, and Object Identifier (OID).

Each Item Data value is associated with exactly one Item definition and as such is bound to a specific datatype. The physical implementation of the OpenClinica data model (aka the database schema) stores all item data values in the item_data.value field, which is a text field of a maximum 4000 characters. Despite the fact that the DBMS treats all item data values as text, OpenClinica enforces strict data typing corresponding with the Items Datatype. Each data value must conform to the canonical format[1] for its data type. If a value with an invalid data type is inserted or updated, it must be rejected and an exception thrown.

An Item can be represented in one or more CRF Versions. Each Item has CRF version-specific properties (specified in the item_form_metadata and response_set tables) that control the allowed value domain and representation/formatting of the Item Data value in different interfaces of the system (e.g. CRF data entry input type, data import, data extract). The properties controlling these representations may change from CRF version to CRF version. The representations may be impacted by other system settings that could change over time or from user to user, such as locale settings in the case of dates, and by edit check constraints such as simple validation checks and OpenClinica Rules.

---

[1] A canonical form means that values of a particular type of resource can be described or represented in multiple ways, and one of those ways is chosen as the favored canonical form. (That form is canonized, like books that made it into the bible, and the other forms are not.) A classic example of a canonical form is paths in a hierarchical file system, where a single file can be referenced in a number of ways:

```
myFile.txt                                # in current working dir

../conf/myFile.txt                        # relative to the CWD

/apps/tomcat/conf/myFile.txt              # absolute path using symbolic links

/u1/local/apps/tomcat-5.5.1/conf/myFile.txt  # absolute path with no symlinks
```

The classic definition of the canonical representation of that file would be the last path. With local or relative paths you cannot globally identify the resource without contextual information. With absolute paths you can identify the resource, but cannot tell if two paths refer to the same entity. With two or more paths converted to their canonical forms, you can do all the above, plus determine if two resources are the same or not, if that is important to your application (solve the aliasing problem).

Note that the canonical form of a resource is not a quality of that particular form itself; there can be multiple possible canonical forms for a given type like file paths (say, lexicographically first of all possible absolute paths). One form is just selected as the canonical form for a particular application reason, or maybe arbitrarily so that everyone speaks the same language. (taken from http://stackoverflow.com/questions/280107/what-does-the-term-canonical-form-or-canonical-representation-in-java-mean)

This page is not approved for publication.

6.2 Canonical Datatypes

The canonical formats are described below. They are derived from CDISC ODM 1.3 , which itself are derived from XML Schema datatypes. This is the basis for our specification, however only a subset of ODM formats are supported in OpenClinica.

OpenClinica Datatype Name	**ODM Format Name**	**Schema Datatype**	**Canonical Format (Allowed String Pattern)**	**Notes**
ST	text	xs:string	any sequence of characters	
INT	integer	xs:integer	-?digit+	
REAL	float	xs:decimal	-?digit+(.digit+)?	
DATE	date	xs:date	YYYY-MM-DD	An OpenClinica DATE is a string with year, month, day separated by a -which is the ISO8601 standard date format. [1] The OpenClinica user interface supports localization of dates, with the default date format in the user interface being DD-MMM-YYYY.[2] All dates must be translated by the application to the canonical format specified here prior to insert into the database. It is possible for OpenClinica instances upgraded from 3.0.x and older versions that some items of type DATE may have existing values that do not match the allowed string pattern. See OpenClinica Troika Item Data Specifications.doc and OpenClinica Date Format Change Specifications.doc for reference.
PDATE	partialDate	xs:date	YYYY-[[MM-]DD]	An OpenClinica partial date (PDATE) is a string with a year, optional month, and optional day separated by a -. No timezone is specified. The OpenClinica user interface supports localization of partial dates, with the default partial date format in the user interface being DD-MMM-YYYY. All partial dates must be translated by the application to the canonical format specified here prior to insert into the database. It is possible for OpenClinica instances upgraded from 3.0.x and older versions that some items of type PDATE may have existing values that do not match the allowed string pattern. See OpenClinica Troika Item Data Specifications.doc and OpenClinica Date Format Change Specifications.doc for reference.
FILE	URI	xs:anyURI	scheme ":" hier-part ["?" query] ["#" fragment]	See http://www.w3.org/Addressing/URL/uri-spec.txt for URI specification.

[1] http://www.iso.org/iso/date_and_time_format

[2] https://www.openclinica.org/dokuwiki/doku.php?id=developerwiki:i18n

This page is not approved for publication.

7 Internationalization and Localization

Overview

OpenClinica supports internationalization and localization. This allows users and developers to adapt the software for a specific region or language by adding locale-specific components and translating text. A significant number of changes related to internationalization and localization have been implemented in OpenClinica 3.1.3. This chapter applies to OpenClinica 3.1.3 and later versions.

By default, OpenClinica is configured with English language support files in CATALINA_HOME/webapps/OpenClinica/WEB-INF/classes/org/akaza/openclinica/i18n. Other languages/locales may be added as described in this guide.

This page is not approved for publication.

7.1 Configure OpenClinica for Internationalization

Configure the Apache Tomcat Connector

Please change the default connector configuration in tomcat/conf/server.xml to ensure proper processing of UTF-8 characters.

The default setting is as follows:

```
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1"
redirectPort="8443"/>
```

Change this to the following:

```
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1"
redirectPort="8443" URIEncoding="UTF-8"/>
```

Without this modification, UTF-8 characters will not be processed correctly by tables filters and the Calendar Widget.

Note that, if you are using SSL, you must apply this modification to the line that starts with:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"...
```

Ensure Operating System Requirements are Met

OpenClinica has been tested on operating systems configured for the English language only. Please install OpenClinica on a supported OS that is configured for the English language. This will help ensure proper behavior of date formats and locale-specific text, as described in the guide below.

This page is not approved for publication.

7.2 Install Desired Translations

Existing Translations

The OpenClinica community provides translations for multiple locales. Each translation is comprised of a series of properties files with the translated text for the language in question, as well as other localized properties like date formats. Existing translations are accessible at https://www.transifex.com/projects/p/openclinica/.

Only English properties files are installed in OpenClinica by default. The additional translations have been contributed by community members. These translations are not all fully tested and may not have all the strings translated. To determine if the localized files you are interested in are complete, please compare them to the files in CATALINA_HOME/webapps/OpenClinica/Web-INF/classes/org/akaza/openclinica/i18n.

OpenClinica currently has 9 properties files:

- admin.properties
- audit_events.properties
- exceptions.properties
- format.properties
- notes.properties
- page_messages.properties
- terms.properties
- words.properties
- workflow.properties

Starting with version 3.1.3, there is an additional properties file, licensing.properties file. Please do not translate this file.

First, you need to duplicate each of 9 properties files and renaming them by appending the locale string. For example, in case of French, this is how the file list should look like:

- admin.properties
- admin_fr.properties
- audit_events.properties
- audit_events_fr.properties

- exceptions.properties
- exceptions_fr.properties
- format.properties
- format_fr.properties
- notes.properties
- notes_fr.properties
- page_messages.properties
- page_messages_fr.properties
- terms.properties
- terms_fr.properties
- words.properties
- words_fr.properties
- workflow.properties
- workflow_fr.properties
- licensing.properties (3.1.3 or above - do not translate this file)

Pre-3.1.3, please do not translate License infomation under 'footer.license' in words.properties file where it notes "# DO NOT CHANGE ANY OF THE TEXT BELOW".

Installing a Translation

Once you are satisfied with your translation, place the localized properties files in your i18n folder (CATALINA_HOME/webapps/OpenClinica/Web-INF/classes/org/akaza/openclinica/i18n) and restart Tomcat. When a user changes the preferred language in the user's web browser to match the locale in the format_xx.properties file, the user will see the translation into that language.

Please note: even though you have no intention of using the original English strings and wish to use only translated GUI, OpenClinica may break if you modify the original properties files. Instead, you need to create localized versions with '_xx' i18n marker as described above.

Which Language will be displayed in the User Interface (UI)?

OpenClinica determines the locale/language to be displayed based on two conditions: user language preferences and the availability of nine OpenClinica properties files in the desired language.

English (en) language is the default language.

The user's browser language setting determines the language priority for OpenClinica. For locale lookup, OpenClinica adhere to W3C rules which are based on RFC 4647. And also this locale must have all of nine OpenClinica properties files available in the i18n folder.

Browser Language Preferences	Installed Localized Properties files	Displayed Language
1. English [en-us] 2. French [fr] 3. German [de] 4. English [en]	English (default) _fr.properties (French) _en-gb.properties (United Kingdom)	English [en] Even though 'en-us' is not installed, you "progressively truncate" the user's top priority language range until you get a match, which would be en = en.

1. German [de] 2. French [fr] 3. English [en]	English (default) _fr.properties (French) _en-gb.properties (United Kingdom)	French [fr] German [de] is not installed so French [fr] is taking effect that is the 2nd on the list.
1. Chinese [zh]	English (default) _fr.properties (French) _en-gb.properties (United Kingdom)	English [en] Chinese [zh] is not installed so the default, English is taking effect.
1. United Kingdom [en-gb]	English (default) _fr.properties (French) _en-gb.properties (United Kingdom)	United Kingdom [en-gb]

Approved for publication by Cal Collins. Signed on 2014-03-26 4:31PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

7.3 Translate OpenClinica into New Languages or Maintain Existing Translations

Completing Translations

The default (English) files in your i18n folder should be considered a complete list of all the strings that should be translated. You can find existing translated files at the Transifex OpenClinica Project site. The Transifex site includes tools to allow you to manage and conduct translations. Sign up to be a translator for your target language. Once approved you can upload translations or build them online using the Transifex tools.

Visit the i18n forum to share information on what you're translating and ask others for help.

If you find your target language does not match with your OpenClinica version, run a diff to identify the changes so you do not need to start from scratch. Here are some free diff tools. Diff files for 3.1.3 are located here.

In order to complete a translation, you must translate the text to the right of the '=' sign into your language.

Default	CRF_attributes = CRF Attributes
es	CRF_attributes = Atributos del CRD

Then save the file with your language extension. For example, if you are translating the strings into Spanish, you would add _es to the names of each of the 9 default files and save them as additional files. For example, words.properties would become words_es.properties. Please make sure to keep the original default files in the i18n folder along with the translated files.

Do not change the contents of the default files. Note also there are some strings in the file which should never be translated. These are identified with an appropriate comment in each of the properties files.

Asian Languages

Asian language translations are managed through GitHub rather than Transifex. Transifex does not seem to handle Asian language well. For example, you won't be able to see human readable characters, and downloaded files will add an unnecessary back slash because it treats file encoding as Latin-1 while it should be UTF-8.

To contribute/update Asian language translations, please submit a pull request to the GitHub repository.

Character Encoding

If you are translating into a language that uses characters not found in Western Europe (specifically, outside the ISO-Latin-1 character set) you will have to save those characters as Unicode escape sequences. For example "" would be "u20AC". To convert your translated text to Unicode escape sequences (Hex value), enter them into a converter such as brahan_converter and copy the strings. Be sure to save your original unconverted text in a comment, because the escape sequences are not editable.

Contributing a Translation

If you complete or improve one of the existing translations or create a new one, please consider sharing this back with the community! It's easy, sign up the Transifex OpenClinica Project site and post your files there.

Localization of Date Formats and Calendar Widget

OpenClinica supports displaying date and time in localized formats via the date format settings in the format.properties file. By default dates are formatted in DD-MMM-YYYY in English, where 'MMM' is an alphabetical abbreviation for the month.

Other date formats may be used by changing the locale-specific date format, with the alphabetical abbreviation or using numeric formats. For instance, you may want to use DD/MM/YYYY in your format_en-gb.properties for the Great Britain locale (note: this particular format is provided as an example only and is NOT recommend due to its ambiguity with common American date formats).

OpenClinica dates are never localized in ODM XML files, web services calls, or other programmatic data formats. In these cases OpenClinica uses the universal ISO 8601 format to represent all dates, date/times, and partial dates. Regardless of the user's locale, OpenClinica also saves all dates in its database in the ISO 8601 format. See OpenClinica Date Format Specifications for more detail.

When localizing OpenClinica, you must ensure that date formats specified in the format.properties file for a given locale are also supported by the OpenClinica javascript calendar widget. This is especially important when using the 'MMM' alphabetical month name as part of the localized date format. Please refer to examples below.

By default, OpenClinica provides a calendar widget in English (en). A file named calendar-en.js is located at ${Catalina.home}webappsyour-openclinica-instanceincludesnew_callang.

OpenClinica supports localization of the calendar widget into other languages. Let's use French (fr) as example. We'll assume French properties files are installed with format_fr.properties having a date format of DD-MMM-YYYY.

In this French scenario, if French is the chosen web browser locale, but there is no calendar-fr.js available, the default English calendar widget will be used. Since server-side date validation expects a French-formatted date, the English date string provided by the calendar widget may fail server-side date format validation. The user would be required to manually change the English date to a French date. For example, the English calendar widget will populate the field with 01-May-2012 but the DD-MMM-YYYY expected for the French locale would be 01-Mai-2012. 01-May-2012 has to be changed to 01-Mai-2012 to pass the validation.

To ensure the calendar widget is compatible with the localized date format, create a calendar-fr.js file which contains the translated contents from calendar-en.js file (follow the instructions in the .js file). Please refer to the examples below for more detail. Then, place calendar-fr.js file into ${Catalina.home}webappsyour-openclinica-instanceincludesnew_callang and restart Tomcat.

Based on the locale 'fr' being used, server-side validation of month names/abbreviations are automatically expected in French by the OpenClinica. The calendar widget translation MUST use the same month names/abbreviations as those expected by Java. Java 6 currently supports this list of languages for the java.text package, and the translations are provided by projects like CLDR from the Unicode Consortium. To ensure your month names/abbreviations exactly match those expected by the Java application, you can download the CLDR translations from here and find the list of months for your language in core/common/main.xml. Please note that Asian languages do not seem to follow this practice. For example, the browser default for the MMM abbreviation for Japanese and Korean are a number without leading zero when less than the month of October. This indicates both languages do not use strings for an abbreviated month.

Examples of Localized Date Formats

To ensure localized date formats work properly, you will need to modify the following items in the format.properties file located under tomcat/webapps/OpenClinica/WEB-INF/classes/org/akaza/openclinica/:

- date_format =
  - is used to build the error message
- date_format_string =
  - is used to display format by UI level
- date_time_format_string =
  - is used to format where time is applicable such as Scheduling Event
  - Note that OpenClinica allows only 24 hour format, and AM/PM markers will not be used
- date_format_calendar=
  - is used to format the date string that is sent from calendar widget when applicable

date_format, date_format_string, date_time_format_string

- Date
  - dd: 2 digits, i.e., 01, 02, 03...
  - d: digit(s) without leading zero, i.e., 1, 2, 3... 10, 11...
- Month
  - MM: month in digits. i.e., 01, 02, 03...
  - MMM: locale default abbreviated month names

        - Mar (English)
        - mars (French)
        - Mr (Deutsch)
        - 3 (Japanese, Korean)
        - ??(Chinese)

          .

          .
    - MMMMM: locale default full month names
        - December (English)
        - dcembre (French)

          .

          .
- Year
    - yyyy: Year with century
    - yy: Year without century

date_format Examples

Upper case M is used for month, while lower case is used for day and year for java class. However, upper case for all three are used to build error messages.

US English (Default): 01-Jan-2012

- date_format = DD-MMM-YYYY
- date_format_string = dd-MMM-yyyy
- date_time_format_string = dd-MMM-yyyy HH:mm:ss

Great Britain English and Spanish: 30/10/2012

- date_format = DD/MM/YYYY
- date_format_string = dd/MM/yyyy
- date_time_format_string = dd/MM/yyyy HH:mm:ss

French: 01-fvr.-2012

- date_format = DD-MMM-YYYY
- date_format_string = dd-MMM-yyyy
- date_time_format_string = dd-MMM-yyyy HH:mm:ss

Deutsch: 20.10.2012

- date_format = MM.DD.YYYY
- date_format_string = dd.MM.yyyy
- date_time_format_string = dd.MM.yyyy HH:mm:ss

Japanese: 2012/1/1

- date_format = YYYY/MMM/D
- date_format_string = yyyy/MMM/d
- date_time_format_string = yyyy/MMM/d HH:mm:ss

Japanese and Chinese: 2012?1?1? (Unicode Hex value needs to be escaped with single quote)

- date_format = YYYY'u5e74'MMM'u6708'd'u65e5'
- date_format_string = yyyy'u5e74'MMM'u6708'd'u65e5'
- date_time_format_string = yyyy'u5e74'MMM'u6708'd'u65e5' HH:mm:ss

Chinese: 2012-01-01

- date_format = YYYY-MM-DD
- date_format_string = yyyy-MM-dd
- date_time_format_string = yyyy-MM-dd HH:mm:ss

Chinese: 2012-??-1

- date_format = YYYY-MMM-D
- date_format_string = yyyy-MMM-d
- date_time_format_string = yyyy-MMM-d HH:mm:ss

Korean: 2012.1.1

- date_format = YYYY.MMM.D
- date_format_string = yyyy.MMM.d
- date_time_format_string = yyyy.MMM.d HH:mm:ss

Korean: 2012? 1? 1? (Unicode Hex value needs to be escaped with single quote)

- date_format = YYYY'uB144'MMM'uC6D4'd'uC77C'
- date_format_string = yyyy'uB144'MMM'uC6D4'd'uC77C'
- date_time_format_string = yyyy'uB144'MMM'uC6D4'd'uC77C' HH:mm:ss

CJK: 2012.01.01

- date_format = YYYY.MM.DD
- date_format_string = yyyy.MM.dd
- date_time_format_string = yyyy.MM.dd HH:mm:ss

date_format_calendar

List of parameters:

- %b: Prints the abbreviated **month name**. Range: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. Example Output: Jan
- %B: Prints the full month name. Range: January, February, March, April, May, June, July, August, September, October, November, December. Example Output: January
- %d: Prints the **day** of the month with the leading zero. Range: 01 to 31. Example Output: 01
- %e: Prints the **day** of the month without the leading zero: Range: 1 to 31. Example Output: 1
- %m: Prints the numeric **month** with the leading zero. Range: 01 to 12. Example Output: 01
- %o: Prints the numeric representation of the **month** without the leading zero. Range: 1 - 12. Example Output: 1
  (Not supported OpenClinica version 3.1.2 or earlier)
- %y: Prints the **year** without the century with the leading zero. Range: 00 - 99. Example: 01
- %Y: Prints the **year** with the century. Example: 2001

date_format_calendar Examples

US English (Default): 01-Jan-2012

- date_format_ = %d-%b-%Y

Japanese and Chinese: 2012?1?1? (Hex escape not needed)

- date_format_ = %Yu5e74%ou6708%eu65e5
  (%o not supported OpenClinica 3.1.2 or earlier)

CJK: 2012.01.01

- date_format_ = %Y.%m-%d

Calendar Widgets

- When you translate Calendar Widgets found at ${Catalina.home}webappsyour-openclinica-instanceincludesnew_callang, you do not need to convert to Unicode escape sequence (Hex value). Translated strings pass without corruption as long as your tomcat connector is configured properly as explained before.

  Example in French:

  // short month names
  Calendar._SMN = new Array
  ("janv.",
  "fvr.",
  "mars",
  "avr.",
  "mai",
  "juin",
  "juil.",
  "aot",
  "sept.",
  "oct.",
  "nov.",
  "dc.");
- Save As the calendar-en.js file with appropriate locale marker, i.e., calendar-fr.js. This file name has to be called from the format-{lang}.properties file under "jscalendar_language_file =".

Localization of Image files

As of OpenClinica 3.1, icons and other image files do not contain any text and do not have to be localized. However it is possible to create localized images in the desired language. Under the folder ${Catalina.home}webappsyour-openclinica-instanceimages, makes a new folder with the name as desired locale code, e.g. "es" for Spanish (es). In this new folder collects all translated image files.

Notes for Developers

1. When you need Locale, please call the method org.akaza.openclinica.i18n.core.LocaleResolver.getLocale (HttpServletRequest request).
org.akaza.openclinica.i18n.core.LocaleResolver class locates in Core module. Its resolveLocale(HttpServletRequest request) method implements OpenClinica business logic of how to determine locale (please refer to Which Language to Be Displayed on User Interface (UI)).

2. If you need Date/Time format in UI, please call the methods for pattern string or SimpleDateFormat in the class org.akaza.openclinica.i18n.util.I18nFormatUtil.
If you need Date/Time format in database, please call methods for date format pattern string in the class org.akaza.openclinica.bean.core.ApplicationConstants.

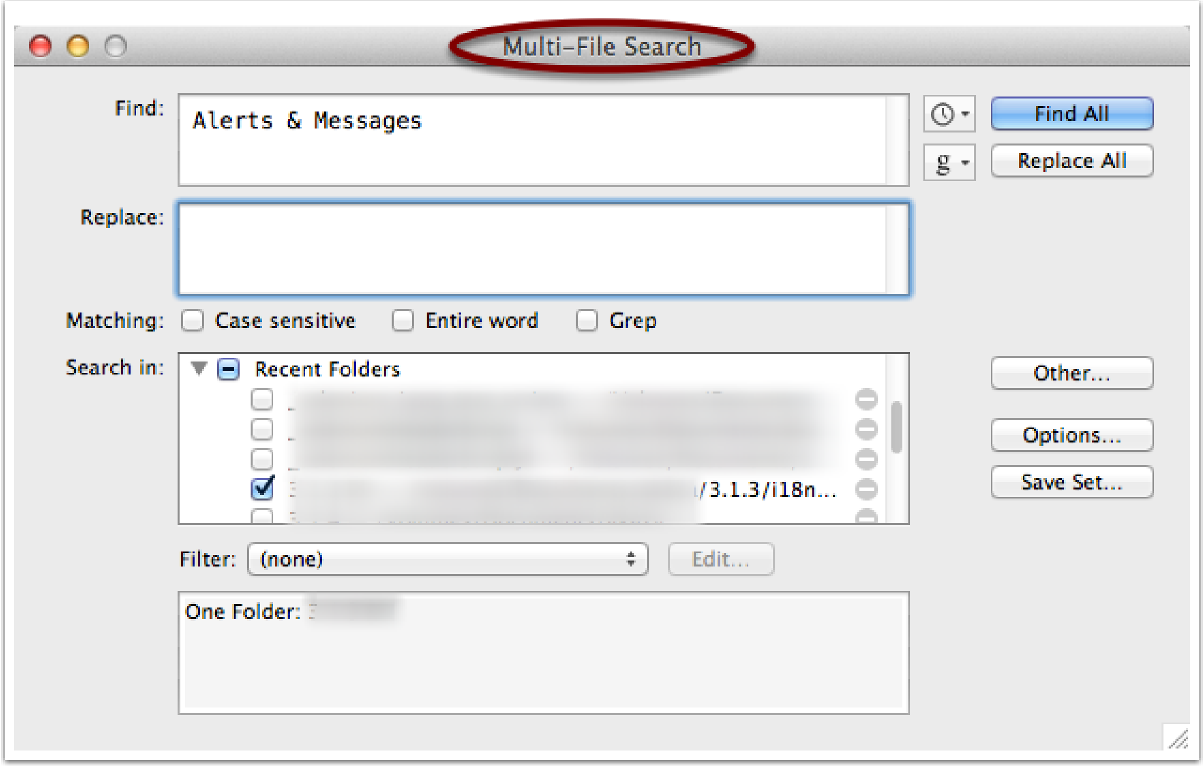
Approved for publication by Cal Collins. Signed on 2015-02-06 8:42AM

Not valid unless obtained from the OpenClinica document management system on the day of use.

7.4 Recommended steps to translate strings

OpenClinica properties files may contain some legacy strings which should have been removed. For this reason, translating page by page on the front end, instead of translating string by string in the properties files, is recommended.

1. Create a new local directory for your work space
2. Copy all the original properties files, except licensing.properties, into the new directory
3. Rename each of the 9 files so the language extension is appended as explained in section 11.2, e.g., words.properties ? words_ja.properties (make sure you are not modifying the original files)
4. Open the page you wish to start translating, e.g., /OpenClinica/MainMenu
5. Chose a word and/or sentence you wish to translate, e.g., "Alerts & Messages"
6. Search the string(s) chosen in Step 2 throughout the work directory you created in the Step 1
    - Some text editors such as TextWrangler (Mac)/NotePad++ (Wind) makes this step easy



Find: Alerts & Messages

Replace:

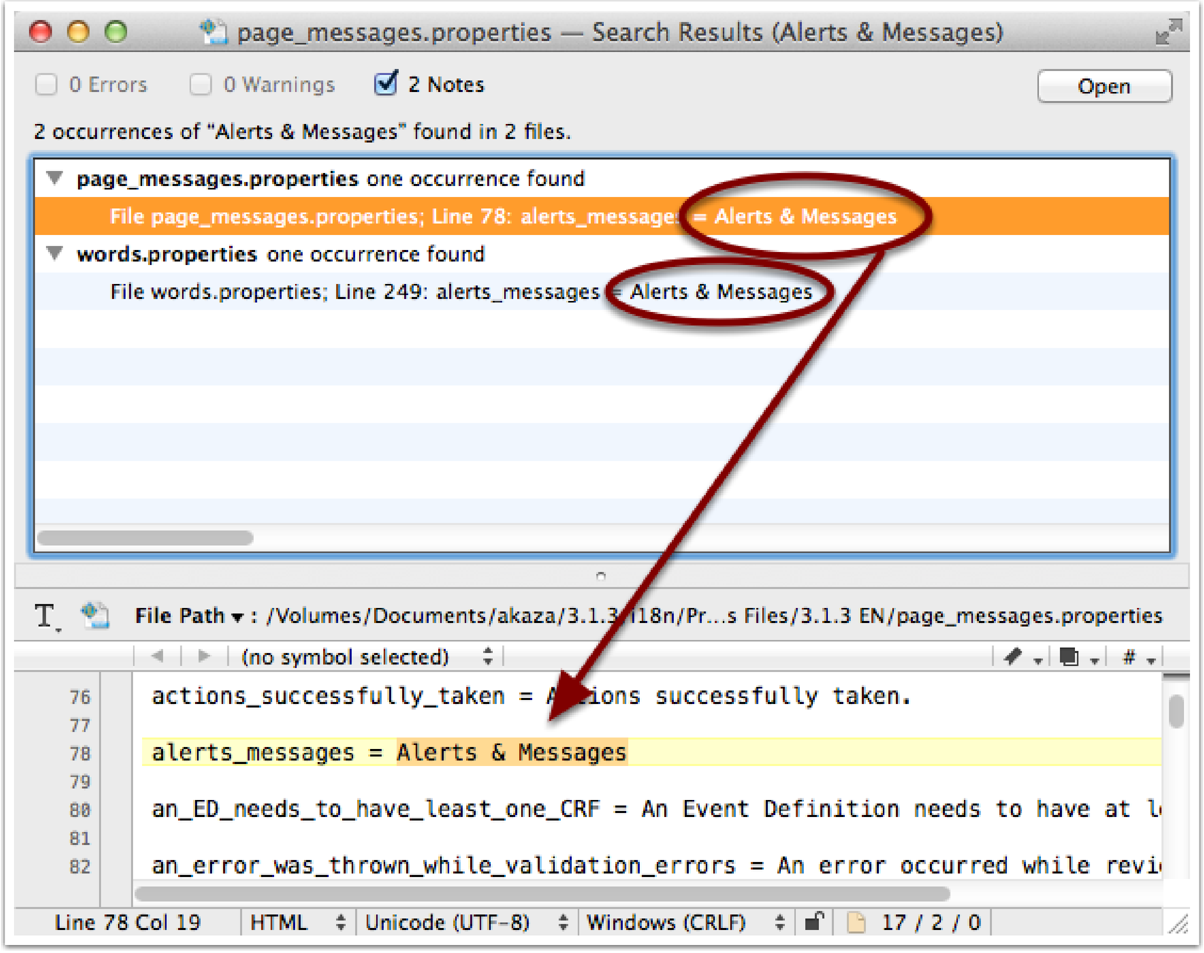
Matching: □ Case sensitive □ Entire word □ Grep

Search in: Recent Folders

□ [redacted]
□ [redacted]
□ [redacted]
☑ [redacted]/3.1.3/i18n…
□ [redacted]

Filter: (none)

One Folder: [redacted]



page_messages.properties — Search Results (Alerts & Messages)
Open

□ 0 Errors □ 0 Warnings ☑ 2 Notes

2 occurrences of "Alerts & Messages" found in 2 files.

- **page_messages.properties** one occurrence found
  - File page_messages.properties; Line 78: alerts_messages = Alerts & Messages
- **words.properties** one occurrence found
  - File words.properties; Line 249: alerts_messages = Alerts & Messages

File Path : /Volumes/Documents/akaza/3.1.3/i18n/Pr…s Files/3.1.3 EN/page_messages.properties

(no symbol selected)

