openCRX Admin Guide

Version 2.3

www.opencrx.org

22-Dec-2008 @ 03:03:38 PM

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 2008 © CRIXP Corp. All rights reserved.

Table of Contents

1 About this Book....8
1.1 Who this book is for....8
1.2 What do you need to understand this book....8
1.3 Tips, Warnings, etc....8
2 Prerequisites....9
3 Security....10
3.1 Introduction....10
3.1.1 Basic Concepts and Conventions....10
3.1.2 Permissions / Access Control....13
3.1.3 The SQL approach to understanding security....15
3.2 Activating Security....16
3.3 Default Settings....17
3.4 Security Settings of New Objects....18
3.5 Checking Permissions....18
3.6 Login Procedure....19
3.6.1 Apache Tomcat / Application Server Login....20
3.6.2 Segment Login....20
3.6.3 Disabling Login....20
3.7 Resetting Security....21
4 Managing Users....22
4.1 Creating Users....22
4.1.1 Create Users as Segment Administrator....23
4.1.2 Import Subjects and Application Login Principals....24
4.1.3 Import Users....25
4.2 Disable/Deactivate Users....26
4.2.1 Disable Users at the level Tomcat /Application Server....26
4.2.2 Disable Users at the level openCRX....26
5 Deployment Scenarios....27
5.1 Typical Deployment Scenarios....27
5.2 Multi Entity Deployment Scenarios....28
5.2.1 Multiple Data Segments in a single DB....28
5.2.2 Multiple DBs....29
5.3 openCRX Custom Applications....29
6 Workflow Controller....30
6.1 Workflow Controller Configuration....32
6.1.1 Startup Configuration in web.xml....32
6.1.2 ServerURL....33
6.1.3 Handler pingrate and autostart....33
6.2 Servlet IndexerServlet....34
6.3 Servlet SubscriptionHandler....34
6.4 Servlet WorkflowHandler....35

6.5 Trouble Shooting Servlets....36
7 Subscribe / Notify Services....37
7.1 Example Subscription – Account Modifications....39
7.2 Example Subscription with Filtering....40
7.3 Trouble Shooting Notification Services....41
8 E-mail Services....42
8.1 Install and Configure Mail Resource and E-Mail Services....43
8.1.1 Installation of JavaMail and JAF....43
8.1.2 Mail Resource for openCRX on Apache Tomcat....43
8.1.2.1 Add resource definition to opencrx-core-CRX.xml....43
8.1.2.2 Mail Resource in web.xml....44
8.1.3 Mail Resource for openCRX on JBoss....45
8.1.3.1 Create mail-service.xml....45
8.1.3.2 Mail Resource in web.xml and jboss-web.xml....46
8.2 Outbound E-mail....47
8.2.1 Outbound E-mail Configuration....47
8.2.2 Export E-mails....49
8.2.3 Send E-mails directly from openCRX....50
8.2.4 Send E-mails as Attachments to your Mail Client....51
8.2.5 Send E-mails to Fax-/SMS-Gateways....52
8.3 Inbound E-mail....53
8.3.1 Import E-mails....53
8.4 Trouble Shooting E-mail Services....53
9 Groupware Services....54
9.1 Directory Service / LDAP....54
9.1.1 Configuring the openCRX LDAP Port....55
9.1.2 LDAP Configuration of Thunderbird....55
9.1.3 LDAP Configuration of MS Outlook....56
9.2 openCRX vcard Servlet....57
9.2.1 Account Selectors....57
9.2.1.1 Connecting MS Outlook to the openCRX vcard servlet....57
9.2.1.2 Connecting Thunderbird to the openCRX vcard servlet....57
9.3 Calendaring....58
9.3.1 Calendar as a Set of Activities....58
9.3.2 Calendar Selectors....59
9.3.3 openCRX Activities mapped to Calendar Events....60
9.3.4 Calendaring / Free Busy....62
9.3.4.1 Free Busy Configuration of Thunderbird....62
9.3.4.2 Free Busy Configuration of MS Outlook....62
9.3.5 Calendaring / iCalendar (ICS)....63
9.3.5.1 ICS Configuration of Thunderbird/Lightning and Sunbird....63
9.3.5.2 ICS Configuration of MS Outlook....64
9.3.5.3 Deleting Events....64
9.3.5.4 iCalender Guard Event....65

9.3.5.5 VEVENT vs. VTODO....65
9.3.6 Calendaring / Timeline....66
9.4 Mailstore / IMAP....67
9.4.1 Configuring the openCRX IMAP Port....69
9.4.2 Configuring the IMAP Maildir Cache....69
9.4.2.1 Maildir Configuration with Apache Tomcat....69
9.4.2.2 Maildir Configuration with JBoss....69
9.4.3 IMAP Configuration of Thunderbird....70
9.4.4 IMAP Configuration of MS Outlook....71
10 openCRX is a REST Service (Web Service)....73
11 Data Import/Export....74
11.1 Importing Data into openCRX....74
11.2 Importing XML Files....74
11.2.1 Importing vCard Files (→ openCRX Contacts)....77
11.2.2 Importing E-Mails....77
11.2.3 Other Options....77
11.3 Exporting Data from openCRX....78
11.3.1 Exporting XML Files....78
11.3.1.1 ICS Configuration of Thunderbird/Lightning and Sunbird....79
11.3.2 Exporting Data to MS Excel / OO Calc Files....79
11.3.3 Exporting openCRX Contacts (→ vCard Files)....80
11.3.4 Exporting openCRX Contacts (→ Outlook Contacts)....80
11.3.5 Exporting openCRX Meetings (→ iCalendar Files)....82
11.3.6 Exporting E-Mails....82
11.3.7 Other Options....82
12 Customizing openCRX....83
12.1 Managing Locales....83
12.2 Managing Packages....84
12.2.1 Enabling/Disabling Root Menu Entries....85
12.3 Role-based UI....86
12.3.1 Model Permissions....86
12.3.2 Custom Layout JSPs....86
13 Integration with Office Application....87
13.1 MS Word, OO Writer, etc....87
13.2 MS Excel, OO Calc, etc....88
13.3 MS Outlook, Thunderbird/Lightning, Sunbird, etc....88
14 Reporting....89
14.1 Standard Reports....89
14.2 Ad-hoc Reporting....90
14.3 Large Scale Reporting....90
15 Miscellaneous Topics....92
15.1 SNMP Monitoring....92
15.2 Tomcat w/ openCRX and LDAP-based Authentication....93
16 Next Steps....94

List of Figures

Figure 1: Security Realms, Principals and Subjects after Initial Setup.....11
Figure 2: Segment Administration.....12
Figure 3: Role Drop Down with list of available Segment Login Principals.....12
Figure 4: openCRX UML Model – Class Diagram SecureObject.....13
Figure 5: System attributes of an openCRX object as shown in the GUI.....14
Figure 6: Table OOCKE1_SEGMENT after default installation (QuickStart).....17
Figure 7: Table OOCKE1_SEGMENT after modification.....17
Figure 8: Result of Check Permissions.....19
Figure 9: Role Drop Down with list of available Segment Login Principals.....20
Figure 10: Operation Actions > Import Login Principals (admin-Root).....24
Figure 11: Operation Actions > Import Users (admin-Standard).....25
Figure 12: Disabling of Segment Login Principal guest by admin-Standard....26
Figure 13: 3-Tier with Apache Tomcat / LWC.....27
Figure 14: 3-Tier with J2EE-compliant Application Server.....27
Figure 15: 4-Tier with Clustered Application and DB Servers.....27
Figure 16: Multiple Data Segments in a single DB.....28
Figure 17: Dedicated DB for each Entity.....29
Figure 18: Accessing the openCRX Workflow Controller.....30
Figure 19: openCRX 2.3 Workflow Controller.....30
Figure 20: Default Configuration of WorkflowController.....31
Figure 21: openCRX Administration – WorkflowController.....32
Figure 22: Workflow Controller Configuration – serverURL.....33
Figure 23: Workflow Controller Configuration – pingrate and autostart.....33
Figure 24: Default Workflow Processes created by WorkflowHandler.....35
Figure 25: Event and Notification Service.....37
Figure 26: Standard Topics included in the openCRX distribution.....38
Figure 27: Create a new Subscription.....39
Figure 28: Create a Subscription with Filters.....40
Figure 29: Flow of e-mail messages between openCRX, MTA and mail client..42
Figure 30: Create a new E-Mail Account – step 1.....47
Figure 31: Create a new E-Mail Account – step 2.....47
Figure 32: Create a new E-Mail Account – step 3.....48
Figure 33: E-mail subject prefix and Web access URL.....48
Figure 34: Example of outbound E-mail Action Log Entries.....49
Figure 35: Send E-Mail from openCRX – Overview.....50

