

openCRX Admin Guide

Version 1.10.0



www.opencrx.org

28-Feb-2007 @ 09:42:07 AM

License

The contents of this file are subject to a BSD license (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.opencrx.org/license.htm>

Copyright 2007 © CRIXP Corp. All rights reserved.



Table of Contents

1	About this Book.....	7
1.1	Who this book is for.....	7
1.2	What do you need to understand this book.....	7
1.3	Tips, Warnings, etc.....	7
2	Prerequisites.....	8
3	Security.....	9
3.1	Introduction.....	9
3.1.1	Basic Concepts and Conventions.....	9
3.1.2	Permissions / Access Control.....	12
3.2	Default Settings.....	14
3.3	Activating Security.....	16
3.4	Security Settings of New Objects.....	16
3.5	Login Procedure.....	17
3.5.1	Application Server Login.....	17
3.5.2	Segment Login.....	17
3.5.3	Disabling Login.....	17
3.6	Resetting Security.....	18
4	Managing Users.....	19
4.1	Creating Users.....	19
4.1.1	Create Users Manually.....	20
4.1.2	Import Subjects and Application Login Principals.....	21
4.1.3	Import Users.....	22
4.2	Disable/Deactivate Users.....	23
4.2.1	Disabling Users at the level Application Server.....	23
4.2.2	Disabling Users at the level openCRX.....	23
5	Deployment Scenarios.....	24
5.1	Typical Deployment Scenarios.....	24
5.2	Multi Entity Deployment Scenarios.....	25
5.2.1	Multiple Data Segments in a single DB.....	25
5.2.2	Multiple DBs.....	26
5.2.3	Multiple Applications.....	26
5.3	openCRX Custom Applications.....	26
6	Workflow Controller.....	27
6.1	Workflow Controller Configuration.....	29
6.1.1	Startup Configuration in web.xml.....	29
6.1.2	ServerURL.....	30
6.1.3	Handler pingrate and autostart.....	30
6.2	Servlet SubscriptionHandler.....	31
6.3	Servlet WorkflowHandler.....	32
6.4	MailImporterServlet.....	33

6.5	Trouble Shooting Servlets.....	34
7	Subscribe / Notify Services.....	35
7.1	Example Subscription – Activity Modifications.....	37
7.2	Example Subscription with Filtering.....	39
7.3	Trouble Shooting Notification Services.....	40
8	E-mail Services.....	41
8.1	Import E-mails from a Mail Client into openCRX.....	42
8.2	Use openCRX as an E-mail Archive/Audit Tool.....	43
8.3	Send E-mail directly from openCRX.....	44
8.4	Export E-mails from openCRX to your Mail Client.....	45
8.5	Install and Configure Mail Resource and E-Mail Services.....	47
8.5.1	Installation of JavaMail and JAF.....	47
8.5.2	Create mail-service.xml.....	47
8.5.3	Mail Resource in web.xml and jboss-web.xml.....	49
8.5.4	Outbound E-mail.....	50
8.5.5	Inbound E-mail.....	53
8.5.6	Trouble Shooting E-mail Services.....	57
9	Data Import/Export.....	59
9.1	Importing Data into openCRX.....	59
9.2	Importing XML Files.....	59
9.2.1	Importing vCard Files (→ openCRX Contacts).....	62
9.2.2	Importing iCalendar Files (→ openCRX Meetings).....	62
9.2.3	Other Options.....	63
9.3	Exporting Data from openCRX.....	63
9.3.1	Exporting XML Files.....	63
9.3.2	Exporting openCRX Contacts (→ vCard Files).....	64
9.3.3	Exporting openCRX Contacts (→ Outlook Contacts).....	65
9.3.4	Exporting openCRX Meetings (→ iCalendar Files).....	66
9.3.5	Other Options.....	66
10	Customizing openCRX.....	67
10.1	Managing Locales.....	67
10.2	Managing Packages.....	68
10.3	Role-based UI.....	69
10.3.1	Model Permissions.....	69
10.3.2	Multiple Applications.....	69
10.3.3	Custom Layout JSPs.....	69
11	Integration with Office Suites.....	70
11.1	MS Office.....	70
11.1.1	MS Word.....	70
11.1.2	MS Excel.....	70
11.1.3	MS Outlook.....	70
11.2	Open Office.....	70
11.2.1	OpenOffice Writer.....	70
11.2.2	OpenOffice Calc.....	70

12 Next Steps.....71**List of Figures**

Figure 1: Security Realms, Principals and Subjects after Initial Setup.....	10
Figure 2: Segment Administration.....	11
Figure 3: Role Drop Down with list of available Segment Login Principals.....	11
Figure 4: openCRX UML Model – Class Diagram SecureObject.....	12
Figure 5: System attributes of an openCRX object as shown in the GUI.....	13
Figure 6: Table kernel_Segment after default installation (QuickStart).....	15
Figure 7: Table kernel_Segment after modification.....	15
Figure 8: Role Drop Down with list of available Segment Login Principals.....	17
Figure 9: Operation Actions > Import Login Principals (admin-Root).....	21
Figure 10: Operation Actions > Import Users (admin-Standard).....	22
Figure 11: Disabling of Segment Login Principal guest by admin-Standard....	23
Figure 12: 3-Tier with Application Server.....	24
Figure 13: 4-Tier with Application Server.....	24
Figure 14: 4-Tier with Clustered Application Servers.....	24
Figure 15: 4-Tier with Servlet Engine.....	24
Figure 16: Multiple Data Segments in a single DB.....	25
Figure 17: Dedicated DB for each Entity.....	26
Figure 18: Accessing the openCRX Workflow Controller.....	27
Figure 19: openCRX 1.10.0 Workflow Controller.....	27
Figure 20: Default Configuration of WorkflowController.....	28
Figure 21: openCRX Administration – WorkflowController.....	29
Figure 22: Workflow Controller Configuration – serverURL.....	30
Figure 23: Workflow Controller Configuration – pingrate and autostart.....	30
Figure 24: Default Workflow Processes created by WorkflowHandler.....	32
Figure 25: openCRX MailImporterServlet.....	33
Figure 26: MailImporterServlet – Configuration.....	33
Figure 27: Event and Notification Service.....	35
Figure 28: Standard Topics included in the openCRX distribution.....	36
Figure 29: Create a new Subscription – step 1.....	37
Figure 30: Create a new Subscription – step 2.....	37
Figure 31: Create a new Subscription – step 3.....	38
Figure 32: Create a new Subscription – step 4.....	38
Figure 33: Create a Subscription with Filters.....	39
Figure 34: Flow of messages between openCRX, mail server and mail client..	41
Figure 35: Import E-Mails from Mail Client.....	42
Figure 36: E-Mail Audit – import all inbound/outbound e-mail messages.....	43

Figure 37: Send E-Mail from openCRX – Overview.....	44
Figure 38: Send E-Mail from openCRX with Actions > Follow Up.....	44
Figure 39: Export E-Mail from openCRX – Overview.....	45
Figure 40: Export E-Mail from openCRX with Actions > Follow Up.....	45
Figure 41: Envelope E-mail with exported E-Mail Activity as attachment.....	46
Figure 42: MS Outlook – Resend This Message to edit/send e-mail.....	46
Figure 43: MS Outlook – Send message.....	47
Figure 44: Create a new E-Mail Account – step 1.....	50
Figure 45: Create a new E-Mail Account – step 2.....	50
Figure 46: Create a new E-Mail Account – step 3.....	51
Figure 47: E-mail subject prefix and Web access URL.....	51
Figure 48: Example of outbound E-mail Action Log Entries.....	52
Figure 49: Envelope E-Mail with attached E-Mail to be imported.....	53
Figure 50: Activity Tracker E-Mail is created automatically.....	54
Figure 51: Activity Tracker E-Mail with newly imported e-mail.....	54
Figure 52: Newly imported e-mail.....	55
Figure 53: Activity Creator Default E-mail Creator.....	56
Figure 54: XML import from 3rd party system – overview.....	60
Figure 55: Interactive import of XML Files.....	60
Figure 56: Interactive import of XML Files.....	61
Figure 57: Operation vCard Import.....	62
Figure 58: Exporting Contact as XML File.....	63
Figure 59: XML Exporter provides XML data file and code tables as ZIP file....	64
Figure 60: Export Contact as vCard.....	64
Figure 61: Export Contact to MS Outlook.....	65
Figure 62: MS Outlook Contact created from openCRX Contact.....	65
Figure 63: Exporting Meeting / Sales Visit as iCalendar File.....	66

List of Listings

Listing 1: File Format Subjects and Application Login Principals.....	21
Listing 2: Example File Subjects and Application Login Principals.....	21
Listing 3: File Format Users.....	22
Listing 4: Example File Users.....	22
Listing 5: web.xml – auto startup of the Workflow Controller.....	29
Listing 6: Iterate all auditees and check for new audit entries.....	31
Listing 7: Mark Audit Entries as visited by Subscription Handler.....	31
Listing 8: Servlets managed by Workflow Controller log to server.log.....	34
Listing 9: File mail-service.xml.....	48
Listing 10: Uncomment mail resource definition in web.xml.....	49

Listing 11: Uncomment mail resource definition in jboss-web.xml.....	49
Listing 12: Importing certificate into keystore cacerts.....	49
Listing 13: Debug Output of MailImporterServlet.....	53
Listing 14: Importing Certificate.....	57
Listing 15: Locales in web.xml.....	67
Listing 16: Activating/Deactivating Locales in web.xml.....	67
Listing 17: Locales in web.xml.....	68
Listing 18: Activating/Deactivating Locales in web.xml.....	68

1 About this Book

This book describes various configuration settings and tasks an openCRX administrator should know about.

openCRX is the leading enterprise-class open source CRM suite. openCRX is based on openMDX, an open source MDA framework based on the OMG's model driven architecture (MDA) standards. This guarantees total openness, standards compliance, a state-of-the-art component-based architecture, and virtually unlimited scalability.

1.1 Who this book is for

The intended audience are openCRX administrators.

1.2 What do you need to understand this book

This book describes some of the settings and configurations an openCRX administrator can use to control the behavior of openCRX.

1.3 Tips, Warnings, etc.

We make use the following pictograms:



Information provided as a "Tip" might be helpful for various reasons: time savings, risk reduction, etc. - it goes without saying that we advise to follow our guides meticulously

meticulous \muh-TIK-yuh-luhs\, *adjective*:
Extremely or excessively careful about details.



You should carefully read information marked with "Important". Ignoring such information is typically not a good idea.



Warnings should not be ignored (risk of data loss, etc.)

2 Prerequisites

This guide assumes that you have access to a properly installed instance of openCRX 1.10.0 (please refer to the QuickStart guide available from <http://www.opencrx.org/documents.htm> for installation instructions).

3 Security

In this chapter we will present a high-level overview of openCRX security and discuss a few select issues.



We do **not** recommend learning about security with mission critical data. **Backup your data before** you make changes if you are not certain what the consequences are! The risk of you being locked out is real and the **resources required to fix broken security settings can not be overestimated!**

The default settings should work for virtually all users; the probability of getting yourself into trouble by changing default settings should not be underestimated. **Read and understand at least the basics of openCRX security BEFORE you make any changes.**

3.1 Introduction

3.1.1 Basic Concepts and Conventions

- Each “real user” is represented by a **Subject** (e.g. “Guest”). Subjects are managed by the openCRX Root administrator (**admin-Root**).
- Each subject has an **Application Login Principal** (also called **login id**). Application login principals are managed by the openCRX Root administrator (**admin-Root**).
- Each application login principal is assigned to a subject (e.g. principal “guest” is assigned to subject “Guest”) and allows a “real user” to login.
- A “real user” can have one or more additional segment login principals. A **Segment Login Principal** has typically the same name as the application login principal (e.g. “guest”) and grants a “real user” login access to a segment. Segment login principals are managed by the openCRX segment administrators (e.g. **admin-Standard** for the Segment Standard).
- Each “real user” which has access to a segment (i.e. has a segment login principal) has (in addition to the segment login principal) a segment user principal, e.g. “guest.User”. The **Segment User Principal** is required to assign objects to an **Owning User**.
- Each segment has a corresponding realm to manage Principals:
 - The application login principals are stored in the realm **Default**.
 - The segment login principals for segment *<segment name>* are stored in the realm *<segment name>* (e.g. principals for the segment *Standard* are stored in the realm *Standard*).
 - Each segment has a segment administrator principal (*admin-*<segment name>**) (e.g. **admin-Standard** for the segment **Standard**).

The following figure shows the situation after the initial setup of openCRX (assuming you worked through the QuickStart guide):

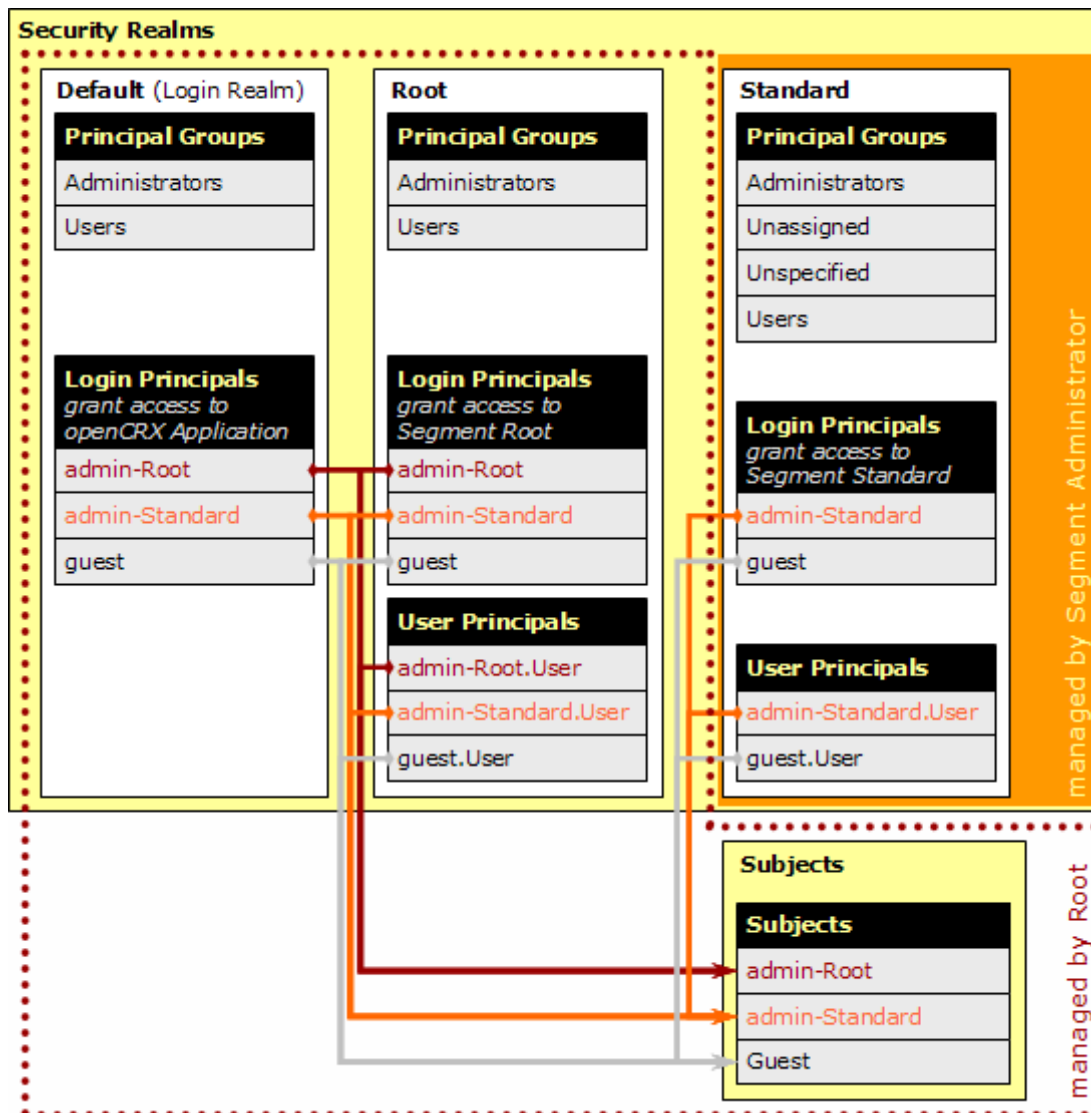


Figure 1: Security Realms, Principals and Subjects after Initial Setup

Summarizing the above:

- there is a realm for each segment (e.g. if you followed the QuickStart guide there is a realm **Standard** corresponding to the segment Standard)
- the realm **Default** acts as login realm, i.e. it contains all principals who are allowed to login to the openCRX application; PrincipalGroups in this realm are only used to configure **Granted Roles** by inheritance (in addition to configuring them directly in the appropriate grid).
- there is a subject for each "real user" and all principals of a user are assigned to the same subject; this allows openCRX to find all principals of a user (--> role selection drop down)

The segment administrator (e.g. **admin-Standard**) creates principals and User Homepages with the operation createUser():

- Each segment login principal has a home page in the corresponding segment (qualifier of principal and home page must match!).
- Each segment login principal is correlated with a contact. This correlation is e.g. required to find all activities and contracts assigned to the logged in principal.

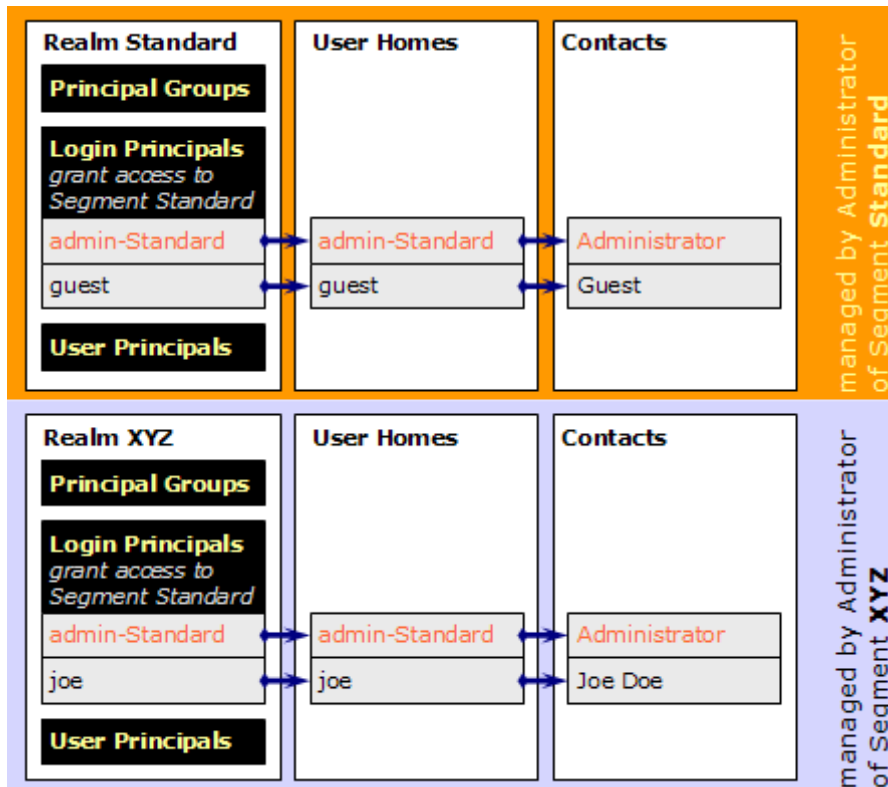


Figure 2: Segment Administration

While each “real user” (typically) has 1 application login principal only, “real users” may have multiple segment login principals (e.g. because a “real user” is allowed to access multiple segments or because a “real user” is allowed to access a particular segment in different roles like *Head of Sales* or *CFO*).

Available segment login principals are listed in the so-called Role Drop Down:



Figure 3: Role Drop Down with list of available Segment Login Principals

3.1.2 Permissions / Access Control

The openCRX security framework makes a distinction between **Ownership Permissions** (i.e. permissions granted on a particular object are based on object ownership) and **Model Permissions** (i.e. permissions are granted on a particular model element). As the latter is not yet implemented we only talk about Ownership Permissions in this guide.

