OPEN Oncology Patient Enrollment Network

CTSU Cancer Trials Support Unit
A SERVICE OF THE NATIONAL CANCER INSTITUTE
Linking practice to progress

Cancer Trials Support Unit

CTSU – A Service of the National Cancer Institute

OPEN RandoNode Request Processing Interface (Updates for RandoNode Version 2.0)

Revision 1

19 November 2009

Document Information

Revision Information for the **Cancer Trials Support Unit** - **OPEN RandoNode Request Processing Interface (Updates for RandoNode Version 2.0)**
Document No**. CTSU/OPEN/DOC## Rev. 1**

Revision History			
#	**Date**	**By**	**Description**
01	31-Jan-2002	Sushmita De	Initial version
Last Saved By **Sushmita De** on **11/19/2009 11:58:00 AM**			
File Location: **C:\CTSU\Sandbox\apps\OPEN\docs\RandoNode\RandoNode_API_2.0.doc.docx** VSS Location: **$/Sandbox/apps/OPEN/docs/RandoNode/RandoNode_API_2.0**			

This document was prepared for:

This document was prepared by:

WESTAT
1650 Research Boulevard
Rockville, Maryland 20850 Phone: (301) 251-1500

Sushmita De *(301) 738-8311*

Contents

Tables

References

#	Document	Location	Description
1.	OPEN RandoNode Request Processing Interface (RandoNode_API_1.0)	https://www.ctsu.org/open/default.asp?fName=open/Randonode/docs	Original RandoNode Request Processing interface document
2.	OPEN RandoNode Request Processing Interface (RandoNode_API_1.1)	https://www.ctsu.org/open/Randonode/docs/RandoNode_API_1.1.pdf	RandoNode 1.1 API version. Used mainly for patient population and duplicate patient verification.
3.	Embedded and Ancillary Study Support In OPEN	VSS	Original document for Embedded and ancillary study support in OPEN. Includes requirements and high level design.

1. Introduction

1.1 Overview

The objective of this document is to recommend the design changes required in RandoNode version 2.0 to support the embedded & ancillary studies, Credentialing Exception Override and transferring Case Notes to RandoNode.

1.2 Target Audience

This document is intended for the development team at the Cooperative Groups implementing and extending the RandoNode for processing patient enrollments under the OPEN portal.

1.3 Requirements

1. **Enrollment of patients to embedded and standalone ancillary studies**:
   1.1. With OPEN 2.0, users will have the ability to register embedded or standalone ancillary studies to a main (treatment) study at the time of registration as well as within x number of days after the registration to the main study. These ancillary studies may be optional or mandatory embedded or standalone ancillary studies.
   1.2. In OPEN, when a main study (treatment study) is configured to contain embedded or standalone ancillary studies, all enrollment data will be submitted to the RandoNode as a single *doRegister()* transaction. This means that during the main study enrollment, if the patient is enrolling to any ancillary studies (embedded or standalone) the patient data along with the eligibility data for the main study and all ancillary studies will be submitted to the RandoNode through a single *doRegister()* web method call. This way, the RandoNode can accept or reject the enrollment using an ALL or NONE criteria.

   This requirement was originated due to the fact that there can be mandatory ancillary studies. If there are mandatory ancillary studies, the registration to the main study must be rolled back at the Group if the patient is found to be ineligible to register to a mandatory ancillary study. Group needs to indicate a failure in response if either the main study or any mandatory ancillary study fails to be enrolled. The single transaction approach will enable rolling back the main study in the event that the patient is found to be ineligible to participate in a mandatory ancillary study.
   1.3. The odmData XML container, metadata XML and OPEN ⇔ RandoNode interface objects will be modified to support the new data transfer requirements to transfer the ancillary study information to the RandoNode.
   1.4. These changes will be backward compatible so that none of the existing RandoNodes will be affected. Only those RandoNodes implementing embedded and ancillary studies will need to process the additional information sent by OPEN. The existing RandoNodes can decide their own timeframe to upgrade their implementation to support embedded and standalone ancillary studies.
   1.5. RandoNode implementation will provide new APIs to support the retrieval of questions and answers based on the ancillary study name.
2. **Case Notes:** OPEN credentialing screen contains a field Case Notes, where registrar can enter additional comments. OPEN will be transferring this data to RandoNode.
3. **Credentialing Exception Override:** OPEN will provide user the capability to override credentialing exceptions. OPEN will transfer this override information to Group RandoNode and RandoNode will decide whether the requested override is valid or not.