Figure 36: Send E-Mail from openCRX with Actions > Follow Up.....50
Figure 37: Send E-Mail as Attachment from openCRX – Overview.....51
Figure 38: Export E-Mail from openCRX with Actions > Follow Up.....51
Figure 39: Thunderbird LDAP Configuration.....55
Figure 40: MS Outlook LDAP Configuration.....56
Figure 41: openCRX Activity Groups.....58
Figure 42: openCRX Activity Filters.....59
Figure 43: An openCRX activity's iCal representation.....60
Figure 44: An openCRX activity in the standard GUI.....61
Figure 45: An openCRX activity filtered to a user's homepage.....61
Figure 46: iCalendar conversion between VEVENT and VTODO.....65
Figure 47: Timeline visualizes time-based events.....66
Figure 48: E-Mail Address UNASSIGNED.....68
Figure 49: Thunderbird IMAP Configuration.....70
Figure 50: Thunderbird IMAP Configuration.....71
Figure 51: XML import from 3rd party system – overview.....75
Figure 52: Interactive import of XML Files.....75
Figure 53: Interactive import of XML Files.....76
Figure 54: Operation vCard Import.....77
Figure 55: Exporting SalesOrder as XML File.....78
Figure 56: XML Exporter provides XML data file and code tables as ZIP file....78
Figure 57: Exporting SalesOrder as Spreadsheet File.....79
Figure 58: Exported Spreadsheet File.....80
Figure 59: Export Contact as vCard.....80
Figure 60: Export Contact to MS Outlook.....81
Figure 61: Exporting Meeting / Sales Visit as iCalendar File.....82
Figure 62: Launch Wizard User Settings.....85
Figure 63: Wizard User Settings – enable/disable Root Menu Entries.....85
Figure 64: RTF Document generated by merging live data with template.....87
Figure 65: Contacts Export Dialog.....88

List of Listings

Listing 1: File Format Subjects and Application Login Principals.....24
Listing 2: Example File Subjects and Application Login Principals.....24
Listing 3: File Format Users.....25
Listing 4: Example File Users.....25
Listing 5: web.xml – auto startup of the Workflow Controller.....32
Listing 6: Servlets managed by Workflow Controller log to server.log.....36
Listing 7: File opencrx-core-CRX.xml.....44
Listing 8: Uncomment mail resource definition in web.xml.....44
Listing 9: Importing certificate into keystore cacerts.....44
Listing 10: File mail-service.xml.....45
Listing 11: Uncomment mail resource definition in web.xml.....46
Listing 12: Uncomment mail resource definition in jboss-web.xml.....46
Listing 13: Importing certificate into keystore cacerts.....46
Listing 14: Importing Certificate.....53
Listing 15: Set org.opencrx.maildir for Apache Tomcat.....69
Listing 16: Set org.opencrx.maildir for Apache Tomcat.....69
Listing 17: Locales in web.xml.....83
Listing 18: Activating/Deactivating Locales in web.xml.....83
Listing 19: Packages in web.xml.....84
Listing 20: Activating/Deactivating Packages in web.xml.....84

1 About this Book

This book describes various configuration settings and tasks that make an openCRX administrator's life easier.

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 2.3 (for installation instructions, please refer to the guides available from http://www.opencrx.org/documents.htm).

You can either follow the openCRX Server Installer documentation (http://www.opencrx.org/server.htm) or you can do a manual installation of openCRX following the QuickStart guide.

3 Security

In this chapter we will present a high-level overview of openCRX security and discuss a few important 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. The **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" who 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 "real user" also has a Principal Group, e.g. "guest.Group".
- 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 installed openCRX Server or followed the QuickStart guide):

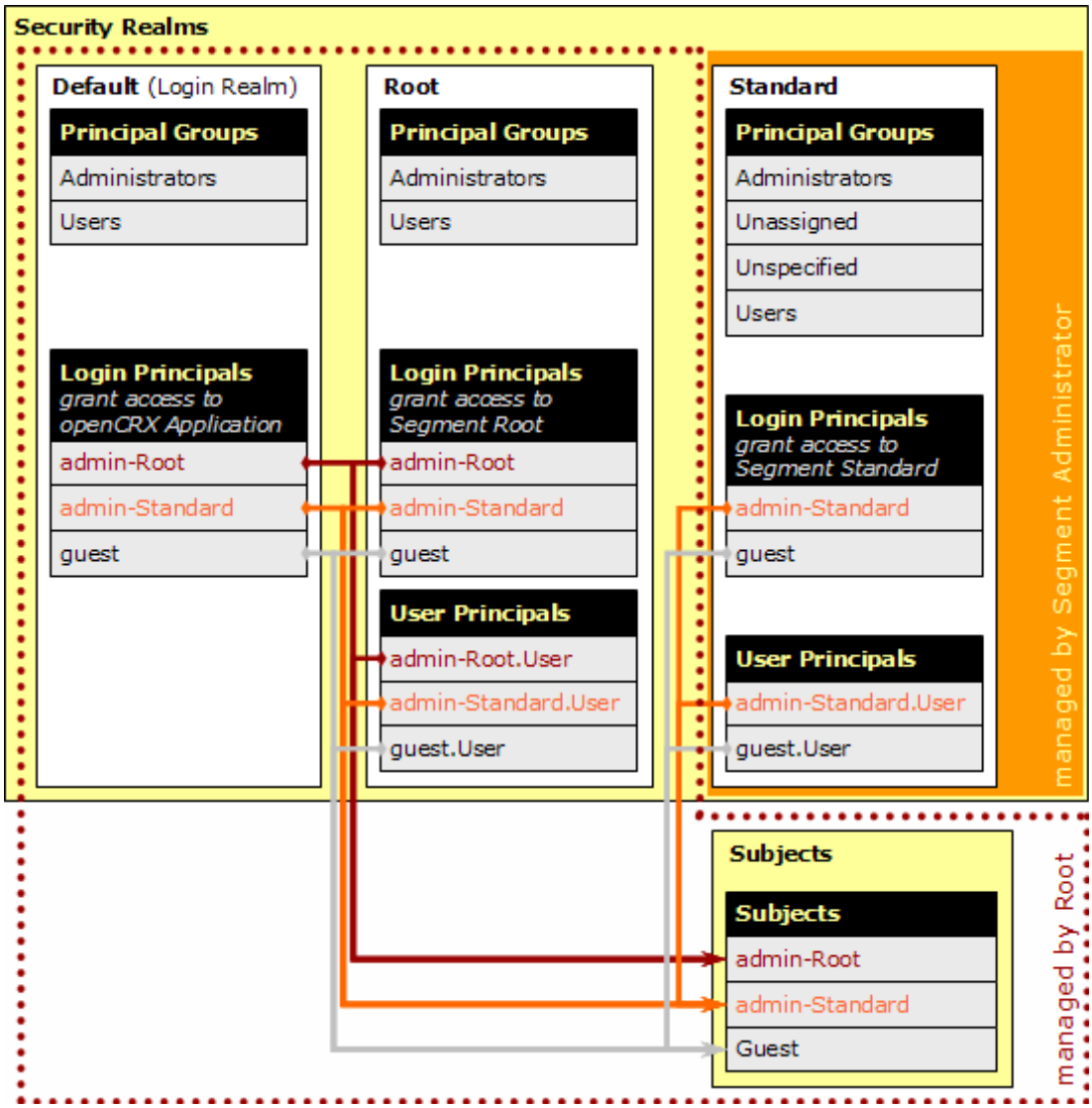


Figure 1: Security Realms, Principals and Subjects after Initial Setup

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; 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 home pages 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 for example required to find all activities and contracts assigned to the logged in principal.