Ownership permissions are used to control browse/delete/update access to openCRX objects by Users and UserGroups. **Ownership access control** was introduced with openCRX v1.4.0. Every openCRX object is a **SecureObject**. The following figure shows an extract from the UML model (if you are interested in all the details and the formally correct and complete specifications you should refer to the latest openCRX UML models):

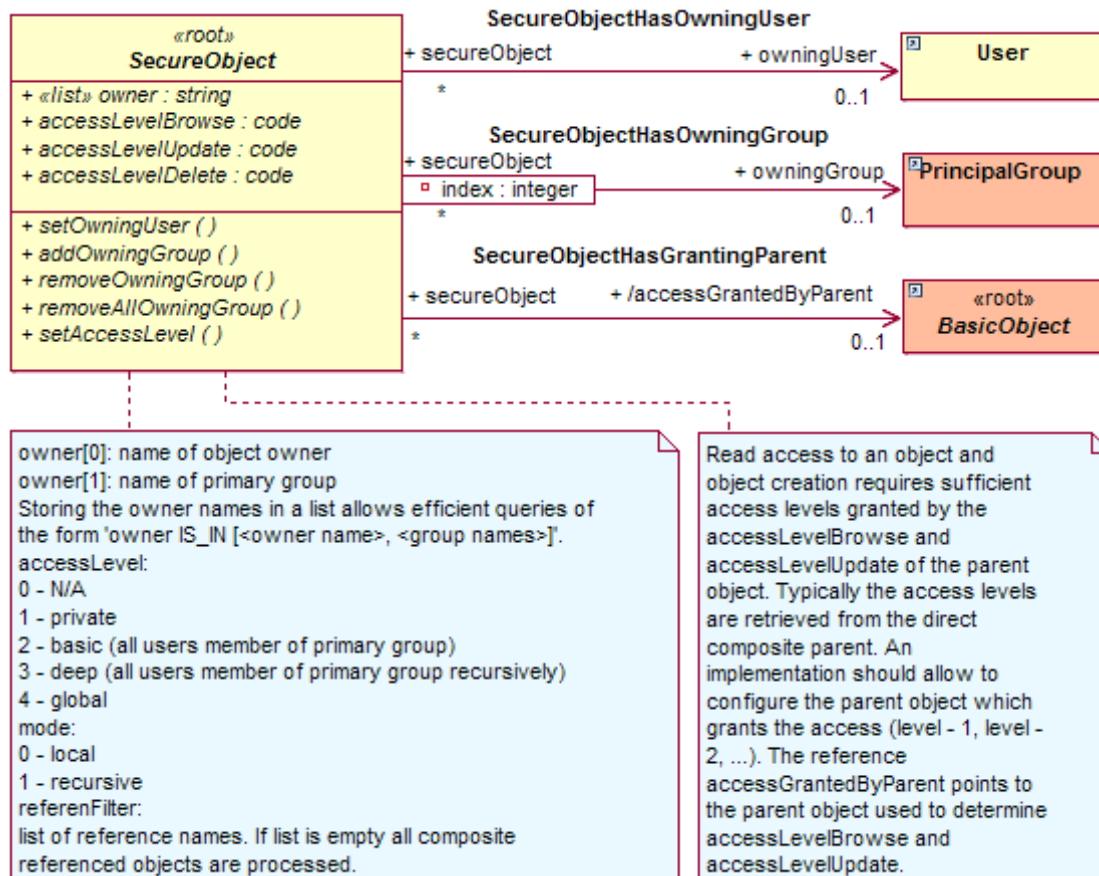


Figure 4: openCRX UML Model – Class Diagram SecureObject



If you see **N/P** in a field instead of a more meaningful value you probably do not have browse access to the respective object (N/P stands for **No Permission**)



If you see **N/A** in a field instead of a more meaningful value the object cannot be retrieved (N/A stands for **Not Available**); maybe the object was deleted or the provider is not accessible/available.

The most important security attributes of an object X are discussed below:

- **Owning User:** this user "owns" object X; the Owning User can always browse/delete/update object X (unless the access level is set to 0 [no access]).
- **Owning Groups:** these groups might enjoy privileged treatment for browsing/deleting/updating object X depending on the relevant access level settings.
- **Access Granted by Parent:** this attribute is set by configuration and refers to the parent object that grants access to object X.
- **Browse Access Level:** this setting determines which users / user groups are granted **browse access to direct composite objects of object X** [i.e. can view/inspect direct composite objects (including all their attributes) of object X].



It is a common misconception that browse access level of an object X controls browse access to this object X – please read the above definition carefully!

- **Delete Access Level:** this setting determines which users / user groups are granted **delete access to object X and all its composite objects (recursively!)** [i.e. can delete object X and all its composite objects (recursively!)].
- **Update Access Level:** this setting determines which users / user groups are granted **update access to object X** [i.e. can change object X; this includes adding composite objects to object X].

General | More Charts | Options | System ▾

Identity: xri:@openmdx.org.opencrx.kernel.home1/provider/CRX/segment/Standard/userHome/admin-Standard

Object

Created at: 2/14/06 3:01 PM Modified at: 7/3/06 9:17 AM
 Created by: admin-Root Modified by: admin-Standard

Security

Owning user: admin-Standard.User Browse access level: [3] deep (group members recursively)
 Access granted by parent: User Homepages Update access level: [2] basic (group members)
 Delete access level: [0] N/A

Alerts | Subscriptions | E-Mail | > ^

+

Owning Groups

Name	User Group	Disabled	Description	Member of Principal Groups	Granted Roles
Administrators	Root\Administrators				Unspecified

Figure 5: System attributes of an openCRX object as shown in the GUI

The following **access levels** are available to control which users / user groups are granted permission to browse/delete/update a particular object X:

Access Level	Meaning
0 – N/A	no access
1 – private	access is granted if the user is owning user of object X
2 – basic	access is granted if at least one of the following conditions is true: <ul style="list-style-type: none"> (a) the user is owning user of object X (b) the user is member of any of the owning groups of object X (c) any of the owning groups of object X is a subgroup** of any group the user is member of
3 – deep	access is granted if at least one of the following conditions is true: <ul style="list-style-type: none"> (a) the user is owning user of object X (b) the user is member of any of the owning groups of object X (c) any of the owning groups of object X is a subgroup** of any group the user is member of (d) any of the owning groups of object X is a subgroup** of any supergroup* of any group the user is member of
4 – global	all users are granted access

* Owing group G_{super} is a supergroup of an owing group G if every user who is member of G is also member of G_{super}

** Owing group G_{sub} is a subgroup of an owing group G if every user who is member of G_{sub} is also member of G

3.2 Default Settings

Default access level settings for non-Root segments (e.g. segment Standard) after a clean install are as follows:

Browse Access Level: **4 – global**
 Update Access Level: **3 – deep**
 Delete Access Level: **1 – private**

object_id	p\$type_name	access_level_browse	access_level_update	access_level_delete
accounts/CRX/Standard	segment	4	3	1
activities/CRX/Standard	segment	4	3	1
admins/CRX/Root	segment	NULL	NULL	NULL
buildings/CRX/Standard	segment	4	3	1
codes/CRX/Root	segment	4	2	2
codes/CRX/Standard	segment	4	3	1
contracts/CRX/Standard	segment	4	3	1
depots/CRX/Standard	segment	4	3	1
docs/CRX/Standard	segment	4	3	1
forecasts/CRX/Standard	segment	4	3	1
homes/CRX/Standard	segment	4	3	1
models/CRX/Standard	segment	4	3	1
products/CRX/Standard	segment	4	3	1
uoms/CRX/Root	segment	4	2	2
uoms/CRX/Standard	segment	4	3	1
workflows/CRX/Standard	segment	4	3	1

Figure 6: Table `kernel_Segment` after default installation (QuickStart)

Due to the setting `access_level_browse = 4 (global)` **any user with access to a particular segment is allowed to browse top level objects** (e.g. browse all accounts, browse all activities).

These default settings are suitable for test environments and deployments in smaller companies/teams with a liberal access policy; for most real-world applications, however, it is more appropriate to set `access_level_browse = 3 (deep)` for non-Root segments. You can do this by changing the values in the **column `access_level_browse`** from 4 to 3 (table `kernel_Segment`). After this change, the table `kernel_Segment` will look as shown in the following figure:

object_id	p\$type_name	access_level_browse	access_level_update	access_level_delete
accounts/CRX/Standard	segment	3	3	1
activities/CRX/Standard	segment	3	3	1
admins/CRX/Root	segment	NULL	NULL	NULL
buildings/CRX/Standard	segment	4	3	1

Figure 7: Table `kernel_Segment` after modification



Segment security settings are loaded during the initialization of the openCRX servlet. Hence, if you change settings you must redeploy openCRX for the new settings to become active.

3.3 Activating Security

The openCRX security provider manages all security data and provides access control services for all requests through the openCRX API. Hence, you can rely on openCRX access control even if you write your own clients or adapters for openCRX.

Security (including Access Control) is not just an add-on, rather it is an integral part of openCRX; **openCRX Access Control is always activated.**



The only “hardening” you might want to do is the one described in the previous chapter: set browse access level to 3 for non-Root segments.

3.4 Security Settings of New Objects

New objects are by default created with the following security settings:

Browse Access Level:	3 – deep
Update Access Level:	2 – basic
Delete Access Level:	2 – basic
Access Granted by Parent	in general: Parent object as modeled exceptions: there are some select exceptions, but they are all pre-configured
Owning User:	User who is creating the object
Owning Groups:	Primary User Group of the user who is creating the object and (meaning <u>as well as</u>) Owning Group(s) of the parent object of the new object (except Users, see below).



Please note that the User Group **Users** (e.g. Standard\\Users) is **not added** to the list of *Owning Groups* of newly created objects unless the creating user's **Primary User Group** is equal to **Users**.



Please note that a User's **Primary User Group** is set by the segment administrator with the operation **Create User**. To change an existing user's primary group, the segment administrator simply executes the operation *Create User* again with a new parameter for primary user group.



In the context of activity management there are various operations that set/change the Owning Groups of objects based on the settings of an assigned **Activity Creator** or assigned **Activity Group** and not based on the settings of the user who executes the operation.

3.5 Login Procedure

The openCRX login procedure consists of 2 levels:

3.5.1 Application Server Login

The application server login procedure depends on various parameters:

- application server (e.g. JBoss, BEA WLS, IBM WebSphere, etc.)
- configuration of application server
 - file-based realm (e.g. **users.properties** for JBoss)
 - DB-based realm (e.g. DatabaseLoginModule for JBoss)
 - LDAP-based realm
 - company-specific / custom-tailored realms

Please note that even though openCRX might be involved in managing some of the above-mentioned realms (e.g. DB-based realm) the application server login is not really under control of openCRX. Many login problems are related to incomplete/faulty application server configuration settings.

3.5.2 Segment Login

Access to segments is managed/controlled by the *ObjectInspectorServlet*. The included *DefaultRoleMapper* identifies all **Segment Login Principals** of a given **Subject** and grants access to the respective segments through the Role Drop Down:

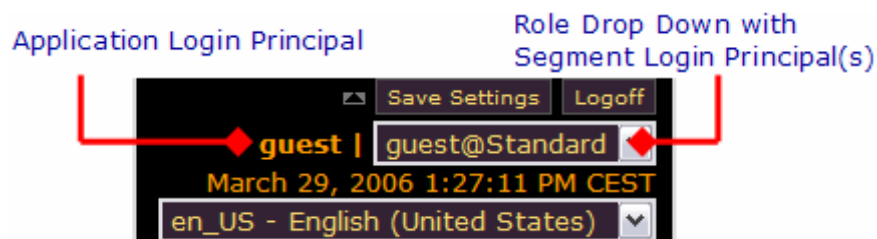


Figure 8: Role Drop Down with list of available Segment Login Principals

It is possible to deploy user-specific implementations of the *DefaultRoleMapper* so that you can adapt the segment login procedure to your requirements.

3.5.3 Disabling Login

Please refer to the chapter "Disable/Deactivate Users".

3.6 Resetting Security



If you get the setting of Update Access Level wrong you may not be able to change the respective object from the GUI anymore (and that includes the security settings of that object!). For example, the only way to recover from setting Update Access Level to **0 – N/A** for a particular object is to edit the data directly in the database!

If you (or one of your users) managed to screw up the security settings in a major way you might be forced to reset all security settings to a well-defined state. Not an easy task – and it typically involves a lot of manual work.

4 Managing Users



Read through the chapter Basic Concepts and Conventions (Security) before reading this chapter. It is quite helpful to have a good understanding of the terms Subject, Application Login Principal, Segment Login Principal, User, etc. before you start reading here.

4.1 Creating Users

The following steps are required to create a new openCRX user:

Who	Steps
Root administrator admin-Root	<ul style="list-style-type: none"> ● create a new Subject <ul style="list-style-type: none"> ▪ set the qualifier to the desired login id ● create a new Principal in realm Default (--> Application Login Principal) <ul style="list-style-type: none"> ▪ set the qualifier to the desired login id ▪ link this Principal to the Subject created in the previous step ▪ make Principal member of the appropriate Principal Group(s), e.g. <i>Users</i>
Segment administrator admin-<SegmentName>	<ul style="list-style-type: none"> ● create a new Contact ● create a new User <ul style="list-style-type: none"> ▪ link this User to the Contact created in the previous step ▪ the Segment Login Principal is created automatically

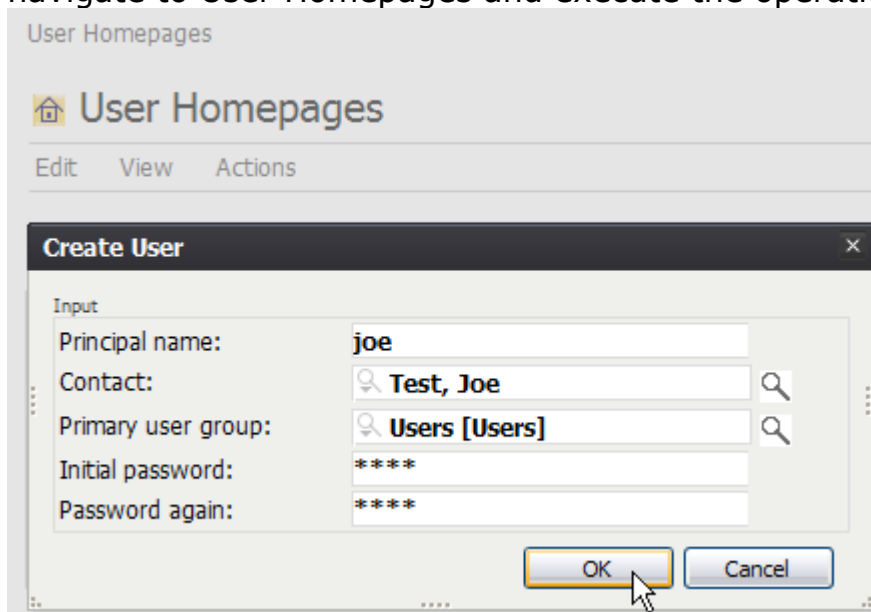
Have a look at *Figure 1: Security Realms, Principals and Subjects after Initial Setup* and *Figure 2: Segment Administration* to see how this all fits together.

4.1.1 Create Users Manually

The openCRX QuickStart guide (<http://www.opencrx.org/documents.htm>) contains a very detailed step-by-step example of how to manually create a new openCRX user.

In addition to the procedure described in the openCRX QuickStart guide where a new **Subject** and a new **Principal** are created manually, there is also fast procedure if subject name and principal name are identical:

- login as Segment administrator (e.g. admin-Standard)
- create a contact for the new user
- navigate to User Homepages and execute the operation Create User:



The screenshot shows the 'User Homepages' interface with a 'Create User' dialog box open. The dialog box has a title bar with 'Create User' and a close button. Below the title bar, there is an 'Input' section with the following fields:

Principal name:	<input type="text" value="joe"/>
Contact:	<input type="text" value="Test, Joe"/> <input type="button" value="Search"/>
Primary user group:	<input type="text" value="Users [Users]"/> <input type="button" value="Search"/>
Initial password:	<input type="password" value="****"/>
Password again:	<input type="password" value="****"/>

At the bottom of the dialog box, there are two buttons: 'OK' and 'Cancel'. A mouse cursor is pointing at the 'OK' button.

- upon execution a new principal and a new subject are created automatically if they do not exist yet

4.1.2 Import Subjects and Application Login Principals

Creating large numbers of subjects/principals by hand can be quite a tedious job. If you prepare a text file containing the appropriate information in the file format as outlined below, the Root administrator (**admin-Root**) can use the operation **Actions > Import Login Principals** to create Subjects and Application Login Principals automatically.

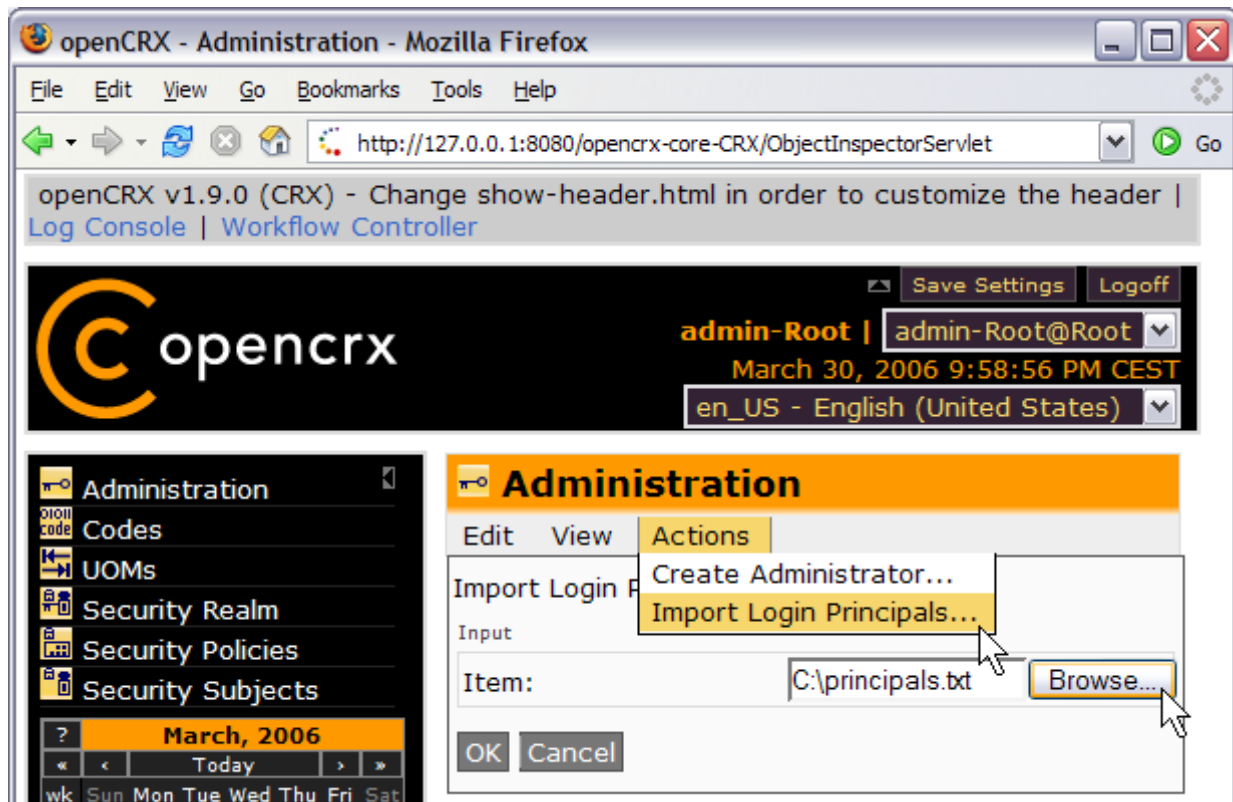


Figure 9: Operation **Actions > Import Login Principals** (**admin-Root**)

Listing 1: File Format **Subjects** and **Application Login Principals**

```
Subject;<subject name>;<subject description>
Principal;<principal name>;<principal description>;<subject name>;<groups>
```

Listing 2: Example File **Subjects** and **Application Login Principals**

```
Subject;joe;Doe, Joe
Subject;mark;Ferguson, Mark
Subject;peter;Lagerfeld, Peter

Principal;joe;Doe, Joe;joe;Users,Administrators
Principal;mark;Ferguson, Mark;mark;Users
Principal;peter;Lagerfeld, Peter;peter;Users
```

4.1.3 Import Users

Similarly to importing Subjects and Application Login Principals from a file you can also import Users from a file. If you prepare a text file containing the appropriate information in the file format as outlined below, the Segment administrator (**admin-<SegmentName>**) can use the operation **Actions > Import Users** to create Users automatically.