```
actions_successfully_taken = Actions successfully taken.

alerts_messages = Alerts & Messages

an_ED_needs_to_have_least_one_CRF = An Event Definition needs to have at le

an_error_was_thrown_while_validation_errors = An error occurred while revie
```

Line 78 Col 19 | HTML | Unicode (UTF-8) | Windows (CRLF) | 17 / 2 / 0

Once you found the string you wish to translate

1. Translate the word(s)

```
alerts_messages = ????????
```

2. Duplicate the line

   ```
   alerts_messages = ????????
   alerts_messages = ????????
   ```

3. Comment out the first line

   ```
   #alerts_messages = ????????
   alerts_messages = ????????
   ```

4. Now convert the translated string into Unicode value using a tool such as this.

   ```
   #alerts_messages = ????????
   alerts_messages = u8b66u544au3068u30e1u30c3u30bbu30fcu30b8
   ```

It's important to leave the readable string in the commented line, in case you need to modify/update the string later.

This page is not approved for publication.

7.5 OpenClinica Data Extract File Format

OpenClinica Data Extract File Format

When data contain non-ASCII characters, you may encounter character viewing issues on extracted files. Here is the rundown:

- CDISC ODM XML 1.3 Full with OpenClinica extensions
  - Converts into Decimal values with Character Entity marker (&#), i.e., &#65297;&#65298;&#65299;&#65300;&#65301;
- CDISC ODM XML 1.3 Clinical Data with OpenClinica extensions
  - Converts into Decimal values with Character Entity marker
- CDISC ODM XML 1.3 Clinical Data
  - Displays as expected - see note below if Windows
- CDISC ODM XML 1.2 Clinical Data with OpenClinica extensions
  - Displays as expected - see note below if Windows
- CDISC ODM XML 1.2 Clinical Data

  - Displays as expected - see notes below if Windows
- View as HTML
  - Displays as expected - see notes below
- Excel Spreadsheet
  - Tab delimited text file. This can be displayed with workaround
- Tab-delimited Text
  - Displays as expected - see notes below if Windows
  - If opening with Microsoft Excel, see this workaround
- SPSS data and syntax
  - Tab delimited text file. This can be displayed - see notes below if Windows
- Datamart in a downloadable format
  - Currently not fully compatible - see notes below
- Datamart
  - Currently not fully compatible - see notes below
- Discrepancy Notes CSV Export
  - Converts into Hex values with Unicode Escape marker (u), i.e., uff11uff12uff13uff14uff15
- Discrepancy Notes PDF Export
  - Currently not compatible 17230

File Encoding Issue

If you encounter issues viewing UTF-8 characters where they are expected to display correctly, you may need to specify the encoding.

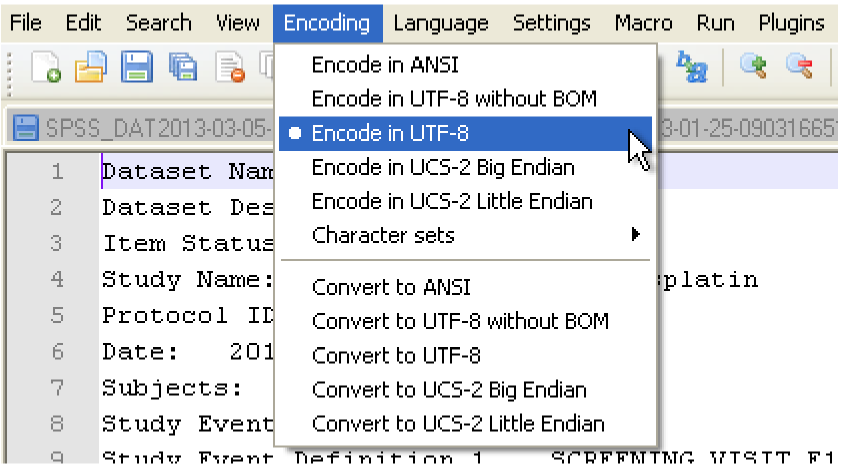
For example, if you open OpenClinica exported **HTML** file and see garbled corrupted characters, you need to set your browser encoding to UTF-8 to view those characters correctly.

Opening files on Windows machines

The native file encoding on Linux, Unix, Mac OS and popular databases such as SQL and Oracle is UTF-8 (Big Endian); however, Microsoft Windows' native file encoding is UTF-16LE (Little Endian). Depending on your text editor, this can become an issue because Java runs on OpenClinica server are UTF-8, not UTF-16LE.

If you open a file that contains non-ASCII characters and the file itself does not declare the encoding at the file binary header, the OS will try to determine with which encoding the file is written. Non-Windows OSes have an UTF-8 character map library in its OS level to determine the character map when opening the file, while Windows does not.

If you see garbled UTF-8 characters in ODM 1.3, ODM 1.2-Ext, ODM 1.2, Tab-delimited Text, and SPSS .dat files, you may need to **SaveAs** with the file encoding specified to UTF-8. Popular text editor such as Notepad++ (Win) and TextWrangler (Mac) will enforce encoding declaration at the file binary header level.



```
Dataset Nam
Dataset Des
Item Status
Study Name:                   platin
Protocol ID
Date:   201
Subjects:
Study Event
Study Event Definition 1      SCREENING VISIT F1
```

BOM Option

BOM (Byte Order Mark) can be critical on Windows environment. Unicode on Linux, Unix, Mac OS and popular databases such as SQL and Oracle is UTF-8, which is Big Endian byte order by default; Windows chooses UTF-16 Little Endian byte order.

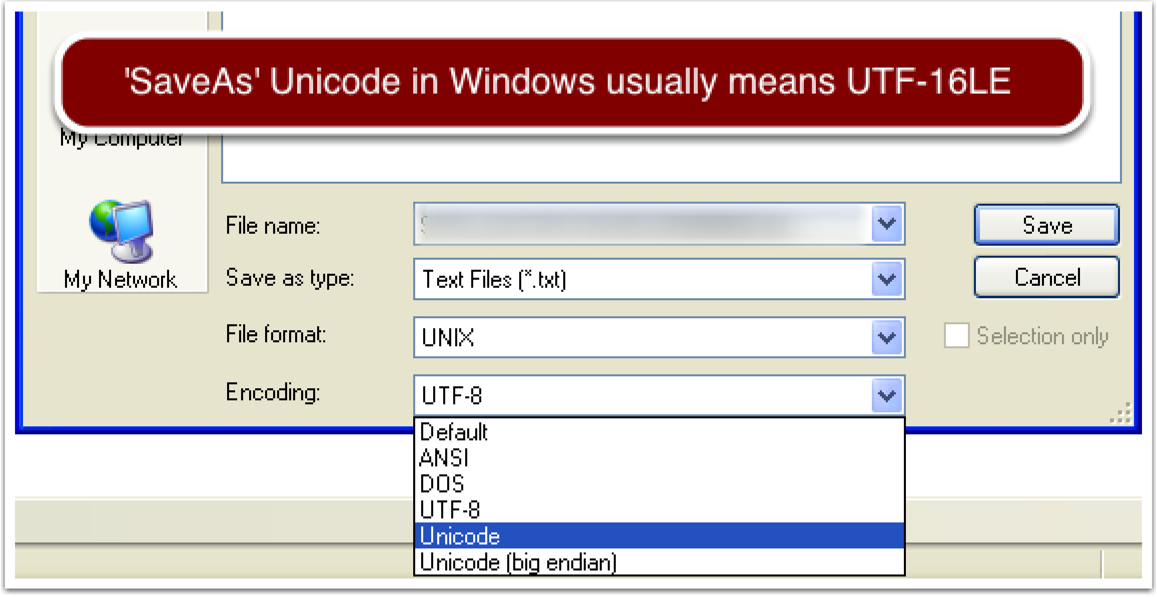
Your text editor should give you the option to SaveAs 'UTF-8 with BOM' and 'UTF-8 without BOM'. In our experience, this is somewhat hit or miss. Logically, it should work better with BOM but sometimes it seems to confuse Windows. You may need to experiment with the option of 'with' and 'without' BOM to find which option works on your Windows environment.

Excel issue

Not only does Excel not offer an encoding option, it doesn't seems to understand UTF-8 encoding. Even on a Mac OS platform, where UTF-8 is the native encoding, Excel cannot display non-ASCII characters unless file is encoded in UTF-16LE.

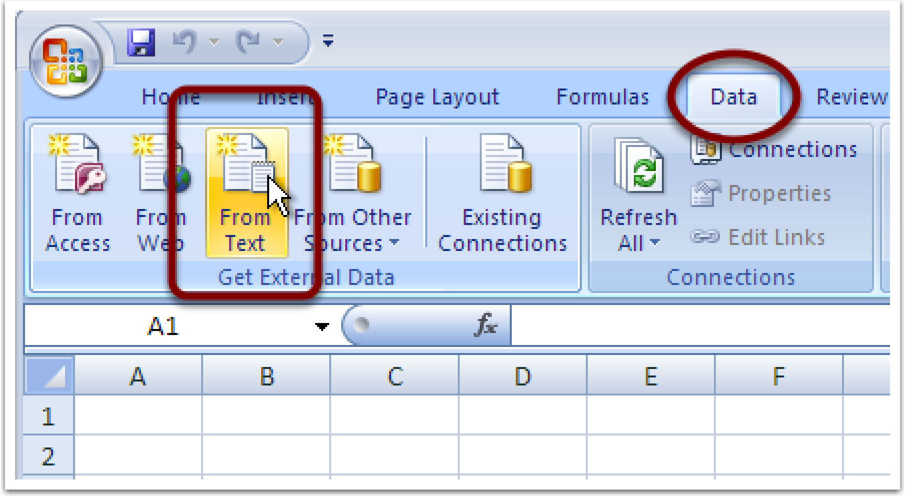
Workaround 1

1. Open the .xls file with a text editor of your choice
2. SaveAs UTF-16LE encoding


'SaveAs' Unicode in Windows usually means UTF-16LE

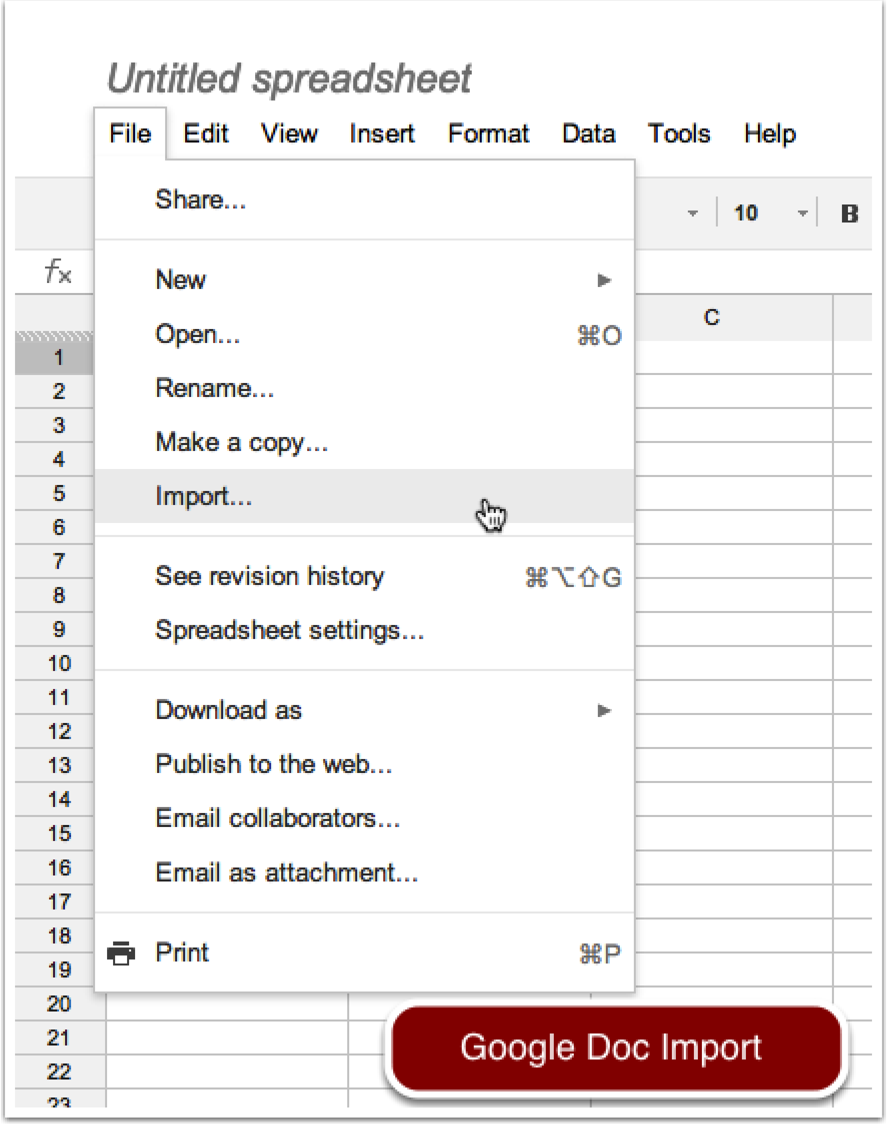
File name:
Save as type: Text Files (*.txt)
File format: UNIX
Encoding: UTF-8

3. Open with Excel application
   - If .tsv instead of .xls, import the data into a new Excel file

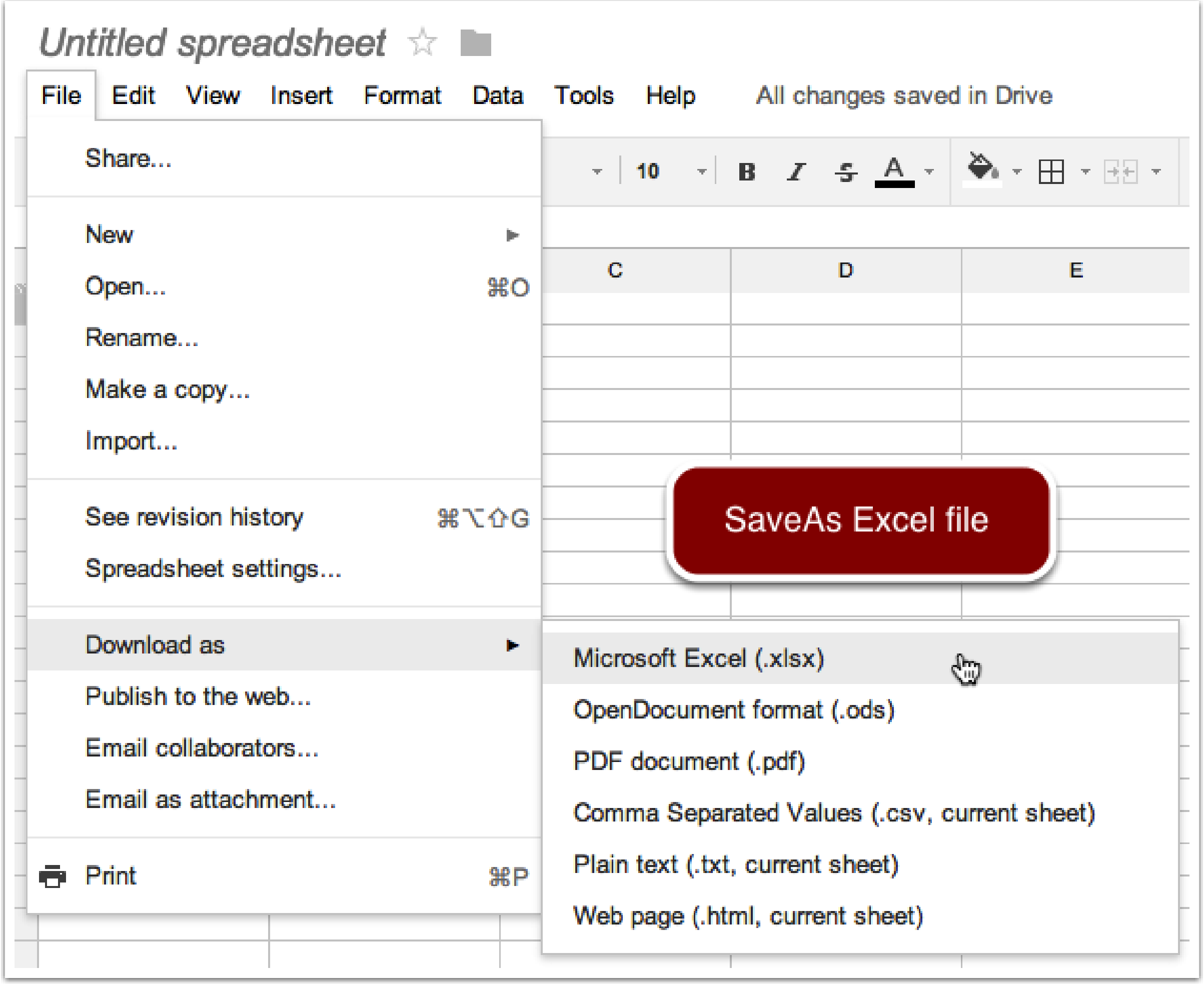


Workaround 2

1. Import to Google Doc Spreadsheet
    - Google Doc is native UTF-8 and correctly identifies UTF-8 encoded files



2. Select 'Download As' ? Microsoft Excel
    - Google Doc successfully embeds the encoding declaration binary header when the file is re-saved to your local directory



3. Open with Excel application

Troubleshooting

- If you see white boxes, e.g., ??????????
  - This is an indication of font problem. Your OS may not have Unicode mapped fonts.
  - This is a typical issue with Windows XP
  - You need to obtain Unicode font and install on your Windows
    - Arial Unicode Font is Microsoft default Unicode font
- If you see one or more white boxes in a recognizable i18n string, e.g., ????
  - This usually means incomplete Unicode font is assigned, often seen when the properly encoded file is opened with Excel
  - Select All and reassign known working Unicode font

Data Mart Issues

Currently, OpenClinica Data Mart function converts non-ASCII characters used for Table Names and Column Names into underscore character to avoid possible database issues.

It was designed this way for occasional non-ASCII character appearances among ASCII characters in a string, such as European word with accented characters. It was never meant for 100% non-ASCII string such as Asian languages.

If 100% non-ASCII string, all the entries become a series of underscore characters, which ends up with duplicated Table/Column names. We are hoping to resolve this issue as soon as possible.

On the other hand, data will not be affected by this. You can have Unicode characters in data string, and Data Mart will work as expected.

In summary:

- Any string that becomes a Table Name needs to be ASCII such as CRF name and Item Group name.
- Any string that becomes a Column names needs to be ASCII such as Multi-select Response Text and Item Name.
- Study Name becomes a series of underscore characters if non-ASCII

If a series of underscore characters become duplicated entries, Data Mart in a Downloadable Format output file will error when importing to Postgres. On the other hand, Data Mart extract operation silently stops during the operation without error message, leaving the data output incomplete (17249).

Data Mart in a Downloadable Format on Windows

Even if your .sql output file from Data Mart in a Downloadable Format does not contain any offensive underscore characters, remember Windows may require you to modify the file encoding as discussed above. This is not an issue when Postgres/pgAdmin III is running on Mac OS and/or Linux OS.

This page is not approved for publication.

8 System Administration

This chapter contains information on general system administration of the OpenClinica software stack.

This page is not approved for publication.

8.1 Configure Postgres to Allow Additional Connections

To allow access to a Postgres DB server from any location other than "localhost," some modifications have to be made to the Postgres server configuration files.

Postgres 8.4 comes already configured to serve the database on all IP's. Postgres 8.2 however only serves the database on localhost by default and not on all of the IP's (note: There is no version of OpenClinica that runs on Postgres 8.3). Regardless of whether you are using Postgres 8.2 or 8.4, to open up additional connections you will need to add an entry in the Posgres conf file so the Postgres server can allow the connection.

The following lists instructions to both (a) serve your database on all IP's, and (b) modify Postgres' configuration to allow additional connection.

Postgres configuration files location:

- For Windows - C:Program FilesPostgreSQL8.4data
  - Note "Program Files" may be "Program Files x86" for your installation.
- For Linux - /opt/PostgreSQL/8.4/data

The two main configuration files which are both located in the data directories referenced above are "postgresql.conf" and "pg_hba.conf."

To get Postgres 8.2 to listen on all IP's:

- Edit the postgresql.conf file and set the listen_address variable to "listen_addresses = '*'"
- Save postgresql.conf
- Restart Postgres (Ensure any tomcat instances pointed to the databases on the server are stopped.)
- Postgres 8.2 is now listening on all IP addresses on the server (Reference Material)

Enable Postgres to accept additional connections:

- Edit the pg_hba.conf file to allow connections (Reference Material).
  - The pg_hba.conf file is like an ACL (Access Control List) for the database. It only allows connections from servers that match the ACL list.
  - The file can restrict access based on IP, username and database and any combination of the above.
  - The file can also setup authentication requirements for that connection.
  - An example entry that will allow all IP's and all usernames to connect to all databases is below.
    - host all all 0.0.0.0/0 md5
  - Most of the time the error message you get will show you what you need to add to the file. (Shown below under Error Messages)
- Once you are done editing the file, restart Postgres to apply the new settings (Ensure any Tomcat instances pointed to the database(s) on the server are stopped).
  - Alternatively, you can reload the pg_hba configuration without restarting by sending the main server process a SIGHUP signal. If you edit the file on an active system, you will need to signal the server (using pg_ctl reload or kill -HUP) to make it re-read the file. This will also load any changes to postgresql.conf.
- The Postgres server should now allow you to connect as configured.

Error Messages:

The following error message means that the server is not running on the host. This error message could also mean that the Postgres is not listening on all IP's--this typically only occurs with Postgres 8.2. To resolve this issue, see above section on getting Postgres 8.2 to listen on all IP's.