2. Changes To RandoNode Enrollment Workflow For Ancillary Studies

2.1 Requirements

With the introduction of embedded and standalone ancillary study support, there will be a few changes in the enrollment workflow followed by `doRegister()` method.

Currently the RandoNode workflow involves the following four steps as shown in Figure 1:

1. Perform duplicate patient check.
2. Validate EC data.
3. Enroll the patient.
4. Return the Patient ID and Treatment Arm to OPEN.

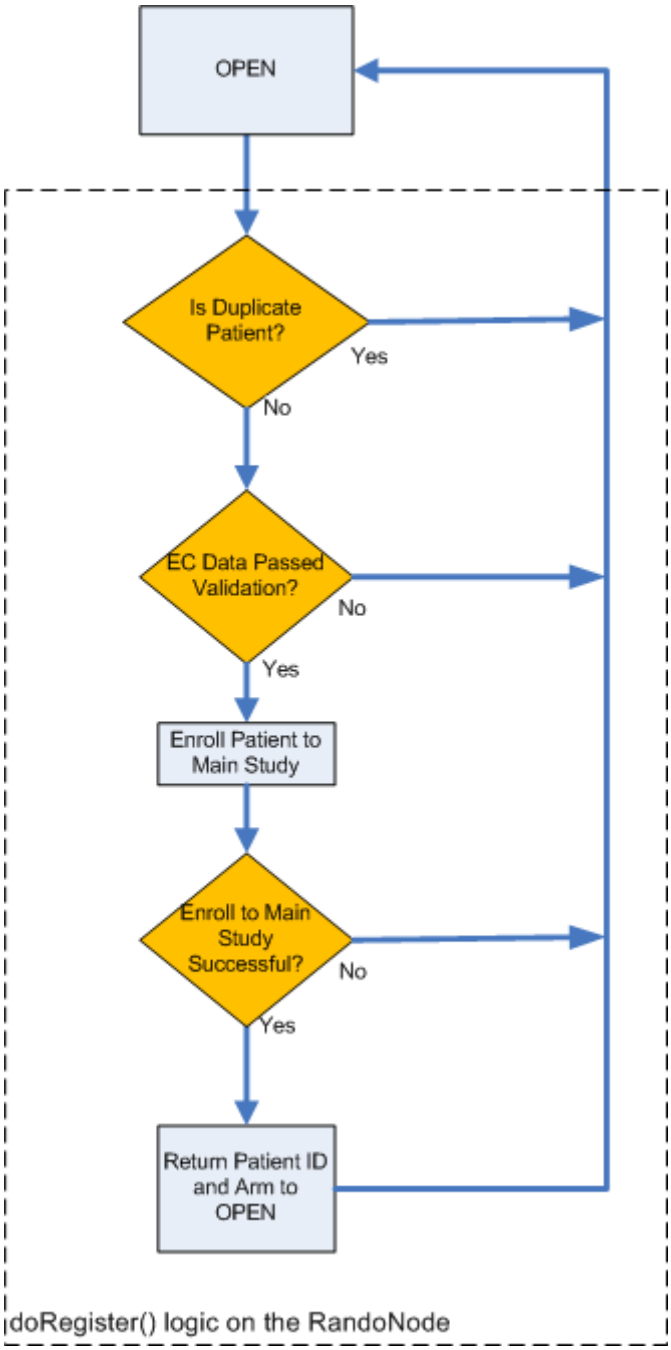

Figure 1: Existing *doRegister()* processing sequence by the RandoNode

With the introduction of embedded and standalone ancillary studies, this workflow will change to the following as shown in Figure 2:

1. Perform duplicate patient check.
2. Validate EC data.
3. Enroll the patient to the treatment study.
4. Enroll the patient to companion studies.

5. If enrollment is a success for main study and mandatory ancillary studies, return the Patient ID(s) and Treatment Arm(s) to OPEN. Enrollment is a failure otherwise.