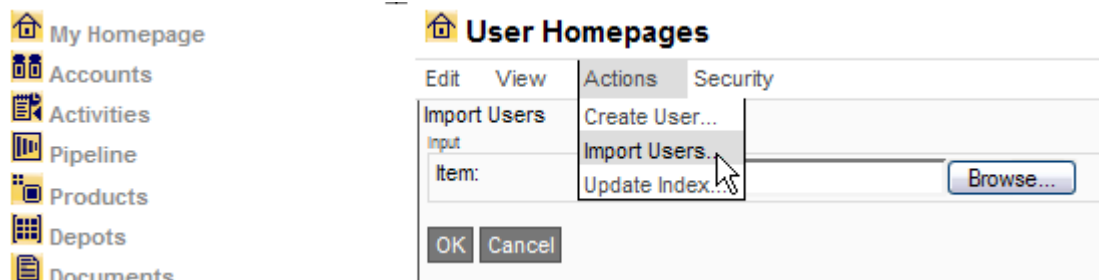


Figure 10: Operation Actions > Import Users (admin-Standard)

Listing 3: File Format **Users**

```
User;<principal>;<account alias>;<account full name>;<primary group>;<password>[;group [, group] ]
```

Please note that **<password>** is a clear-text value. The optional parameter **group** can be used to make the **Segment Login Principal** member of the respective Principal Groups of that segment.

Listing 4: Example File **Users**

```
User;joe;JD;Doe, Joe;Users;2%jOd.IT
User;mark;Fergi;Ferguson, Mark;Users;maFe&.3-
User;peter;Pete;Lagerfeld, Peter;Users;PlF*;ReGaL
```



Contacts are not created automatically; existing Contacts are first search by **<account alias>**. If no matching account alias is found, Contacts are search by **<account full name>**. If still no matching account is found, the UserHome is **not** created.

Users are only imported/created if the referenced Principals exist.

4.2 Disable/Deactivate Users

There are various ways of disabling/deactivating users. To fully understand your options it is helpful if you are familiar with the openCRX Login Procedure.

4.2.1 Disabling Users at the level Application Server

Depending on the configuration of your application server you can disable users at that level. For example, if you rely on file-based realms with JBoss you can simply remove users from the file **users.properties** to prevent access to openCRX. If you block access at the level Application Server such users are locked out from accessing any application and any openCRX segment. However, as the Application Server Login procedure is not entirely controlled by openCRX you must consult the documentation of your respective AppServer or your AppServer Admin for details.

4.2.2 Disabling Users at the level openCRX

The segment administrator (e.g. **admin-Standard**) can prevent a user from accessing a particular openCRX segment by either disabling the respective **Segment Login Principal** or by deleting it altogether. Disabling is the preferred option to prevent access temporarily. If a user has multiple Segment Login Principals you must disable all of them to prevent access to the openCRX application.

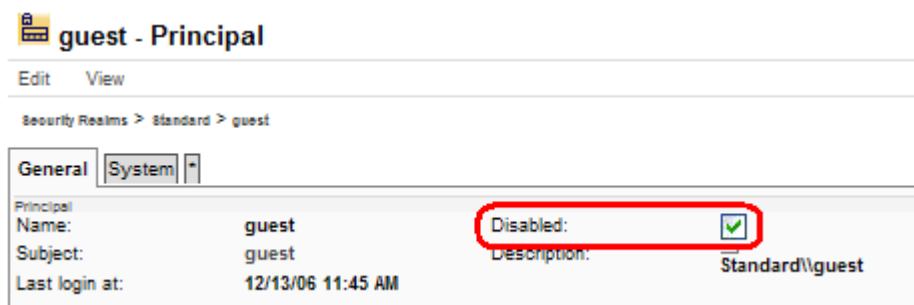


Figure 11: Disabling of Segment Login Principal **guest** by **admin-Standard**

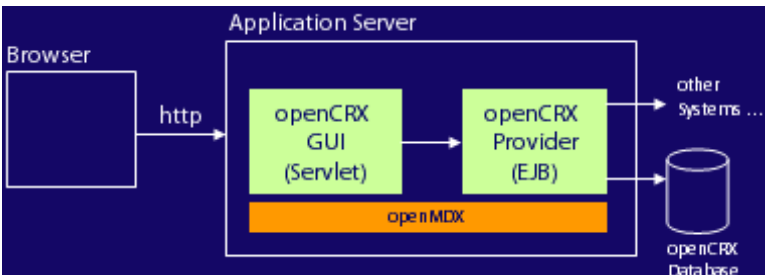
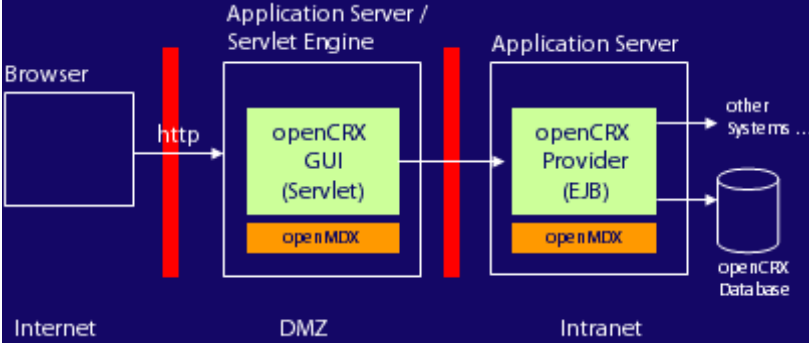
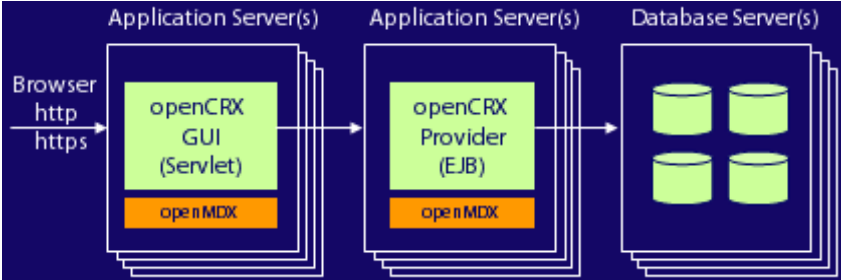
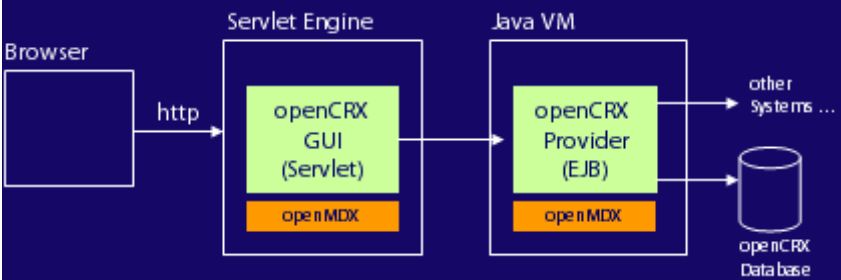


You should **not** delete a particular Subject as long as it is referenced by any Principal. Otherwise you'll end up with "dangling" Subject references.

5 Deployment Scenarios

5.1 Typical Deployment Scenarios

The following table lists some of the pros and cons of the 4 most common deployment scenarios:

<p>3-Tier with AppServer</p> <ul style="list-style-type: none"> ▪ simple setup and management (one application server) ▪ limited security (not recommended for Internet deployment) ▪ limited scalability and availability (no clustering) ▪ Transaction Service 	 <p><i>Figure 12: 3-Tier with Application Server</i></p>
<p>4-Tier with AppServer</p> <ul style="list-style-type: none"> ▪ simple setup (two application servers) ▪ allows to setup Internet, DMZ, Intranet security zones ▪ limited scalability and availability (no clustering) ▪ Transaction Service 	 <p><i>Figure 13: 4-Tier with Application Server</i></p>
<p>4-Tier with Clustered AppServers</p> <ul style="list-style-type: none"> ▪ complex setup (four and more application servers) ▪ allows to setup Internet, DMZ, Intranet security zones ▪ full scalability and availability ▪ Transaction Service 	 <p><i>Figure 14: 4-Tier with Clustered Application Servers</i></p>
<p>4-Tier with Servlet Engine</p> <ul style="list-style-type: none"> ▪ complex setup (four and more application servers) ▪ allows to setup Internet, DMZ, Intranet security zones ▪ full scalability and availability ▪ Transaction Service 	 <p><i>Figure 15: 4-Tier with Servlet Engine</i></p>

5.2 Multi Entity Deployment Scenarios

The open source MDA platform openMDX supports a multitude of deployment scenarios and persistency configurations. The most common multi entity deployment scenarios are discussed in the following sections.

5.2.1 Multiple Data Segments in a single DB

The setup “Multiple Data Segments in a single DB” provides adequate security for many use cases and is relatively easy to manage. As all the data is stored in a single database, however, security configuration mistakes (e.g. principals linked to the wrong subject, etc.) can lead to situations where a user is granted access to the data of a particular company/client that should not be accessible (please note that **human error** is the root cause here, not a malfunction of openCRX). Furthermore, this setup is not recommended if users can get direct access to the database, e.g. with third party reporting tools as those tools typically bypass the openCRX API.

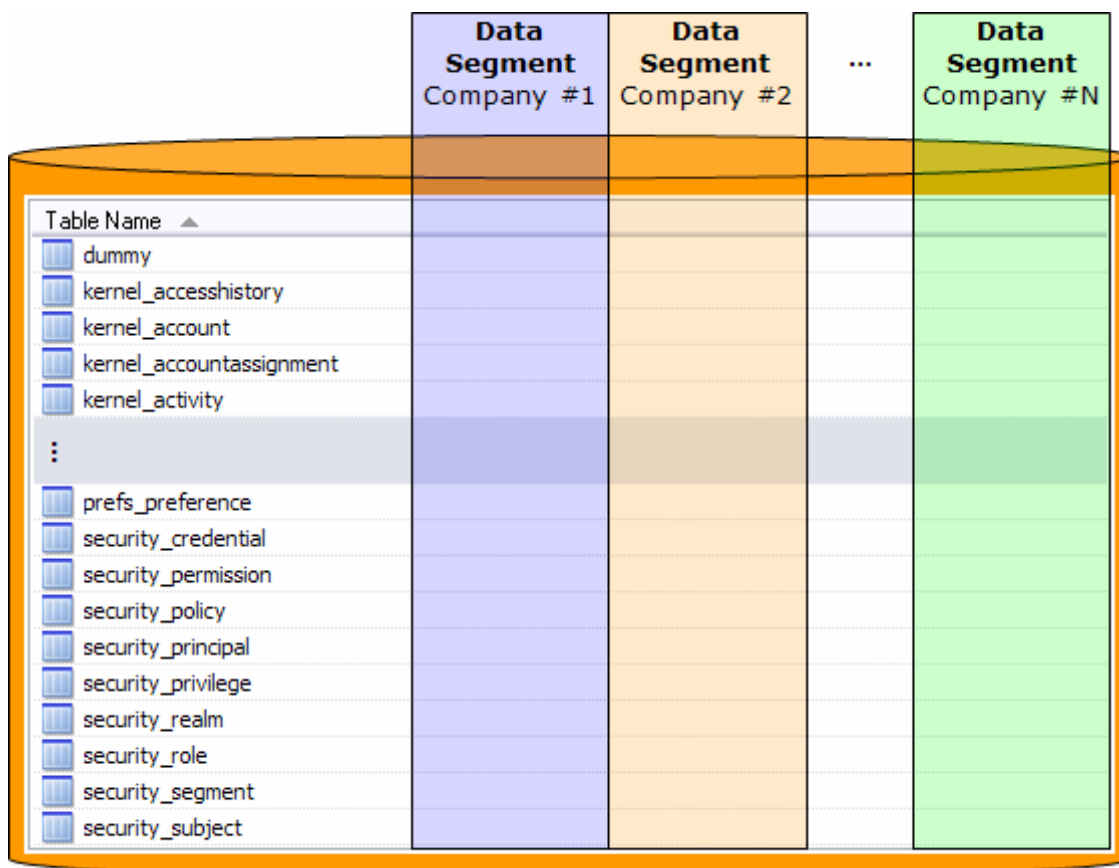


Figure 16: Multiple Data Segments in a single DB

5.2.2 Multiple DBs

The highest level of security is provided by setting up a dedicated database for each entity so that data sets of the various entities are physically separated:

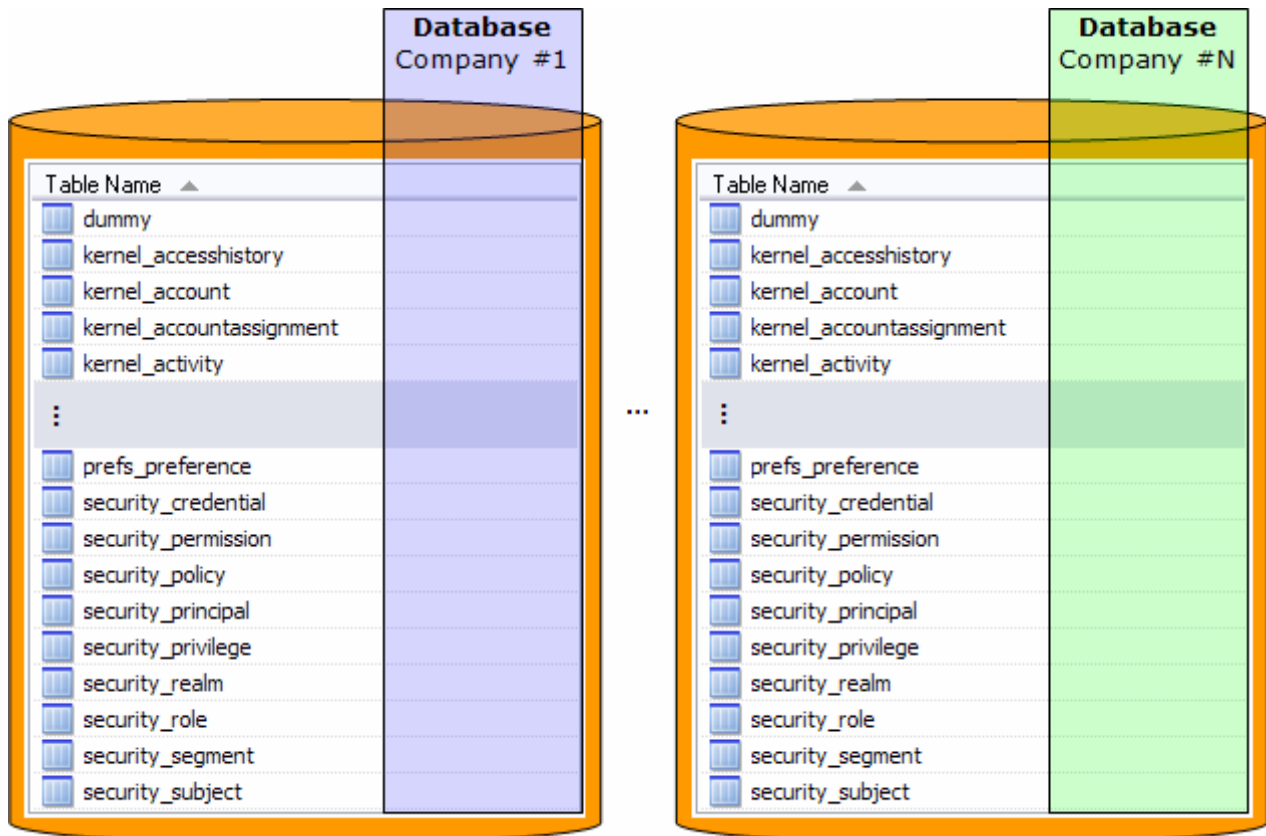


Figure 17: Dedicated DB for each Entity

5.2.3 Multiple Applications

Multiple (differently) customized Web-EARs can access the same App-EAR. The build process creates the following EARs:

- One App EAR for the provider.name specified in build.properties
- One Web EAR for the provider.name and web.application.name in build.properties, whereas the Web EAR delegates to the App EAR.
- With `ant -Dweb.application.name=<appname> -Ddata.dir=<custom dir>` additional Web EARs can be created. They Web EAR has the name <appname> containing the customizing info from <custom dir>.

5.3 openCRX Custom Applications

<documentation pending>

6 Workflow Controller

With the **Workflow Controller** the openCRX Root administrator (**admin-Root**) can enable/disable various servlets (configured in **web.xml**) included in the openCRX distribution. This chapter gives an overview over the currently available servlets and explains how to start/stop them.

You can access the Workflow Controller by navigating to the URL

`http://127.0.0.1:8080/opencrx-core-CRX/WorkflowController`

or starting the Workflow Controller Wizard as shown in the figure below:

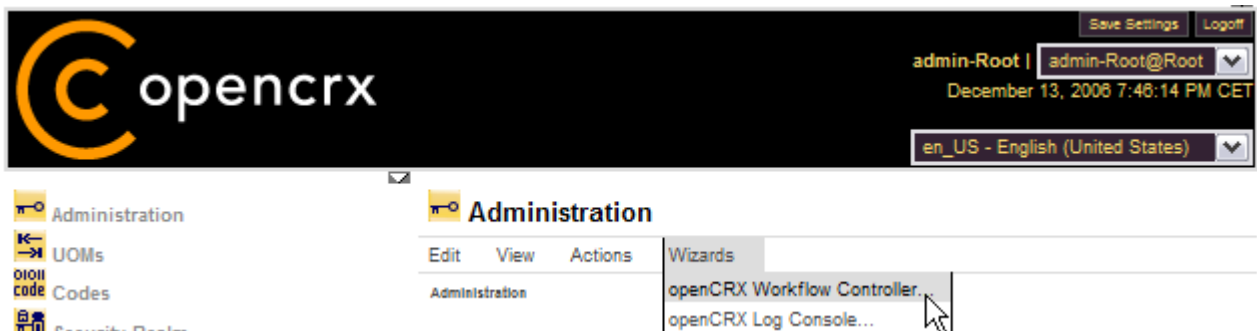


Figure 18: Accessing the openCRX Workflow Controller



You should **connect to the Workflow Controller with http**. If you use SSL-secured connections to start/stop servlets you must ensure that your server's certificate is available in cacerts.

The following figure shows the Workflow Controller of openCRX 1.10.0 (the look and feel may be different for other openCRX versions):



Figure 19: openCRX 1.10.0 Workflow Controller



Please note that access is granted to the openCRX Root administrator (**admin-Root**) only. Hence, if you see the openCRX login screen instead of the Workflow Controller you must first login as Root administrator. Also, ensure that openCRX is properly initialized before you connect to the Workflow Controller.



The first time the Workflow Controller is started it will create a default configuration:

WorkflowController - Component Configuration

Edit View Security

Administration > WorkflowController

General System

Name: WorkflowController
Description:

Properties

New Edit

Name	Description	String value
MailImporterServlet.CRX.Standard.autostart	MailImporterServlet autostart	false
MailImporterServlet.CRX.Standard.pingrate	MailImporterServlet pingrate	2
serverURL	Server URL	http://127.0.0.1:8080/openocrx-core-CRX
SubscriptionHandler.CRX.Standard.autostart	SubscriptionHandler autostart	false
SubscriptionHandler.CRX.Standard.pingrate	SubscriptionHandler pingrate	2
WorkflowHandler.CRX.Standard.autostart	WorkflowHandler.autostart	false
WorkflowHandler.CRX.Standard.pingrate	WorkflowHandler.pingrate	2

Figure 20: Default Configuration of WorkflowController

If you ever need to recreate this default configuration, you can do so with the following steps:

- stop the WorkflowController
- delete the Configuration with the name WorkflowController
- start the WorkflowController

You can manually start (stop) servlets that are managed by the Workflow Controller by clicking on "Turn On" ("Turn Off"). Please note that you can control servlets on a segment by segment basis. For example, if you created a segment "MySegment" in addition to the segment "Standard" you can start/stop servlets of the segment "MySegment" without interfering with the servlets of the segment "Standard".