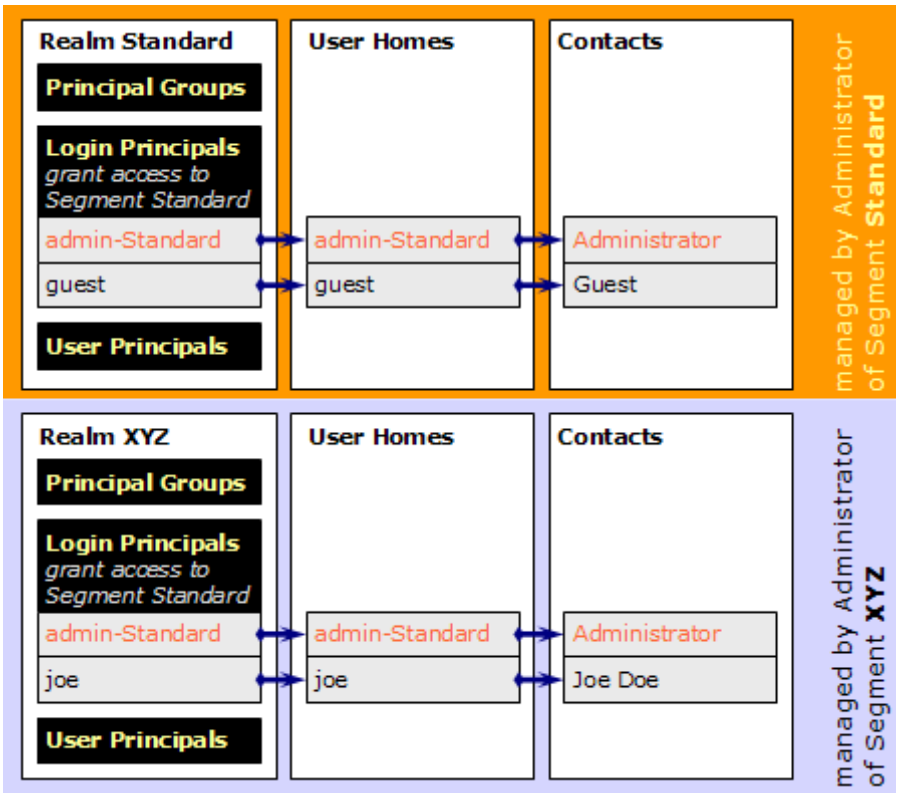


Figure 2: Segment Administration

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:

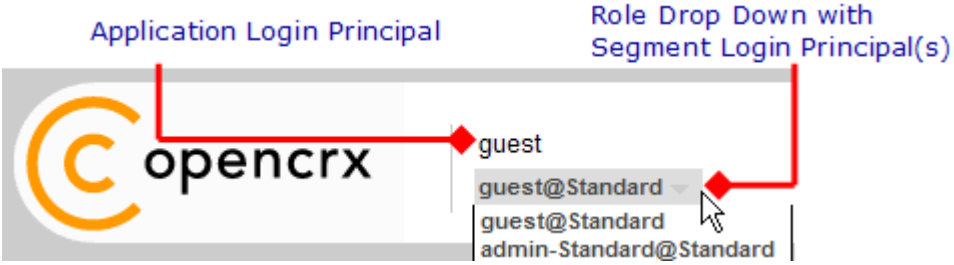


Figure 3: Role Drop Down with list of available Segment Login Principals

3.1.2 Permissions / Access Control

The openCRX security framework makes a clear distinction between **Ownership Permissions** (permissions granted on a particular object) and **Model Permissions** (permissions granted on a particular model element). As the latter are not implemented (yet) 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 (i.e. **Ownership access control**). 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):

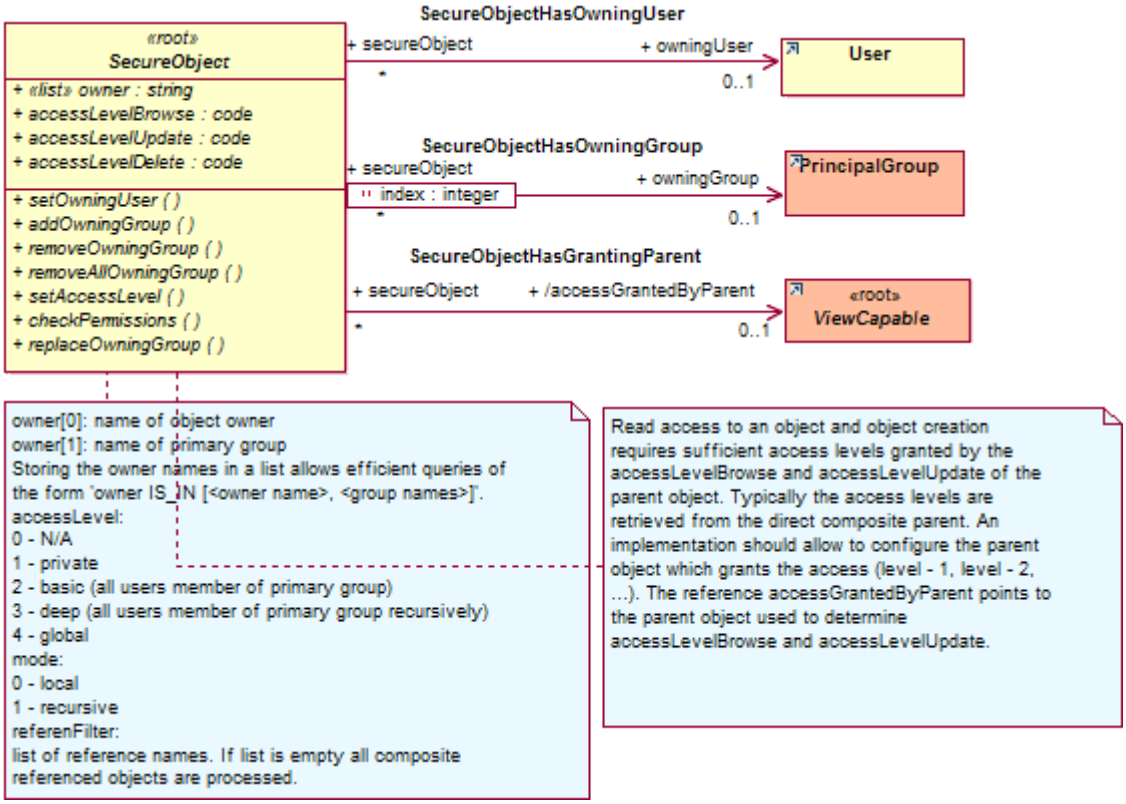


Figure 4: openCRX UML Model – Class Diagram SecureObject

If you see **N/P** in a reference 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 reference 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 respective 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 [in which case nobody has access – probably not a desirable situation]).
- **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. who can view/inspect direct composite objects of object X (including all their attributes)].

  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. who 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. who can change object X; this includes adding composite objects to object X].

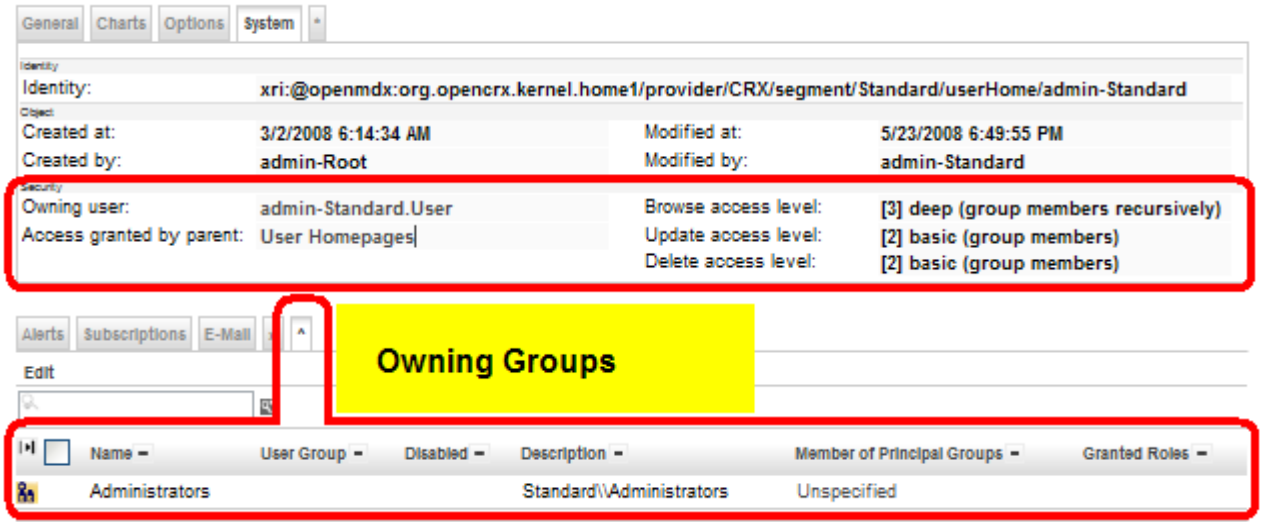


Figure 5: System attributes of an openCRX object as shown in the GUI

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: (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: (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

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

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

3.1.3 The SQL approach to understanding security

If you are familiar with SQL, the following approach to understanding security might be helpful. Let's put ourselves into the role of the AccessControl Plugin; accessing an object (read mode) results in a SELECT statement as follows:

SELECT * FROM T WHERE owner IN (p1, p2, ...)