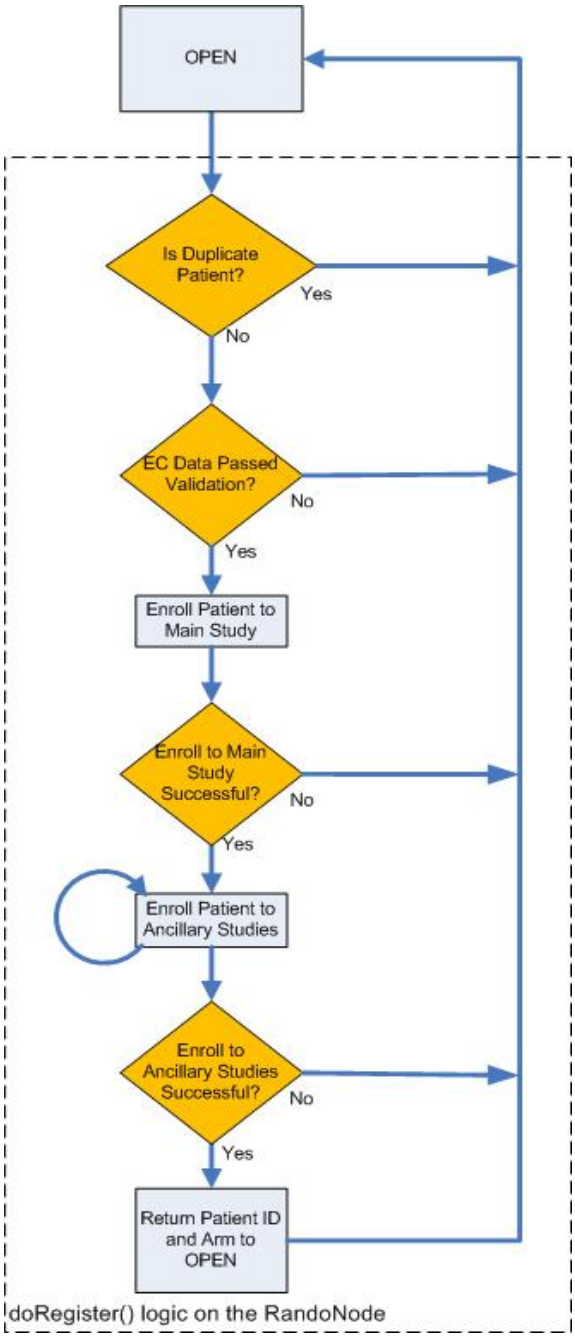

Figure 2: Modified *doRegister()* processing sequence for ancillary studies

3. Credentialing Exception Override Changes

OPEN will send the credentialing exception code and credentialing exception reason attributes to RandoNode within `OpenRegistration` object. RandoNode needs to verify this exception code against their database to verify the success of the credentialing.

4. Change Details

The following are the summary of changes for RandoNode version 2.0.

4.1 Changes To Form Metadata

Since embedded or standalone ancillary study questions will be attached to the main study EC form, modules need to be marked with the study name within meta data. MetaData xml file will be modified to tag the *ItemGroupDef* (aka question module).

The "*Purpose*" attribute of the *ItemGroupDef* will be used for tagging the *ItemGroupData*. The study ID(RSS protocol number) will be used as the tag.