6.1 Workflow Controller Configuration

In addition to configuring the Startup option of the Workflow Controller you can also configure various options related to the servlets managed by the Workflow Controller. The configuration of the Workflow Controller is accessible to the openCRX Root administrator (**admin-Root**) by navigating to Administration and then clicking on the icon of the WorkflowController:

The screenshot shows the openCRX Administration web interface. On the left is a navigation menu with items: Administration, UOMs, Codes, Security Realm, Security Policies, Security Subjects, and History. Below the menu is a calendar for December 2006, with the 13th highlighted. The main content area is titled 'Administration' and contains a 'Configurations' section. This section has 'New' and 'Edit' buttons, a search field with a question mark icon, and a table of configurations. The table has columns for 'Name' and 'Description'. One configuration is listed: 'WorkflowController'.

Figure 21: openCRX Administration – WorkflowController

6.1.1 Startup Configuration in web.xml

You can start the Workflow Controller **manually** by navigating to the URL

<http://127.0.0.1:8080/opencrx-core-CRX/WorkflowController>

or starting the Workflow Controller Wizard. However, it is also possible to start the Workflow Controller **automatically** by activating the corresponding option in the file **web.xml**:

Listing 5: web.xml – auto startup of the Workflow Controller

```
<!-- WorkflowController -->
<servlet id="WorkflowController">
  <servlet-name>WorkflowController</servlet-name>
  <servlet-class>org.opencrx.kernel.workflow.servlet.WorkflowControllerServlet</servlet-class>
  ...
  <!-- activate if WorkflowController should be initialized at startup-->
  <load-on-startup>10</load-on-startup>
</servlet>
```



With the value of **load-on-startup** (10 above) you can control the order of starting up servlets in case there is more than one.

6.1.2 ServerURL

Adapt the value of `serverURL` to your environment (e.g. `http://127.0.0.1:8080/opencrx-core-CRX`):

The screenshot shows the 'WorkflowController - Component Configuration' page. The 'Properties' section is expanded, and the 'serverURL' property is highlighted with a red box. The value is 'http://127.0.0.1:8080/opencrx-core-CRX'.

Name	Value
serverURL	http://127.0.0.1:8080/opencrx-core-CRX
WorkflowHandler.CRX.Standard.pingrate	1

Figure 22: Workflow Controller Configuration – serverURL

6.1.3 Handler pingrate and autostart

Use the **pingrate** to define the interval (in minutes) between successive calls of the respective handler and **autostart** (true/false) to start the respective handler automatically:

The screenshot shows the 'WorkflowController - Component Configuration' page. The 'Properties' section is expanded, and the 'SubscriptionHandler.CRX.Standard.pingrate' and 'SubscriptionHandler.CRX.Standard.autostart' properties are highlighted with a red box. The values are '1' and 'true' respectively.

Name	Value
SubscriptionHandler.CRX.Standard.pingrate	1
SubscriptionHandler.CRX.Standard.autostart	true

Figure 23: Workflow Controller Configuration – pingrate and autostart

6.2 Servlet SubscriptionHandler

The openCRX **SubscriptionHandler** is the backbone of the openCRX Subscribe / Notify Services. The Subscription Handler does not require any configuration by the openCRX administrator other than setting the pingrate and autostart options, i.e. it is designed to work "out of the box".

Turning on the SubscriptionHandler of a particular segment is required if you want that segment to provide **Alerts** and **E-mail Notifications** to its Users. The polling frequency can be set by the Root administrator (see Figure 23: Workflow Controller Configuration – pingrate and autostart).

The SubscriptionHandler checks openCRX audit entries on a regular basis and – if matching Subscriptions exist – executes the Workflow Process referenced by the Subscription using *Userhome.executeWorkflow()*.

Listing 6: Iterate all auditees and check for new audit entries

```
for (
    Iterator j = auditSegments.iterator();
    j.hasNext();
) {
    Auditee auditee = (Auditee)j.next();
    AuditEntryFilter filter = basePkg.createAuditEntryFilter();
    // Not visited elements are marked with VISITOR_ID:-
    // Visited elements are marked with VISITOR_ID:<time stamp of visit>
    filter.thereExistsVisitedBy(
        FilterOperators.IS_IN,
        new String[]{VISITOR_ID + ":-"}
    );
    try {
        this.handleSubscriptions(
            ...
            auditee.getAudit(filter)
        );
    }
    catch (ServiceException e) {
        System.out.println(new Date() + ": openCRX/SubscriptionHandler:
exception occured " + e.getMessage() + ". Continuing");
    }
}
private static final String VISITOR_ID = "SubscriptionHandler";
```



Please note that activating the Subscription Handler on a DB with lots of (unvisited) Audit Entries (e.g. after an upgrade from openCRX v1.8.1 to openCRX v1.9.1 or later) can put some heavy load on your system until all the Notifications have been generated. You might want to use the following statement to mark all the existing entries as visited:

Listing 7: Mark Audit Entries as visited by Subscription Handler

```
UPDATE kernel_AuditEntry_N
set visited_by = 'SubscriptionHandler:-'
where (object_idx = 0) and (visited_by is null)
```

Userhome.executeWorkflow() – implemented by the openCRX plugin – creates an entry in *Userhome.wfProcessInstance* (accessible through the grid Workflow Process Instances) and executes synchronous workflows immediately. Beyond creating entries for asynchronous workflows, *executeWorkflow()* does not do anything with them (the Servlet WorkflowHandler is specialized in dealing with asynchronous workflows – see below for details).

6.3 Servlet WorkflowHandler

The openCRX **WorkflowHandler** is responsible for executing WfProcessInstances based on asynchronous WfProcesses like:

- org.opencrx.mail.workflow.ExportMailWorkflow
- org.opencrx.mail.workflow.SendMailNotificationWorkflow
- org.opencrx.mail.workflow.SendMailWorkflow

The execution frequency can be set by the Root administrator (see Figure 23: Workflow Controller Configuration – pingrate and autostart). Please note that the WorkflowHandler is required for outbound E-Mail Services.

The WorkflowHandler executes Workflow Process Instances that have not been executed yet.



The first time the WorkflowHandler is started it will create various default Workflow Processes:

Workflows

Edit View Actions Security

Workflows

Workflow Processes Topics ^

New Edit

? [input] [checkbox] [X] [stop] [list] [check] [list]

Name	Priority	Is synchronous	De
org.opencrx.kernel.workflow.PrintConsole	0	<input checked="" type="checkbox"/>	Print to
org.opencrx.kernel.workflow.SendAlert	0	<input checked="" type="checkbox"/>	Send a
org.opencrx.mail.workflow.ExportMailWorkflow	0	<input type="checkbox"/>	Export
org.opencrx.mail.workflow.SendMailNotificationWorkflow	0	<input type="checkbox"/>	Send r
org.opencrx.mail.workflow.SendMailWorkflow	0	<input type="checkbox"/>	Send r

Figure 24: Default Workflow Processes created by WorkflowHandler

If you ever need to recreate these default Workflow Processes, you can do so with the following steps:

- stop the Servlet WorkflowHandler
- delete the Workflow Processes that were originally created by the WorkflowHandler (or at least the ones that still exist)
- start the Servlet WorkflowHandler



All WfProcesses with undefined/unknown runtime length should be defined as asynchronous. This is particularly true for WfProcesses that might block. The current setup ensures that blocking WfProcesses cannot block openCRX.

6.4 MailImporterServlet

The **MailImporterServlet** provides generic E-mail Services. The servlet regularly connects to e-mail boxes and fetches messages to be imported into openCRX.

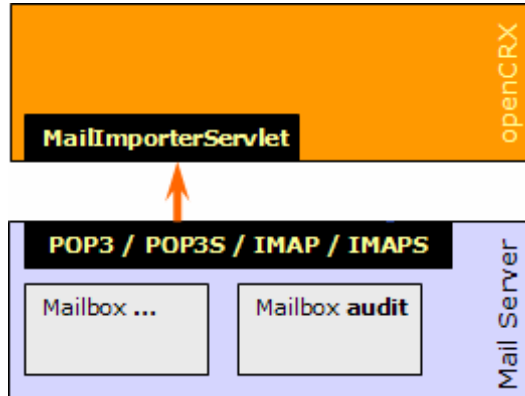


Figure 25: openCRX MailImporterServlet

The configuration of the MailImporterServlet is explained in detail in the chapter E-mail Services). The Root administrator (**admin-Root**) can set a few select options in the MailImporterServlet's ComponentConfiguration. The figure below shows the default configuration as it is created the first time the MailImporterServlet is started:

MailImporterServlet - Component Configuration

Edit View Security

General System *

Name: **MailImporterServlet**
 Description:

Properties ^

New Edit

	Name	Description	Domain	Boolean value	String value
01	CRX.Standard.caseInsenstiveAddressLookup.Default	Case insensitive address lookup		<input checked="" type="checkbox"/>	
01	CRX.Standard.deleteImportedMessages.Default	Delete imported messages		<input type="checkbox"/>	
ab	CRX.Standard.mailServiceName.Default	Mail service name			/mail/provider/CRX
ab	CRX.Standard.mailbox.Default	Mailbox name			INBOX

Figure 26: MailImporterServlet – Configuration

Polling frequency and autostart can also be set (see Figure 23: Workflow Controller Configuration – pingrate and autostart).

6.5 Trouble Shooting Servlets

All the openCRX servlets controlled by the Workflow Controller log their actions to the server log file (e.g. `D:\jboss-4.0.3SP1\server\default\log\server.log` on JBoss). The following log file extract shows, for example, that both the Subscription Handler and the Workflow Handler seem to be working fine, whereas the MailImporterServlet cannot connect to the mail box (due to missing configuration):

Listing 8: Servlets managed by Workflow Controller log to server.log

```
2006-04-04 14:04:25,936 INFO [STDOUT] Tue Apr 04 14:04:25 CEST 2006: openCRX/SubscriptionHandler: CRX/Standard
2006-04-04 14:04:25,967 INFO [STDOUT] Tue Apr 04 14:04:25 CEST 2006: openCRX/WorkflowHandler: CRX/Standard
2006-04-04 14:04:26,451 DEBUG [org.jboss.resource.connectionmanager.IdleRemover]
run: IdleRemover notifying pools, interval: 450000
2006-04-04 14:04:27,092 INFO [STDOUT] DEBUG: setDebug: JavaMail version 1.3.3
2006-04-04 14:04:27,092 INFO [STDOUT] DEBUG POP3: connecting to host "mail.changeme.com", port 110, isSSL false
2006-04-04 14:04:52,795 INFO [STDOUT] S: EOF
2006-04-04 14:05:15,811 INFO [STDOUT] C: QUIT
2006-04-04 14:05:15,811 INFO [STDOUT] S: EOF
```

openCRX Exceptions (like NullPointers, etc.), however, are still logged to the application log file as configured during the installation (see QuickStart guide).

It is always worth checking whether the Workflow Handlers actually are active; they must be started by the Root administrator. You can find out by connecting to the Workflow Controller (see *Figure 19: openCRX 1.10.0 Workflow Controller*).



After restarting the application server all servlets managed by the WorkflowController are turned off, i.e. the Root Administrator must explicitly turn them on again (if desired) unless the respective servlet's `autostart` option is set to `true` in the WorkflowController's configuration **and** the WorkflowController's `Startup` option is set to `true` in the file `web.xml`. The servlets do not automatically resume the state they were in before the application server was shut down.

7 Subscribe / Notify Services

openCRX features a powerful event subscription and notification service:

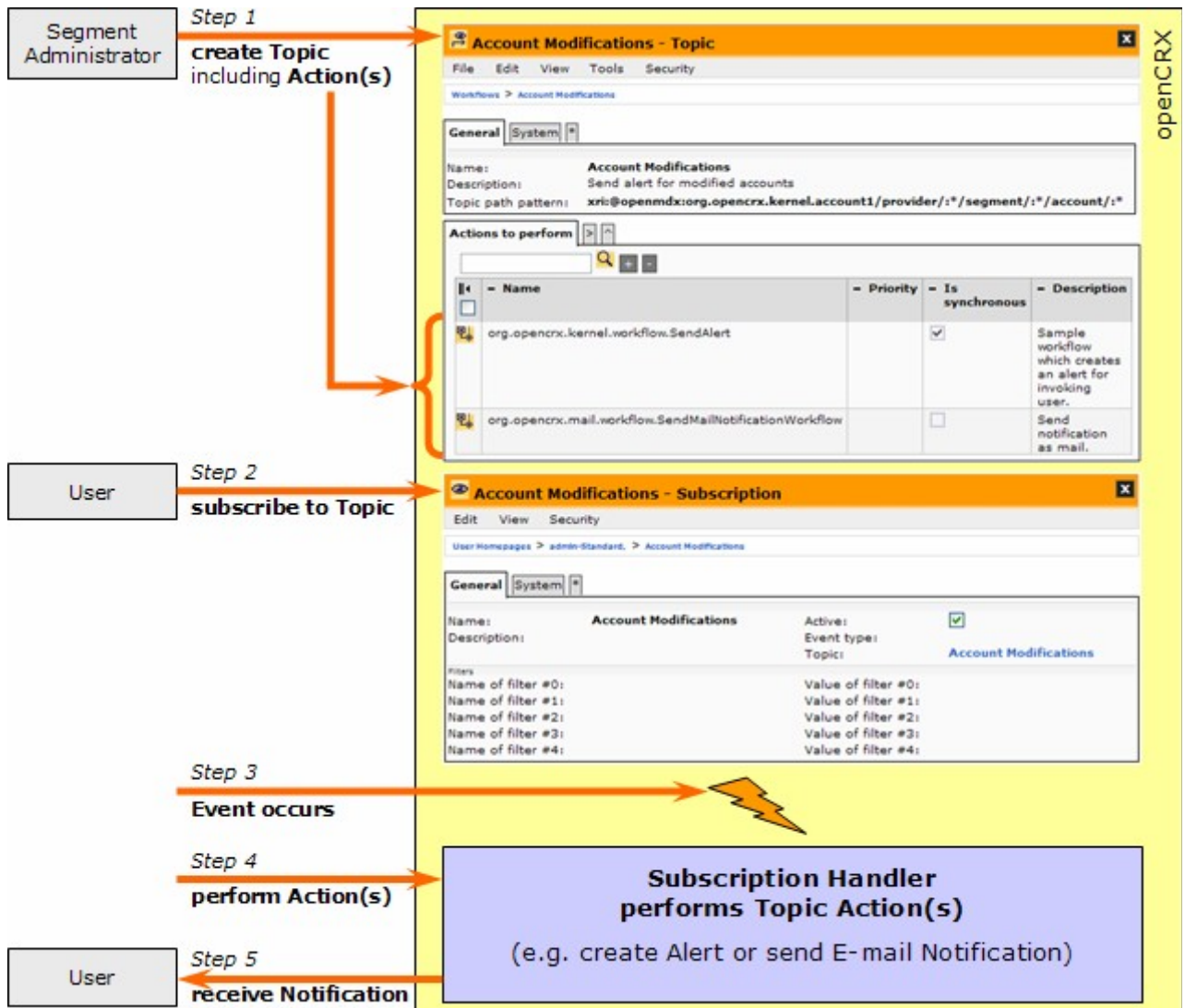


Figure 27: Event and Notification Service

Once a topic is created, openCRX users can subscribe to it. Users manage their subscriptions individually on their UserHomes. If a topic has subscribed users and a monitored event occurs then the predefined actions are performed. If the action is set to – for example – creating an alert for subscribed users, then each subscribed user will receive an alert on her UserHome.



Please note that event and notification services depend on the **Servlet SubscriptionHandler**, i.e. **you must turn on the openCRX Subscription Handler** for the respective segment with the **Workflow Controller**, otherwise **Topic Actions** are not executed, i.e. no Alerts and E-mail Notifications.



Furthermore, as outbound E-Mail Services depend on the WorkflowHandler, you must activate the Workflow Handler to receive E-Mail Notifications.

The openCRX distribution includes quite a few default topics (see *Figure 28: Standard Topics included in the openCRX distribution*) to get you started:

- Topic **Account Modifications** sends an alert to the UserHome of subscribed users whenever an account is modified.
- Topic **Activity Follow Up Modifications** sends an alert to the UserHome of subscribed users whenever a Follow Up of an Activity is modified.
- Topic **Alert Modifications** sends an e-mail notification to subscribed users – assuming outbound e-mail services are configured correctly – whenever an Alert is created/modified.



Please note that newly created Segments do neither contain Workflow Processes nor Topics (i.e. the respective grids are empty). Both Workflow Processes and Topics are created by the Subscription Handler of the respective segment when it is started for the first time.

Workflow Processes		Topics	
New			
Delete			
?			
X			
✓			
☰			
	- Name	- Description	- Topic path pattern
	Account Modifications	Send alert for modified accounts	xri:@openmdx:org.opencrx.kern
	Activity Follow Up Modifications	Send alert for modified activity fo	xri:@openmdx:org.opencrx.kern
	Activity Modifications	Send alert for modified activities	xri:@openmdx:org.opencrx.kern
	Booking Modifications	Send alert for modified bookings	xri:@openmdx:org.opencrx.kern
	Competitor Modifications	Send alert for modified competitc	xri:@openmdx:org.opencrx.kern
	Competitor Modifications	Send alert for modified competitc	xri:@openmdx:org.opencrx.kern
	Compound Booking Modifications	Send alert for modified compoun	xri:@openmdx:org.opencrx.kern

Figure 28: Standard Topics included in the openCRX distribution

Users can easily custom-tailor their subscriptions with filters and by selecting event types like *Object Creation*, *Object Replacement*, and *Object Removal*.

7.1 Example Subscription – Activity Modifications

In this example we will create a subscription to the standard Topic Activity Modifications for the user “guest”.

- Login as **guest**, click on the grid Tab **Subscriptions** (on the Homepage) and then use the Creator Menu **New > Subscription** to create a new Subscription:

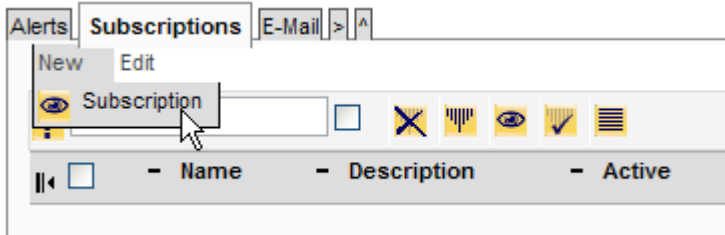


Figure 29: Create a new Subscription – step 1

- Enter **Name** and **Description** of the subscription to be created and check the box **Active**. Click on the edit icon of the field **Event type** and then select the option **[1] Object Creation**:

Untitled - Subscription

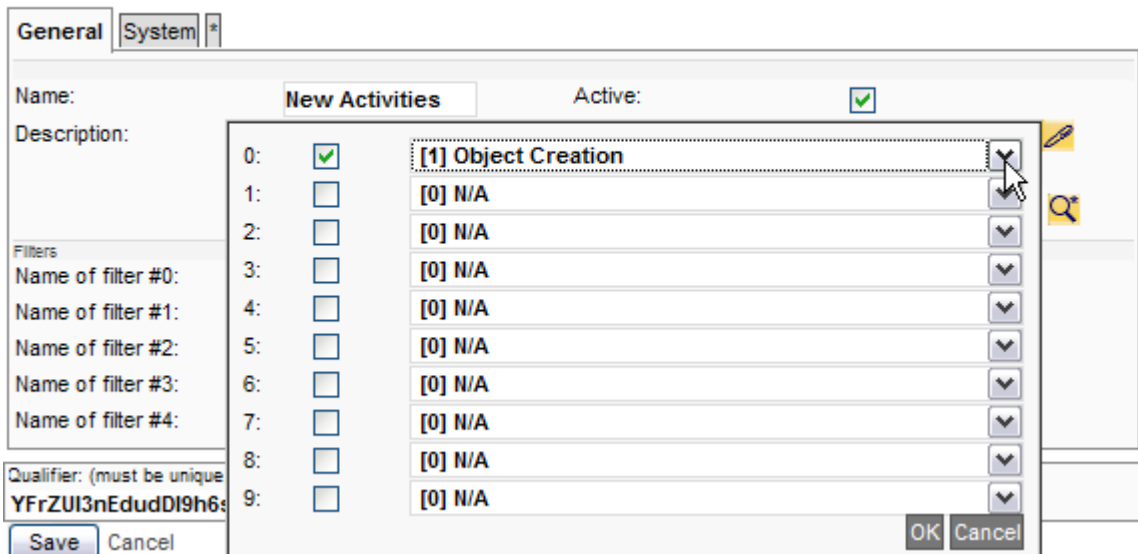


Figure 30: Create a new Subscription – step 2

- Next you click OK to accept; selecting option **[1] Object Creation** will ensure that we will get alerts for new accounts only (if you leave the field Event type empty you will get alerts for all event types, i.e. Object Creation, Object Modification, and Object Removal).