```
psql: could not connect to server: Connection refused

        Is the server running on host "192.168.15.163" and accepting

        TCP/IP connections on port 5432?
```

The following error message means that the server does not have an entry in pg_hba.conf for your connection. The message helps indicate the type of entry you would need to add.

```
psql: FATAL:  no pg_hba.conf entry for host "192.168.15.56", user "postgres", database
"openclinica", SSL off
```

The above message is saying your IP is "192.168.15.66", your user is "postgres" and you are trying to connect to database "openclinica." Based on this information, your new pg_hba.conf configuration line would be the following:

```
 host    postgres    openclinica        192.168.15.66/32      md5
```

This page is not approved for publication.

8.2 Email Configuration and Troubleshooting Tips

This page provides example settings for how to configure email functionality within OpenClinica.

OpenClinica can work with several different email server configurations. Some of the most common configurations are below. These configurations have been tested and are known to work with OpenClinica 3.1.2.

SMTP Relay (no user authentication)

```
mailHost=IPHOSTNAME OF YOUR E-MAIL SERVER
mailPort=25
mailProtocol=smtp
mailUsername=
mailPassword=
mailSmtpAuth=false
mailSmtpStarttls.enable=false
mailSmtpsAuth=false
mailSmtpsStarttls.enable=false
mailSmtpConnectionTimeout=1000
mailErrorMsg=developers@openclinica.com
```

SMTP Relay (user authentication)

```
mailHost=IPHOSTNAME OF YOUR E-MAIL SERVER
mailPort=25
mailProtocol=smtp
mailUsername=USERNAME
mailPassword=PASSWORD
mailSmtpAuth=true
mailSmtpStarttls.enable=false
mailSmtpsAuth=false
mailSmtpsStarttls.enable=false
mailSmtpConnectionTimeout=1000
mailErrorMsg=developers@openclinica.com
```

SMTP SSL - GMAIL

```
mailHost=smtp.gmail.com
mailPort=465
mailProtocol=smtps
mailUsername=USERNAME@gmail.com
mailPassword=PASSWORD
mailSmtpAuth=false
mailSmtpStarttls.enable=false
mailSmtpsAuth=true
mailSmtpsStarttls.enable=false
mailSmtpConnectionTimeout=1000
mailErrorMsg=developers@openclinica.com
```

Gmail reference material

SMTP TLSSTARTTLS - GMAIL

```
mailHost=smtp.gmail.com
mailPort=587
mailProtocol=smtp
mailUsername=USERNAME@gmail.com
mailPassword=PASSWORD
mailSmtpAuth=true
mailSmtpStarttls.enable=true
mailSmtpsAuth=false
mailSmtpsStarttls.enable=false
mailSmtpConnectionTimeout=1000
mailErrorMsg=enterprise-support@akazaresearch.com
```

Gmail reference material

Troubleshooting Tips

While the above configurations have been tested and work, some people may nonetheless find issues sending email. If you are still having trouble sending email with your OpenClinica instance, the information below may assist you.

Make sure the mail server is accessible from your OpenClinica server:

Servers typically have firewalls in place that filter network traffic. You can use the following "telnet" command to ensure that your server can communicate on its port designated for email. For instance if you are using SMTP, most likely your mail server port is 25.