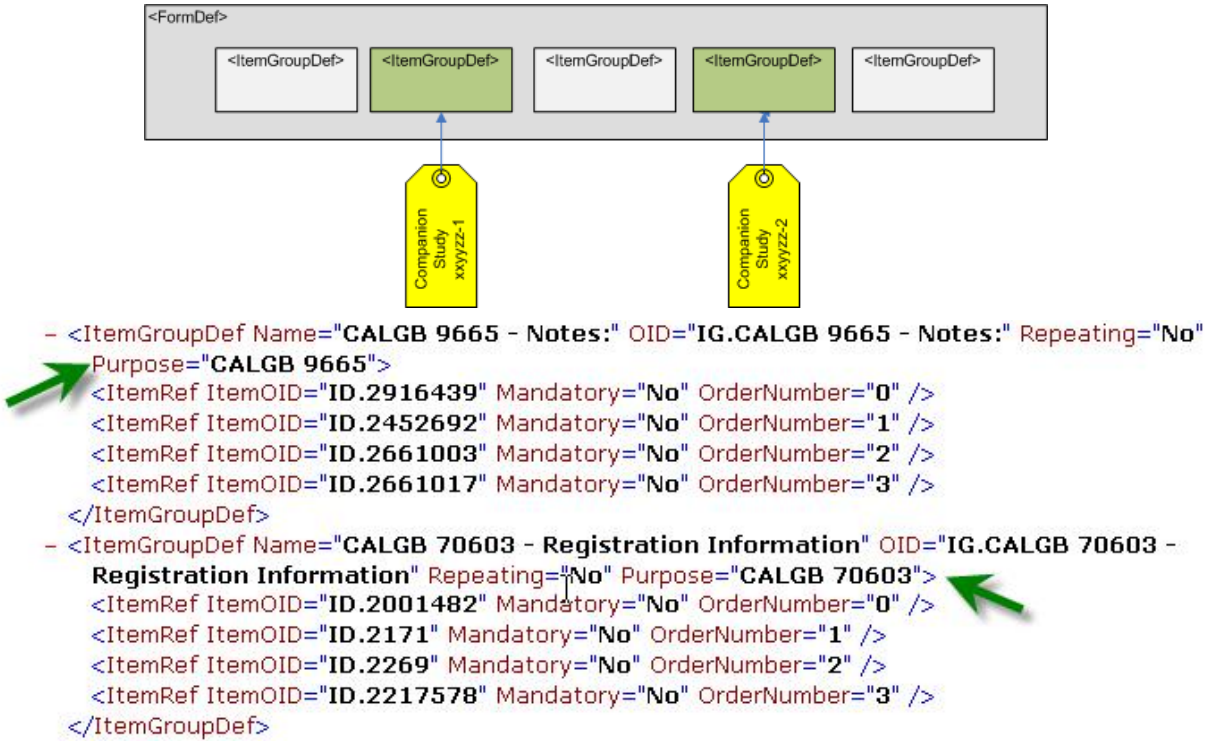

Figure 3: Modified Meta Data

4.2 Attribute Changes To Existing Classes

4.2.1 Changes To OpenRegistration Class

4.2.1.1 OpenRegistration New Attributes

Existing *OpenRegistration* object will be modified to accommodate the new requirements. A new attribute *action* will be introduced to indicate the action required by RandoNode for a study. *OpenRegistration* additionally will contain a nested array of *OpenRegistration* objects named as *ancillaryRegistrationArray*. Each entry of *ancillaryRegistrationArray* object will indicate the registration detail of an ancillary study. There will be additional changes to transfer data for credentialing override and case notes. New attributes of the *OpenRegistration* object are listed in the source code below.

```
public class OpenRegistration{
                  :
                  :
private String groupProtocolNumber; /* Group specific protocol numbers
                                       specially for ancillary studies */
private String credentialingExceptionCode; /* Exception code provided by
                                users to override credential exceptions */
private String credentialingExceptionReason; /* Credential exception
                                              reason provided by user */
private String caseNotes;          /* Case notes entered by registrar */
private String action;              /* Action needs to be taken */
private OpenRegistration[] ancillaryRegistrationArray;/* OpenRegistration
                                        array for ancillary studies */
}
```

Code 1: *OpenRegistration* new attributes

The details of the new attributes with the possible values are listed in the table below.

Attribute Name	Data Type	Description	Example Values
action	characters No specific length limit.	Action needs to be taken by the RandoNode for credentialing or enrollment.	Action should be one of these four valid values: **DO_NOT_PROCESS** – Do not process the registration data this flag belongs to. In case of a delayed ancillary study enrollment, since OPEN will be sending even associated registered enrollment data along with the enrollment needs to be processed, this flag will indicate that the main registration and already registered ancillaries should not be processed. **PROCESS –** This flag indicates that the corresponding enrollment needs to be processed. Used in *doCredential(), dovalidate()* methods, not used in *doRegister()* method. **ENROLL** – Request to enroll a patient. This value applies only to *doRegister ()* method. **WITHDRAW** – This flag is not supported as of now, is a future place holder. Indicates to withdraw a patient. This value applies only to *doRegister* () method for ancillary studies. Currently OPEN does not support withdrawing patient from ancillary study. Going forward this feature may need to be implemented depending on the Groups' requirements.
ancillaryRegistrationArray	OpenRegistration	This attribute is an array of OpenRegistration objects. Each entry of the array contains data for an associated ancillary study.	This array elements are existing OpenRegistration objects. This array will contain null in request if the corresponding study does not have an ancillary study.
groupProtocolNumber	v50	This is the group specific protocol	Contains alphanumeric characters. Group protocol number may not be available

		number.	for all the protocols. Assigned by the Group especially to the ancillary studies which do not have PIO number. This attribute is populated with the value "NULL" when do not have a Group specific number.
credentialingException Code	V256	Override code for credentialing exception.	Value will be "NULL" in case credentialing override is not required.
credentialingException Reason	V1000	Override reason for credentialing exception.	Value will be "NULL" in case credentialing override is not required.
caseNotes	4000	Notes or comments entered by a registrar for a specific enrollment.	Value will be "NULL" in case registrar does not enter a note.