- Type **activity** into the field **Topic**; the auto-completer will show a drop down from which you select **Activity Modifications [Activity Name]** as shown below:

Untitled - Subscription

General System *

Name: Active:

Description:

Event type: [1] Object Creation

Topic: activity

Activity Follow Up Modification

Activity Modifications [Activity]

Filters

Name of filter #0: Value of filter #0:

Name of filter #1: Value of filter #1:

Name of filter #2: Value of filter #2:

Name of filter #3: Value of filter #3:

Name of filter #4: Value of filter #4:

Qualifier: (must be unique, i.e. leave unchanged unless you know what you are doing...)

YFrZUI3nEdudDI9h6sk4dA

Figure 31: Create a new Subscription – step 3

- Click the button **Save** to commit the new Subscription. The grid **Subscriptions** should now contain an entry for the new Subscription:

Alerts **Subscriptions** E-Mail > ^

New ▶

	Name	Description	Active	Event type	Topic
	New Activities		<input checked="" type="checkbox"/>	[1] Object Creation	Activity Modifications

Figure 32: Create a new Subscription – step 4

- This completes the creation of this new Subscription.



Please note that the Root administrator must start the Subscription Handler – otherwise you will not get any Alerts/Notifications.

7.2 Example Subscription with Filtering

In combination with openCRX security the subscription filter feature enables you to provide highly specific subscriptions. Imagine the following situation: there are 2 ActivityTrackers **DivisionA:ProjectX** and **DivisionA:ProjectY** and some of your users are interested in receiving notifications related to activities of ProjectX only, some users want to receive notifications related to activities of ProjectY only, and some users want to receive notifications from both projects. Such a situation could be handled as follows:

- create a **PrincipalGroups** **DivisionA.ProjectX** and **DivisionA.ProjectY**
- assign **PrincipalGroup** **DivisionA.ProjectX** to **ActivityTracker** **DivisionA:ProjectX**; like this new activities assigned to this Tracker will also be assigned the **PrincipalGroup** **DivisionA.ProjectX**
- assign **PrincipalGroup** **DivisionA.ProjectY** to **ActivityTracker** **DivisionA:ProjectY**; like this new activities assigned to this Tracker will also be assigned the **PrincipalGroup** **DivisionA.ProjectY**
- an Activity Modification subscription of a user wanting notifications related to ProjectX **and** ProjectY would look as follows:

Untitled - Subscription

The screenshot shows a web-based configuration interface for a subscription. It has two tabs: 'General' and 'System'. The 'General' tab is active. The form contains the following fields:

- Name:** ActivityModifications
- Description:** ProjectX only
- Active:**
- Event type:** (empty field)
- Topic:** Activity Modifications [Acti]
- Filters:**
 - Name of filter #0:** owner
 - Value of filter #0:** Standard:DivisionA.ProjectX, Standard:DivisionA.ProjectY
 - Name of filter #1:** (empty field)
 - Value of filter #1:** (empty field)

Figure 33: Create a Subscription with Filters

Enter the name of the attribute (**owner** in our example) into the name field and then enter the value(s) to match into the value field (**Standard:DivisionA.ProjectX** and **Standard:DivisionA.ProjectY** in our case)




Multiple values of a named filter are combined with **OR**.

Multiple named filters are combined with **AND**.

7.3 Trouble Shooting Notification Services

The following table lists some of the common issues and how to fix them:

Problem	Solution
The grids Workflow Processes and/or Topics are empty.	<ul style="list-style-type: none"> ▪ verify that the Subscription Handler of the respective segment was started at least once (Workflow Processes and Topics are created automatically by the Subscription Handler if they are not present) ▪ click on the filter button  to see all rows without filtering (maybe you defined a default filter in the past?)
I started the Subscription Handler but I never receive any Alerts / Notifications	<ul style="list-style-type: none"> ▪ verify that you started the correct Subscription Handler (each segment has its own Subscription Handler) ▪ in case you upgraded to a new version of openCRX and forgot to delete Workflows and Topics provided by openCRX, stop the Subscription Handler, delete Workflow Processes and Topics, and then start the Subscription Handler again ▪ check the openCRX log files to find out whether bad/corrupt data might be causing problems (e.g. NullPointerException during Workflow execution)
I receive Alerts on my Subscriptions but no Notification E-mails	<ul style="list-style-type: none"> ▪ verify that JavaMail is properly installed and the mail service properly configured ▪ verify your e-mail settings (see E-mail Services for details) ▪ verify that the Servlet WorkflowHandler of the respective segment is turned on

8 E-mail Services

Please note that you can use your favorite e-mail client with openCRX. None of our E-mail services are platform dependent and they work with any e-mail client and with any mail server as long as they support standard protocols like SMTP, POP3, POP3S, IMAP, IMAPS, etc.

Inbound and outbound E-mail services are based on JavaMail. Installation of JavaMail is not required to run openCRX, but it is required if you want to make use of openCRX E-Mail services.

The following figure shows the flow of mail messages between openCRX, mail server, and mail client as it is supported with openCRX 1.10.0:

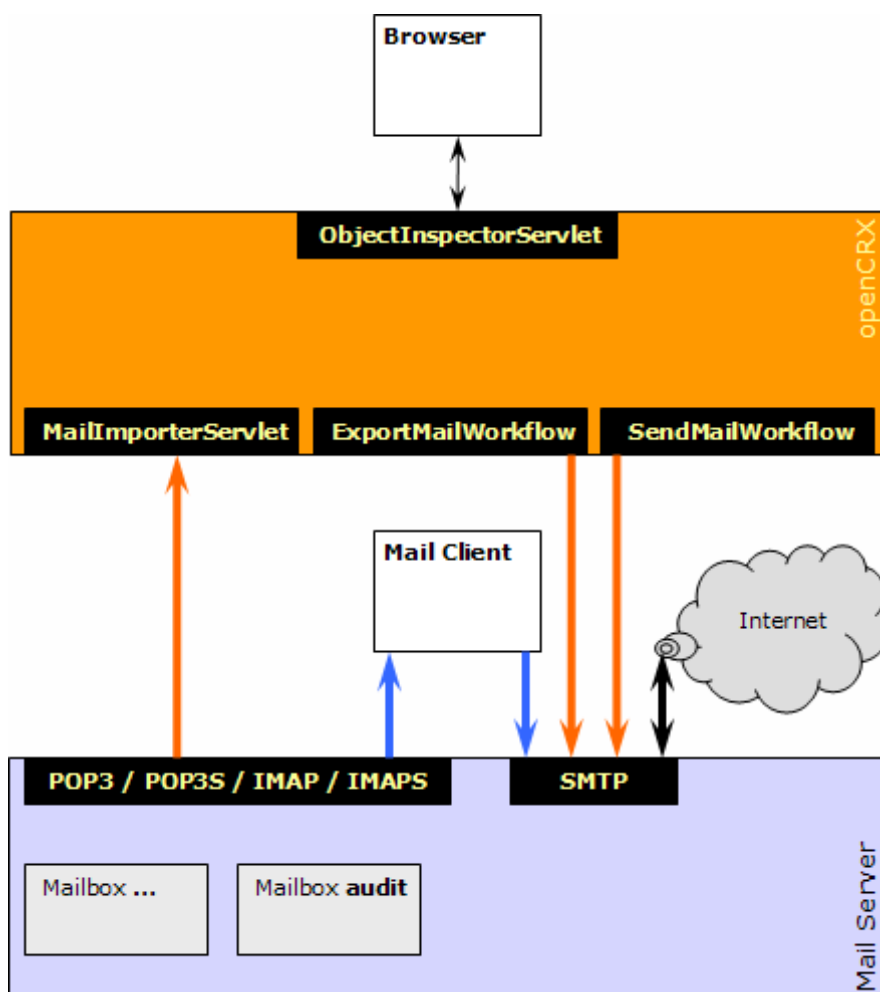


Figure 34: Flow of messages between openCRX, mail server and mail client

In the following few sections we will first discuss various important use cases and subsequently show how to configure openCRX in order to make use of the available functionality.

8.1 Import E-mails from a Mail Client into openCRX

Instead of offering platform specific plugins for a multitude of mail clients like MS Outlook, MS Outlook Express, Thunderbird, Evolution, Eudora, Elm, etc. openCRX features a platform neutral e-mail importer. The advantages are:

- works with any mail client (including your favorite one)
- no clumsy installation of plugins, i.e. you can get this to work on your company's laptop regardless of how "hardened" and locked down the system is
- supports single message import and bulk import
- imports headers, body, and attachments
- automatically creates links to sender and recipient(s) if the respective e-mail addresses are present in openCRX

The following figure shows an overview of how you can import e-mails from your mail client into openCRX:

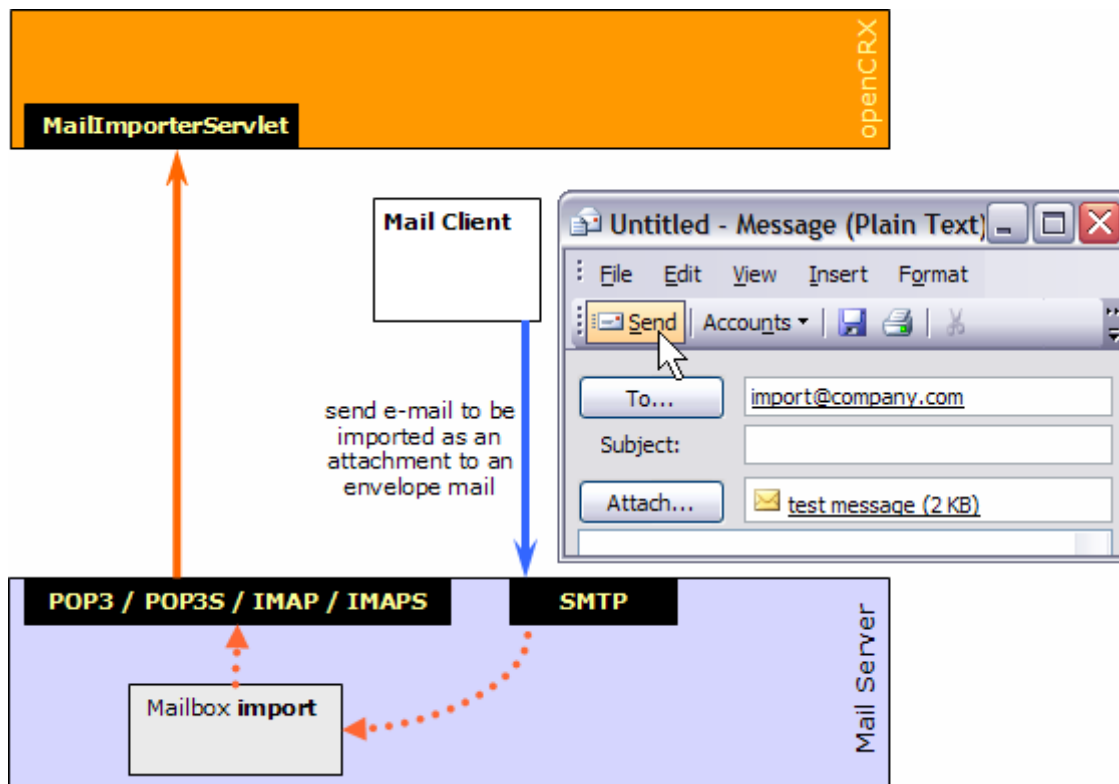


Figure 35: Import E-Mails from Mail Client

The whole setup is quite straightforward; in a first step you configure the **MailImporterServlet** (see Inbound E-mail) so that it fetches e-mails from a mailbox, e.g. named "import". Optionally, you can create a custom-tailored **Activity Creator** to handle imported E-mails exactly the way you like, but in most cases the provided **Default E-mail Creator** is sufficient. This default **Activity Creator** assigns imported E-Mails to the **Activity Tracker** E-Mails.

To import an e-mail message from your mail client into openCRX:

1. Create a new message to be sent to your importer mailbox, e.g. by entering `import@company.com` into the **TO** field of the new message.
2. Optionally you can specify the name of the **Activity Creator** in the **Subject** of the new message (this must be an exact match!).
3. Next you **attach** the message(s) to be imported to that new message (yes, you can attach multiple e-mail messages and if those messages contain attachments themselves they will also be imported) and send it.

Once delivered to the appropriate mailbox (called "import" in our example) the **MailImporterServlet** will fetch it from there and then import the messages attached to that envelope message. This process works for messages in any of your mail client's folders, e.g. Inbox, Outbox, Sent, Trash, etc.

8.2 Use openCRX as an E-mail Archive/Audit Tool

openCRX can easily keep track of all your e-mail traffic, inbound and/or outbound (and given the increasingly more stringent rules on e-mail retention – Sarbanes-Oxley, etc. – it is probably worthwhile considering the advantages of importing **all** e-mail messages by default).

The following figure shows a configuration where the mail server puts a copy of each received message (inbound traffic) and all sent messages (outbound traffic) into the mailbox audit; configuring such audit accounts can easily be done with most Mail Transport Agents (MTAs) like `qmail`, `postfix`, etc. With the appropriate configuration (see Inbound E-mail), the **MailImporterServlet** can import all messages from that audit mailbox and attach it to an Activity Tracker of your choice:

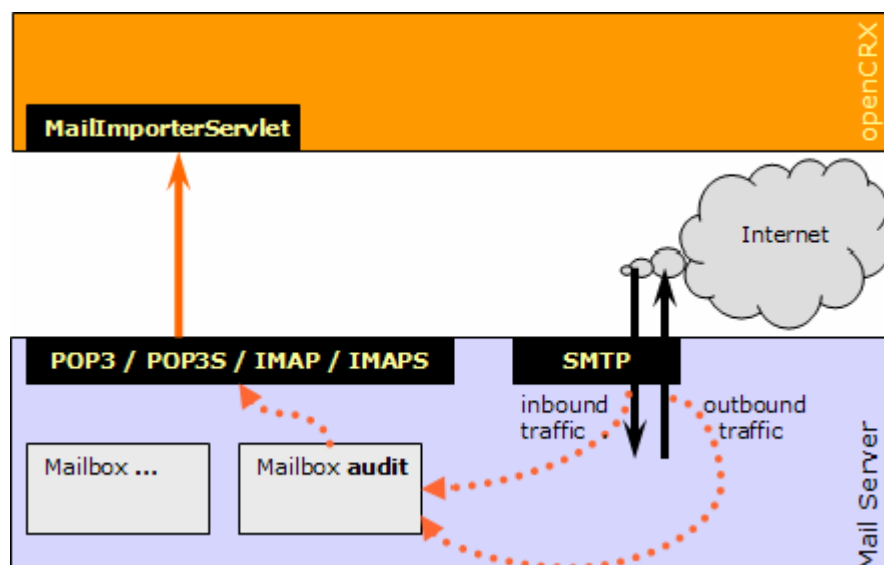


Figure 36: E-Mail Audit – import all inbound/outbound e-mail messages

8.3 Send E-mail directly from openCRX

Any openCRX **E-Mail Activity** can be sent as e-mail directly from openCRX:

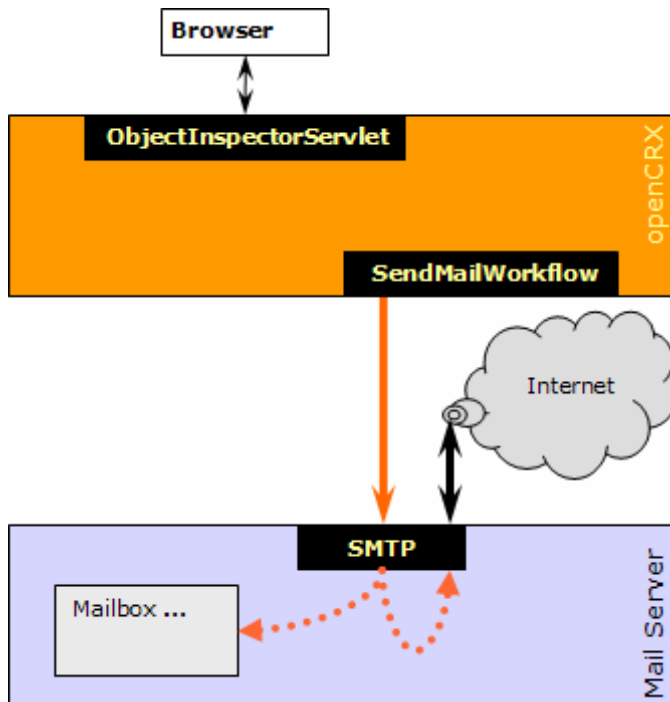


Figure 37: Send E-Mail from openCRX – Overview

The idea behind this functionality is less that you will use openCRX as a mail client, rather the **SendMailWorkflow** is an important element of the openCRX campaign management functionality. **E-Mail Activities** of type “E-Mails” are controlled by the **Activity Process E-mail Process**. Send E-Mail Activities to all recipients by executing the operation **Actions > Follow Up** and then selecting the Transition **Send as mail** as shown below:

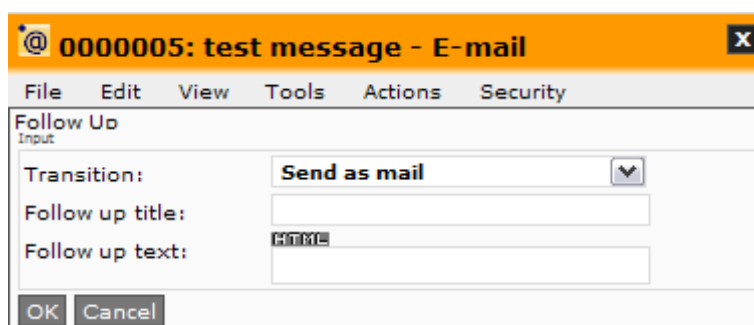


Figure 38: Send E-Mail from openCRX with Actions > Follow Up



Important

Please note that the transition “**Send as mail**” is only available after the Transition “**Assign**” has been executed.



Tip

Media attached to E-Mail Activities are sent as e-mail attachments.

8.4 Export E-mails from openCRX to your Mail Client

Any openCRX **E-Mail Activity** can be exported to your mail client:

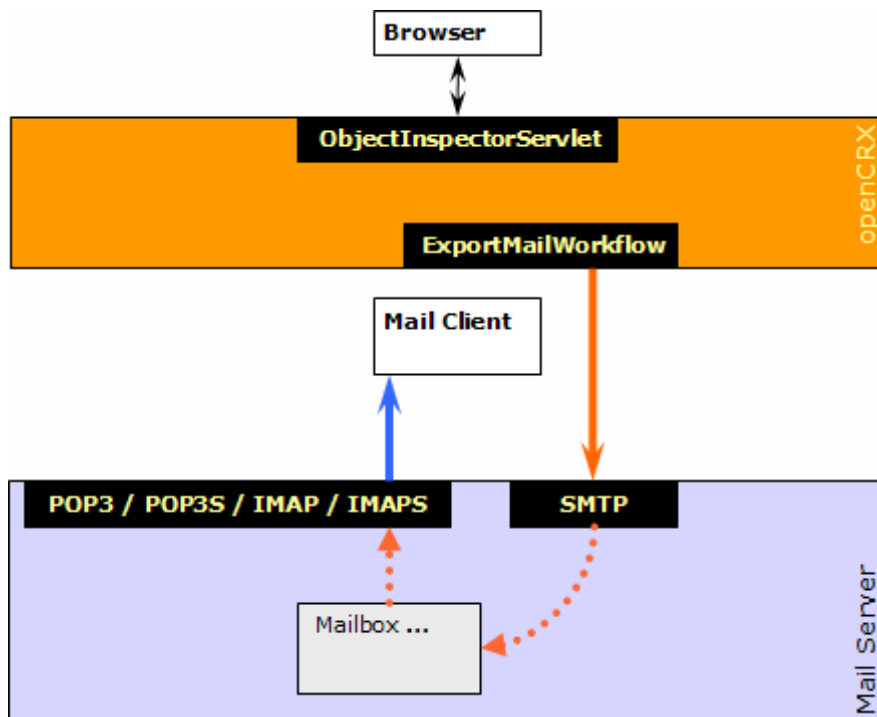


Figure 39: Export E-Mail from openCRX – Overview

The idea behind this functionality is that you might want to put some finishing touches on an e-mail before you actually send it from your mail client. **E-Mail Activities** of type “E-Mails” are managed by the standard **Activity Process E-mail Process**, i.e. they can be exported to the user's default mail account by executing the operation **Actions > Follow Up** and then selecting the Transition **Export as mail attachment**:

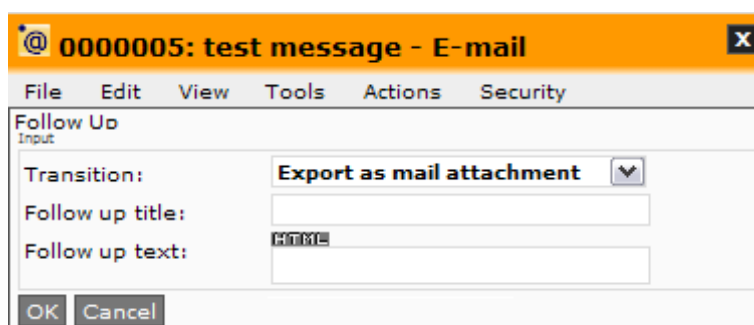


Figure 40: Export E-Mail from openCRX with Actions > Follow Up



Important

Please note that the transition “**Export as mail attachment**” is only available after the Transition “**Assign**” has been executed.