- owner is a column that is present in all (multi-valued) tables xACCOUNT_, xADDRESS_, etc.) and it contains a list of principals who are permitted to access the respective object in read-mode
- the set P = {p1, p2, ...} is calculated by the AccessControl Plugin before accessing the object and it corresponds to the principals who are assigned to the current user based on the object's AccessLevel as shown

in the following table:

Access Level	Set P = {p1, p2, ...}
0 – N/A	P = {}
1 – private	P = P_p where P_p = {all groups directly assigned to the principal p}
2 – basic	P = P_p + P_{upper} where P_p = {all groups directly assigned to the principal p} P_{upper} = {all groups that contain at least one group contained in P_p}
3 – deep	P = P_p + P_{upper} + P_{lower} where P_p = {all groups directly assigned to the principal p} P_{upper} = {all groups that contain at least one group contained in P_p} P_{lower} = {all groups contained in P_{upper}}
4 – global	the where-clause "WHERE owner IN (p1, p2, ...)" is not required, i.e. the SELECT statement reduces to SELECT * FROM T

3.2 Activating Security

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

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 you own clients or adapters for openCRX.

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

3.3 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	ACCESS_LEVEL_BROWSE	ACCESS_LEVEL_UPDATE	ACCESS_LEVEL_DELETE
accounts/CRX/Standard	4	3	1
activities/CRX/Standard	4	3	1
admins/CRX/Root			
buildings/CRX/Standard	4	3	1
codes/CRX/Root	4	2	2
codes/CRX/Standard	4	3	1
contracts/CRX/Standard	4	3	1
depots/CRX/Standard	4	3	1
docs/CRX/Standard	4	3	1
forecasts/CRX/Standard	4	3	1
homes/CRX/Standard	4	3	1
models/CRX/Standard	4	3	1
products/CRX/Standard	4	3	1
uoms/CRX/Root	4	2	2
uoms/CRX/Standard	4	3	1
workflows/CRX/Standard	4	3	1

Figure 6: Table OOCKE1_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** (i.e. any user can browse all accounts, all activities, all documents, etc.).

These default settings are suitable for test environments and deployments in smaller companies/teams with a generous 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 OOCKE1_SEGMENT). After this change, the table OOCKE1_SEGMENT will look as shown in the following figure:

OBJECT_ID	ACCESS_LEVEL_BROWSE	ACCESS_LEVEL_UPDATE	ACCESS_LEVEL_DELETE
accounts/CRX/Standard	3	3	1
activities/CRX/Standard	3	3	1
admins/CRX/Root			

Figure 7: Table OOCKE1_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.

3.4 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 as well as) 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 Checking Permissions

You can check security permissions on any SecureObject with the operation Check Permissions. Provide the principal name as a parameter. The following figure shows the result of the operation on a Contact:

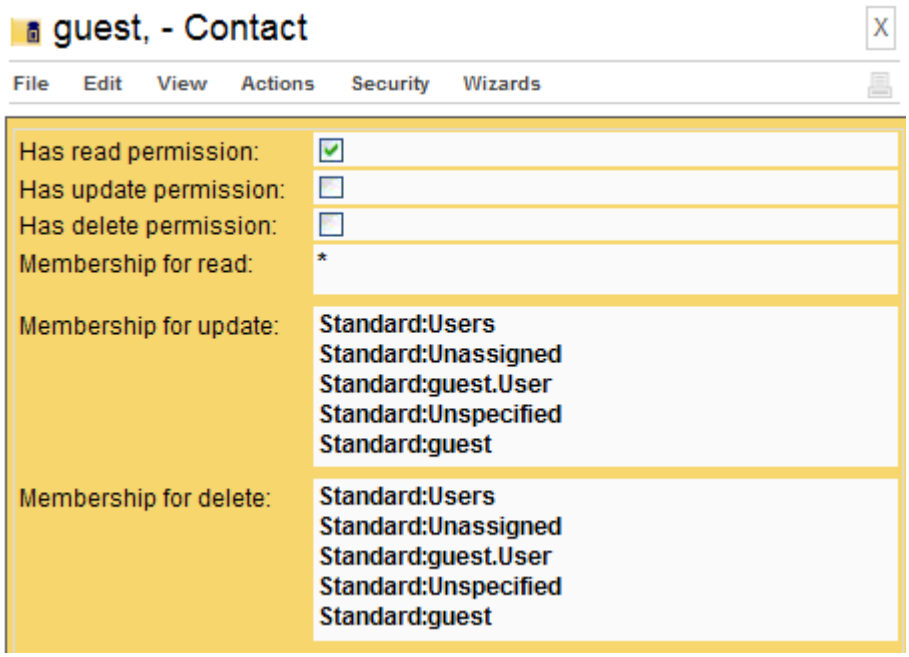


Figure 8: Result of Check Permissions

The meaning of the above result is as follows:

Has read permission:	☑ principal can browse this object ☐ principal cannot browse this object
Has update permission:	☑ principal can modify/update this object ☐ principal cannot modify/update this object
Has delete permission:	☑ principal can delete this object ☐ principal cannot delete this object
Membership for read:	principal has read permission if the intersection of the resulting list of groups and the list of owning groups of the respective SecureObject is not empty
Membership for update:	principal has modify/update permission if the intersection of the resulting list of groups and the list of owning groups of the respective SecureObject is not empty
Membership for delete:	principal has delete permission if the intersection of the resulting list of groups and the list of owning groups of the respective SecureObject is not empty

3.6 Login Procedure

The openCRX login procedure consists of 2 levels:

3.6.1 Apache Tomcat / Application Server Login

The Apache Tomcat / application server login procedure depends on various parameters:

- Servlet container (Apache Tomcat, JBoss, BEA WLS, IBM WAS, etc.)
- configuration of Apache Tomcat / application server
  - file-based realm (e.g. **tomcat-users.xml** for Tomcat or **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 Apache Tomcat / application server login is not really under control of openCRX. Many login problems are related to incomplete/faulty configuration settings of the servlet container.

3.6.2 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:

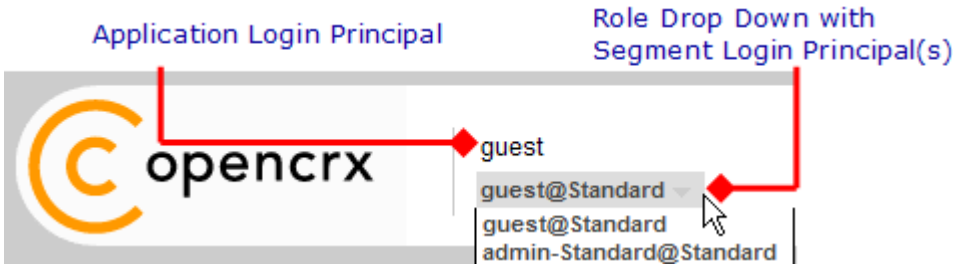


Figure 9: Role Drop Down with list of available Segment Login Principals

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

3.6.3 Disabling Login

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

3.7 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!

It is simply not possible to disable openCRX Security.

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.

Educate your users about openCRX security. You might also consider disabling some of the more powerful operations and/or security attributes in the default GUI.

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

Even though you can create users with a variety of methods, "behind the scenes" the following steps are always required to create a new openCRX user:

Who	Steps

Root administrator
admin-Root
	• create a new Subject
	  ▪ set the qualifier to the desired login id
	• create a new Principal in the realm Default (--> Application Login Principal)
	  ▪ set the qualifier to the desired login id
	  ▪ link this Principal to the Subject created in the previous step
	  ▪ make this Principal member of the appropriate Principal Group(s), e.g. *Users*

Segment administrator
admin-<SegmentName>
	• create a new Contact
	• create a new User
	  ▪ link this User to the Contact created in the previous step
	  ▪ the Segment Login Principal is created automatically
	  ▪ the userhome is created automatically
	• run Edit > User Settings from the new user's homepage

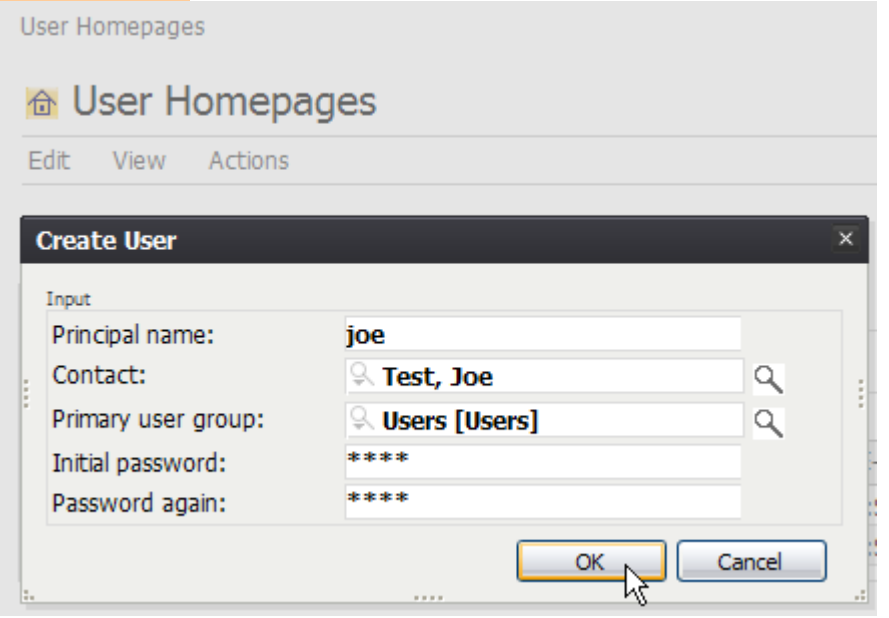
Depending on how you create a new user, some of the above steps might be taken care of by a wizard. If you want to have full control over the user creation process, however, then you can certainly create new users following the above instructions step by step.