Table 1: *OpenRegistration* New Attributes

4.2.1.2 Changes To OpenRegistration Status

Main *OpenRegistration* object will indicate the enrollment status of main study as well as of the mandatory ancillary studies. Main study *status* attribute should indicate *FAILURE* in case main study or any of the mandatory ancillary studies does not meet enrollment criteria. Optional ancillary enrollment status will be indicated by the attributes contained within the individual nested *OpenRegistration* objects.

Figure 4: Modified *OpenRegistration* class

4.2.2 Changes To OpenRequest.operation Attribute

`OpenRequest` will be modified to indicate a new valid value for operation attribute. A new operation value PROCESS_ANCILLARY_STUDY will be introduced for the `doRegister()` and `doCredential`() methods to indicate that this is a delayed enrollment and main study does not need to be processed within the list of studies present within request data. All other times, including when main study and ancillary studies are enrolled together, operation tag value will indicate REGISTER_PATIENT for `doRegister`() and DO_CREDENTIAL for `doCredential`() method.

4.3 Changes To Existing Methods

4.3.1 Changes To doRegister() Web Method

When the RandoNode receives the `doRegister()` method call, existing implementation validates data for the main study and determines the eligibility. From RandoNode 2.0 onwards, RandoNode will first verify if the `OpenRegistration.ancillaryRegistrationArray` contains any elements. In case, the main study contains ancillary study records and the action is ENROLL, ancillary study data should be validated (and randomized if required) as well. The `doRegister()` workflow changes has already been

discussed in section 2 of this document.

4.3.2 Changes To doCredential() Web Method

The `doCredential`() method needs to be updated to verify the credentialing for each ancillary registrations. This is an opportunity for the Groups for separate ancillary credentialing. Example code for the same is provided within the RandoNode starter kit.

4.3.3 Changes To doValidate() Web Method

The `doValidate`() method does not require any changes for demography validation. Checklist data validation should be updated to include ancillary data validation as well. Example code for the same is provided within the RandoNode starter kit.

4.4 New Utility Methods

Few new utility methods were introduced in ClinicalDataUtil.java and MetaDataUtil.java. Please see the detail below.

API Name	Class	Description
getStudiesByItem	MetaDataUtil.java	Retrieves the list of study names for a given Item oid.
getStudyByItemItemGroup	MetaDataUtil.java	Retrieves the study name for a given Item oid and ItemGroup oid.
getStudyByItemGroup	MetaDataUtil.java	Retrieves the study name for a given item group oid.
getItemsByStudy	MetaDataUtil.java	Retrieves the list of Items by Study.
getItemsByItemGroupStudy	MetaDataUtil.java	Retrieves the list of Items by ItemGroup Oid and Study.
getItemsByItemOidItemGroupStudy	MetaDataUtil.java	Retrieves the list of Items by Item Oid, ItemGroup Oid and Study.
getItemDataListByItemItemGroupStudy	ClinicalDataUtil.java	Retrieves ItemData by Item Oid, ItemGroup Oid and Study.
getItemGroupsByStudy	MetaDataUtil.java	Retrieves ItemGroups by Study.
getAncillaryStudies	ClinicalDataUtil.java	Retrieves ancillary studies present within a request.
getAncillaryStudiesToProcess	ClinicalDataUtil.java	Retrieves the list of ancillary studies need to be processed.
getActionByAncillaryStudy	ClinicalDataUtil.java	Retrieves the action needs to be taken for a given ancillary study.

Table 2: New Utility methods

4.5 How To Install/Upgrade To RandoNode 2.0.0.0 Build

Please refer to the document readme.doc distributed with RandoNode 2.0.0.0 installation package for installation/up gradation guide. Readme.doc document can be located within the RandoNode.Zip distributed to groups.