Tip

Media attached to E-Mail Activities are sent as e-mail attachments.



Please note that the exported message is sent as an attachment to the e-mail address specified as the user's default e-mail address. See Outbound E-mail for information on how to specify such an outbound e-mail address.

The following figure shows an example envelope e-mail with the attachment “test message.msg” representing the exported E-Mail Activity:

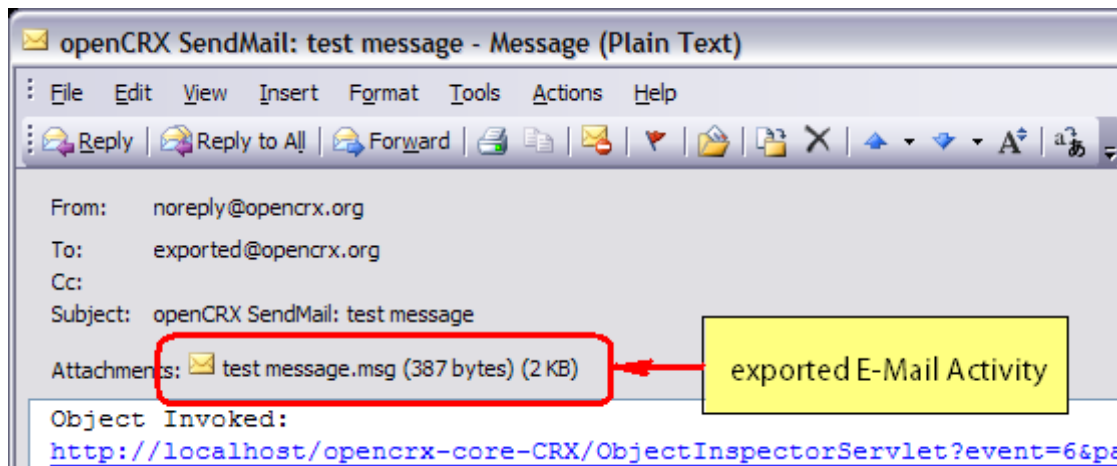


Figure 41: Envelope E-mail with exported E-Mail Activity as attachment

With a few simple steps you can typically open and edit the exported message and then send it. The following example shows how to do this with MS Outlook (but the steps are similar for other mail clients):

- double-click on the attachment “test message.msg” to open it
- in the menu bar of the newly opened message “test message” select **Actions > Resend this message** as shown in the figure below:

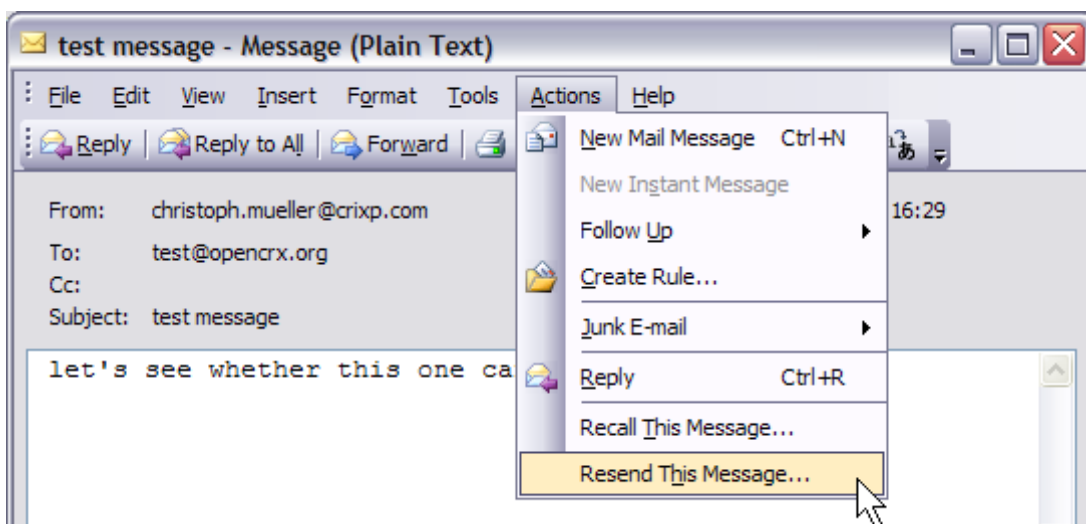


Figure 42: MS Outlook – Resend This Message to edit/send e-mail

- Now you can edit/change the test message as you like.

- Once you are ready to send the message, click the button Send:

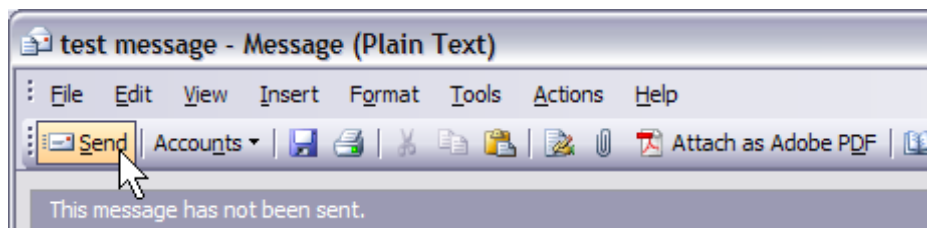


Figure 43: MS Outlook – Send message

8.5 Install and Configure Mail Resource and E-Mail Services

The following chapters explain how to install JavaMail and how to configure the Java mail service and various in- and outbound E-mail services.



Please note that outbound E-Mail Services depend on the Servlet WorkflowHandler of the respective segment being turned on.

8.5.1 Installation of JavaMail and JAF

Detailed installation instructions are provided at the JavaMail home:

<http://java.sun.com/products/javamail/FAQ.html>

And here is the short version:

- Download JavaMail (at least version 1.4) from <http://java.sun.com/products/javamail/downloads/index.html>
- Put the file **mail.jar** into the directory `JAVA_HOME\jre\lib\ext`
- Download JAF (JavaBeans Activation Framework, version 1.1 or newer) from <http://java.sun.com/products/javabeans/glasgow/jaf.html>
- Put the file **activation.jar** into the directory `JAVA_HOME\jre\lib\ext`



Verify that you put the files into the above-mentioned directory, which is **not** an openCRX installation directory!

8.5.2 Create mail-service.xml

Next you need to create the file **mail-service.xml** in the directory (on JBoss) `d:\pgm\jboss-4.0.5.GA\server\default\deploy` (Windows) or `/opt/jboss/server/default/deploy` (Linux). In the sample file below you must at least adapt the **highlighted strings** to your own environment:

Listing 9: File *mail-service.xml*

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE server>
<server>
  <mbean code="org.jboss.mail.MailService" name="jboss:service=Mail">
    <attribute name="JNDIName">java:/mail/provider/CRX</attribute>
    <attribute name="User">test</attribute>
    <attribute name="Password">test</attribute>
    <attribute name="Configuration">
      <configuration>
        <!-- Outbound -->
        <property name="mail.transport.protocol" value="smtp"/>
        <property name="mail.smtp.starttls.enable" value="true"/>
        <property name="mail.smtp.auth" value="true"/>
        <property name="mail.smtp.host" value="mail.dot.org"/>
        <property name="mail.smtp.port" value="25"/>
        <property name="mail.smtp.user" value="test"/>
        <property name="mail.from" value="noreply@dot.org"/>
        <!-- Inbound POP3 -->
        <property name="mail.store.protocol" value="pop3"/>
        <property name="mail.pop3.host" value="mail.dot.org"/>
        <property name="mail.pop3.port" value="110"/>
        <property name="mail.pop3.auth" value="true"/>
        <property name="mail.pop3.user" value="test"/>
        <property name="mail.pop3.password" value="test"/>
        <!-- Inbound POP3S -->
        <!--
        <property name="mail.store.protocol" value="pop3s"/>
        <property name="mail.pop3s.host" value="mail.dot.org"/>
        <property name="mail.pop3s.port" value="995"/>
        <property name="mail.pop3s.auth" value="true"/>
        <property name="mail.pop3s.user" value="test"/>
        <property name="mail.pop3s.password" value="test"/>
        -->
        <!-- Inbound IMAP -->
        <!--
        <property name="mail.store.protocol" value="imap"/>
        <property name="mail.imap.host" value="mail.dot.org"/>
        <property name="mail.imap.port" value="143"/>
        <property name="mail.imap.auth" value="true"/>
        <property name="mail.imap.user" value="test"/>
        <property name="mail.imap.password" value="test"/>
        -->
        <!-- Inbound IMAPS -->
        <!--
        <property name="mail.store.protocol" value="imaps"/>
        <property name="mail.imaps.host" value="mail.dot.org"/>
        <property name="mail.imaps.port" value="993"/>
        <property name="mail.imaps.auth" value="true"/>
        <property name="mail.imaps.user" value="test"/>
        <property name="mail.imaps.password" value="test"/>
        -->
        <property name="mail.debug" value="true"/>
      </configuration>
      <depends>jboss:service=Naming</depends>
    </attribute>
  </mbean>
</server>

```

Additional information about configuration options of JavaMail are available from the JavaMail home:

<http://java.sun.com/products/javamail/FAQ.html>



It is advisable to leave the debug option set to true until you have the Mail Services working as desired. In debug mode you will get helpful information about all the activities of the Mail Service on the console and/or in the respective log file including connection attempts, messages in the INBOX, etc.

8.5.3 Mail Resource in web.xml and jboss-web.xml

In the file **web.xml** in the directory (JBoss on Windows)

d:\pgm\jboss-4.0.5.GA\server\default\deploy\opencrx-core-CRX-Web.ear\opencrx-core-CRX.war\WEB-INF\

or (JBoss on Linux)

/opt/jboss/server/default/deploy/opencrx-core-CRX-Web.ear/opencrx-core-CRX.war/WEB-INF/

you must uncomment the following section:

Listing 10: Uncomment mail resource definition in **web.xml**

```
...
<!-- Wizards, Workflows (e.g. MailWorkflow), etc. can use mail resources.
     Configure a mail resource for each used mail resource. -->
<resource-ref id="mail_opencrx_CRX">
  <res-ref-name>mail/provider/CRX</res-ref-name>
  <res-type>javax.mail.Session</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
...
```

Similarly, in the file **jboss-web.xml** in the directory (JBoss on Windows)

d:\pgm\jboss-4.0.5.GA\server\default\deploy\opencrx-core-CRX-Web.ear\opencrx-core-CRX.war\WEB-INF\

or (JBoss on Linux)

/opt/jboss/server/default/deploy/opencrx-core-CRX-Web.ear/opencrx-core-CRX.war/WEB-INF/

you must uncomment the following section:

Listing 11: Uncomment mail resource definition in **jboss-web.xml**

```
...
<!-- Configure a resource-ref for each mail resource configured in web.xml -->
<resource-ref>
  <res-ref-name>mail/provider/CRX</res-ref-name>
  <jndi-name>java:/mail/provider/CRX</jndi-name>
</resource-ref>
...
```

Restart the application server for these changes to become active.



If you want to enable TLS / SSL connections to your mail server you must (in addition to enabling the appropriate options in the file **mail-service.xml**) also import the mail server's public key into the file cacerts of your JRE:

Listing 12: Importing certificate into keystore **cacerts**

```
keytool -keystore cacerts -import -storepass changeit -file mailserver.cer
```

8.5.4 Outbound E-mail

openCRX users can configure e-mail accounts on their Homepage indicating where they would like to receive e-mail notifications (e.g. generated by subscriptions):

- Click on **My Homepage** and select the grid Tab **E-Mail**.
- Next you click on the creator menu **New > E-Mail Account** to create a new E-mail Account:

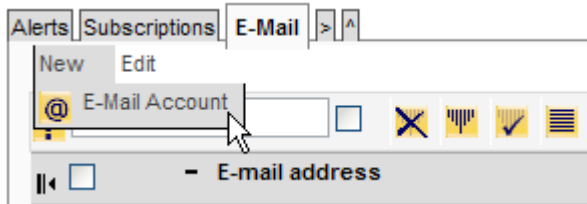


Figure 44: Create a new E-Mail Account – step 1

- Now you can configure your E-Mail Account for outbound e-mail service:
Untitled - E-Mail Account

Figure 45: Create a new E-Mail Account – step 2

The various fields have the following meanings:

- **E-mail address:** enter your e-mail address (i.e. the address where you would like to receive e-mail notifications)
- **Reply address:** enter a reply address (is also used for the From field)
- **Default:** check if this is your default e-mail address (notifications will only be sent to your default e-mail address)
- **Outgoing Mail Service:** leave this empty unless the default configuration does not suit you (the default name of the mail service is `/mail/<provider>/<segment>`)
- **Incoming Mail Service:** leave this empty unless the default configuration does not suit you (the default name of the mail service is `/mail/<provider>/<segment>`)



If a user does not define the name of the mail service in his **E-Mail Account** settings the default name `/mail/<provider>/<segment>` is used; if there is no resource with this name the fallback name `/mail/<provider>` is used (and if this name does not exist either, then an error is logged).

You might also want to consult the chapter Mail Resource in `web.xml` and `jboss-web.xml` for details on how to define Mail Resources.

- Click the button **Save** to commit the new E-Mail Account. The grid **E-Mail** should now contain an entry for the new E-Mail Account:

	- E-mail address	- Reply address
@	user@mycompany.com	noreply@mycompany.com

Figure 46: Create a new E-Mail Account – step 3

- On your Homepage you can provide additional information related to E-Mail Notifications:

guest, - My Homepage

General	More Charts	Options	System	*
E-mail Options				
E-mail subject prefix:	crx-CRX	Web access URL:	http://www.my.com/opencrx-core-	

Figure 47: E-mail subject prefix and Web access URL

The meaning of the two fields is as follows:

- E-mail subject prefix:** enter a string that might help you identify or filter e-mails from your openCRX server (optional, i.e. you can also leave this empty) – the entered string is prepended to the subject line of generated e-mails
- Web access URL:** enter the URL of the openCRX instance at hand; if entered correctly, generated e-mails will contain URLs that allow you to connect to your openCRX server with a single click

You can easily test your e-mail settings if you create a subscription for Activity Modifications (see Example Subscription – Activity Modifications) and then work through the following steps:

- create a new account (e.g. a new contact)
- navigate to your Homepage and check whether you actually received an alert related to the newly created account

- next click on the Grid Tab **Pending / Completed Workflows** on your homepage (unhide it by clicking on [**>**] if it is not visible)
- there should be (at least) two entries (you might have to sort the column **Started on** to locate recent entries):
 - `org.opencrx.kernel.workflow.SendAlert` (which generated the Alert)
 - `org.opencrx.mail.workflow.SendMailNotificationWorkflow` (which was responsible for sending the E-mail Notification)
- click on the icon of the respective grid icon to inspect the corresponding Workflow Process object
- the grid **Action Log Entries** contains the message body of the e-mail that was sent or an error message if the workflow failed (please note that even if you see a "timeout" error message the e-mail might have been sent; timeouts are typically caused by e-mail servers with high latency – try sending out notifications through a mail server that is responsive).

The screenshot shows the 'Action Log Entries' window with a 'Properties' tab selected. Below the title bar is a 'New' button and a search bar. The main area contains a table with the following data:

Name	Correlation	Description
openCRX SendMail: test 2		Object Invoked: http://localhost/opencrx-core-CRX/ObjectInspector By Workflow: http://localhost/opencrx-core-CRX/ObjectInspector Triggered By Event: OBJECT_CREATION Triggered By Subscription: mxxecMQBEdqQW4VT5Pz34w http://localhost/opencrx-core-CRX/ObjectInspector
Can not send mail: MessagingException		Can't send command to SMTP host; nested exception is: java.net.SocketException: Software caused connec
Can not send mail: MessagingException		Could not connect to SMTP host: mail.crixp.com, pc

Figure 48: Example of outbound E-mail Action Log Entries

8.5.5 Inbound E-mail

The configuration of inbound E-mail is explained in detail in the chapters MailImporterServlet and Import E-mails. You can test your settings as follows:

- Send an e-mail to the account specified in the file mail-service.xml (e.g. import@company.com). Attach the message to be imported to this “envelope” e-mail (please note that the attachment is imported by the MailImporterServlet, not the “envelope” e-mail):

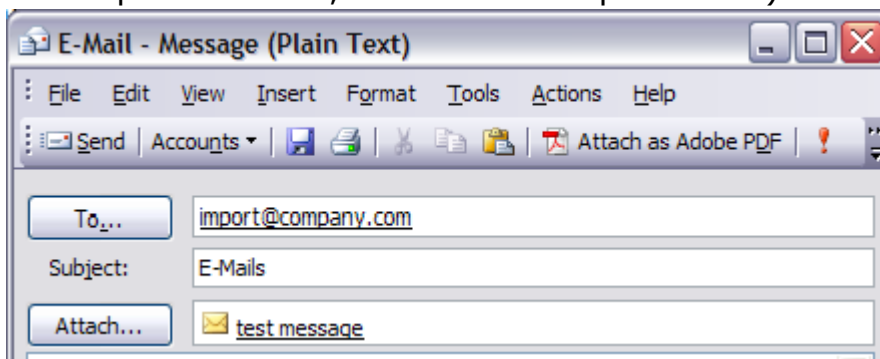


Figure 49: Envelope E-Mail with attached E-Mail to be imported

- Once the Workflow Controller triggers the MailImporterServlet you will see the debug output of the servlet on the console (on JBoss also in the file **server.log** if you set the debug option to true in mail-service.xml):

Listing 13: Debug Output of MailImporterServlet