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.

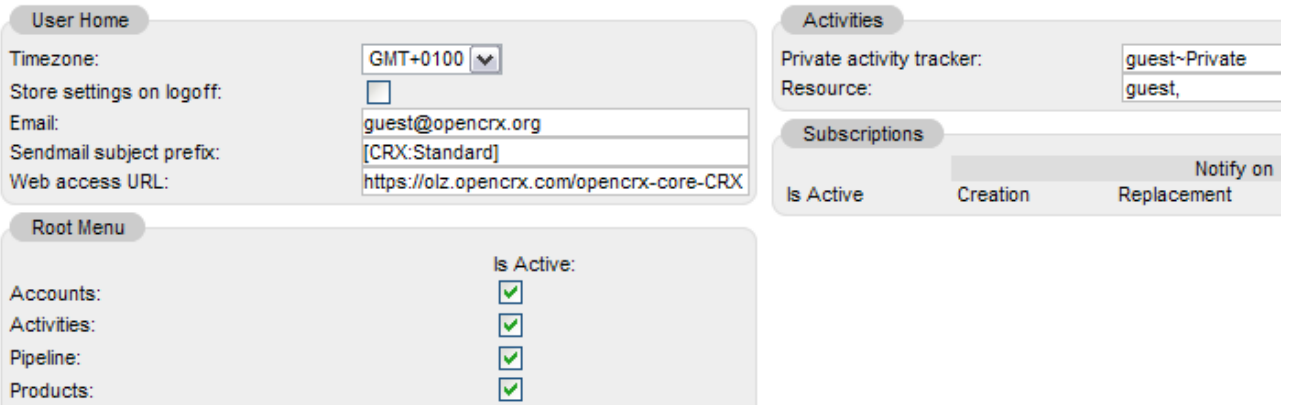
4.1.1 Create Users as Segment Administrator

The Segment administrator can create new users with the following steps:

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



- upon execution a new principal and a new subject are automatically created if they do not exist yet (please note that both the subject name and principal name are set to the value of the parameter "Principal name")
- remain logged in as Segment administrator and navigate to the newly created Homepage of the new user
- execute the operation Edit > User Settings and adjust the settings:



- click Apply to save the settings and then click Cancel to leave the wizard; now the new user should be able to login

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.

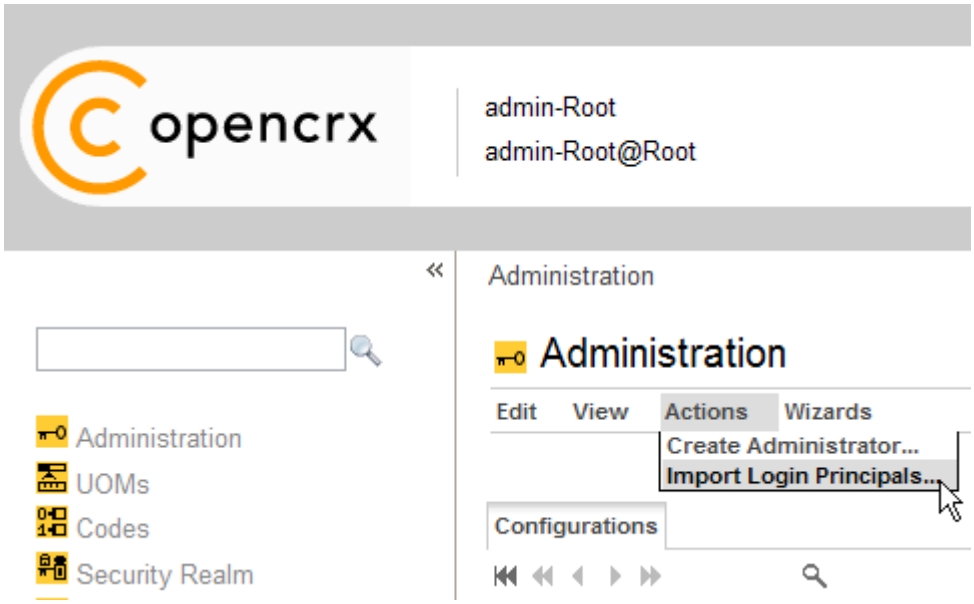


Figure 10: Operation Actions > Import Login Principals (admin-Root)

*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.

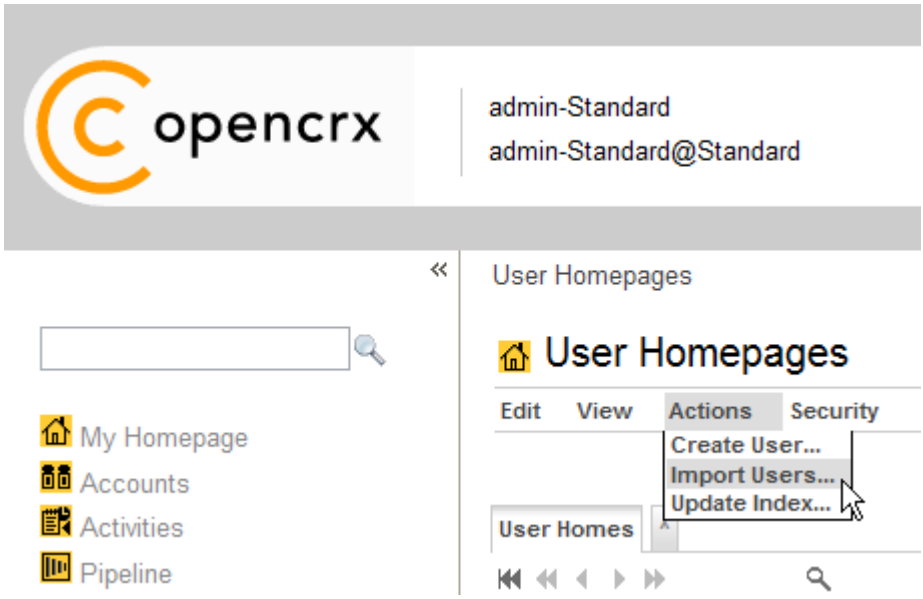


Figure 11: Operation Actions > Import Users (admin-Standard)

*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 Disable Users at the level Tomcat /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, you can simply remove users from the file **tomcat-users.xml** (with Apache Tomcat) or **users.properties** (with JBoss) to prevent access to openCRX. If you block access at the level Tomcat / application server such users are locked out from accessing any application and any openCRX segment. However, as the Tomcat/ application server Login procedure is not entirely controlled by openCRX you might have to consult the documentation of your respective application server or ask your application server administrator for details.

4.2.2 Disable 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.

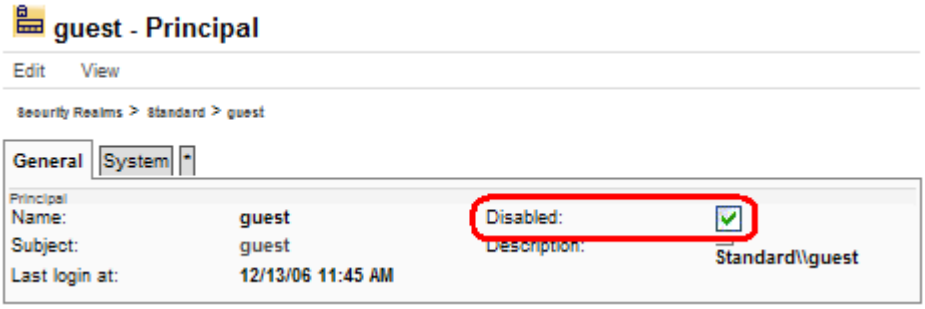


*Figure 12: Disabling of Segment Login Principal **guest** by **admin-Standard***

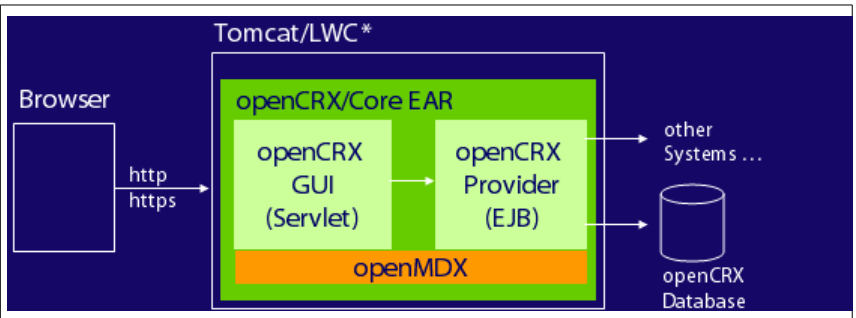
You should **not** delete a particular Subject as long as it is referenced by any Principal. Otherwise you'll end up with "dangling" Subject references.