```
telnet $IP 25
```

Simply replace $IP with the IP address or hostname of your email server. If you see "Connected to $IP" the mail server is accessible from your server.

If you see "telnet: Unable to connect to remote host: Connection refused" the problem lies with your network access and you will not be able to send email until your server can communicate with the mail server through the specified port. Please consult your IT department to enable the network access you need.

Test email delivery:

You may use the following to test email delivery via telnet. This will allow you to rule out the OpenClinica application as the cause of the issue preventing email from working. This test is usually the most helpful in determining your issue with email delivery. (Note: this will not work for SMTPS or STARTTLS mail servers.)

Testing for SMTP no user authentication - http://www.yuki-onna.co.uk/email/smtp.html

Testing for SMTP with user authentication - http://www.webpan.com/Customers/Email/SMTP_Authentication_Telnet_Test.htm

If you are able to receive the test email sent via telnet on the OpenClinica server, then OpenClinica should have no issues with email delivery.

If you get an error message or issue with the telnet test on the OpenClinica server please contact your IT department or mail provider for help in resolving.

Other ideas or suggestions:

If you would like to submit any new configuration examples or troubleshooting tips based on your own experiece, we would be more then happy to post them on this page. To submit new configurations or tips please post to the users email list with a Subject of "E-mail Additions" and we will add them to this page. Thanks!

This page is not approved for publication.

8.3 Backup and Restore OpenClinica

This page covers how to backup and restore OpenClinica data on the PostgreSQL database. OpenClinica keeps data needed to run the system in two locations, the actual database and the openclinica.data folder. Note: If changes to either of the variables below (which reference other folders) where made in the datainfo.properties configuration file you will need to backup these folders in your configured location in order to have a complete backup.

- attached_file_location
- exportFilePath

Backup OpenClinica

A full backup of OpenClinica data can be accomplished by the following steps:

1. Backup the DB
2. Backup the openclinica.data folder

Backup the DB

Here are the steps for backing up your DB for both Linux and Windows:

Linux

Before you backup the DB stop Tomcat. Once Tomcat has stopped, take a backup of the DB using the "pg_dump" command from PostgreSQL. To accomplish this run the following commands:

- /etc/init.d/tomcat stop **(Confirm Tomcat is stopped before continuing.)**
- /opt/PostgreSQL/8.4/bin/pg_dump -F c -U postgres openclinica > openclinica.backup **(When prompted for a password enter your "postgres" DB superusers password.)**

You now have a complete backup of your OpenClinica DB.

In case you need it, here is some pg_dump reference material.

Windows

Before you backup the DB stop Tomcat. Once stopped we will take a backup using the "pg_dump" command from PostgreSQL. To accomplish this run the following commands:

- Open a command prompt and type "net stop tomcat6" and hit "enter" to stop tomcat.
- C:Program Files (x86)PostgreSQL8.4binpg_dump.exe -F c -U postgres openclinica > openclinica.backup (When prompted for a password enter your "postgres" db superusers password)
  - On 32-Bit editions of windows the path will be the following for pg_dump.exe C:Program FilesPostgreSQL8.4binpg_dump.exe

You now have a complete backup of your OpenClinica DB.

In case you need it, here is some pg_dump reference material.

Backup the openclinica.data folder

Here are the steps to backup the openclinica.data folder in Linux or Windows:

Linux

The following will create a backup of your openclinica.data folder into the current directory:

- cp -rf /usr/local/tomcat/openclinica.data .

Windows

The following illustrates how to get a backup of your openclinica.data folder on Windows:

- Open "My Computer" or "Explorer"
- Navigate to "c:octomcat"
- Copy the folder labeled "openclinica.data" within"c:octomcat" to your backup location.

Restore OpenClinica

Restoring OpenClinica is quite simple. The recommended method is simply follow the OpenClinica installation guide for your OS. Once you have a blank OpenClinica instance ready, use the following steps to restore your data. By installing a blank OpenClinica instance first you can more easily rule out potential problems which may be caused by configuration issues.

The basic steps to restore your data into OpenClinica are:

1. Install a new blank OpenClinica instance.
2. Restore the OpenClinica DB from backup.
3. Restore the openclinica.data folder from backup.

Step 1: Install a new blank OpenClinica instance

Linux

- Please follow this guide to install a new OpenClinica instance on Linux.

Once you see the OpenClinica login screen you are ready to proceed to Step 2.

Windows

- Please follow this guide to install a new OpenClinica instance.

Once you see the OpenClinica login screen you are ready to proceed to Step 2.

Step 2: Restore the OpenClinica DB

To restore the OpenClinica DB:

- Stop Tomcat
- Drop the current DB
- Create a new DB
- Restore your database backup file into the DB

The following describes how to do this for both Linux and Windows:

Linux

Run the following commands:

- /etc/init.d/tomcat stop **(Confirm tomcat is stopped before continuing)**
- /opt/PostgreSQL/8.4/bin/dropdb -U postgres openclinica **(When prompted for a password enter your "postgres" db superusers password)**

- **/opt/PostgreSQL/8.4/bin/psql -U postgres -c "CREATE DATABASE openclinica WITH ENCODING='UTF8' OWNER=clinica" (When prompted for a password enter your "postgres" db superusers password)**
- "cd" to change directory to where your backup DB file is located.
- **/opt/PostgreSQL/8.4/bin/pg_restore -U postgres -d openclinica < $DBBACKUPFILE (When prompted for a password enter your "postgres" db superusers password)**

The backup of your DB has now been restored.

dropdb - Reference Material

pg_restore - Reference Material

Windows

- Open a command prompt and type "net stop tomcat6" to stop tomcat.
- In the command prompt "cd" to the following directory "C:Program Files (x86)PostgreSQL8.4bin"
  - On 32-Bit editions of windows the path will be the following C:Program FilesPostgreSQL8.4bin
- In the command prompt now run the following commands.
  - dropdb -U postgres openclinica **(When prompted for a password enter your "postgres" db superusers password)**
  - **psql -U postgres -c "CREATE DATABASE openclinica WITH ENCODING='UTF8' OWNER=clinica" (When prompted for a password enter your "postgres" db superusers password)**
  - **pg_restore -U postgres -d openclinica < $DBBACKUPFILE (When prompted for a password enter your "postgres" db superusers password)**

The backup of your DB has now been restored. This can also be done via PGAdminIII.

dropdb - Reference Material

pg_restore - Reference Material

Restore the openclinica.data folder

Restoring the openclinica.data folder is quite simple--just copy the folder to your new installation location after deleting the current default folder. Below list the steps to accomplish this for Linux and Windows.

Linux

- Issue the following command to go to the root folder of the openclinica.data directory "cd /usr/local/tomcat"
- Issue "rm -rf openclinica.data" to remove the current default installations openclinica.data folder.
- Issue "cp -rf $OPENCLINICA.DATAFOLDER BACKUP /usr/local/tomcat/" to restore your openclinica.data folder.
- Next issue a "chown -R tomcat /usr/loca/tomcat/*". This ensures your permissions are setup correctly for OpenClinica to access the folder.

Your openclinica.data folder is now restored.

Windows

- Open "My Computer" or "Explorer"
- Navigate to "c:octomcat"
- Delete the current "openclinica.data" folder within "c:octomcat"
- Copy the openclinica.data folder you backed up to "c:octomcat"

Your openclinica.data folder is now restored.

Current Community Contributed Content

http://en.wikibooks.org/wiki/OpenClinica_User_Manual/BackupAndRestore

Other Useful Links

http://www.postgresql.org/docs/8.4/static/backup.html

http://www.cyberciti.biz/tips/howto-backup-postgresql-databases.html

http://archives.postgresql.org/pgsql-admin/2001-03/msg00143.php

http://wiki.postgresql.org/wiki/Automated_Backup_on_Windows

http://www.wisdombay.com/articles/article000013.htm

Other Ideas or Suggestions

If you would like to submit any new configuration examples or troubleshooting tips based on your own experience, we would be more then happy to post them on this page. To submit new configurations or tips please post to the users email list with a Subject of "DOCS SITE:Backup and Restore" and we will add them to this page. Thanks!

This page is not approved for publication.

8.3.1 Community Contributed Backup Scripts

You will need to have 7zip installed for this method to work.

You may download the files reference in this article here

Dear All,

I would like to share my OC daily backup script and OC daily archive backup script to the community, both of them should works with both OC 3.0.x and OC 3.1.x (please note that both of scripts are for OC installed on Windows only).

Please see the description of individual script file below:

1. OC_Daily_Backup.txt (please rename its extension to .cmd after copied to OC server)

Description:

- Configurable OC backup related parameters

- Expect to scheduled execute on daily basis

- Steps are VACUUM DB (db maintenance), OC DB backup, OC configuration file backup, and OC data directory backup

- After successfully executed, it will create single zip file with the following name format OC_Daily_[Year]-[Month]-[Date]-[Date Init]--[Hour]-[Min].zip

(i.e. OC_Daily_2012-02-29-Wed--001-49.zip) in configured OC backup path

- The script also perform zip self test and generate test results file with the following name format OC_Daily_[Year]-[Month]-[Date]-[Date Init]--[Hour]-[Min]_Zip_Status.txt

(i.e. OC_Daily_2012-02-29-Wed--001-49_Zip_Status.txt) in configured OC backup path, sample of file is in attached.

2. OC_Daily_Backup_Archive.txt (please rename its extension to .cmd after copied to OC server)

Description:

- Configurable OC backup archive related parameters

- Expect to scheduled execute on periodically basis (every xx days)

- Steps are added all existing OC daily backup zip files and OC daily backup zip status files into single archive file with the following name format OC_Archive_[Year]-[Month]-[Date]- --[Hour]-[Min].zip

(i.e. OC_Archive_2012-02-29--01-48.zip) in configured OC backup archive path

- After successfully executed, it will delete all existing daily backup zip file(s) and zip status file(s)

- The script also perform zip self test and generate test results file with the following name format OC_Archive_[Year]-[Month]-[Date]- --[Hour]-[Min]_Zip_Status.txt

(i.e. OC_Archive_2012-02-29--01-48_Zip_Status.txt) in configured OC backup archive path, sample of file is in attached.

Please feel free to edit/modify them according to your environment or your need, any comment/feedback are welcome J

Hope it helps & cheers,

Montri

8.4 Loading the Demo Database

The OpenClinica Demo Database provides some pre-configured example studies that can help you learn about OpenClinica. The download package is available here and contains a database as well as some supporting files.

The demo database can be loaded either via your system's command line or with the free graphical tool, pgAdmin (download here). If you want to load the demo database from the command line, you can follow these Backup and Restore instructions.

If you would prefer to use pgAdmin, we will be posting instructions here soon. In the meantime, feel free to ask any questions on the OpenClincia Forums.

Approved for publication by Warren Vandeventer. Signed on 2017-02-01 3:55PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

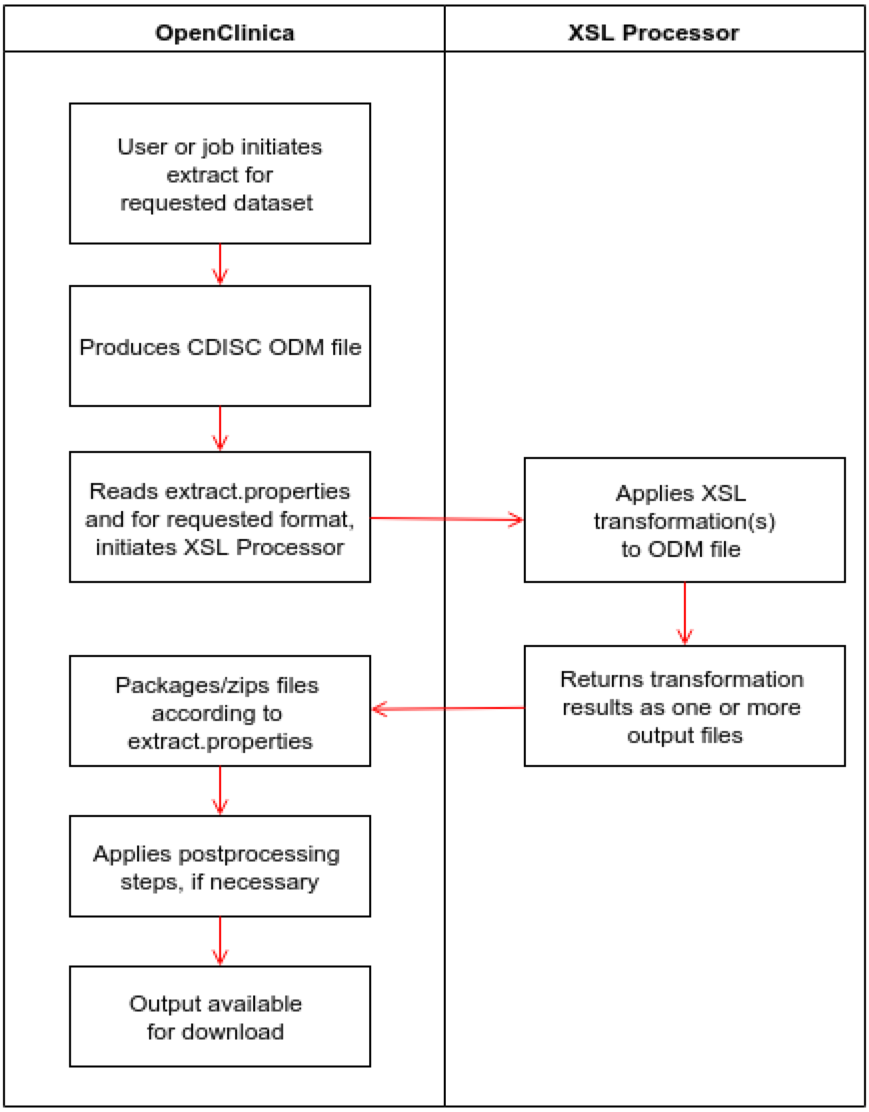
9 Build a Data Extract or Reporting plug-in

Overview

OpenClinica's Extract Data architecture lets you develop data extract formats that meet your precise requirements. It does this by:

- Specifying available formats, their associated stylesheets, and associated properties (like filename, archival settings, and whether to compress the file) in a properties file (the extract.properties file)
- Using XSL stylesheet transformations to read native CDISC ODM XML and output the data in a transformed format.
- Optionally, enabling postprocessing of the transformed data to output to certain non-text file formats and destinations.

Add an extract format to your OpenClinica environment


OpenClinica
XSL Processor
User or job initiates extract for requested dataset
Produces CDISC ODM file
Reads extract.properties and for requested format, initiates XSL Processor
Applies XSL transformation(s) to ODM file
Returns transformation results as one or more output files
Packages/zips files according to extract.properties
Applies postprocessing steps, if necessary
Output available for download

- Locate XSL files for the format you want to add. You can find packages on the OpenClinica Extensions site and in Lindsay Steven's Github repository.
- Add your files to the xslt directory in your OC environment, normally your_OC_data_directory/xslt
- Edit your extract.properties and add a new extract form.
- Restart OpenClinica and test it out!

Create a new extract format

You can add your own transformations to get data into a wide variety of formats using the XSLT language.

- Familiarize yourself with OpenClinica's implementation of CDISC ODM.
- Create your XSL file. While you can start from scratch, you'll save time if you work off one of the existing OpenClinica extract files, from the Extensions site, github, or CDISC's Define.xsl.
- If your requirements include outputting several files at once (such as a data file and load script), look at the SPSS format in extract.properties to see how you can include multiple XSL files and have them produce multiple output files.
- Postprocessing: To do things that XSLT cannot do by itself, like produce PDF files or load the data into external relational databases for ad-hoc reporting, a postprocessor framework is available to generate binary output formats or send data to a target destination. Two postprocessors are included: output to a database using JDBC connectivity and generate PDF

files using XSL-FO. The postprocessing step is transparent to end-users; they simply get their files for download or alternatively receive a message that the data has been loaded into the database. Instructions for use are provided in the extract.properties.

- Add the XSL to your OpenClinica environment as described above.

Use your extract format

Initiate an extract for your study from the Download Data screen or via a job and select your new output format. Execution follows a five step process:

1. OpenClinica generates CDISC ODM XML version 1.3 with OpenClinica Extensions
2. OpenClinica applies the XSL transformation and generates output file(s) according to the settings in extract.properties for the specified format
3. Optionally, if postprocessing is enabled for the requested format, OpenClinica runs the post processing action according to the settings in extract.properties.
4. OpenClinica provides user notification with success or failure message.
5. The data is available for download.

Other notes

- A framework exists in the code to add additional postprocessors via the addition of Java classes with references to those class names in the extract.properties file.
- Do not replace the extract XSLs that come with OpenClinica. If you do, your changes will be overwritten with the original contents every time OpenClinica is restarted.
- The Java code in the OpenClinica Extract Data module outputs study metadata and clinical data in only one format: CDISC ODM (version 1.3, with OpenClinica Extensions). OpenClinica's vendor extensions in the ODM file ensure that we can extract all possible data related to a study and its clinical data, even if not supported by the core ODM standard. This includes audit trail, discrepancy, and electronic signature information.
- Transformations are powered by the Saxon XSLT and XQuery processor.

Sharing

If you improve an existing extract format, create your own, or add a new postprocessor, please share it with the community!

Approved for publication by Cal Collins. Signed on 2016-03-11 3:34PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

10 CDISC ODM Representation in

OpenClinica

The Clinical Data Interchange Standards Consortium (CDISC) is a clinical research standards body formed to encourage maximum sharing of information and minimum duplication of efforts. One of the standards CDISC has created and endorsed is the Operational Data Model (ODM), which facilitates the archive and interchange of the metadata and data for clinical research. ODM is represented in XML format and is designed to collect data from many different sources into one document.

Purpose of this document

OpenClinica provides and/or consumes CDISC ODM XML representations in its Extract Data and Import Data modules and other parts of the software. This document describes how OpenClinica represents study metadata and data that is stored in its internal database as CDISC ODM XML documents. It assumes a working knowledge of CDISC ODM 1.3 and of OpenClinica, and attempts to describe how each OpenClinica field or element is represented in ODM, and under what conditions. The document is best read when accompanied by the CDISC ODM standard. It is geared towards developers, but is also intended for data managers who want to know more about the capabilities of ODM XML export in OpenClinica. Additionally, parts of this document will find its way into the online documentation, for all end users.

OpenClinicas ODM representation has changed iteratively from version to version of OpenClinica, and the appendix to this document charts these changes since version 2.5 and the addition of the custom extension to the ODM, introduced with OpenClinica 3.0.

Scope of this document

This document provides a detailed specification of the OpenClinica CDISC ODM XML version 1.3 with OpenClinica Extensions as implemented in the OpenClinica 3.1 and later releases.

Definitions and acronyms

- CDISC Clinical Data Interchange Standards Consortium
- DOM Document Object Model
- ODM CDISC Operational Data Model
- XML Extensible Markup Language
- XSL Extensible Stylesheet Language, a language for displaying an XML file as a given type.

10.1 CDISC ODM - General Issues

CDISC defines its Operational Data Model, version 1.3, as a vendor neutral, platform independent format for interchange and archive of clinical trials data. The model includes the clinical data along with its associated metadata, administrative data, reference data and audit information. All of the information that needs to be shared among different software systems during the setup, operation, analysis, submission or for long term retention as part of an archive is included in the model.

An XML document must meet certain basic criteria to be considered conformant to the ODM standard. These are briefly discussed below:

10.1.1 Syntatic Constraints

The syntactic constraints defined by the ODM standard are

1. The ODM file must be a well-formed XML file. See the XML standard for details.

2. The ODM file must conform to the XML Namespace standard. See the XML Namespace standard for details.

3. The ODM file must contain only elements and attributes defined in the ODM standard schema or in a valid vendor extension schema, and must satisfy the rules about element nesting and the formats of attribute values and element bodies.

4. The ODM file must contain a prolog and a single (top-level) ODM element.

5. The namespace for version 1.3 of the ODM is http://www.cdisc.org/ns/odm/v1.3.

OpenClinica ODM Exports from the Extract Data module meets these constraints.

Currently, the ODM study definition file (available from the View Study page at the URL /DownloadStudyMetadata?studyId=#) does not meet these requirements for the following reasons:

1. The file generated from the View Study page is only a fragment of XML, and does not contain the initial tag which defines the character set and version, i.e. <?xml version="1.0" encoding="UTF-8"?>.
2. The file generated does not contain references to any XML Namespaces, including the namespace for version 1.3 of the ODM itself.
3. The file generated does contain elements defined in the ODM standard schema, but lacks the single top-level ODM element.
4. The files suffix is txt instead of xml.

OpenClinica ODM Data Import meets the above constraints, but note that OpenClinica parses everything within the ClinicalData element only, and it does not read anything in the Study element, and, as such, cannot import Study metadata at this time.

10.1.2 System Conformity

A computing system that processes information in ODM format can claim conformance to this standard only if it obeys the following rules.

1. Generated ODM files must satisfy all the correctness rules in the standard, both syntactic and semantic.

2. A receiving system must be able to read any ODM file that satisfies all the correctness rules in this standard, both syntactic and semantic.

3. ODM files must validate against the ODM schema for the ODM version indicated in the ODM root element.

4. Information included in generated ODM files must be accurate according to the rules of this standard as defined in this specification.

5. A receiving system must interpret information read from an ODM file accurately according to the rules of this standard as defined in this specification.

6. Generated ODM files need not include information that is not normally handled or stored by the generating system.

7. A receiving system may selectively ignore information read from an ODM file if that information is not normally handled or stored by the receiving system.

8. A receiving system may constrain the range of data values, keys, names, and so on, that it is capable of handling or storing.

9. Systems that receive ODM clinical data files but do not normally support one or more of the datatypes specified in section 2.14, should accept clinical data of the unsupported types as text.

10. All system limitations (rules 6 through 9) must be documented.

11. If conformity is dependent on certain modes or settings, this must also be documented.

OpenClinicas support for generated and received meets these constraints, including documentation of limitations (#10). The limitations are described in the Limitations section of this document.

10.1.3 Vendor Extensions

Requirements for Vendor extensions to the ODM model are:

1. The vendor must supply a XML Schema fully describing their extended ODM format.

2. Extended ODM files should reference the proper extension Schema.

3. The extension may add new XML elements and attributes, but may not render any standard ODM elements or attributes obsolete. Vendor extensions cannot be used for information that is normally expressed using other ODM elements.

4. All new element and attribute names must use distinct XML namespaces to insure that there are

no naming conflicts with other vendor extensions.

5. Removing all vendor extensions from an extended ODM file must result in a meaningful and accurate standard ODM file.

6. Vendors should be able to produce ODM files free of any vendor extensions upon request.

OpenClinica meets these requirements in its use of vendor extensions.

10.2 Limitations on Support for Generated and Received ODM Files

OpenClinica has the following limitations on its support for generated and received ODM files:

OpenClinica acts as a receiving system via its Data Import functionality. In this case it selectively ignores information as allowed by the conformity requirements. In particular:

- Receiving files for data import ignores any information under the Study AdminData, ReferenceData, and Association nodes. When receiving files OpenClinica only deals with ClinicalData
- OpenClinica does not support the typed data transmission construct in ClinicalData (see ODM section 2.5.1 - Typed Data Transmission). All files generated by OpenClinica use the generic ItemData tag rather than ItemData[TYPE]. If when receiving files (ie in Data Import), OpenClinica encounters ClinicalData using ItemData[TYPE] tags, these entities are ignored. They will pass through schema validation, but the data will not be imported, as OpenClinica wont be able to generate the necessary Java objects to put data into the database.
- OpenClinica does not support the same set of Item Data types as specified in the ODM definition. For generation of ODM, the types supported are defined in the table Item Data Types in this document. For receiving ODM, since only ClinicalData is recognized, and Item Data Types defined as part of the incoming ODM file will be ignored. Incoming ItemData values will be validated based on the OpenClinica data type of the specified Item OID.

10.3 Entities and Elements

Entities and elements in OpenClinica use the same names as their counterparts in ODM. For example, the ODM definitions for study event and Study Event Definition are valid for the entities of the same name in OpenClinica (see Section 2.6, Entities and Elements, of the ODM specification):

- A *study event* is a reusable package of forms usually corresponding to a study data-collection event.
- A *Study Event Definition* describes a particular type of study event (mostly by listing the types of forms it can contain).

- The clinical data of a study will typically have many actual study events corresponding to each StudyEventDef,

Where the usage of these entity names in OpenClinica diverges from the ODM definition, it will be noted in this document.

Approved for publication by Cal Collins. Signed on 2014-04-02 2:56PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

10.4 OIDs and Clinical Data Keys

Approved for publication by Ben Baumann. Signed on 2014-03-24 9:56AM

Not valid unless obtained from the OpenClinica document management system on the day of use.

10.4.1 Generation of OIDs

OpenClinica uses Object Identifiers (OIDs, sometimes referred to as OpenClinica IDs) to link objects to one another, and enforce uniqueness within an instance of OpenClinica (with the exception of Rule OIDs, which must only be unique within a study). We currently utilize OIDs for the following objects in OpenClinica:

- CRFs
- CRF Versions
- Item Groups
- Items
- Measurement Units
- Rules
- Study Event Definitions
- Studies
- Sites
- Study Subjects

Each OID generated by OpenClinica starts with a prefix based on the type of entity, followed by *n* characters taken from the start of the proper name of the object (alpha-numeric only and converted to all caps), plus a random number at the end if the OID already exists in the database. This random number is generally 3-4 characters long (345, 5678, etc). Each part of the OID is separated by underscores. Using this scheme, we generate the following OIDs for objects:

- CRF: F + first 12 alpha-numeric characters in the CRF Name + random number if necessary (e.g. F_PHYSEXAM_7133)
- CRF Version: the CRF OID + the first 10 alpha-numeric characters in the CRF Version Name +

random number if necessary
- Item Group: IG + first 5 alpha-numeric characters in the CRF Name + all the alpha-numeric characters in the Item Group Label + random number if necessary
- Item: I + first 5 alpha-numeric characters in the CRF Name + first 26 alpha-numeric characters in the Item Label + random number if necessary
- Measurement Unit: MU + first 37 alpha-numeric characters in the measurement unit name + random number if necessary
- Rule: generated by user input at the Import Rules stage of OpenClinica. Must be alpha-numeric and all caps (underscores allowed) with a maximum 40 characters. Each Rule OID must be unique within the study that it is uploaded to.
- Study Event Definition: SE + first 28 alpha-numeric characters in study event definition name + random number if necessary
- Study: S + first 8 alpha-numeric characters in the Studys Unique Protocol ID + random number if necessary
- Site: S + first 8 alpha-numeric characters in the Studys Unique Protocol ID + random number if necessary
- Study Subject: SS + all alpha-numeric characters in the Study Subject ID + random number if necessary. Each Study Subject OID will be unique within the entire OpenClinica instance. In ODM, Study Subjects are given a SubjectKey to identify a specific subject that is unique within the study, while OpenClinica Study Subject OIDs are unique across all studies in the instance.

10.4.2 OID Scope, Context and Uniqueness Constraints

ODM defines a system of OID uniqueness constraints for Element Identifiers and references, which only require uniqueness within certain contexts. For example, the Study OID has to be unique within a series of ODM documents, while an OID for ItemGroup has to be unique within its Study only. Aside from that, notable requirements for ODM OID uniqueness include the following:

- ODM OIDs for each element type inside a MetaDataVersion must be unique for that scope of the MetaDataVersion.
- MetaDataVersion OIDs must be unique within the containing Study.
- ArchiveLayout OIDs, while not used in OpenClinica yet, must be unique within a single FormDef.
- Study OIDs must be unique within the containing ODM.
- Measurement Unit OIDs must be unique within the containing Study.
- User, Location and Signature OIDs, while not used in OpenClinica yet, must be unique within the containing AdminData.

OpenClinica generally adheres to these contraints, but OpenClinica OIDs are also unique across all studies in a single instance of OpenClinica. Because of this, an entity (such as a CRF) loaded into a separate instance of OpenClinica may not have the same OIDs in the new instance since those OIDs may already be assigned. This has drawbacks for reuse of CRFs and Rules across instances (such as porting a study definition from test to production) because the OIDs of items, groups, forms, and events that are referenced in the Rules may change when loaded into the new instance.

10.4.3 Clinical Data Keys

The ODM standard uses the concept of Internal Clinical Data Keys to uniquely address clinical data entities within the model. The following table details the key, or combination of entity identifiers, that you would need to uniquely and specifically address a clinical data entity.

Kind of Entity	**Identifying Keys (ODM)**	**Identifying Keys (OpenClinica ODM)**
study	StudyOID	Same as ODM
subject	above plus SubjectKey	Same as ODM
study event	above plus StudyEventOID and StudyEventRepeatKey	Same as ODM
Form	above plus FormOID and FormRepeatKey	Same as ODM, however repeating forms are not supported so no FormRepeatKey is necessary
Item group	above plus ItemGroupOID and ItemGroupRepeatKey	Same as ODM
item	above plus ItemOID	Same as ODM
annotation	keys for the annotated entity plus SeqNum	Not used in OpenClinica

For example, an XPath query to retrieve a specific item data value in an OpenClinica ODM Extract would be of the form:

```
/odm:ODM/odm:ClinicalData[@StudyOID='S_P12345_2818']/odm:SubjectData[@SubjectKey='SS_101']/odm:StudyEventData[@StudyEventOID='SE_INITIALT' and @StudyEventRepeatKey='1']/odm:FormData[@FormOID='F_AGEN_V10']/odm:ItemGroupData[@ItemGroupOID='IG_AGEN_DOSETABLE-F_AGEN_V10' and @ItemGroupRepeatKey='1']/odm:ItemData[@ItemOID='I_AGEN_AGENT_NAME']/@Value
```

In the image below you can see that the latter half of the XML file (the part contained in the <ClinicalData> tags) links to specific tables in the OpenClinica database. We then link back to the Study metadata through those OIDs. Internally we dont use OIDs in those tables, but instead the conventional methods of primary keys and foreign keys in the database is good enough. For more on how ODM, OIDs, and Clinical Data Keys are implemented in OpenClinica see the blog post, "Thoughts on Code: OpenClinica and Open Standards with CDISC".

10.5 File Properties, Transactions and Element Ordering

ODM and OpenClinica support versioning of files with a FileOID attribute which is meant to uniquely identify the document. The OID is generated in OpenClinica Extract Data module, from the file name which contains a timestamp down to the millisecond. No effort is made to parse or connect this FileOID upon Bulk Data Import, however.

ODM and OpenClinica both support the FileType attribute, which is automatically set to Snapshot in Extract Data.

ODM also supports an optional Archival attribute, but OpenClinica does not generate or support this attribute currently.

In ODM, each data attribute has an optional TransactionType attribute. This attribute can be one of Insert, Update, Remove or Upsert. OpenClinicas Extract Data always outputs TransactionTypes of Insert. OpenClinicas Data Import module always processes data as if it were a TransactionType of Upsert, and ignores any TransactionType attributes specified in the ODM.

ODM also supports the concept of Element Ordering: where all time stamps on each element must precede the CreationDateTime of the ODM file. Currently, OpenClinica does not have any logic which monitors date correctness in ODM files, during Import or Extract.

10.6 Study Metadata Versions and Sites

Within the study metadata file generated by OpenClinica, we contain CRF Versions inside of each FormRef called inside the StudyEventDef element of the ODM document. Since CRF Versions can appear in different study event definitions in OpenClinica, we allow duplicate calls to the same definition in this part of the file.

OpenClinica also supports inclusion of multiple instances of the MetadataVersion tag in its documents. This means that one ODM document can hold multiple sets of study metadata, detailing a study and one or more of its sites, for example. Each MetadataVersion has an automatically-generated OID, which can then be attached to any group of ClinicalData that is also contained in the same document.

The OID is generated automatically with the ODM Metadata Version number that is set in the Dataset, typically v1.0.0-, plus the Study OID. Part of the metadata which is central to all Studies is generated with only the ODM Metadata Version number, and then other versions (Sites) can access the metadata using an Include tag, which generally looks like this:

```
<Include StudyOID ="S_OID_HERE" MetaDataVersionOID="v1.0.0"/>
```

The Include tag allows Studies to share a collection of metadata, such as CRFs, CRF versions, etc. Typically, a parent Study would contain this metadata, and each Site would then reference it via the Include tag.

Approved for publication by Cal Collins. Signed on 2014-04-02 3:36PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

10.7 Data Representations in ODM XML (Extract)

When OpenClinica outputs ODM XML, the five basic XML entities (gt, lt, quot, amp, apos) are escaped using XML Entity notation (For example: "bread" & "butter" => &quot;bread&quot; &amp; &quot;butter&quot;).

Whitespace is represented literally linebreaks and tabs in ItemData values and other fields will be preserved. Note that, while tabs and carriage returns are limited in the data entry side of the application, (tabs will automatically shift focus from one Item to the next, for example) all spaces and linebreaks are saved to the database, and will export into ODM XML.

Items saved in the database with non-ASCII characters will be extracted to XML entities using their ASCII decimal value equivalents; please see the next section, "OpenClinica Data Representations in ODM XML (Extract)" for an example of this.

Approved for publication by Cal Collins. Signed on 2014-04-02 3:37PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

10.8 Data Representations in ODM XML (Import)

ItemData values being imported are validated against data type and length.

During data import the XML Entity representation of characters in ItemData values are treated as literals. So an Item Data value of

"bread" & "butter"

would be exported as

&quot;bread&quot; &amp; &quot;butter&quot;

and if it were then imported back into OpenClinica would be saved as

&amp;quot;bread& amp;quot; & amp;amp; & amp;quot;butter& amp;quot;

Whitespace, including linebreaks and tabs in ItemData values are preserved, but the validations on different data types react differently to whitespace. Single-select and multi-select values do not accept whitespace upon import, instead generating the hard validation check error message, This is not in the correct response set. Whitespace generated in date types passes validation, but is stripped out and does not re-export; tabs are kept in other data types, but returns are not kept.

For example, a set of Items in XML that is imported as the following:

```
<ItemData ItemOID="I_GROU_TC_ADV_PRIMARY_06" Value="Heart Attack      "/>

 <ItemData ItemOID="I_GROU_TC_ADV_PRIMARY_07" Value="2009-12-16       "/>

 <ItemData ItemOID="I_GROU_TC_ADV_PRIMARY_08" Value="11/02/2009         "/>
```

Will be returned as the following:

```
 <ItemData ItemOID="I_GROU_TC_ADV_PRIMARY_06" Value="Heart Attack"/>

 <ItemData ItemOID="I_GROU_TC_ADV_PRIMARY_07" Value="2009-12-16"/>

 <ItemData ItemOID="I_GROU_TC_ADV_PRIMARY_08" Value="11/02/2009"/>
```

Items imported with non-ASCII characters will be exported using escaped ASCII codes. For example, the following line

```
<ItemData ItemOID="I_GROU_TC_ADV_PRIMARY_06" Value="??? ????"/>
```

Will be returned as the following:

```
<ItemData ItemOID="I_GROU_TC_ADV_PRIMARY_06" Value="&#1087;&#1088;&#1086;
&#1099;&#1091;&#1080;&#1091;  "/>
```

Note that, if the above ASCII values are copied and inserted back into the database, they are still saved as UTF-8 values, and will be exported the same way, without any degradation (as in the above example with XML escaped characters).

Approved for publication by Cal Collins. Signed on 2014-04-02 3:37PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

10.9 Item Data Types

OpenClinica supports a subset of the Item Data Types defined in ODM. The data type mapping is shown below, along with the allowed string pattern used to validate item values for a given data type. Note that a listing of no definition in the table below means that the data type is not supported in OpenClinica.

Item Data Types

CDISC Item Data Type	***OpenClinica Data Type***	***Allowed Values (Data Import)***	***Representation of Values (Extract)***
text	ST	Any sequence of characters up to the maximum allowed number of characters (currently 4000). If the value is greater than the width is specified in the items width_decimal property (or 255, whichever is less), a discrepancy note will be created but the data will be allowed.	
partialDate	PDATE	A date represented according to the XML schema date datatype, which is based on the ISO8601 standard (YYYY-MM-DD).	A date represented according to the XML schema date datatype, which is based on the ISO8601 standard (YYYY-MM-DD). Partial Dates can be YYYY-MM or YYYY and will be exported as YYYY-MM or YYYY.
text	FILE	Files cannot be imported into ODM at this time.	A string representing the file name of the stored file, up to the maximum allowed number of characters (currently 4000).
integer	INT	-?digit+ If the value is greater than the width specified in the items width_decimal property (or 255, whichever is less), a discrepancy note will be created but the data will be allowed.	
float	REAL	-?digit+(.digit+)? If the value is greater than the width is specified in the items width_decimal property (or 255, whichever is less), a discrepancy note will be created but the data will be allowed. Float values will only be rounded for calculations, based on the decimal specified in the items width_decimal property if it exists. If no width_decimal is provided it will round to the 4th decimal place. For example, if someone entered a value like 6.987398 into a field that is not a calculation, the number will not be rounded to the 4th decimal place.	
date	DATE	A date represented according to the XML schema date datatype, which is based on the ISO8601 standard (YYYY-MM-DD).	A date represented according to the XML schema date datatype, which is based on the ISO8601 standard (YYYY-MM-DD).
time	No definition		
datetime	No definition		
string	No definition		
boolean	No definition		
double	No definition		
hexBinary	No definition		
base64Binary	No definition		
hexFloat	No definition		
base64Float	No definition		

partialTime	No definition		
partialDatetime	No definition		
durationDatetime	No definition		
intervalDatetime	No definition		
incompleteDatetime	No definition		
URI	No definition		

10.10 Mapping of OpenClinica Elements to ODM

CDISC ODM	OpenClinica Data Schema (table.column_name)	OpenClinica CRF Spreadsheet	Example Values (Comments)	Added to 3.1?
ODM				
FileOID	dataset.name-D-creationDateTime		Intervention_DatasetD20100924143323+0300	
CreationDateTime			2010-09-24T14:33:23+03:00	
FileType			Snapshot (generated automatically)	
ODMVersion			1.3	
xsi:schemaLocation			http://www.cdisc.org/ns/odm/v1.3 OpenClinica-ODM1-3-0-OC2-0.xsd	
Study				
OID	study.oc_oid		S_DEMO123	
StudyName	study.name		Demonstration Study	
StudyDescription	study.summary		This study is used for demonstrational purposes	
ProtocolName	study.unique_identifier		Demo123	
MeasurementUnit				
OID	measurement_unit.oc_oid		MU_HG	
Name	measurement_unit.name	UNITS	Hg	
Symbol				
TranslatedText	item.units	UNITS	Hg	
MetaDataVersion				
OID	dataset.odm_metadataversion_oid		v1.0.0 (defined at dataset creation time)	
Name	dataset.odm_metadataversion_name		MetaDataVersion_v1.0.0	
StudyEventRef				
StudyEventOID	study_event_definition.oc_oid		Systematically generated when a study event definition is created. Refer to section 2.4.1	
OrderNumber	study_event.sample_ordinal		The order the event definitions appear in the system. These values can be rearranged by certain users.	
Mandatory	event_definition_crf.required_crf		If at least one CRF in the event definition is set to Required, the event itself becomes required.	
StudyEventDef				
OID	study_event_definition.oc_oid		Systematically generated when a study event definition is created. Refer to section 2.4.1	
Name	study_event_definition.name		Initial Visit	
Repeating	study_event_definition.repeating		Yes, No	
Type	study_event_definition.type		Scheduled, Unscheduled, Common. The values conform to CDISC terms.	
FormRef				
FormOID	crf_version.oc_oid		F_DEMOGRAPHICS_1	
Mandatory	event_defintion_crf.required_crf		Yes, No	
FormDef				
OID	crf_version.oc_oid		F_DEMOGRAPHICS_1	
Name	crf.name + crf_version.name	CRF_NAME + VERSION	Demographics-Dynamics - 1	
Repeating			No (generated automatically)	
ItemGroupRef				
ItemGroupOID	item_group.oc_oid		IG_DEMOG_UNGROUPED	
Mandatory	Same as ItemRef Mandatory		Yes, No	
OpenClinica:FormDetails				
FormOID	crf_version.oc_oid		F_DEMOG_1	
ParentFormOID	crf.name	CRF_NAME	DEMOG	Y
VersionDescription	crf_version.description	VERSION_DESCRIPTION	This CRF version contains 2 additional items	Y
RevisionNotes	crf_version.revision_notes	REVISION_NOTES	I revised this CRF and created this new version.	Y
OpenClinica:SectionDetails				
OpenClinica:Section				
SectionLabel	section.label	SECTION_LABEL		
SectionTitle	section.title	SECTION_TITLE		
SectionSubtitle	section.subtitle	SUBTITLE		
SectionInstructions	section.instructions	INSTRUCTIONS		
SectionPageNumber	section.page_number_label	PAGE_NUMBER		
OpenClinica:PresentInEventDefinition				Y
StudyEventOID	study_event_definition.oc_oid		Systematically generated when a study event definition is created. Refer to section 2.4.1	Y
IsDefaultVersion	event_definition_crf.default_version_id		True, False	Y
PasswordRequired	event_definition_crf.electronic_signature		True, False	Y

DoubleDataEntry	event_definition_crf.double_entry		True, False	Y
HideCRF	event_definition_crf.hide_crf		True, False	Y
SourceDataVerification	event_definition_crf.source_data_verification		100% Required, Partial Required, Not Required, Not Applicable	Y
ItemGroupDef				
OID	item_group.oc_oid		IG_DEMOG_UNGROUPED	
Name	item_group.name	GROUP_LABEL	Ungrouped (if the items are ungrouped)	
Repeating	item_group_metadata.repeating_group	GROUP_LAYOUT	Yes, No	
Comment	item_group_metadata.header	GROUP_HEADER		
SASDatasetName			UNGRO044 (generated automatically; take the item_group.name, switch to all caps, and truncate to eight characters, adding integers and alpha characters to the end to make unique)	Y
ItemRef				
ItemOID	item.oc_oid		I_THAP_BLOODPRODTYP	
OrderNumber	item_form_metadata.ordinal		1, 2, 3, etc.	
Mandatory	item_form_metadata.required	REQUIRED	Yes, No	
OpenClinicaItemGroupDetails				
ItemGroupOID	item_group.oc_oid		IG_DEMOG_UNGROUPED	
OpenClinica:PresentInForm				
FormOID	crf_version.oc_oid			
ShowGroup	item_group_metadata.show_group	GROUP_DISPLAY_STATUS	HIDE, SHOW	Y
OpenClinica:ItemGroupRepeat				Y
RepeatNumber	item_group_metadata.repeat_number	GROUP_REPEAT_NUMBER		Y
RepeatMax	item_group_metadata.repeat_max	GROUP_REPEAT_MAX		Y
ItemDef				
OID	item.oc_oid		I_THAP_BLOODPRODTYP	
Name	item.name	ITEM_NAME	BloodProdTyp	
DataType	item_data_type.name	DATA_TYPES	Text, float, integer, etc.	
Length	item_form_metadata.width_decimal	WIDTH_DECIMAL*	(For ODM float data types, the length is based on the width parameter specified in the WIDTH_DECIMAL field in the spreadsheet (if it is provided). For other data types, length is calculated based on the maximum length of all the values for the item.)	
SignificantDigits	item_form_metadata.width_decimal	WIDTH_DECIMAL*	(For ODM float data types, the SignificantDigits is based on the decimal parameter specified in the WIDTH_DECIMAL field in the spreadsheet. For other data types, SignificantDigits is not used.)	
SASFieldName			Blood301 (again, generated from the item.name, but truncated to eight characters with an integer at the end to enforce uniqueness)	
Comment	item.description	DESCRIPTION_LABEL		
OpenClinica:FormOIDs	crf_version.oc_oid		F_THAP_V10 (comma-separated list of CRF version OIDs where the item is found)	
Question				
OpenClinica:QuestionNumber	item_form_metadata.question_number	QUESTION_NUMBER		Y
TranslatedText	item_form_metadata.header + item_form_metadata.left_item_text + item_form_metadata.right_item_text	LEFT_ITEM_TEXT RIGHT_ITEM_TEXT HEADER SUB_HEADER	Amount of blood product (ml)	
MeasurementUnitRef				
MeasurementUnitOID	measurement_unit.oc_oid		Systematically generated when a CRF Version is uploaded containing an item with a value in the UNITS cell.	
CodeListRef				
CodeListOID	response_set.response_set_id		CL_233 (OID generated automatically by adding the prefix CL_ to the primary key)	
RangeCheck				
Comparator	item_form_metadata.regexp (if "func:")	VALIDATION	LE, GE, EQ, etc.	
SoftHard			Soft, Hard (based on study parameter values governing discrepancy note generation)	
CheckValue	Item_form_metadata the value in the function. For example, 95 if the check is for greater than 95,	VALIDATION	100	
ErrorMessage				
TranslatedText	item_form_metadata.regexp_error_msg (if "func:")	VALIDATION_ERROR_MESSAGE	Please specify a value between 0 and 100	
OpenClinica:ItemDetails				
ItemOID	item.oc_oid		Systematically generated when a CRF Version is uploaded to OpenClinica	
OpenClinica:PresentInForm				
FormOID	crf_version.oc_oid			
ParentItemOID	item.oc_oid		Items can contain links to other items that are their parents. The value would reflect an OID for an item present in the same section of the same form.	Y
ColumnNumber	item_form_metadata.column_number	COLUMN_NUMBER	1, 2, 3 etc.	Y
PageNumber	item_form_metadata.page_number	PAGE_NUMBER	1, 2, 3 etc.	Y
PHI	item.phi_status	PHI	1, 0	Y
ShowItem	item_form_metadata.show_item	ITEM_DISPLAY_STATUS	SHOW, HIDE	Y
OrderInForm	item_form_metadata.ordinal		The items order shown in UI from view crf page.	
OpenClinica:LeftItemText	item_form_metadata.left_item_text	LEFT_ITEM_TEXT	Height	Y
OpenClinica:RightItemText	item_form_metadata.right_item_text	RIGHT_ITEM_TEXT	Height	Y
OpenClinica:ItemHeader	item_form_metadata.header	HEADER	This field captures Height	Y
OpenClinica:ItemSubHeader	item_form_metadata.subheader	SUB_HEADER	Please provide the Height in inches.	Y
OpenClinica:SectionLabel	section.label	SECTION_LABEL		Y
OpenClinica:ItemResponse	response_type.name	RESPONSE_TYPE		Y
CodeList				
OID	response_set.response_set_id		CL_233 (OID generated automatically by adding the prefix CL_ to the primary key)	

Name	response_set.label	RESPONSE_LABEL	yn	
DataType	item_data_type.name	DATA_TYPES	Text, float, integer, etc.	
SASFormatName			$YN54X (generated, with Name in all caps with a $ and an extra suffix to be unique)	
CodeListItem				
CodedValue	response_set.options_values	REPONSE_VALUES_OR_CALCULATIONS	1, 2, 3, etc.	
Decode				
TranslatedText	response_set.options_text	RESPONSE_OPTIONS_TEXT	Yes, No, etc.	
OpenClinica:MultiSelectList				
OpenClinica:ID	response_set.response_set_id		MSL_54 (generate an OID on the fly by adding the prefix MSL_ to the iprimary key)	
OpenClinica:Name	response_set.label	RESPONSE_LABEL	OptionsList	
OpenClinica:DataType	response_type.name		text, (the DataType value is always text)	
OpenClinica:ActualDataType	response_type.name	DATA_TYPE	text, float, integer	
OpenClinica:MultiSelectListItem				
OpenClinica:CodedOptionValue	response_set.options_values	RESPONSE_VALUES_OR_CALCULATIONS	1, 2, 3, etc.	
OpenClinica:StudyGroupClassList				
OpenClinica:ID	study_group_class.study_group_class_id		SGC_1 (SGC_ prefix automatically appended)	
OpenClinica:Name	study_group_class.name		Age Classification	
OpenClinica:Status	status.name		Available	
OpenClinica:Type	group_class_types.name		Arm, Demographic, etc.	
OpenClinica:SubjectAssignment	study_group_class.subject_assignment		Optional	
OpenClinica:StudyGroupItem				
OpenClinica:Name	study_group.name		Young Adult	
OpenClinica:Description	study_group.description		12-18	
OpenClinica:StudyDetails				
StudyOID	study.oc_oid			
SiteName	study.name		Present only when extracting site level ODM.	Y
ParentStudyName	study.name		Gives the name of study which this site is part of. Tag is present only when the ODM is retrieved at site level.	Y
OpenClinica:StudyDescriptionStatus				Y
OfficialTitle	study.official_title			Y
SecondaryIDs	study.secondary_identifier			Y
DateCreated	study.date_created		2011-01-01	Y
StartDate	study.date_planned_start		2011-01-01	Y
StudyCompletionDate	study.date_planned_end		2011-12-31	Y
OpenClinica:StudySystemStatus	study.status_id		Available, Pending, Locked, Frozen	Y
OpenClinica:PrincipalInvestigator	study.principal_investigator		John Smith	Y
OpenClinica:DetailedDescription	study.protocol_description		This is a demonstration study.	Y
OpenClinica:Sponsor	study.sponsor		Drug Company A	Y
OpenClinica:Collaborators	study.collaborators		This can be a list of different organizations	Y
OpenClinica:StudyPhase	study.phase		N/A, Phase I, Phase I/Phase II, Phase II, Phase II/Phase III, Phase III, Phase III/Phase IV, Phase IV	Y
OpenClinica:ProtocolType	study.protocol_type		Interventional, Observational	Y
OpenClinica:ProtocolVerificationDate	study.protocol_date_verification		2011-01-01	Y
OpenClinica:Purpose	study.purpose		Treatment, Prevention, Diagnosis, Supportive Care, Screening, Health Services Research, Basic Science, Other	Y
OpenClinica:Allocation	study.allocation		Randomized Clinical Trial, Non-Randomized Clinical Trial, N/A	Y
OpenClinica:Masking	study.masking		Open, Single Blind, Double Blind	Y
OpenClinica:Control	study.control			Y
OpenClinica:InverventionModel	study.interventions			Y
OpenClinica:ConditionsAndEligibility	study.eligibility			Y
OpenClinica:Conditions	study.conditions			Y
OpenClinica:Keywords	study.keywords			Y
OpenClinica:EligibilityCriteria				Y
OpenClinica:Sex	study.gender			Y
OpenClinica:Age				Y
MinimumAge	study.age_min			Y
MaximumAge	study.age_max			Y
OpenClinica:HealthyVolunteersAccepted	study.healthy_volunteers			Y
OpenClinica:ExpectedTotalEnrollment	study.expected_total_enrollment			Y
OpenClinica:FaciltyInformation				Y
OpenClinica:FacilityName	study.facility_name			Y
OpenClinica:FacilityCity	study.facility_city			Y
OpenClinica:FacilityState	study.facility_state			Y
OpenClinica:PostalCode	study.facility_zip			Y
OpenClinica:FacilityCountry	study.facility_country			Y
OpenClinica:FacilityContactName	study.facility_contact_name			Y
OpenClinica:FacilityContactPhone	study.facility_contact_phone			Y
OpenClinica:FacilityContactEmail	study.facility_contact_email			Y
OpenClinica:RelatedInformation				Y
OpenClinica:MEDLINEIdentifier	study.medline_identifier			Y
OpenClinica:ResultsReference	study.results_reference			Y
OpenClinica:URLReference	study.url			Y
OpenClinica:URLDescription	study.url_description			Y
OpenClinica:StudyParameterConfiguration				Y
OpenClinica:StudyParameterListRef				Y
StudyParameterListID	study_parameter.name			Y
Value	study_parameter_value.value			Y
OpenClincia:StudyParameterListItem				Y
CodedParameterValue	study_parameter.name			Y
RuleImport			Rules are a separate vendor extensions from the OpenClinica: namespace.	Y
RuleAssignment				Y
RuleRef				Y
OID	rule.oc_oid			Y
ShowAction	rule_action.action_type			Y
ifExpressionEvaluates	rule_action.expression_evaluates_to			Y
HideAction	rule_action.action_type			Y
ifExpressionEvaluates	rule_action.expression_evaluates_to		TRUE or FALSE	Y
EmailAction	rule_action.action_type			Y
ifExpressionEvaluates	rule_action.expression_evaluates_to		TRUE or FALSE	Y
DiscrepancyNoteAction	rule_action.action_type			Y
ifExpressionEvaluates	rule_action.expression_evaluates_to		TRUE or FALSE	Y

InsertAction	rule_action.action_type			Y
ifExpressionEvaluates	rule_action.expression_evaluates_to		TRUE or FALSE	Y
Run				Y
AdministrativeEditing	rule_action.administrative_data_entry		TRUE or FALSE	Y
InitialDataEntry	rule_action.initial_data_entry		TRUE or FALSE	Y
DoubleDataEntry	rule_action.double_data_entry		TRUE or FALSE	Y
Batch	rule_action.batch		TRUE or FALSE	Y
Message	rule_action.message		The values do not match, please verify the data provided.	Y
DestinationProperty				Y
OID	rule_action_property.oc_oid		This will be an OID. If the item is part of the same group, then the user should	Y
RuleDef				Y
OID	rule.oc_oid		All capital letters or integers. This is user defined	Y
Name	rule.name			Y
Description	rule.description		Describe the rule to put it into a context you will recognize.	Y
Expression	rule_expression.value		The expression is used to validate data values to then decide if an action should be taken.	Y
AdminData				Y
StudyOID	study.oc_oid			Y
User				Y
OID			Generated automatically by taking the value in user_account.user_id and appending USR_ to the front	Y
FullName	user.user_name		jsmith	Y
FirstName	user.first_name		John	Y
LastName	user.last_name		Smith	Y
Organization	user.institutional_affiliation		Enterprise Organization	Y
ClinicalData				Y
StudyOID	study.oc_oid		S_DEMO123	
MetaDataVersionOID	dataset.odm_metadataversion_oid		v1.0.0	
SubjectData				
SubjectKey	study_subject.oc_oid		SS_101	
OpenClinica:StudySubjectId	study_subject.label		101	
OpenClinica:UniqueIdentifier	subject.unique_identifier			
OpenClinica:Status	status.name		available	
OpenClinica:SecondaryId	study_subject.secondary_label			
OpenClinica:YearOfBirth	subject.date_of_birth		2001 (depends on the study parameters; if study does not require date of birth or records full date of birth, this is not included)	
OpenClinica:DateOfBirth	subject.date_of_birth		2001-01-01 (depends on the study parameters; if subject is year only or if date of dirth is not recorded, this is not included)	
OpenClinica:Sex	subject.gender		m, f	
StudyEventData				
StudyEventOID	study_event_definition.oc_oid		SE_OBSERVAT	
StudyEventRepeatKey			1, 2, 3, etc. (optional, not included if not repeating)	
OpenClinica:StudyEventLocation	study_event.location		Boston, MA	
OpenClinica:StartDate	study_event.date_start		2010-10-10	
OpenClinica:EndDate	study_event.date_end		2010-12-31 (optional, not included if left blank)	
OpenClinica:Status	status.name		signed, available, etc.	
OpenClinica:SubjectAgeAtEvent			13 (calculated automatically)	
FormData				
FormOID	crf_version.oc_oid		F_GROU_V22	
OpenClinica:Version	crf_version.name			
OpenClinica:InterviewerName	event_crf.interviewer_name			
OpenClinica:InterviewDate	event_crf.date_interviewed		2010-10-10	
OpenClinica:Status	status.name			
ItemGroupData				
ItemGroupOID	item_group.oc_oid		IG_GROU_GROUP_1	
TransactionType			Insert (generated by default)	
ItemGroupRepeatKey	item_data.ordinal if repeating		1, 2, 3, etc. (optional)	
ItemData				
ItemOID	item.oc_oid		I_GROU_TC_ADV_PRIMARY_03	
Value	item_data.value			
IsNull			Yes (only generated if the item is null)	
OpenClinica:ReasonForNull	item_data.value		(only generated if the item is null)	
OpenClinica:AuditLogs				Y
EntityID			Form OID, Item OID, Event OID, Study Subject OID	Y
OpenClinica:AuditLog				Y
ID			System generated ID taken from the audit_log_event.audit_id and appending AL_ to the beginning	Y
UserID			References the value for User OID	Y
DateTimeStamp	audit_log_event.audit_date		1000-11-16T10:15:37	Y
AuditType	audit_log_event_type.name		The audit log type	Y
OldValue	audit_log_event.old_value			Y
NewValue	audit_log_event.new value			Y
OpenClinica:DiscrepancyNotes				Y
EntityID			Item OID, Event OID, Study Subject OID	Y
OpenClinica:DiscrepancyNote				Y
ID			System generated by appending DN_ to the value from discrepancy_note.discrepancy_note_id	Y
Status	resolution_status.name		New, Updated, Resolution Proposed, Closed, Not Applicable	Y
NoteType	discrepancy_note_type.name		Query, Failed Validation Check, Reason for Change, Annotation	Y
DateUpdated	discrepancy_note.date_created for a child note			Y
NumberOfChildren			Adds up the number of child notes that exist in the thread.	Y
OpenClinica:ChildNote				Y

Status	resolution_status.name		New, Updated, Resolution Proposed, Closed, Not Applicable	Y
DateCreated	discrepancy_note.date_created			Y
OpenClinica:Description	discrepancy_note.description		Value outside of range	Y
OpenClinica:DetailedNote	discrepancy_note.detailed_note		This value was outside of the range that is accepted. Please follow up with this subject and see if they should be enrolled or not.	Y
UserRef				Y
UserOID				Y
OpenClinica:SubjectGroupData				
OpenClinica:StudyGroupClassID	study_group_class.study_group_class_id		SGC_1	
OpenClinica:StudyGroupClassName	study_group_class.name		Age Classification	
OpenClinica:StudyGroupName	study_group.name		Young Adult	

Approved for publication by Cal Collins. Signed on 2014-04-02 3:48PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

10.11 CDISC ODM XML Schema

The schema for CDISC ODM, as implemented natively by OpenClinica (ODM 1.3, including vendor extensions) can be found at https://dev.openclinica.com/tools/odm-doc/. This page provides an overview of the schemas/namespaces that are used in an OpennClinica ODM document. Click on the 'schema location' for each individual schema to get the raw XSD. You can also see a visualization of the data structures here.

Approved for publication by Cal Collins. Signed on 2014-04-02 3:40PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

10.12 Global Case Report Form in ODM repesentation

Starting in OpenClinica 3.1.4, the print module has been revamped. The Printable CRF functionality uses javascript and CSS code applied to CDISC ODM data served by the REST API format to print the CRFs with and without data.

The CDISC-ODM model is based around a study protocol, and represents CRFs as part of a protocol. It does not cover CRFs when they are not part of a protocol, even if they are CRFs in the OpenClinica instance. To do this, and therefore allow printable views of non-study specific CRFs, we generate in the REST API calls a "fake" set of OIDs, starting with "OC_FORM_LIB" in the CDISC-ODM xml.

As an example:

```
<GlobalVariables><StudyName>OC_FORM_LIB_STUDY</StudyName><StudyDescription>

        </StudyDescription><ProtocolName/></GlobalVariables><MetaDataVersion OID="v1.0.0"
Name="MetaDataVersion_v1.0.0"><Protocol>
        </Protocol><StudyEventDef OID="OC_FORM_LIB_SE" Name="OC_FORM_LIB_SE_NAME"
Repeating="NO" Type="null">
        <FormRef FormOID="F_0000ARTESTRE_12"        Mandatory="null"/>
        </StudyEventDef>
        <FormDef OID="F_0000ARTESTRE_12" Name="1.2" Repeating="No">
        <OpenClinica:FormDetails FormOID="F_0000ARTESTRE_12"
ParentFormOID="F_TEST_FIELD_R">....
```

11 Developing for OpenClinica

How to contribute and what to work on

Read the overview of how to contribute here.

Basics of the OpenClinica Code

A typical feature might consist of a database change as well as middle-tier and back end integration.

For database changes, OpenClinica uses a 3rd party library called liquibase to upgrade the database & schema from version to version. Liquibase does a good job in maintaining the OpenClinica versions and making sure scripts are run at install/upgrade time. If the code requires a database change, such as adding a table, altering a table etc, it is important to put in liquibase script.

The business layer code implementation is based on a typical UI-Controller-transaction management model. The UI layer is common uses jsps, with JSTL libraries along with javascript and Jquery libraries.

There are 2(3) kinds of controllers used all across the OpenClinica application

1. Base controller is SecureController --> This is used widely all across the application and extends SingleThreadedModel (which is deprecated Java Servlet API 2.4). If you are developing a controller, please avoid extending this. As there is no reason to use SingleThreadedModel in OpenClinica application and this would choke up the system resources when many simultaneous requests are made.
2. Base Controller as CoreSecureController--> In order to avoid the performance issues stated above with SecureController, we removed the SingleThreaded model and came up with this servlet controller for data entry module. This can be used as an alternative.
3. The Spring MVC controller--> this would be the best way available in OpenClinica. The suggested approach to avoid all the problems stated above. You can look at rules framework to

see the examples of existing approach.

Similarly on the transaction management side, there are 2 broader approaches:

1. Hibernate-spring transaction model
2. JDBC/PSQL approach

All the legacy code is written in preparedStatements using JDBC. and 90% of the existing classes use this. For example, if you have to deal with any of the objects such as Study or CRF, it would be easier to use the existing beans. However, if you have a new set of tables and have a transaction model attached to it. use the Hibernate/Spring approach. There are already existing implementation for hibernate part of the code,

As you might be aware, we are extensively utilizing javascript libraries for implementing the printable forms module, and this is the preferred paradigm for future development.

Thanks and please do not hesitate to ask questions or seek guidance on the forum.

Setting up Your Development Environment

Clone from GitHub

Clone OpenClinica from https://github.com/OpenClinica/OpenClinica

Developing with the Eclipse IDE

This guide covers the installation and configuration of a development environment using Eclipse, which is the IDE used by the OpenClinica development team. Other IDEs can be used, although their configurations instructions are not covered here.

Prerequisites

Name	Recommended Version	Link
Java Development Kit (JDK)	1.7.x	www.oracle.com
Eclipse	Eclipse IDE for Java EE Developers - Indigo (3.7.1)	www.eclipse.org/downloads
Apache Maven	3.0.x	maven.apache.org/download.html
Apache Tomcat	7.x	tomcat.apache.org/download-60.cgi

Install prerequisites

1. Install the JDK, export the `JAVA_HOME` environment variable to point to the JDK directory;
2. Install Maven, export the `M2_HOME` environment variable to point to the installation directory;
3. Extract the Tomcat installation file.
4. Add the `bin` directories of the JDK and Maven installations to your `PATH` environment variable.

```
# On Windows set JAVA_HOME=<path to the JDK installation>
set M2_HOME=<path to the Maven installation>
set PATH=%JAVA_HOME%bin;%M2_HOME%bin;%PATH%

# On Unix export JAVA_HOME=<path to the JDK installation>
export M2_HOME=<path to the Maven installation>
export PATH=$JAVA_HOME/bin:$M2_HOME/bin:$PATH
```

To check if this configuration is ok, run `mvn -version` in your command prompt.

Build the project

Go to the directory where you checked the code out and run

```
mvn clean install -Dmaven.test.skip=true
```

Configure Eclipse

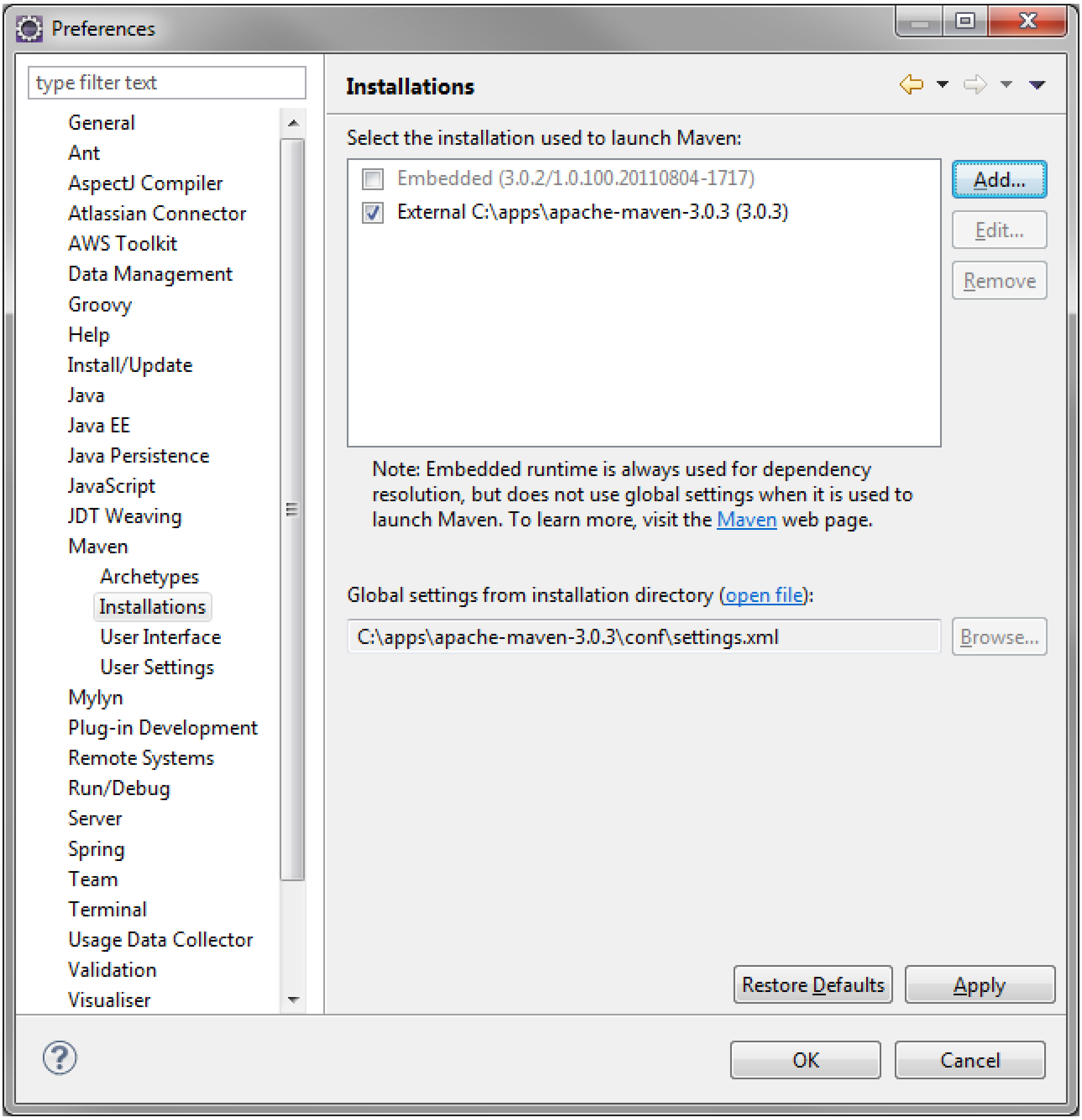
Extract the contents of the Eclipse installation file, then edit the `eclipse.ini` file. Add the `-vm` option to point to the `javaw` executable in the JDK (not the one in the JRE), and change the minimum and maximum memory settings (respectively `-Xms` and `-Xmx`).

```
[...]
-vm
<point to $JAVA_HOME/bin/javaw>
-vmargs
-Dosgi.requiredJavaVersion=1.5
-Xms256m
-Xmx1024m
```

Start Eclipse, create a new workspace and go to `Help > "Install New Software..."`. In the field `"Work with"`, enter the plugin installation URL as listed in the table below. After installing each plugin, you will be prompted to restart your Eclipse. Restart it and install the next plugin in the list.

Plugin name	URL
m2eclipse	download.eclipse.org/technology/m2e/releases
MercurialEclipse	cbes.javaforge.com/update
Spring IDE	dist.springsource.com/release/TOOLS/update/e3.7 - Check only Spring IDE Core (required)

After restarting Eclipse, go to `Window > Preferences`. In the left hand side menu, select `Maven > Installations`, and point to your local Maven installation.



Select the installation used to launch Maven:

□ Embedded (3.0.2/1.0.100.20110804-1717)

☑ External C:\apps\apache-maven-3.0.3 (3.0.3)

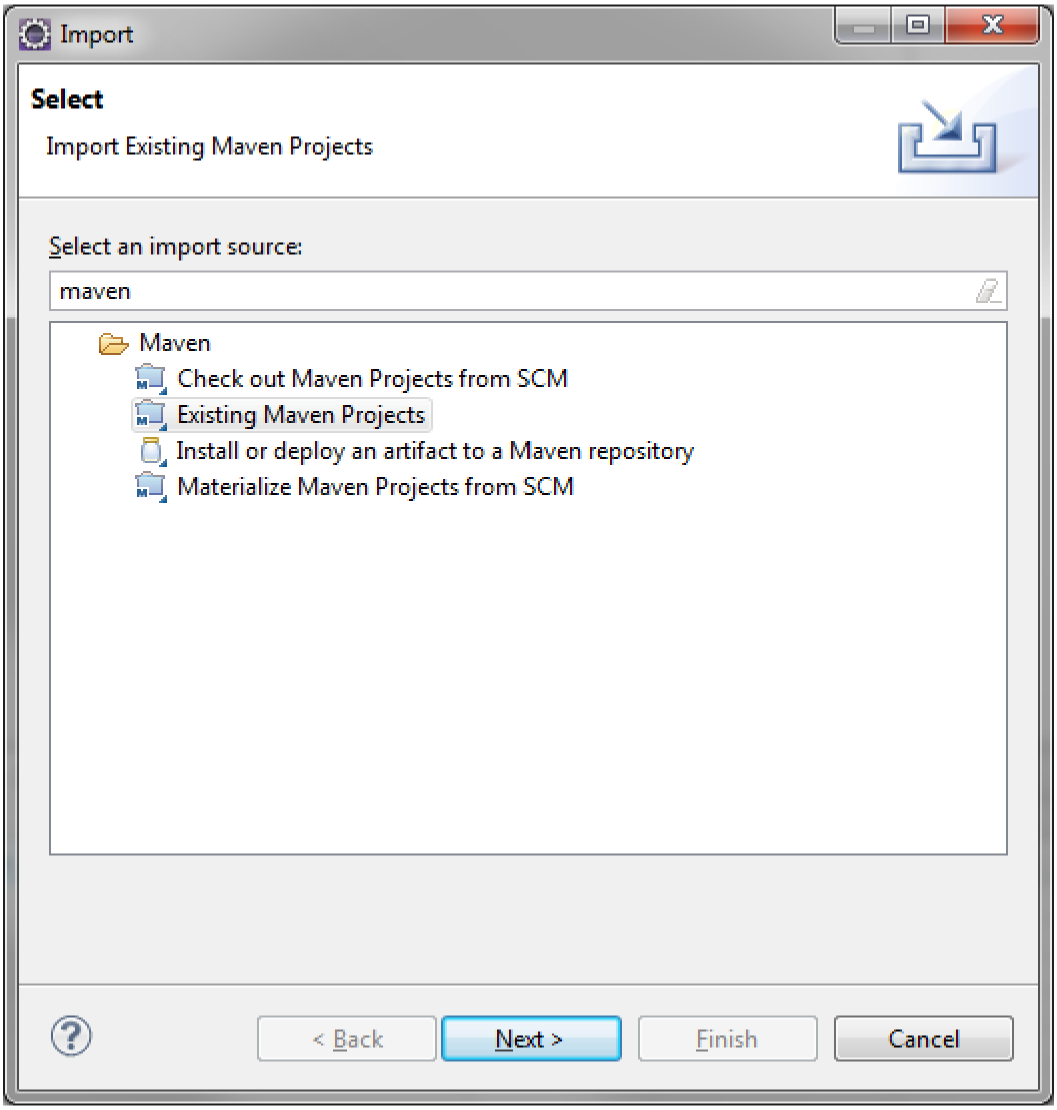
Note: Embedded runtime is always used for dependency resolution, but does not use global settings when it is used to launch Maven. To learn more, visit the Maven web page.

Global settings from installation directory (open file):

C:\apps\apache-maven-3.0.3\conf\settings.xml

Configure the project

Click File > Import > Existing Maven Projects.



Import

Select

Import Existing Maven Projects

Select an import source:

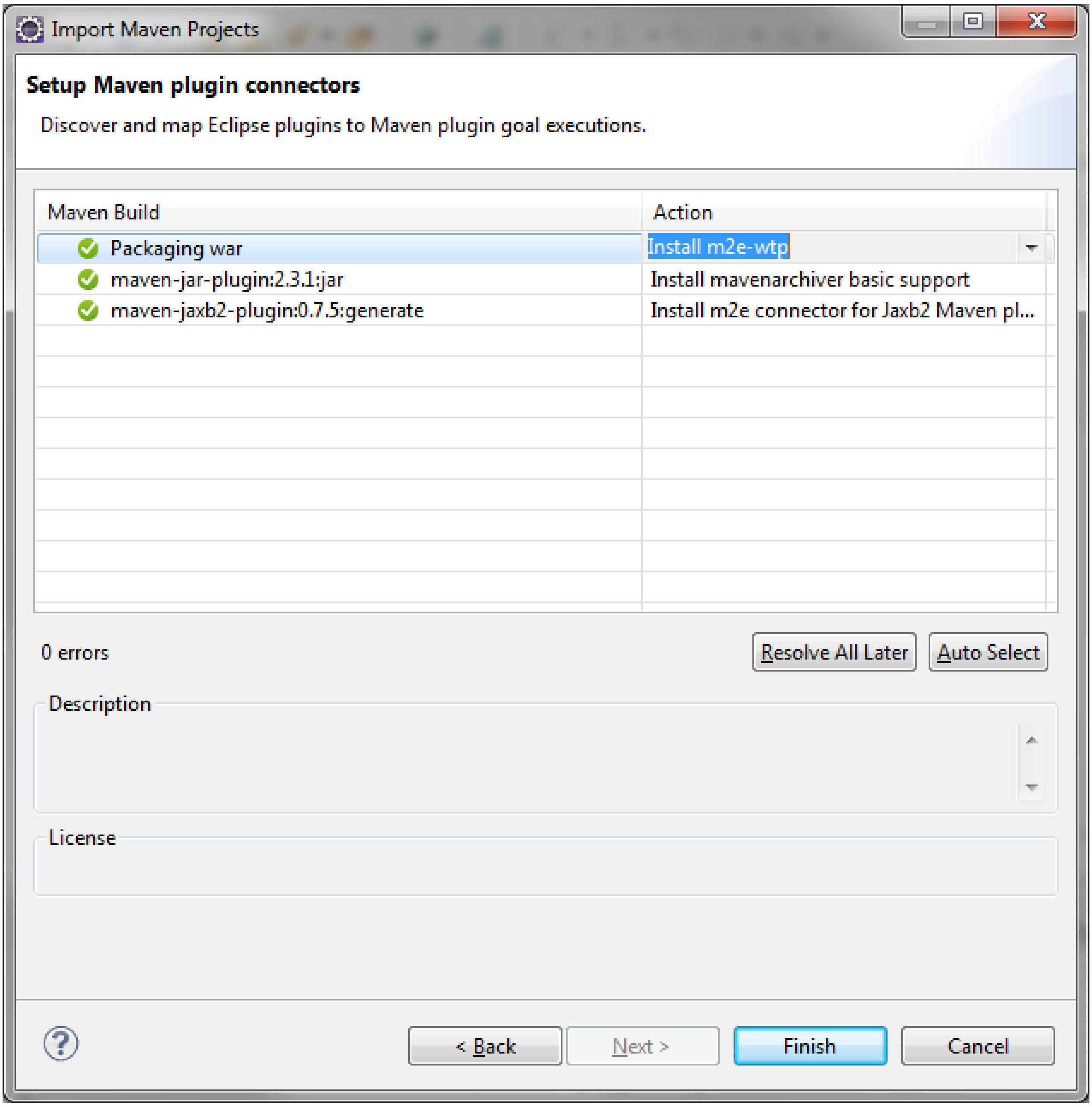
maven

- Maven
  - Check out Maven Projects from SCM
  - Existing Maven Projects
  - Install or deploy an artifact to a Maven repository
  - Materialize Maven Projects from SCM

Point to the directory where the code is, mark all the projects, and select the following plugin connections actions:

Maven Build	Action
Packaging war	Install m2e-wtp
maven-jar-plugin:<version>:jar	Install mavenarchiver basic support
maven-jaxb2-plugin:<version>:generate	Install m2e connector for Jaxb2 Maven plugins

Your configuration should look like in the screenshot below.



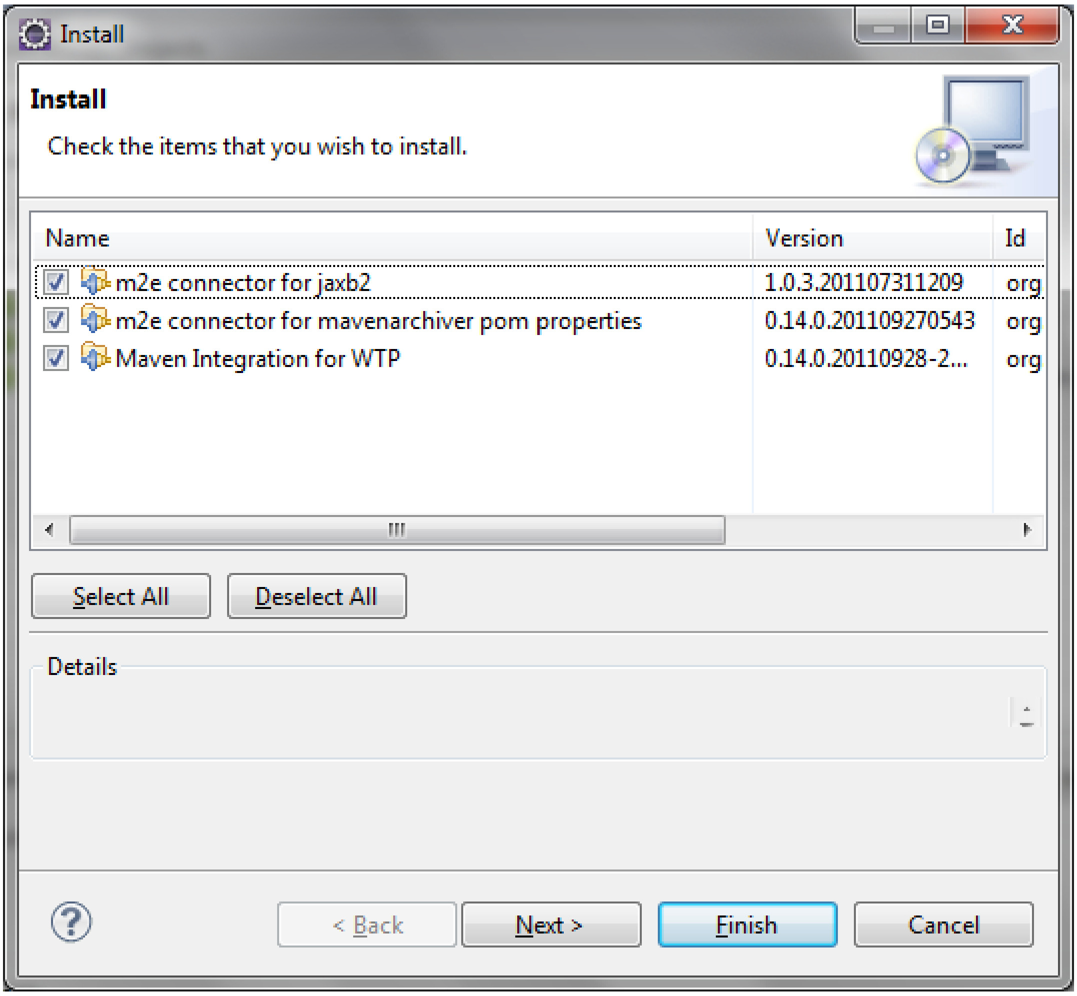
Setup Maven plugin connectors

Discover and map Eclipse plugins to Maven plugin goal executions.

Maven Build	Action
✓ Packaging war	Install m2e-wtp
✓ maven-jar-plugin:2.3.1:jar	Install mavenarchiver basic support
✓ maven-jaxb2-plugin:0.7.5:generate	Install m2e connector for Jaxb2 Maven pl...

0 errors

Eclipse will prompt you to download and install the connectors. Click "Next".



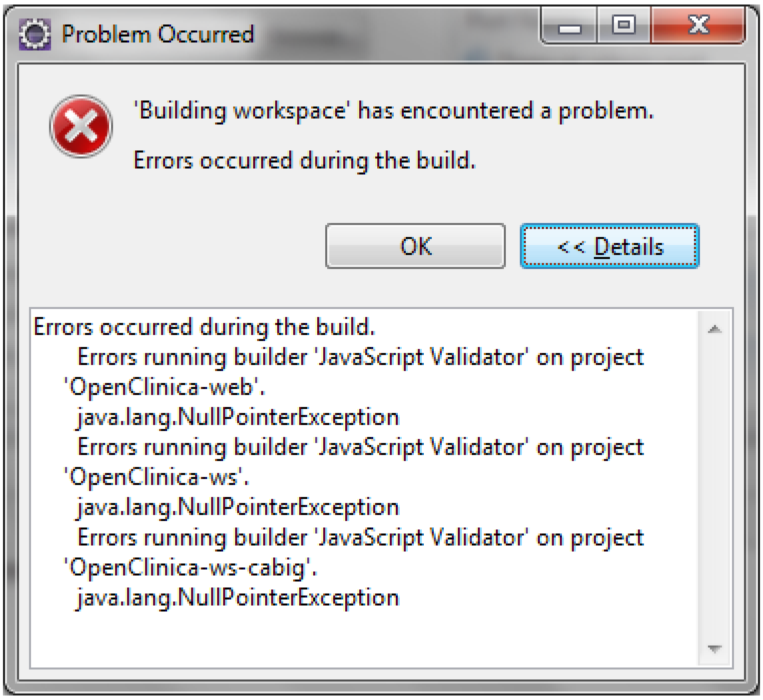
Install

Check the items that you wish to install.

Name	Version	Id
☑ m2e connector for jaxb2	1.0.3.201107311209	org
☑ m2e connector for mavenarchiver pom properties	0.14.0.201109270543	org
☑ Maven Integration for WTP	0.14.0.20110928-2...	org

Confirm the installation of the m2e plugins. Restart Eclipse so the changes will take effect. You should then see the OpenClinica projects configured in your IDE.

Eclipse may show a JavaScript related error message when build your project. This is likely a bug on the IDE or its plugins, but has no impact in the environment setup.

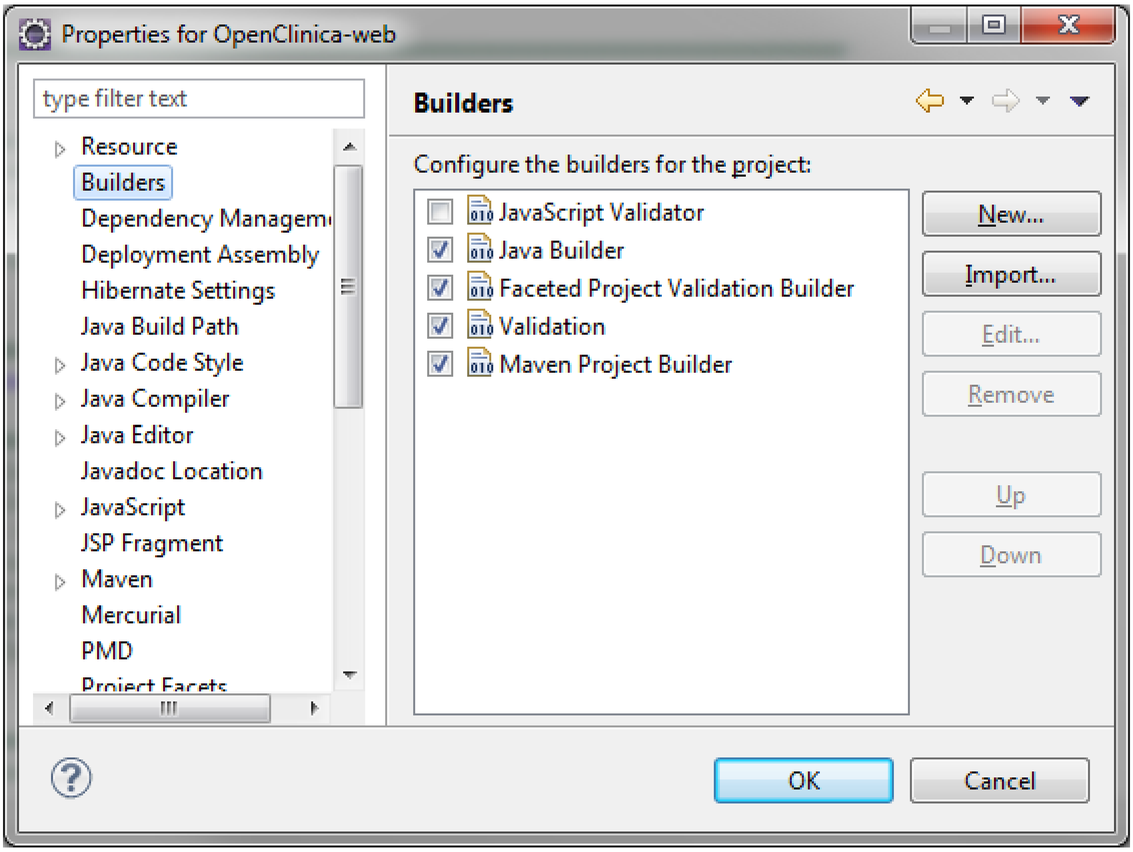


'Building workspace' has encountered a problem.

Errors occurred during the build.