```

18:42:57,779 INFO [STDOUT] DEBUG: JavaMail version 1.3.3
18:42:57,795 INFO [STDOUT] DEBUG: java.io.FileNotFoundException: D:\jdk1.4.2\jre\lib\javamail.providers
(The system cannot find the file specified)
18:42:57,795 INFO [STDOUT] DEBUG: !anyLoaded
18:42:57,795 INFO [STDOUT] DEBUG: not loading resource: /META-INF/javamail.providers
18:42:57,810 INFO [STDOUT] DEBUG: successfully loaded resource: /META-INF/javamail.default.providers
18:42:57,810 INFO [STDOUT] DEBUG: Tables of loaded providers
18:42:57,810 INFO [STDOUT] DEBUG: Providers Listed By Class Name: {
com.sun.mail.smtp.SMTPSSLTransport=javax.mail.Provider
[TRANSPORT,smtps,com.sun.mail.smtp.SMTPSSLTransport,Sun Microsystems, Inc],
com.sun.mail.smtp.SMTPTransport=javax.mail.Provider
[TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Sun Microsystems, Inc],
com.sun.mail.imap.IMAPSSLStore=javax.mail.Provider
[STORE,imaps,com.sun.mail.imap.IMAPSSLStore,Sun Microsystems, Inc],
com.sun.mail.pop3.POP3SSLStore=javax.mail.Provider
[STORE,pop3s,com.sun.mail.pop3.POP3SSLStore,Sun Microsystems, Inc],
com.sun.mail.imap.IMAPStore=javax.mail.Provider
[STORE,imap,com.sun.mail.imap.IMAPStore,Sun Microsystems, Inc],
com.sun.mail.pop3.POP3Store=javax.mail.Provider
[STORE,pop3,com.sun.mail.pop3.POP3Store,Sun Microsystems, Inc]}
18:42:57,810 INFO [STDOUT] DEBUG: successfully loaded resource: /META-INF/javamail.default.address.map
18:42:57,810 INFO [STDOUT] DEBUG: !anyLoaded
18:42:57,810 INFO [STDOUT] DEBUG: not loading resource: /META-INF/javamail.address.map
18:42:57,810 INFO [STDOUT] DEBUG: java.io.FileNotFoundException:
D:\jdk1.4.2\jre\lib\javamail.address.map
(The system cannot find the file specified)
18:42:57,810 INFO [STDOUT] DEBUG: setDebug: JavaMail version 1.3.3
18:42:57,826 INFO [STDOUT] DEBUG POP3: connecting to host "mail.company.com", port 995, isSSL true
18:42:58,654 INFO [STDOUT] S: +OK Dovecot ready.
18:42:58,654 INFO [STDOUT] C: USER xxxxxxxx
18:42:58,670 INFO [STDOUT] S: +OK
18:42:58,670 INFO [STDOUT] C: PASS xxxxxxxx
18:42:58,701 INFO [STDOUT] S: +OK Logged in.
18:42:58,717 INFO [STDOUT] C: STAT
18:42:58,748 INFO [STDOUT] S: +OK 1 3204
18:42:58,748 INFO [STDOUT] C: NOOP
18:42:58,779 INFO [STDOUT] S: +OK
18:42:58,842 INFO [STDOUT] C: TOP 1 0
18:42:58,889 INFO [STDOUT] S: +OK
...

```

- If the MailImporterServlet successfully imported your test mail it will be attached to the **Activity Tracker E-Mails** (the MailImporterServlet creates this Activity Tracker automatically if it does not exist yet). Navigate to **Activities > Activity Trackers** and then click on the icon of **E-Mails**:

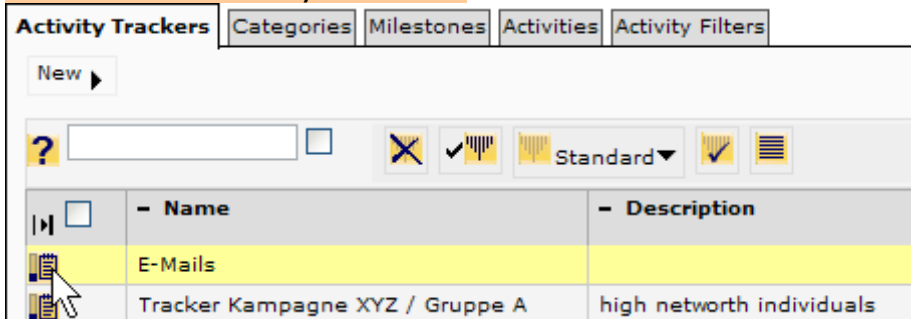


Figure 50: Activity Tracker E-Mail is created automatically

- By default, all imported e-mails are attached to the **Activity Tracker E-Mails** – you should also see the successfully imported test mail in the grid Activities:

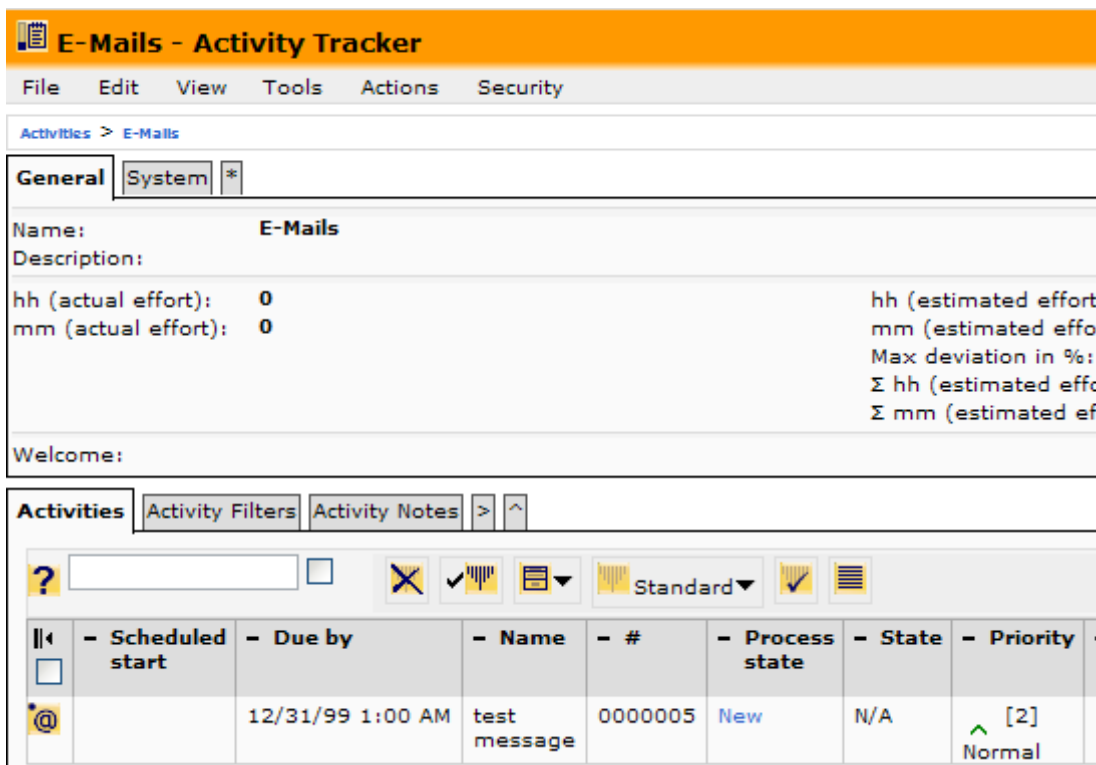


Figure 51: Activity Tracker E-Mail with newly imported e-mail

- Click on the icon of the newly imported e-mail to load it into the Inspector:

The screenshot shows the 'Inspector' window for an e-mail object. The title bar reads '@ 0000005: test message - E-mail'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Tools', 'Actions', and 'Security'. The breadcrumb path is 'Activities > 0000005: test message'. The main content area is divided into two sections: 'General' and 'Recipients'.

General section:

Name:	test message	State:	N/A
#:	0000005	Priority:	[2] Normal
Description:		Scheduled start:	
Assigned to:		Scheduled end:	
Miscellaneous #1:		Scheduled end (origin):	
Miscellaneous #2:		Due by:	12/31/99 1:00 AM
Miscellaneous #3:			
Activity type:	E-Mails	Actual start:	
Sender:	christoph.mueller@crixp.com	Actual end:	
Read receipt:		hh (actual effort):	0
Last Transition:		mm (actual effort):	0
Process state:	New	Sent:	4/9/06 7:19 PM
% complete:	0	Delivery receipt:	
Reporting contact:	admin-Standard,		
Subject:	test message		
Message:	let's see whether this one can be imported...		
Reporting account:			
Detailed description:			

Recipients section:

Buttons: New, Delete

Search: ? [input field] [checkbox] [X] [checkmarks] [list icon] [checkmark] [list icon]

	- E-mail Address(es)	- Type
@	test@opencrx.org	TO

Figure 52: Newly imported e-mail

The mail importer will automatically link imported e-mails with corresponding objects (if they exist in openCRX) and create various additional useful objects:

- e-mail address of sender --> Sender
- e-mail addresses of recipients --> Recipients
- e-mail headers --> Notes
- e-mail attachments --> Media



By default, the MailImporterServlet creates an **Activity Creator** **Default E-mail Creator** and an Activity Tracker **E-Mails**. The latter is referenced in the grid **Activity Groups** of the former:

*** Default E-Mail Creator - Activity Creator**

File Edit View Tools Actions Security

Activities > Default E-Mail Creator

General Details System *

Name: **Default E-Mail Creator**

Activity type: **E-Mails**

Description:

Activity Groups > ^

Search: [] [Q] [+] [-]

	- Name	- Scheduled date	- Desc
	E-Mails		

Figure 53: Activity Creator Default E-mail Creator

This is the reason why newly imported e-mails are shown in the grid Activities of the **Activity Tracker E-Mails**.



You can easily change the contents of the grid **Activity Groups** so that newly imported e-mails will be attached to different Activity Tracker(s). It is also possible to create additional **Activity Creators** with different behavior (just make sure that these Activity Creators create Activities of type **E-Mails**). With the the **subject line** of your envelope e-mail you can indicate which **Activity Creator** should be used to import your e-mail. If you omit the subject line the **Default E-mail Creator** is used.



Once the MailImporterServlet works as desired you can switch off the debugging output by adding a boolean property named "debug" (and setting it to false) to the respective grid of the MailImporterServlet.

8.5.6 Troubleshooting E-mail Services

The following table lists some of the common issues and how to fix them:

Problem	Solution
the grids Workflow Processes and/or Topics are empty	<ul style="list-style-type: none"> verify that the Subscription Handler of the respective segment was started at least once (Workflow Processes and Topics are created automatically by the Subscription Handler if they are not present) click on the filter button  to see all rows without filtering (maybe you defined a default filter in the past?)
the grid Workflow Processes does not contain an entry for the MailImporterServlet	<ul style="list-style-type: none"> verify that the MailImporterServlet was started at least once (see chapter MailImporterServlet for details). depending on your segment access levels this entry might be visible to the Segment Administrator only (e.g. admin-Standard); see also <i>Figure 7: Table kernel_Segment after modification</i>
openCRX does not initiate TLS session with mail server	<p>It seems that JavaMail sometimes does not (even try to) establish a TLS session when connecting to a mail server (smtp) if the certificate of the mail server has not been imported into the keystore. If the mail server requires TLS for authentication (e.g. SASL) and authentication is required to relay messages such failure to establish a TLS session will prevent openCRX from properly sending outbound mail.</p> <div data-bbox="742 1518 845 1742" style="border: 1px solid black; padding: 5px; width: fit-content;">  <p>Tip</p> </div> <p>If you intend to use TLS/SSL to secure the connection to the outbound e-mail server (smtp) we recommend you import the mail server certificate into the keystore.</p> <p><i>Listing 14: Importing Certificate</i></p> <pre data-bbox="742 1809 1476 1859">cd \$JAVA_HOME/lib/security keytool -import -alias <dom> -file <name>.cer -keystore cacerts</pre> <p>Replace <dom> with the domain of the mail server (e.g. mail.company.com) and <name> with the name of the certificate file.</p>

Problem	Solution
I receive Alerts on my Subscriptions but no Notification E-mails	<ul style="list-style-type: none">▪ verify that JavaMail is properly installed▪ verify your e-mail settings (see E-mail Services for details)▪ verify that the Servlet WorkflowHandler of the respective segment is turned on▪ verify that you actually have a subscription for the topic Alert Modifications (with openCRX v1.10 other topics do not include the action SendMailNotificationWorkflow anymore, i.e. you will receive alerts only without a subscription to the topic Alert Modifications)

9 Data Import/Export

There are many ways of importing data (from other systems into openCRX) and exporting data (from openCRX to other systems). Generally speaking, there is no best way of doing imports/exports because depending on how much weight you put on the pros and cons of the various methods you may come to a different conclusion. Some issues to consider are:

- one-time import/export vs. multiple imports/exports
- file-based/batched vs. connection-based/real-time
- allow manual process steps vs. fully automated
- ...

In this chapter we will cover some of the basic options you can choose from, but there are obviously other (and sometimes better) options to consider.



Tip

While it may be tempting to connect to the openCRX database for “quick and dirty” imports/exports, you should really consider using the openCRX API. On the one hand, importers/exporters accessing the database directly are **bypassing openCRX security** (this is actually more of a warning than a tip...). On the other hand, the **openCRX database schema is subject to change** (whereas the API is stable).

9.1 Importing Data into openCRX

The task of importing data is handled by importers. In principle, you can import almost anything into openCRX, it’s really only a matter of writing an appropriate importer.



Warning

You must ensure that (legal) values are assigned to all mandatory (i.e. non-optional) attributes of openCRX objects created/updated during the import; in particular, all code attributes are mandatory!

The Open Source distribution of openCRX includes importers for vCard (see Importing vCard Files) and iCalendar files (see Importing iCalendar Files) in addition to the XML importer.

9.2 Importing XML Files

You can import virtually any data into openCRX as long as you provide it in the form of schema-compliant XML files. The openCRX schema files can be found in the file `opencrx-1.10.0\jre-1.4\core\lib\opencrx-kernel.jar` (unzip and

look for xmi subdirectories). Alternatively, you can export example objects as XML files and look at the produced XML files (although the generated XML file also contains all the derived and optional attributes; hence, you will have to prune the generated XML file before you can use it as a template).

Some of the configuration information and data provided with openCRX are also provided in the form of XML files and imported during system setup (e.g. units of measurement are loaded from `opencrx-core-CRX-Web.ear\opencrx-core-CRX.war\WEB-INF\config\data\Root\ uom_SI_and_Paper.xml`).

An XML import from a third party system might typically involve the following steps:

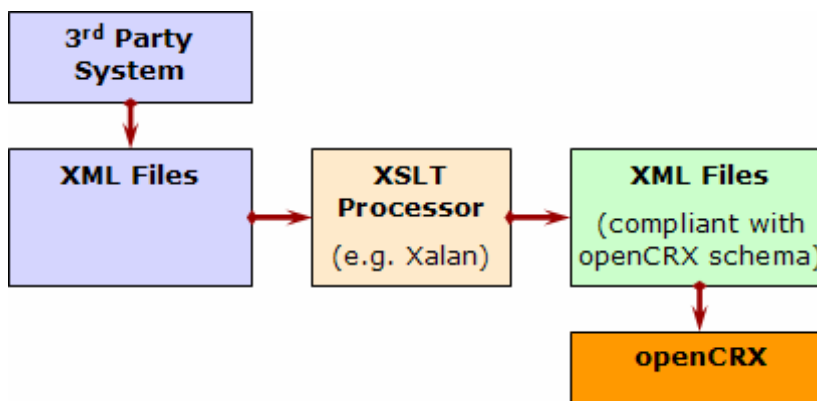


Figure 54: XML import from 3rd party system – overview

You can import schema-compliant XML files with the following methods:

- **Interactive / on-demand**

Navigate to your user home and execute the operation `File > Import`:

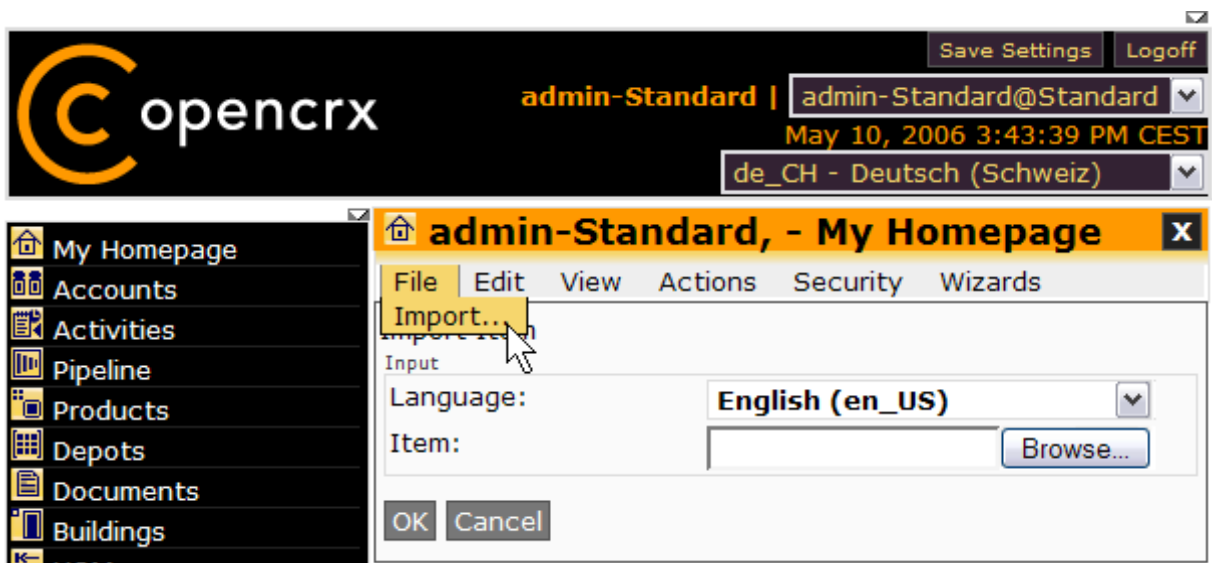


Figure 55: Interactive import of XML Files

Click on the button **Browse...** and navigate to XML file to be imported. Next you click OK to start the import. Please note that this method is very suitable for small XML files and *on-the-fly* imports. If you are dealing with larger XML files, however, you should consider the bulk import described below.

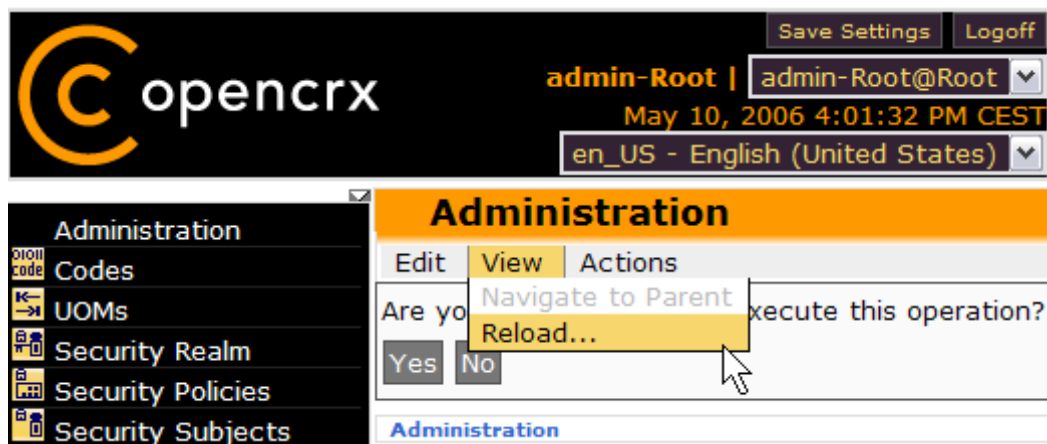
● Bulk Import

Use the bulk import for large XML files or if you need to import multiple XML files in an automated fashion. Put your XML file(s) into the following directory (you might have to expand the EAR file to do so):

`opencrx-core-CRX-Web.ear\opencrx-core-CRX.war\WEB-INF\config\data\<<SegmentName>`

where `<SegmentName>` can be Root, Standard, or whatever your Segment is named.

Next you login as openCRX Root administrator (**admin-Root**) and execute the operation **View > Reload**. Click Yes to start the operation.



● *Figure 56: Interactive import of XML Files*



Once the import is started you can close the browser, i.e. there is no need to keep the session alive until the import is completed. Some information regarding the progress of the import is written to the console.



In case you have data dependencies between/among your XML files (e.g. some files contain Contact data while others contain address data which is composite to Contact data) you should make sure that **parent data are imported before child data** gets imported. This should be relatively easy to achieve as XML files are imported in alphabetical order.

9.2.1 Importing vCard Files (→ openCRX Contacts)

vCard is file format standard for personal data interchange, specifically electronic business cards (additional information is for example available from <http://en.wikipedia.org/wiki/VCard>).

These are the steps to import a vCard file:

- click on the provider Accounts
- select the operation File > Import vCard to unhide the import dialog:

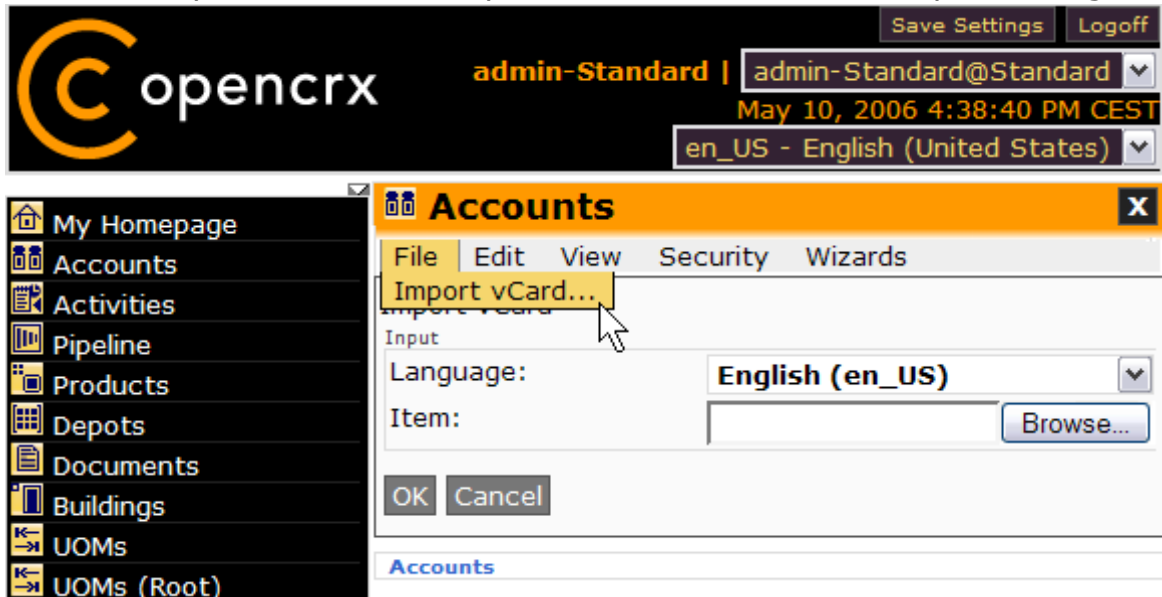


Figure 57: Operation vCard Import

- select the appropriate language
- click the Browse button and navigate to the vCard file you want to import
- click the OK button to start the import operation

9.2.2 Importing iCalendar Files (→ openCRX Meetings)

iCalendar is a standard for calendar data exchange (additional information is for example available from <http://en.wikipedia.org/wiki/ICalendar>).

These are the steps to import an iCalendar file:

- click on the provider Activities
- select the operation File > Import iCal to unhide the import dialog
- select the appropriate language
- click the Browse button and navigate to the iCalendar file to import
- click the OK button to start the import operation

9.2.3 Other Options

There are various other options to consider. You could for example develop a custom-tailored **JSP Wizard** to import data on demand or on a regular basis (e.g. controlled by the openCRX WorkflowController).

Sometimes it is more appropriate to develop a specific openCRX client to handle imports, and in a typical enterprise class environment you will probably consider developing adapters to connect/integrate openCRX with 3rd party systems on a real-time basis.

9.3 Exporting Data from openCRX

The task of exporting data is handled by exporters. The Open Source distribution of openCRX includes exporters for vCard and iCalendar files in addition to the XML exporter.

This allows you to export contacts and meetings/sales visits or any other object from openCRX. vCard and iCalendar files can be imported by a large variety of other applications, including Microsoft Outlook. This chapter shows how to export data.

9.3.1 Exporting XML Files

Navigate to the object to be exported as XML file and execute the operation **File > Export XML** as shown below:

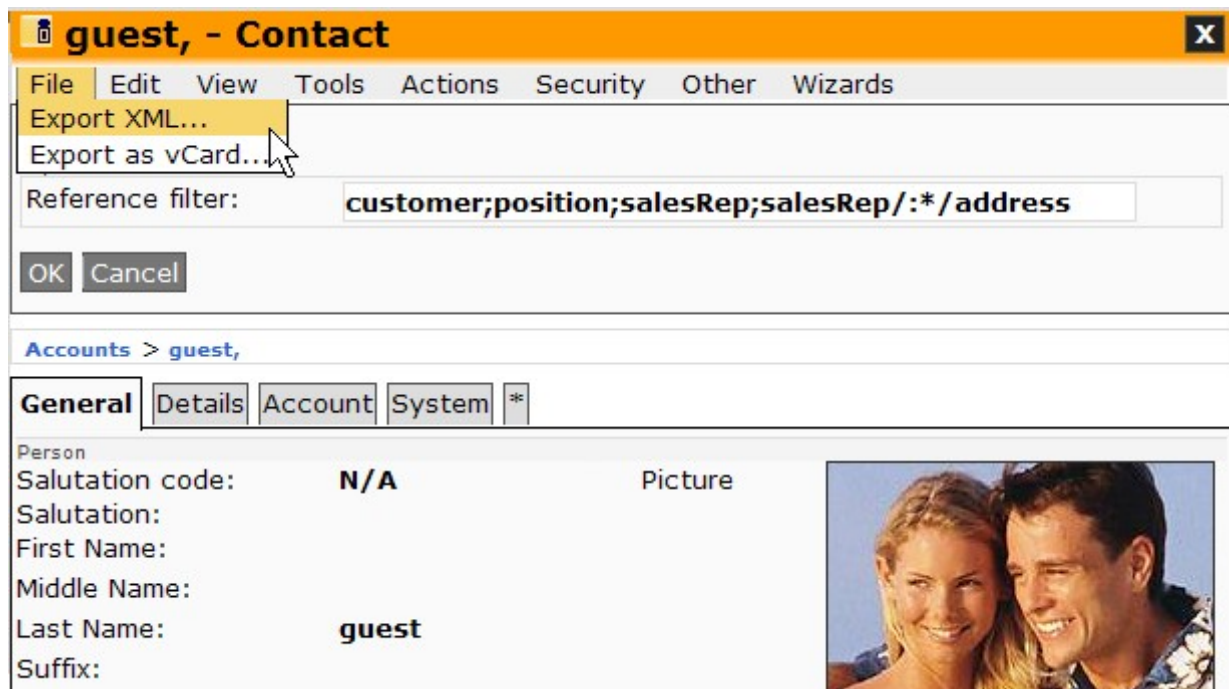


Figure 58: Exporting Contact as XML File

In order to better control which additional objects (composites, referenced objects, ...) the XML exporter should export together with the object loaded in the Inspector, you can (optionally) provide a **reference filter**. The default reference filter is `:*/*` meaning that all composites and referenced objects up to 2 levels deep will be exported together with the main object (this should be sufficient for most use cases). Limiting the scope of the export by providing a reference filter as in the example above might be necessary to limit the size of the XML file.

If the export is successful the exporter will terminate with status OK and you will be provided with a link to a zip file containing the raw data and all the referenced code tables:

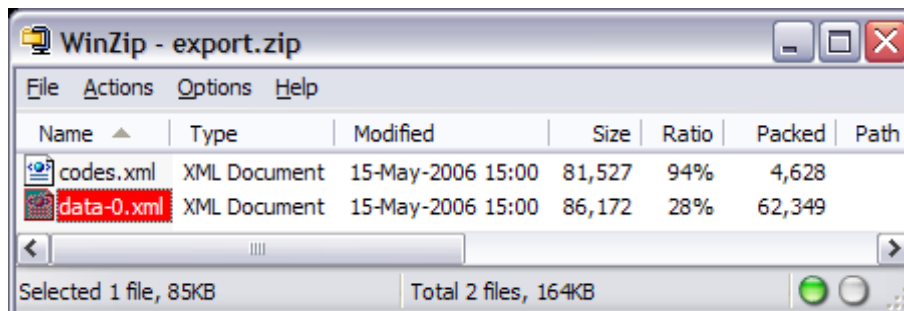


Figure 59: XML Exporter provides XML data file and code tables as ZIP file

9.3.2 Exporting openCRX Contacts (→ vCard Files)

These are the steps to export a contact to a vCard file:

- navigate to the contact you want to export
- select the operation **File > Export as vCard** to unhide the export dialog
- select the appropriate language
- click to button [OK] to start the export operation

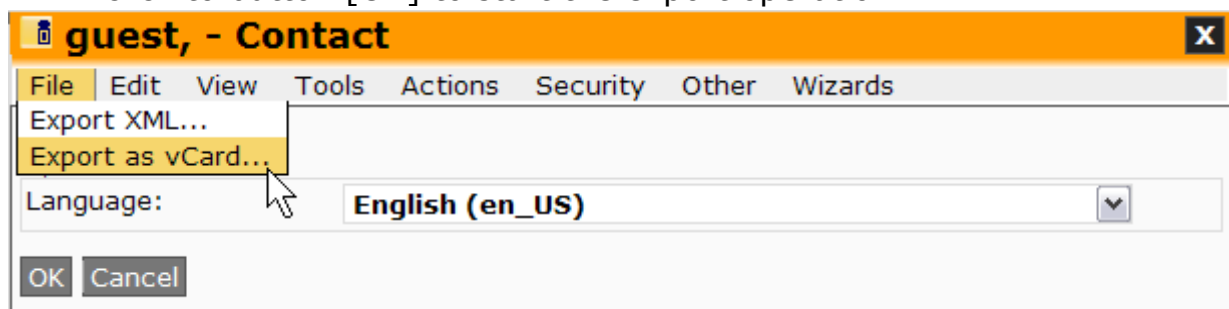


Figure 60: Export Contact as vCard

If the export operation was successful the result will contain a link to the vCard file. Click on that link to download the vCard file from the openCRX server.

9.3.3 Exporting openCRX Contacts (→ Outlook Contacts)

Navigate to the contact you want to export to MS Outlook and execute the Wizard **Export to MS Outlook**:



Figure 61: Export Contact to MS Outlook



Please note that this wizard requires that Internet Explorer and MS Outlook are both installed on your computer (the export involves an ActiveX control). With other browser you might consider exporting contacts to MS Outlook by using vCards.

The Wizard creates a new MS Outlook Contact from the openCRX Contact. If you want to add it permanently to your MS Outlook file, simply click the button [Save and Close], otherwise close the Contact window to discard:

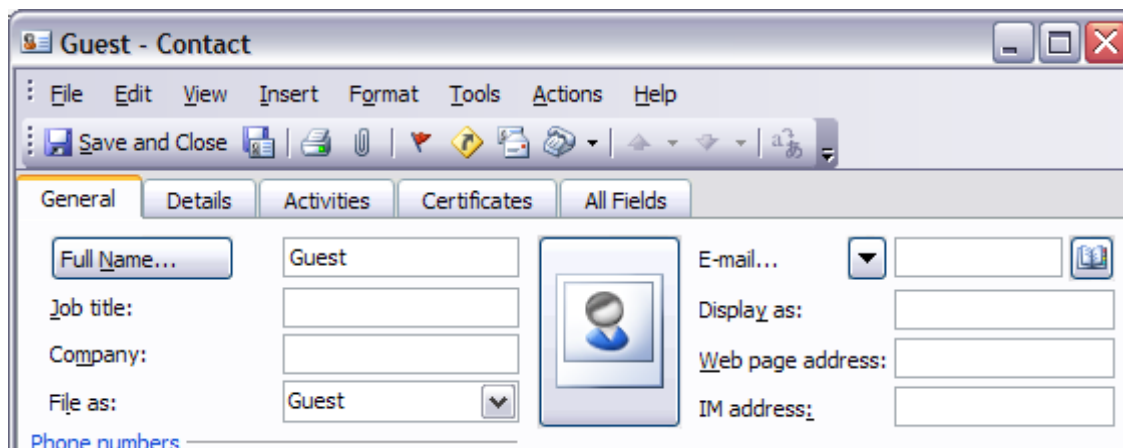


Figure 62: MS Outlook Contact created from openCRX Contact

An animation is available at

<http://www.opencrx.org/opencrx/1.9/new.htm#ops>

9.3.4 Exporting openCRX Meetings (→ iCalendar Files)

These are the steps to export a meeting (or a sales visit) to an iCalendar file:

- navigate to the meeting (or sales visit) you want to export
- select the operation **File > Export as iCal...** to unhide the export dialog
- select the appropriate language
- click to button [OK] to start the export operation

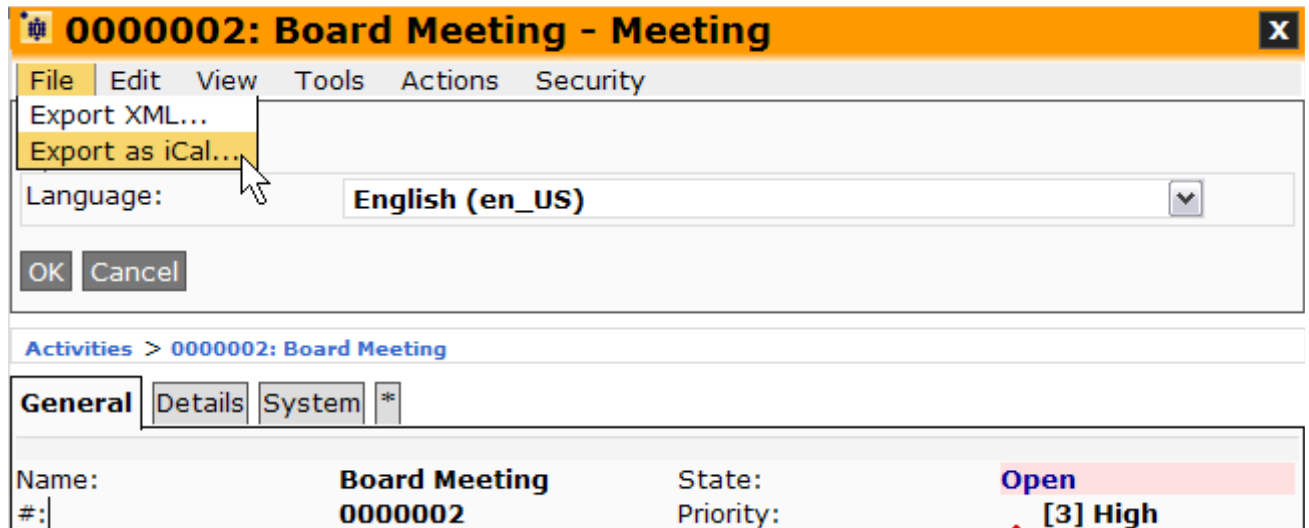


Figure 63: Exporting Meeting / Sales Visit as iCalendar File

9.3.5 Other Options

There are various other options to consider. You could for example develop a custom-tailored JSP Wizard to export data on demand or on a regular basis (e.g. controlled by the openCRX WorkflowController).

Sometimes it is more appropriate to develop a specific openCRX client to handle exports, and in a typical enterprise class environment you will probably consider developing adapters to connect/integrate openCRX with 3rd party systems on a real-time basis.

10 Customizing openCRX

Please refer to the guides available at <http://www.opencrx.org/documents.htm> for detailed information regarding UI customization and localization.

10.1 Managing Locales

The default installation of openCRX activates all locales that are included in the Open Source distribution. The openCRX administrator may wish to deactivate certain locales from the locale list. This chapter shows how you can achieve this.

The locale list is contained in the file

`opencrx-core-CRX-Web.ear\opencrx-core-CRX.war\WEB-INF\web.xml`

Look for the section `<!-- locales -->` to find a list of available locales:

Listing 15: Locales in web.xml