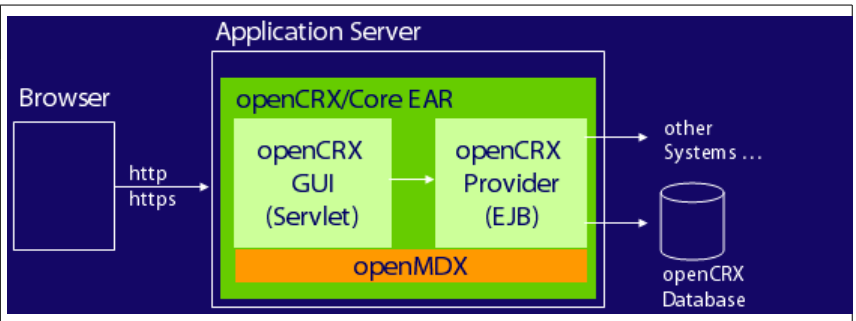
5 Deployment Scenarios

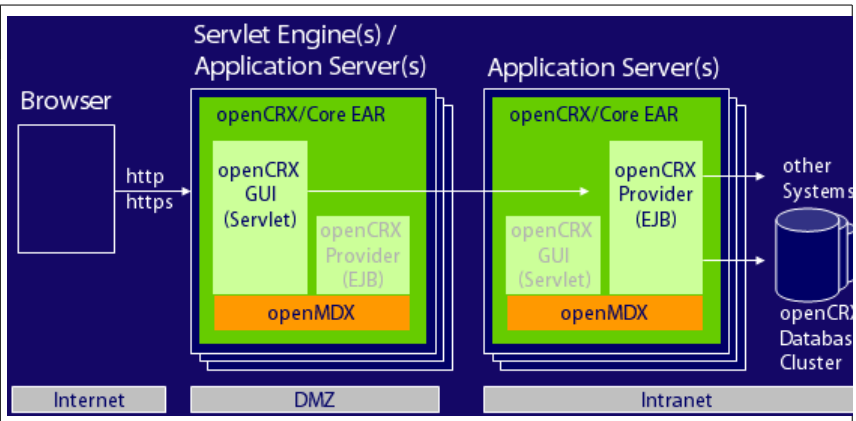
openCRX supports a multitude of deployment scenarios.

5.1 Typical Deployment Scenarios

The following table lists some of the pros and cons of the most common openCRX deployment scenarios. Please note that the list is by no means complete:

3-Tier with Tomcat/LWC	
▪ Tomcat engine extended by openMDX LightweightContainer (LWC) so that EARs with EJBs and WARs can be deployed ▪ simple setup and management (Server Installer) ▪ limited scalability and availability (no clustering) ▪ **highest performing 3-Tier deployment with full transaction service**	*Figure 13: 3-Tier with Apache Tomcat / LWC*

3-Tier with AppServer	
▪ J2EE-compliant Application Server (JBoss is supported out of the box) ▪ simple setup and management (one application server) ▪ limited scalability and availability (no clustering) ▪ Transaction Service	*Figure 14: 3-Tier with J2EE-compliant Application Server*

4-Tier with Clusters	
▪ demanding setup (four and more application servers) ▪ multiple security zones for **highest security** (Internet, DMZ, Intranet) ▪ **maximum availability** ▪ **fully fault tolerant** ▪ **unlimited scalability** ▪ Transaction Service	*Figure 15: 4-Tier with Clustered Application and DB Servers*

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.) might 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 real 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.

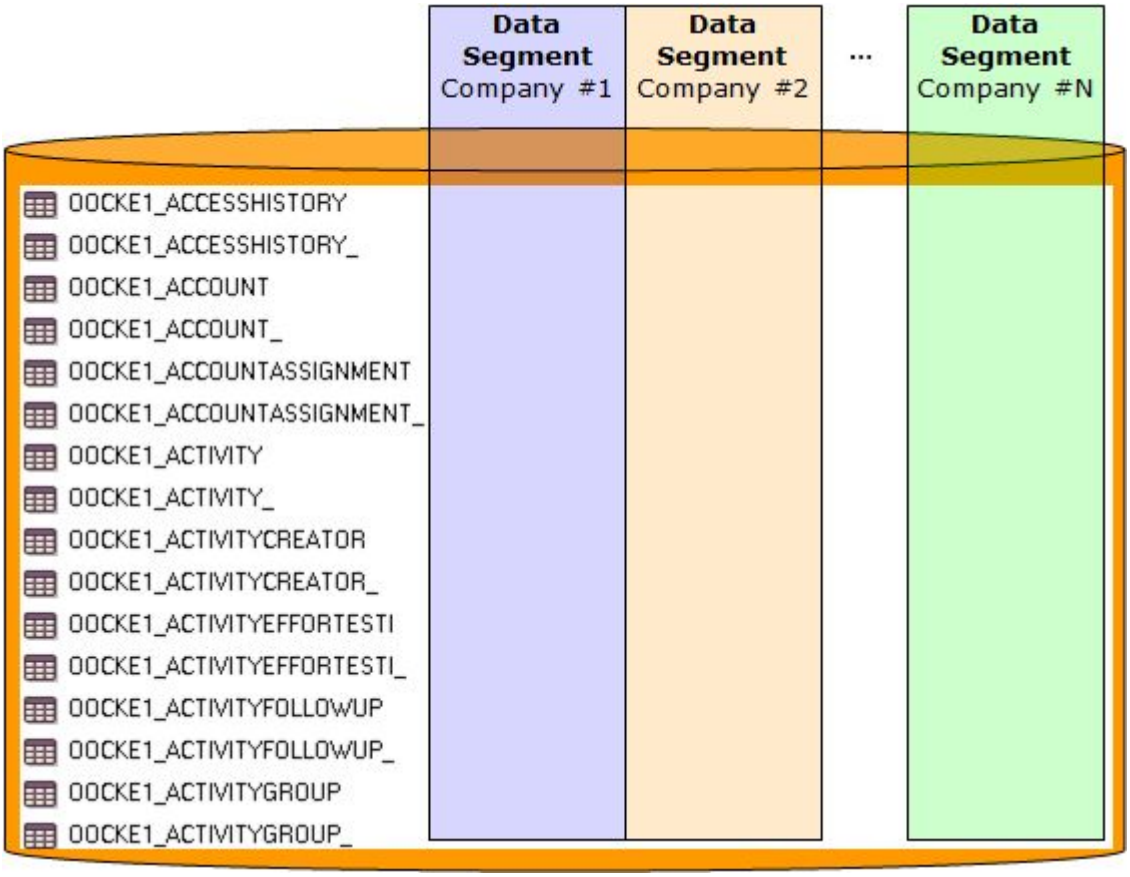


Figure 16: Multiple Data Segments in a single DB

Detailed instructions on how you can create and configure new segments are provided in the openCRX QuickStart guide.