```
Errors occurred during the build.
   Errors running builder 'JavaScript Validator' on project
  'OpenClinica-web'.
   java.lang.NullPointerException
   Errors running builder 'JavaScript Validator' on project
  'OpenClinica-ws'.
   java.lang.NullPointerException
   Errors running builder 'JavaScript Validator' on project
  'OpenClinica-ws-cabig'.
   java.lang.NullPointerException
```

To fix this error message, right-click your OpenClinica-web project and select "Properties". On the left hand side menu, select "Builders". Disable the JavaScript Validator builder for this project. Repeat the procedure to disable it for the ws and ws-cabig projects.

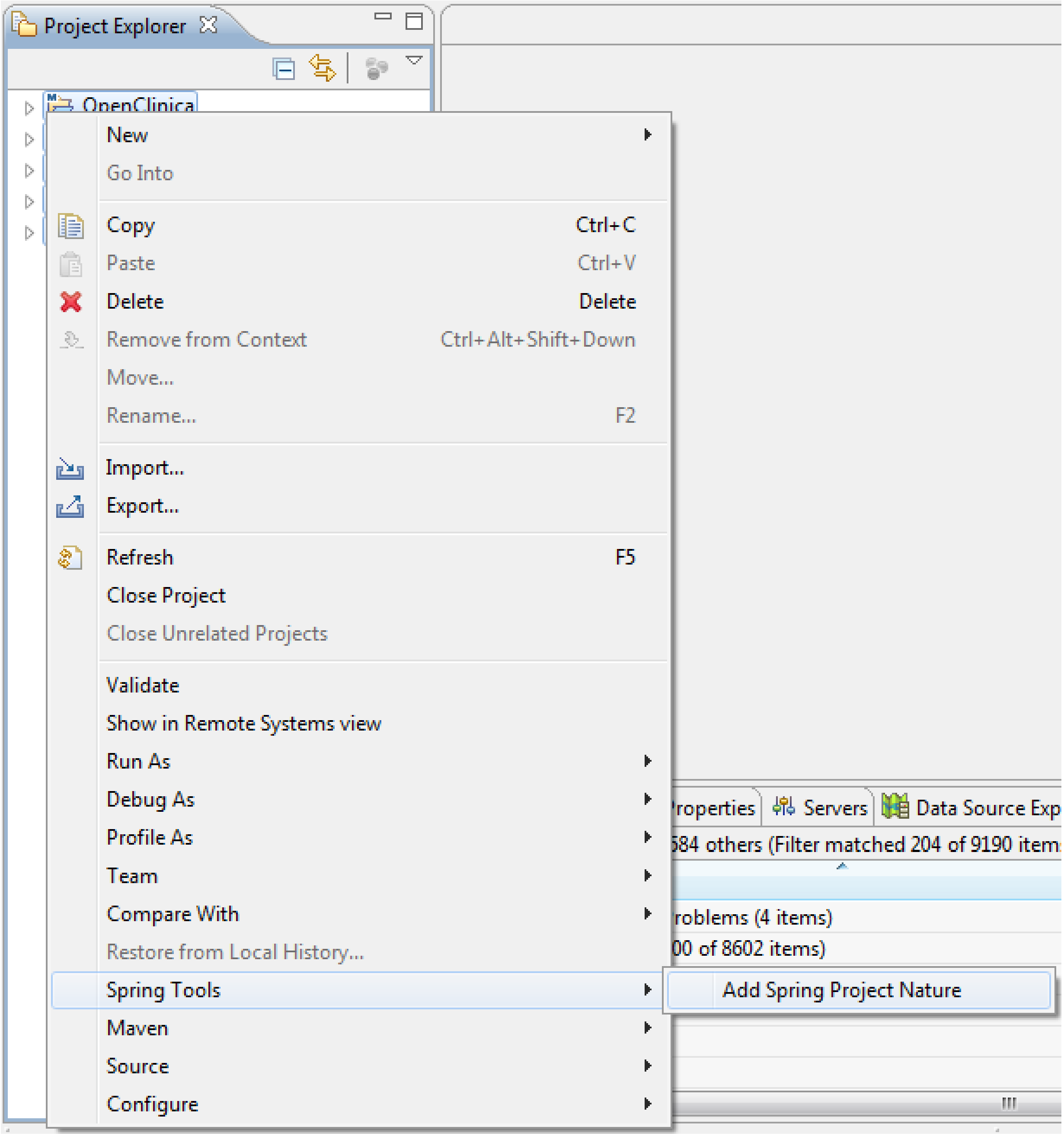


Configure the builders for the project:

□ JavaScript Validator
☑ Java Builder
☑ Faceted Project Validation Builder
☑ Validation
☑ Maven Project Builder

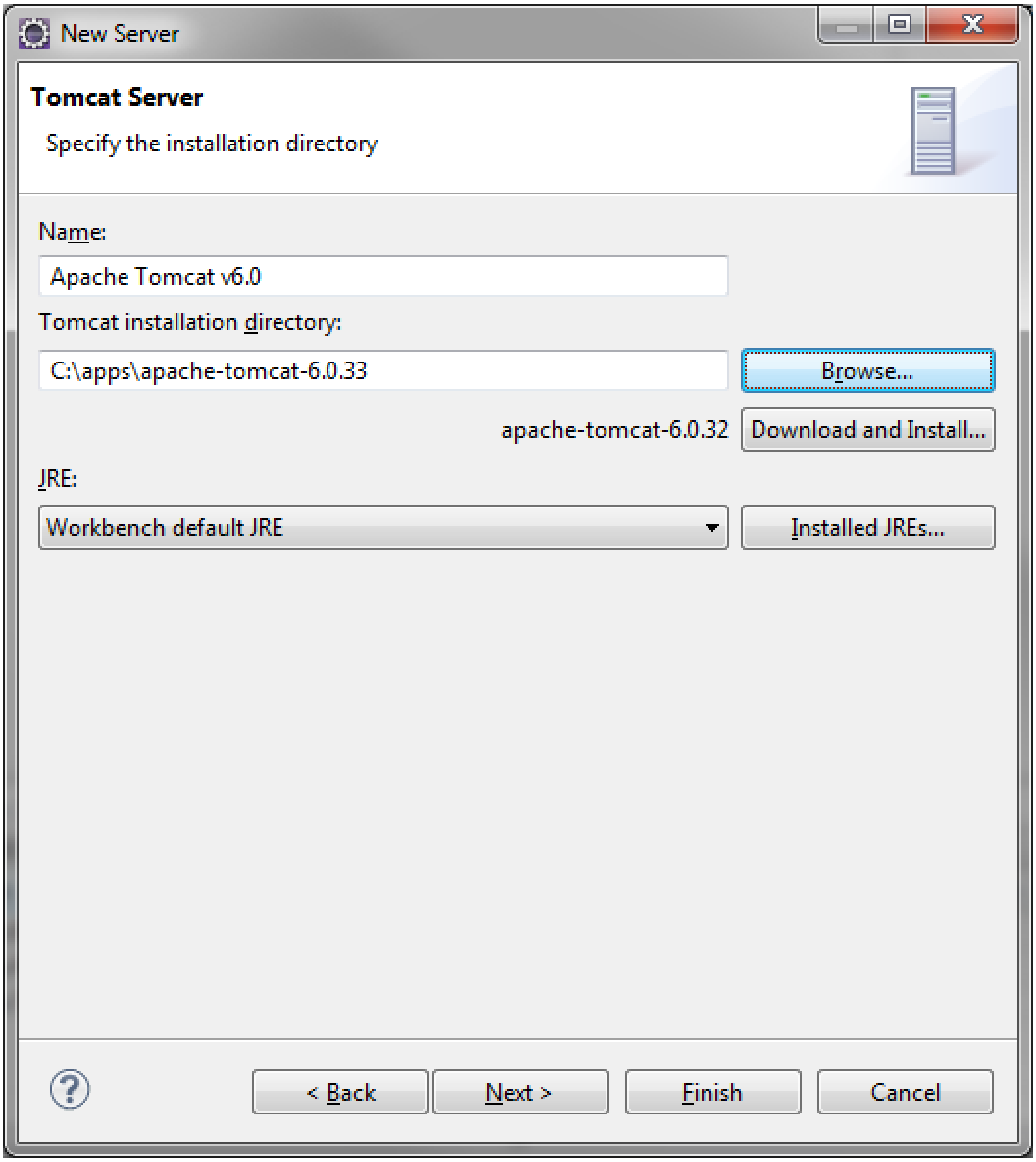
You may see some warnings and errors not related to Java files (e.g., XML validation). To turn those alerts off and to have a faster build, go to "Window" > "Preferences", then under "Validation" check "Suspend all validators".

Enable the Spring plugin by selecting all the projects in the "Project Explorer" view, right click them and select "Spring Tools" > "Add Spring Project Nature".



Create a server

In the "File" menu, select "New", then under "Servers", select "Server". Expand "Apache", click "Tomcat v6.0 Server" and click "Next". Under "Tomcat installation directory", browse to the directory where you extracted Tomcat. Click "Next":



Name: Apache Tomcat v6.0

Tomcat installation directory: C:\apps\apache-tomcat-6.0.33

apache-tomcat-6.0.32

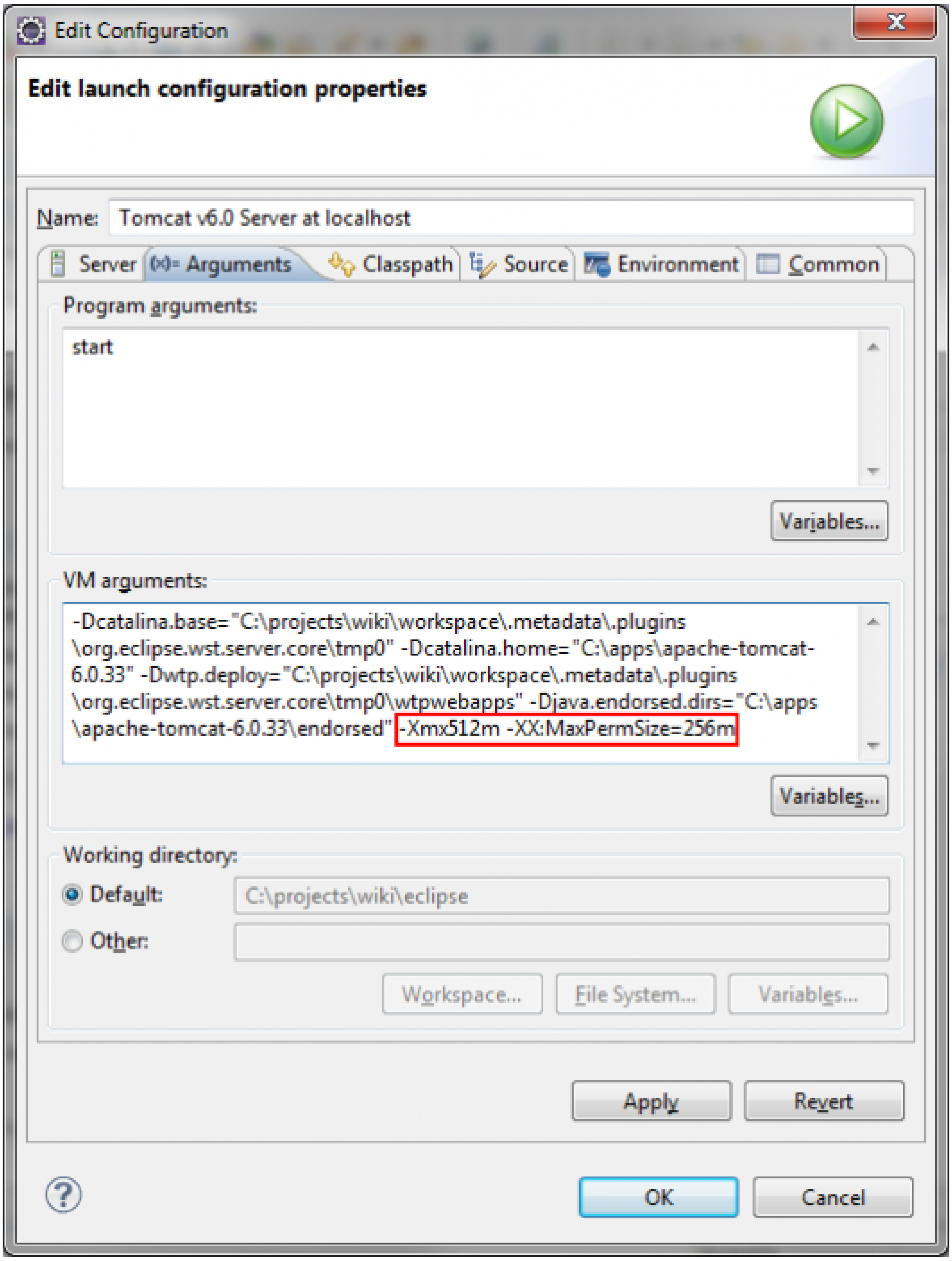
JRE: Workbench default JRE

Select in the list on the right hand side the projects you want to deploy in the server, click "Add >", then "Finish".

Configure the server

Double-click the name of your server in the "Servers" view to open the server configuration screen. Under the "General Information" section, click the link "Open Launch Configuration", then click the tab "Arguments". Add the following properties to the field "VM arguments".