```
<!-- locales -->
<init-param>
  <param-name>locale[0]</param-name>
  <param-value>en_US</param-value>
</init-param>
<init-param>
  <param-name>locale[1]</param-name>
  <param-value>de_CH</param-value>
</init-param>
<init-param>
  <param-name>locale[2]</param-name>
  <param-value>es_MX</param-value>
</init-param>
...
```

You can deactivate locales by simply commenting them out. The following example shows how to deactivate the locale `de_CH`.

Listing 16: Activating/Deactivating Locales in web.xml

```
<!-- locales -->
<init-param>
  <param-name>locale[0]</param-name>
  <param-value>en_US</param-value>
</init-param>
<!--
<init-param>
  <param-name>locale[1]</param-name>
  <param-value>de_CH</param-value>
</init-param>
-->
<init-param>
  ...
```



Please note that you must **not** deactivate the base locale (that is the locale with the id 0, typically `en_US`) as the base locale contains a lot of customizing information not present in other locales.

10.2 Managing Packages

The default installation of openCRX activates all packages that are included in the Open Source distribution. The openCRX administrator may wish to deactivate certain packages if they are not used. This chapter shows how you can achieve this.

The package list is contained in the file

`opencrx-core-CRX-Web.ear\opencrx-core-CRX.war\WEB-INF\web.xml`

Look for the section `<!-- Admin -->` to find a list of available packages:

Listing 17: Locales in web.xml

```
<!-- Admin -->
<init-param>
  <param-name>rootObject[0]</param-name>
  <param-value>xri:@openmdx:org.opencrx.kernel.admin1/provider/CRX/segment/${SEGMENT}</param-value>
</init-param>
<!-- Home -->
<init-param>
  <param-name>rootObject[1]</param-name>
  <param-value>xri:@openmdx:org.opencrx.kernel.home1/provider/CRX/segment/${SEGMENT}/userHome/${USER}</param-value>
</init-param>
<!-- Accounts -->
<init-param>
  <param-name>rootObject[2]</param-name>
  <param-value>xri:@openmdx:org.opencrx.kernel.account1/provider/CRX/segment/${SEGMENT}</param-value>
</init-param>
...

```

You can deactivate packages by simply commenting them out. The following example shows how to deactivate the package depot1:

Listing 18: Activating/Deactivating Locales in web.xml

```
...
</init-param>
<!-- Depots -->
<!--
<init-param>
  <param-name>rootObject[6]</param-name>
  <param-value>xri:@openmdx:org.opencrx.kernel.depot1/provider/CRX/segment/${SEGMENT}</param-value>
</init-param>
-->
<!-- Documents -->
<init-param>
  <param-name>rootObject[6]</param-name>
  <param-value>xri:@openmdx:org.opencrx.kernel.document1/provider/CRX/segment/${SEGMENT}</param-value>
</init-param>
<!-- Buildings -->
...

```



Warning

Please note that you must **renumber** all the packages listed after the package you deactivated so that the package numbering does not have any gaps (i.e. **package numbering starts at 0 and it must be consecutive**).



Tip

It is also possible to change the order of the active packages by renumbering them. However, you must still ensure both that the numbering starts at 0 and that the numbering is consecutive.

10.3 Role-based UI

Requires Model Permissions (which are not implemented yet). The same goal can easily be achieved with Multiple Applications, however.

10.3.1 Model Permissions

Model permissions are not implemented yet.

10.3.2 Multiple Applications

Please refer to the chapter Multiple Applications.

10.3.3 Custom Layout JSPs

openCRX is distributed with 2 default layout JSPs located in the directory `opencrx-core-CRX-Web.ear\opencrx-core-CRX.war\WEB-INF\config\layout\en_US:`

- **show-Default.jsp**

This layout JSP renders all pages that show information (typically an Inspector containing information about the current object and all the grids containing associated information). This layout JSP is generic (it is provided by openMDX/portal) and it can handle any object.

- **edit-Default.jsp**

Similarly, this layout JSP renders all pages that are used to edit objects.

If you have a need for specialized screens for a particular object in edit and/or show mode, you can write your own layout JSP and deploy it to the above-mentioned directory. The file name of your custom layout JSP determines which objects (or rather: objects of which class) will be handled by your custom layout JSP.

Example:

Let's assume you want to replace the default edit screen for openCRX Contacts (i.e. class **org.opencrx.kernel.account1.Contact**) with a custom layout JSP. Name your file

edit-org.opencrx.kernel.account1.Contact.jsp

and deploy it to the directory `...\WEB-INF\config\layout\en_US`. After restarting your application server your new layout JSP will be active.



If you develop localized JSPs you can create new directories for the respective locales and then deploy your localized JSPs there. The fallback algorithms are comparable to those in ui customization.

11 Integration with Office Suites

openCRX provides various technologies that enable you to easily integrate common office suites like Open Office or Microsoft Office.

11.1 MS Office

11.1.1 MS Word

See information published at
<http://www.opencrx.org/opencrx/1.10/new.htm#rtf>

11.1.2 MS Excel

See information published at
<http://www.opencrx.org/opencrx/1.10/new.htm#xls>

11.1.3 MS Outlook

See chapters E-mail Services, Importing iCalendar Files, Importing vCard Files, Exporting openCRX Meetings (iCalendar Files), Exporting openCRX Contacts (vCard Files), Exporting openCRX Contacts (Outlook Contacts), and information published at <http://www.opencrx.org/opencrx/1.9/new.htm#ops>.

11.2 Open Office

11.2.1 OpenOffice Writer

See information published at
<http://www.opencrx.org/opencrx/1.10/new.htm#rtf>

11.2.2 OpenOffice Calc

See information published at
<http://www.opencrx.org/opencrx/1.10/new.htm#xls>

12 Next Steps

You might want to have a look at some of the additional documentation published at <http://www.opencrx.org/documents.htm>.