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:

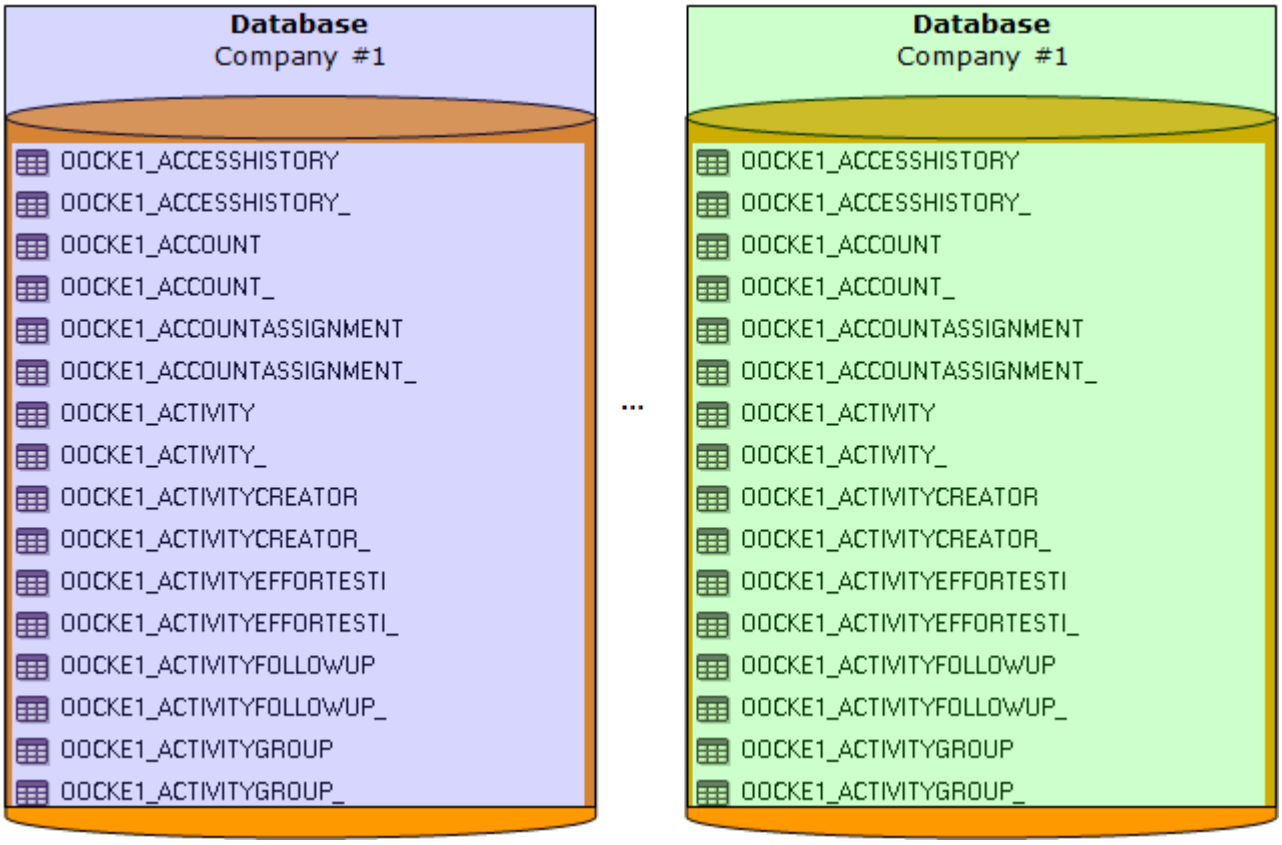


Figure 17: Dedicated DB for each Entity

5.3 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:

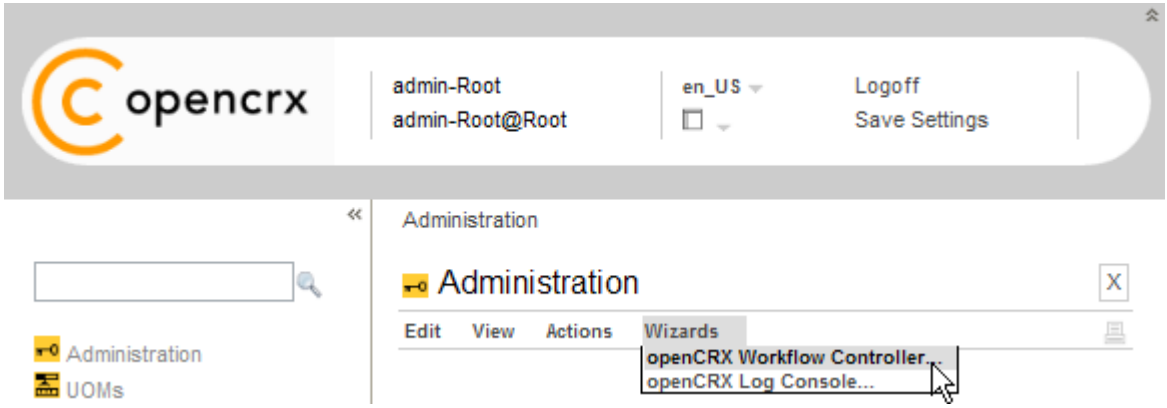


Figure 18: Accessing the openCRX Workflow Controller

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 2.3:

Figure 19: openCRX 2.3 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:

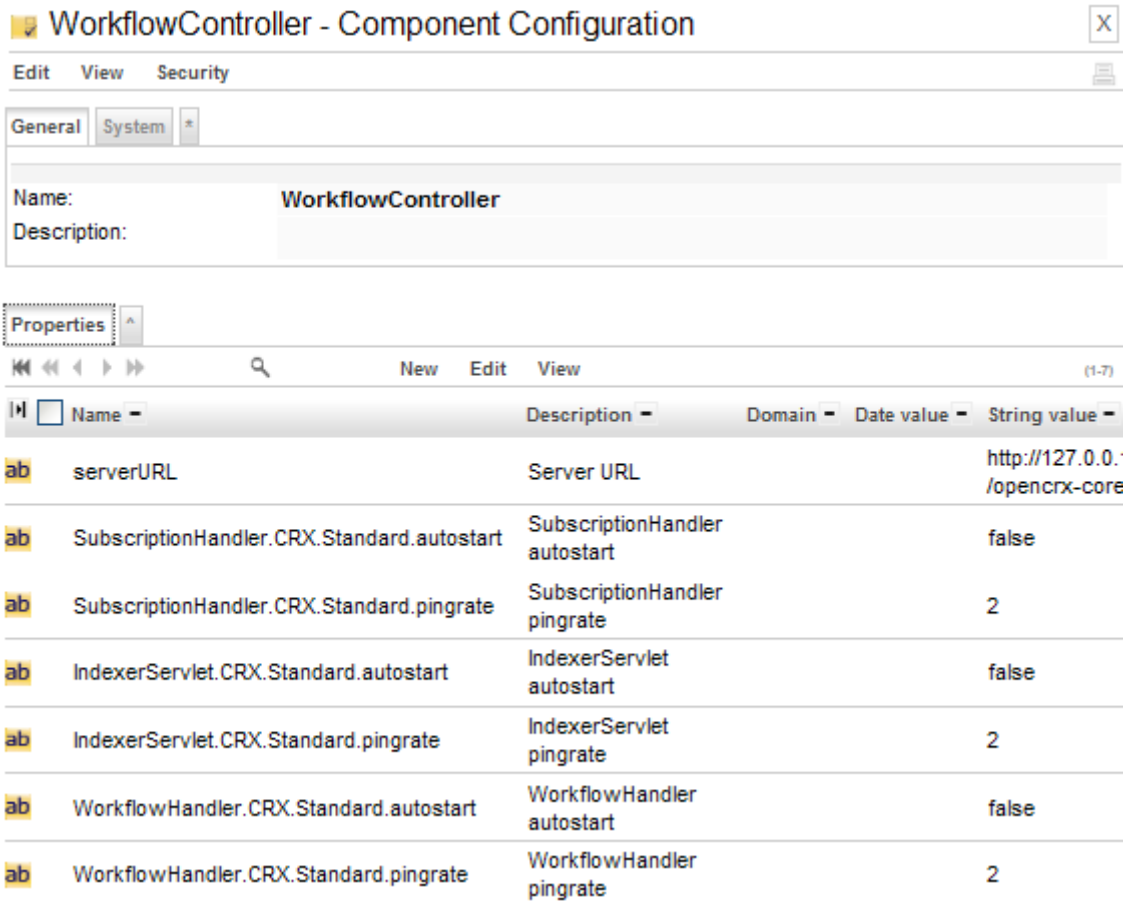


Figure 20: Default Configuration of WorkflowController

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 “OtherSegment” in addition to the segment “Standard” you can start/ stop servlets of the segment “OtherSegment” 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 available to the openCRX Root administrator (admin-Root) by navigating to Administration and then clicking on the icon of the WorkflowController:

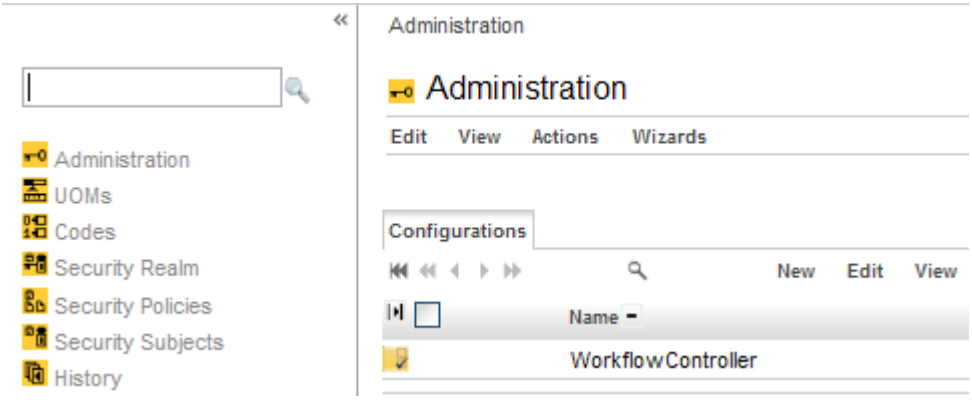


Figure 21: openCRX Administration – WorkflowController

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 servlet.