```
-Xmx512m -XX:MaxPermSize=256m
```



Name: Tomcat v6.0 Server at localhost

Program arguments:

```
start
```

VM arguments:

```
-Dcatalina.base="C:\projects\wiki\workspace\.metadata\.plugins
\org.eclipse.wst.server.core\tmp0" -Dcatalina.home="C:\apps\apache-tomcat-
6.0.33" -Dwtp.deploy="C:\projects\wiki\workspace\.metadata\.plugins
\org.eclipse.wst.server.core\tmp0\wtpwebapps" -Djava.endorsed.dirs="C:\apps
\apache-tomcat-6.0.33\endorsed" -Xmx512m -XX:MaxPermSize=256m
```

Working directory:

● Default: C:\projects\wiki\eclipse

○ Other:

Under the "Timeouts" section, set both "Start" and "Stop" to 999. Save the changes to the server by clicking "File" > "Save".

Configure JRebel (Optional)

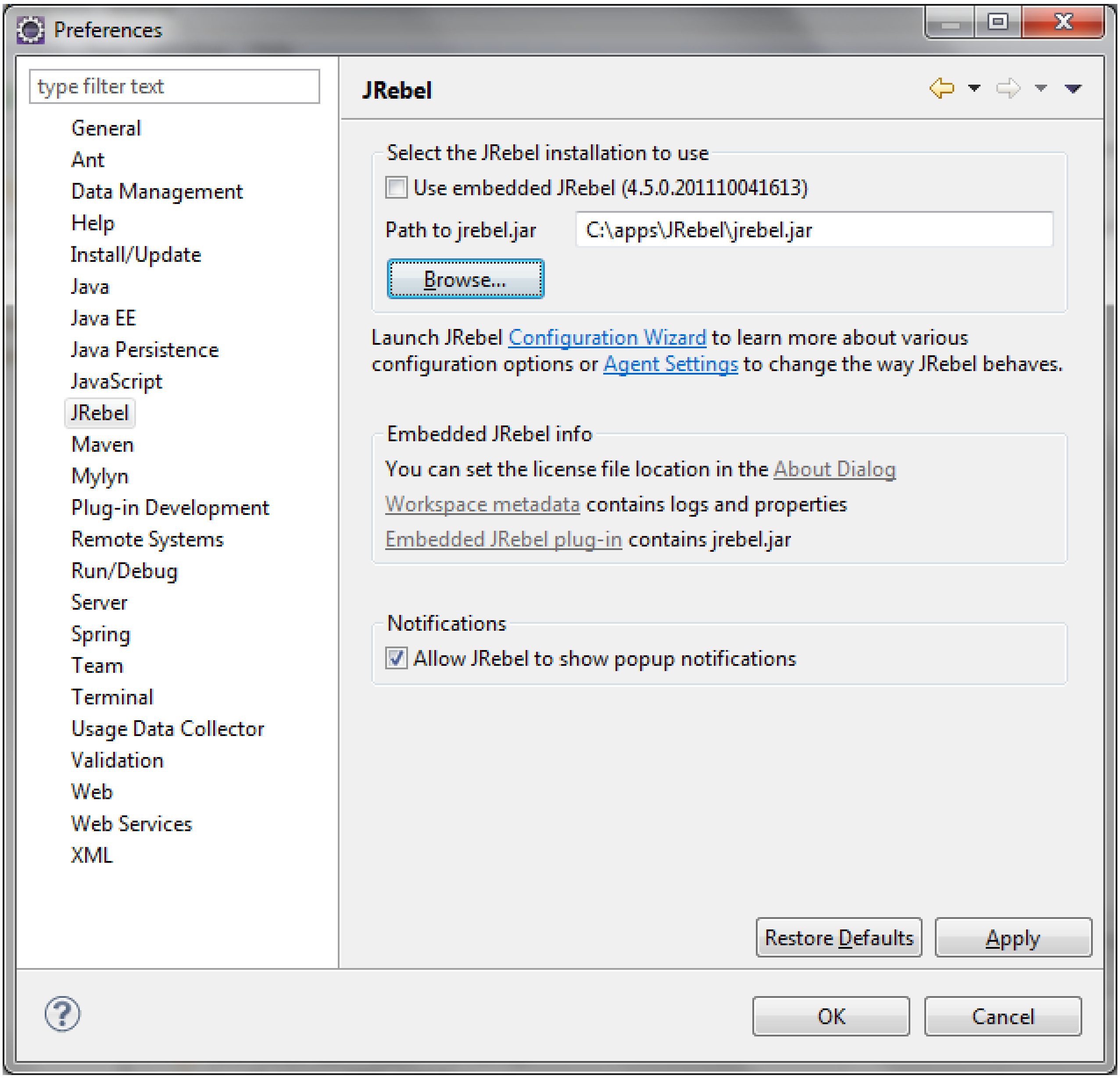
Prerequisites

Download the JRebel installer corresponding to your operating system (not the Eclipse plugin, which will be installed in the next step) from http://zeroturnaround.com/jrebel/current/. Copy the license file (.lic) to the same directory where you installed JRebel. This license file must be in the same directory as the jrebel.jar file. JRebel will work in trial mode without the license file.

Configure Eclipse with the JRebel Eclipse plugin.

Download the JRebel Eclipse plugin from http://www.zeroturnaround.com/update-site. Select all the items under the first group "JRebel".

Restart Eclipse when prompted. The JRebel Configuration Wizard screen should appear after Eclipse starts. Click "Cancel". Go to "Window" > "Preferences", and under "JRebel", uncheck "Use embedded JRebel". Click "Browse" and locate the jrebel.jar file in the JRebels installation directory.



JRebel

Select the JRebel installation to use

□ Use embedded JRebel (4.5.0.201110041613)

Path to jrebel.jar C:\apps\JRebel\jrebel.jar

Launch JRebel Configuration Wizard to learn more about various configuration options or Agent Settings to change the way JRebel behaves.

Embedded JRebel info

You can set the license file location in the About Dialog

Workspace metadata contains logs and properties

Embedded JRebel plug-in contains jrebel.jar

Notifications

☑ Allow JRebel to show popup notifications

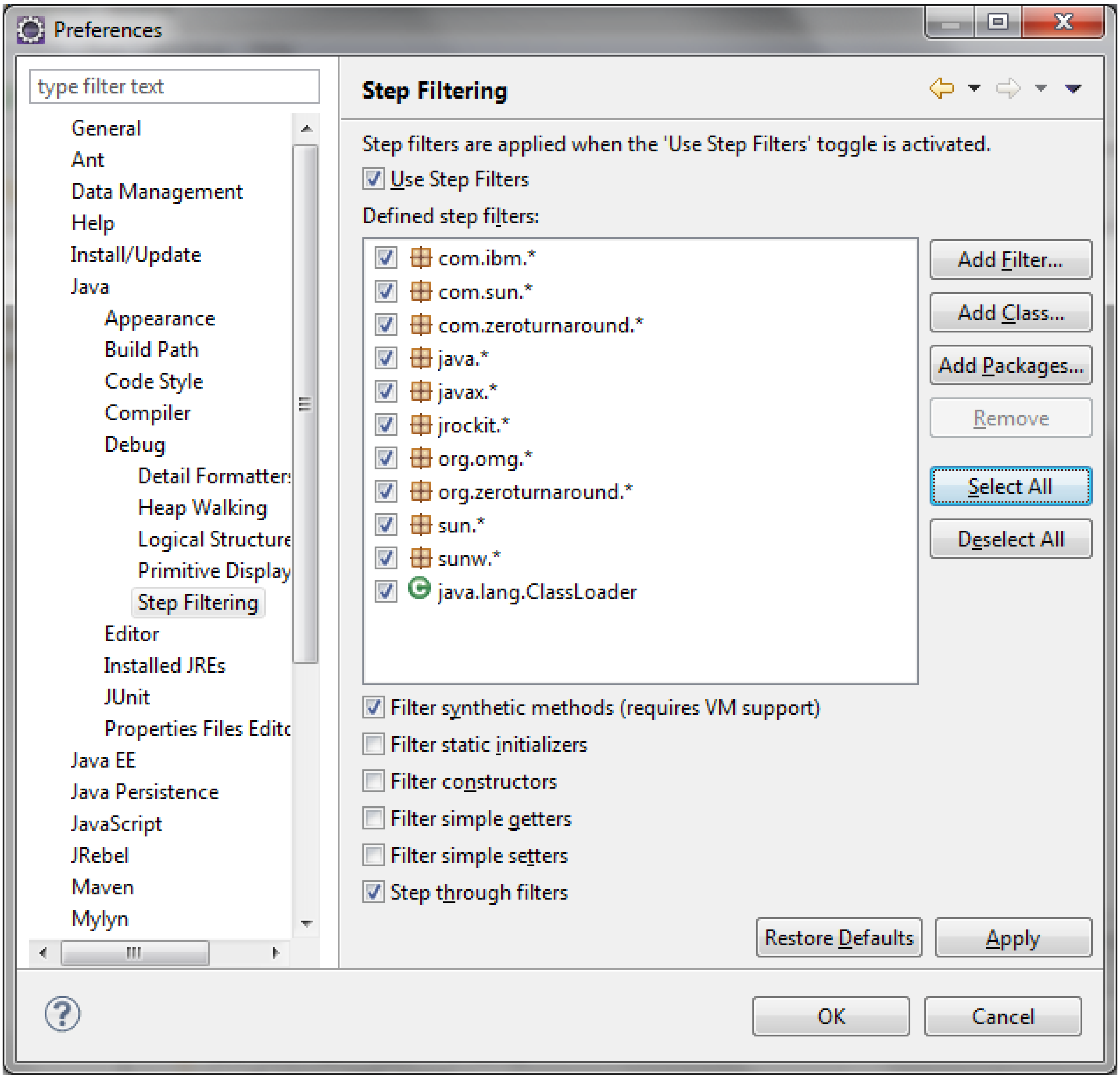
On the left hand side, select "Java" > "Debug" > "Step filtering". Check "Use Step Filters", "Filter synthetic methods" and "Step through filters".

Use the "Add Filter..." button to add the following filters:

```
com.zeroturnaround.*
```

```
org.zeroturnaround.*
```

Click "Select All" to check all filters in the filters list. Your configuration should look like the screenshot below.



Step Filtering

Step filters are applied when the 'Use Step Filters' toggle is activated.

☑ Use Step Filters

Defined step filters:

- ☑ com.ibm.*
- ☑ com.sun.*
- ☑ com.zeroturnaround.*
- ☑ java.*
- ☑ javax.*
- ☑ jrockit.*
- ☑ org.omg.*
- ☑ org.zeroturnaround.*
- ☑ sun.*
- ☑ sunw.*
- ☑ java.lang.ClassLoader

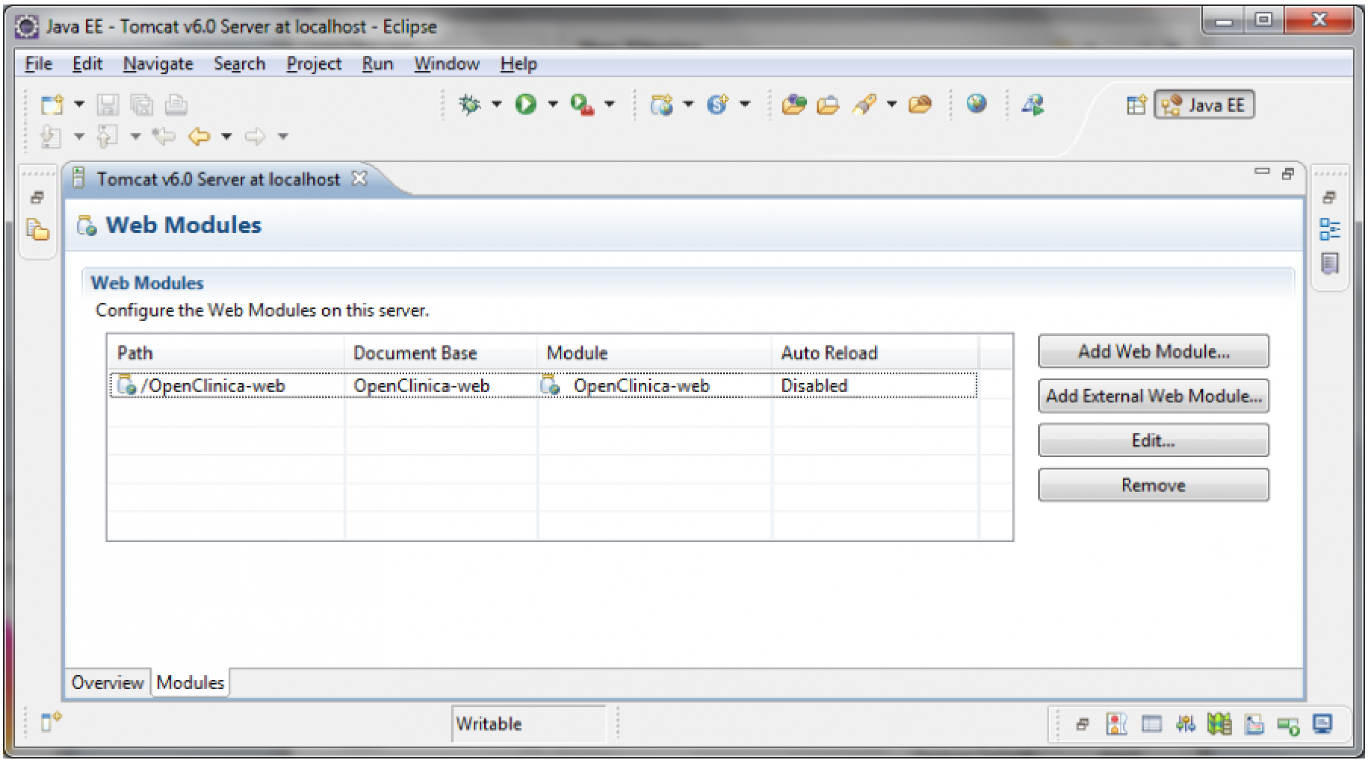
☑ Filter synthetic methods (requires VM support)
☐ Filter static initializers
☐ Filter constructors
☐ Filter simple getters
☐ Filter simple setters
☑ Step through filters

Click "Ok" to close the preferences window. Ensure that "Project" > "Build automatically" is checked.

Configure server for JRebel

Double-click the entry in the "Servers" view corresponding to your Tomcat server to open the server configuration page. Under the section "JRebel Integration", check "Enable JRebel agent".

On the bottom-left corner of the server configuration page, click the tab "Modules". A list of web modules deployed to the server is displayed. For each deployed module, select it in the list and click "Edit...". Uncheck the box "Auto reloading enabled". Once all the modules are configured, click "File" > "Save" to save the server configuration. Your modules configuration should look like the screenshot below.


Java EE - Tomcat v6.0 Server at localhost - Eclipse
File
Edit
Navigate
Search
Project
Run
Window
Help
Java EE
Tomcat v6.0 Server at localhost
Add Web Module...
Add External Web Module...
Edit...
Remove
Overview
Modules
Writable

Web Modules

Web Modules

Configure the Web Modules on this server.

Path	Document Base	Module	Auto Reload
/OpenClinica-web	OpenClinica-web	OpenClinica-web	Disabled

Configure Maven for JRebel

Create or edit your ~/.m2/settings.xml file, where ~ corresponds to the users home directory. If you dont have one, copy it from the $M2_HOME/conf/settings.xml. Add the following XML fragment within the <settings> tag of your file:

```
<profiles> <!-- This tag may already exist in your file -->
...
<profile>
    <id>JRebel</id>
    <properties>
        <useJRebel>true</useJRebel>
    </properties>
</profile>
...
</profiles>


<activeProfiles> <!-- This tag may already exist in your file -->
    <activeProfile>JRebel</activeProfile>
</activeProfiles>
```

Open a command prompt and build your project with the command mvn clean install -Dmaven.test.skip=true. Refresh the project in Eclipse.

Check the JRebel configuration

Start your server in the "Servers" view of Eclipse. Once the server is started, switch to the "Console" view. Scroll to the very beginning of the log file and make sure JRebels header is there.

```
#############################################################

 JRebel 4.5.1 (201110191452)

 (c) Copyright ZeroTurnaround OU, Estonia, Tartu.

[...]

#############################################################
```

The log file should also describe files monitored or changed by JRebel

```
JRebel-Spring: Monitoring Spring bean definitions in [...]
```

You can now change Java classes or JSP files and have JRebel redeploying them without restarting the server.

Customize Properties With a Build Profile

A development filter file can be used to override the default properties during development. This filter is applied only to the developers environment and has no effect on other developers, instances or distribution packages.

Un-comment out the filters on lines 545-548 in the top-level "pom.xml". Otherwise, developer-specific Maven build profile instructions will not work.

Create a Filter File

A filter file can be used to create an OpenClinica package with configuration values different from the default.

For example, here are the steps required to create a sandbox configuration.

- Create a file named sandbox.properties in the source/main/filters directories of all modules. There is a file named default.properties in these directories that can be used as a template. Or, use an empty file and define on it only the values that are different from the ones defined in default.propertes (i.e., if a property is not specified in sandbox.properties, the value defined in default.properties will be used).
- Make sure the filter file was created for all OpenClinica modules (core, web, and ws). The build will fail if it cannot find the filter file for a module.
- Build OpenClinica using the sandbox configuration in the command line, by typing:

```
mvn clean package -Dconfig.id=sandbox
```

Now, create a filter file following the example above. Just dont build the project with Maven yet, as a slightly different configuration should be used. For the file name, follow the dev-<username>-<optional_identifier>.properties convention (e.g., the user John Doe could create a file named dev-jdoe.properties or dev-jdoe-quickfix.properties).

Its important to create the filter file for all Maven modules (core, web, and ws) otherwise the Maven build will fail.

Build from source

Use this file to override properties defined in the default.properties file during the build. Its possible to copy all the content on the default.properties file to it, but thats not recommendable - copy only the properties that will be changed, as all missing properties will still be taken from default.properties.

Create a Maven Build Profile

A Maven build profile will add the filter to the build process. Open the Mavens user settings file (normally located on ${user.home}/.m2/settings.xml, where ${user.home} correspond to the users home directory) and add create a profile tag within the profiles tag of the file, following the example below (make sure the value of the <config.id> tag matches the name of the filter file created previously):

```
    <profile>

        <id>config-local</id>

        <activation>

            <activeByDefault>true</activeByDefault>

        </activation>

        <properties>

            <config.id>dev-jdoe</config.id>

        </properties>

    </profile>
```

Build the Project

Build the whole project using

```
mvn clean install
```

in the root project level. Notice that the -Dconfig.id parameter should not be used here, as this configuration was already defined in the build profile.

Optional step: Check the properties files in the target directories to make sure the properties were replaced according to the defined filter.

Restart Eclipse (to reload the profile configuration), refresh the project files and perform a clean build. Start Tomcat and the changes made to the filter file should be reflected in the application.

Switch to another configuration

The steps to switch to a different configuration are:

1. Create the filter files for all modules
2. Change the value of the <config.id> property in the settings.xml file
3. Restart Eclipse, if open - Eclipse seems to read the settings.xml file only during startup

To use the default configuration (i.e., all values read from default.properties only), simply comment the whole config-local profile in the settings.xml file.

Development Tools

Mercurial Source Control

- Mercurial PowerPoint Slides - by Doug Rodrigues, OpenClinica, 16-Nov-2011
- TortoiseHg
- MercurialEclipse
- MacHG
- HgInit: Subversion Re-education - Explains the conceptual differences between Subversion and Mercurial)
- Mercurial SCM - Mercurials home page. Contains very interesting information in the user guide and wiki.
- Mercurial Cheat Sheet

11.1 Using the OpenRosa API in OpenClinica (experimental)

Starting with version 3.5, OpenClinica will begin to support the OpenRosa API, which will let you run Enketo, ODK Collect, or any of a number of OpenRosa-compliant data capture clients. If you're not familiar with Enketo, ODK, or OpenRosa, here's a primer. You can also see this blog post on how OpenClinica LLC is using Enketo and the OpenRosa API in its products.

To try it yourself:

1. Build/install the latest OpenClinica code from github or distros
2. Install an OpenRosa client, such as enketo or ODK Collect
3. Add the property 'PortalURL' to datainfo.properties with the URL of the OpenRosa Client

   ```
   PortalURL = http://www.example.com
   ```

4. Issue a /formList request from the client to the OpenClinica:

   ```
   GET /OpenClinica-web/rest2/openrosa/{studyOID}/formList
   ```

5. Currently supported methods include

   ```
   GET /OpenClinica-web/rest2/openrosa/{studyOID}/formList
   GET /OpenClinica-web/rest2/openrosa/{studyOID}/formXml
   POST /OpenClinica-web/rest2/openrosa/{studyOID}/submission
   ```

Note: These APIs are still experimental, may not work at all, and are certainly not suitable for production use. OpenClinica Participate uses them but the hosted environment includes tools to ensure network security and robustness for production use that are not in the enketo package. We aim to eventually support OpenRosa API as part of the standard OpenClinica API and welcome feedback, testing, and code contriutions. In particular, this is still experimental because:

- Form submission will not really work because it does not create a study subject and study event where the data can go. This will be added in the near future.
- The API is not particularly secure. There is no production-quality authentication mechanism, other than to secure network access to known safe clients.

Alternative CRF design model using XForm

Starting with OpenClinica 3.8, you can use an alternative model for CRF Design based on the OpenRosa XForm specification, instead of the spreadsheet-based OpenClinica CRF Template. To enable:

- Configure your OpenClinica datainfo.properties to activate the Xform feature
  - If you will be uploading images, the default size limit for total images uploaded is 5MB
  - If you will be uploading a large number of images or particularly large images, you should add the following to datainfo.properties:
    - pformMaxSubmissionSize=100000000
    - This maximum submission size setting is a safe bet to cover most image upload needs (this specific setting allowed for 15 files of 4.72MB each to be loaded. Feel free to adjust the number as needed to meet your needs.)
- When adding a CRF or CRF version, select the 'Upload as Xform' tab
- Paste your OpenRosa-compliant XForm code into the textarea (you can use an OpenRosa-compliant form design tool such as XLSForm to generate the XML from a spreadsheet)
- Upload any media files (images, videos, audio, etc) that you want to be embedded in the form

The primary use case right now is to support having images & video embedded in participant forms. It also enables you to use a growing list of features available in the enketo form engine that are not available in the traditional OpenClinica CRF engine. Your CRFs should still function in the traditional OpenClinica CRF engine, though they may not look as pretty or support all the features of a typical OpenClinica CRF because only a minimal amount of required metadata is parsed from the XForm into the OpenClinica form metadata model. Most of the display/layout information as well as any edit checks and skip logic is left only in the XForm.

Functional approval by Laura Keita. Signed on 2016-05-12 3:37PM

Approved for publication by Cal Collins. Signed on 2016-05-13 8:13AM

Not valid unless obtained from the OpenClinica document management system on the day of use.