6.1.2 ServerURL

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

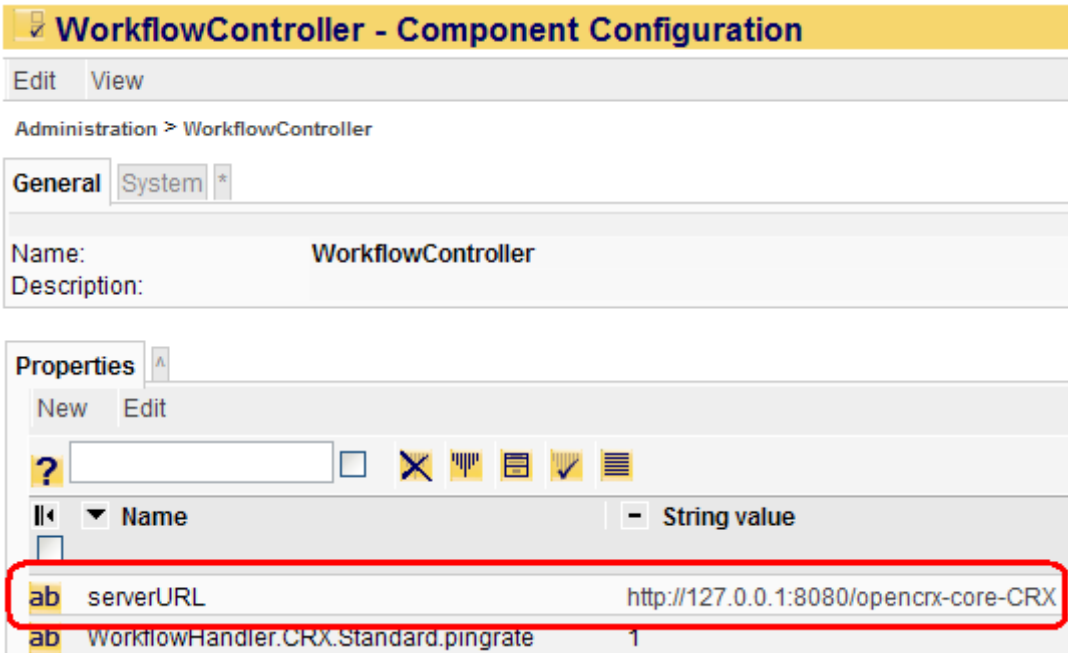


Figure 22: Workflow Controller Configuration – serverURL

6.1.3 Handler pingrate and autostart

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

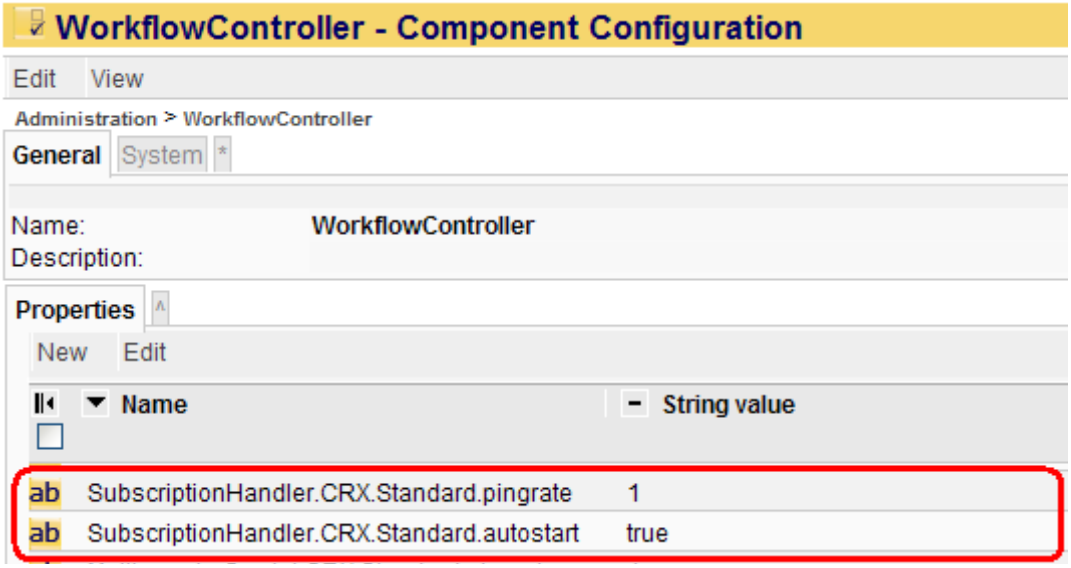


Figure 23: Workflow Controller Configuration – pingrate and autostart

6.2 Servlet IndexerServlet

The openCRX IndexerServlet updates index entries (used for keyword/index based search introduced with openCRX v2) by indexing all objects which do not have an IndexEntry newer than the modification date of the object. The IndexerServlet creates an index by invoking the operation updateIndex() on the object to be indexed.

Please note that indexing can put some heavy load on your database server. Hence, you might consider turning off (or at least lowering the frequency of calling) the IndexerServlet during busy hours.
If you are looking for a way to define advanced schedules for calling the openCRX indexer you might consider cURL in combination with a scheduler provided by your operating system (e.g. Scheduled Tasks on Windows, cron on Linux).
With curl calling the indexer boils down to calling curl with the appropriate URL as a parameter. The following example shows how to call the indexer for the provider CRX and the segment Standard:

```
curl "http://localhost:8080/opencrx-core-CRX/IndexerServlet/execute?
provider=CRX&segment=Standard"
```

6.3 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()*.

Userhome.executeWorkflow() – implemented by the openCRX plugin – creates an entry in *Userhome.wfProcessInstance* (accessible through the grid Workflow Process Instances). **Synchronous workflows** are executed immediately, **asynchronous workflows** are left alone (the Servlet WorkflowHandler is specialized in dealing with asynchronous workflows – see below for details).

6.4 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:

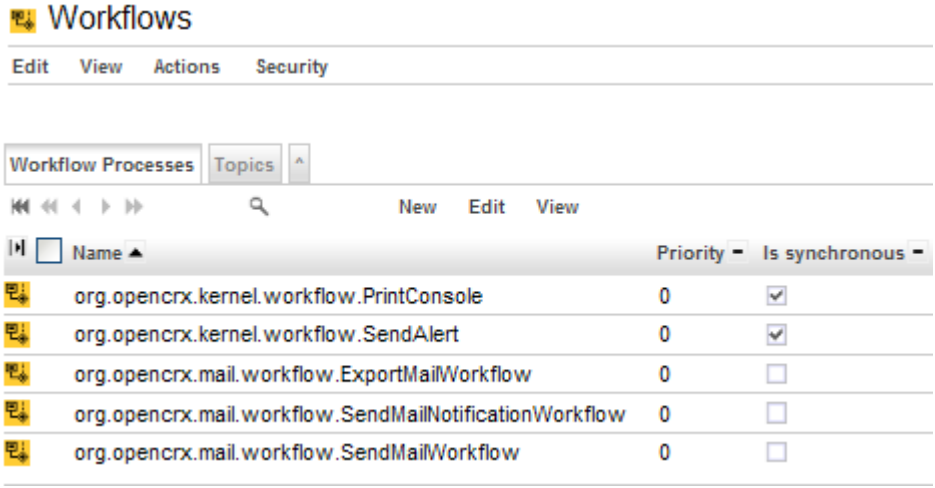


Figure 24: Default Workflow Processes created by WorkflowHandler

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 default setup ensures that blocking WfProcesses cannot block openCRX.

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-4.2.1.GA\server\default\log**server.log** on JBoss or D:\apache-tomcat-6.0.14\logs**catalina.<date>.log** on Tomcat). The following log file extract shows, for example, that the three Servlets IndexerServlet, SubscriptionHandler, and WorkflowHandler seem to be working fine:

Listing 6: Servlets managed by Workflow Controller log to server.log

```
20:25:18,388 INFO  [STDOUT] Tue Mar 04 20:25:18 CET 2008: Indexer CRX/Standard
20:27:18,400 INFO  [STDOUT] Tue Mar 04 20:27:18 CET 2008: SubscriptionHandler CRX/Standard
20:27:18,400 INFO  [STDOUT] Tue Mar 04 20:27:18 CET 2008: WorkflowHandler CRX/Standard
```

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 2.3 Workflow Controller*).

After restarting the application server all servlets managed by the WorkflowController are inactive, 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:

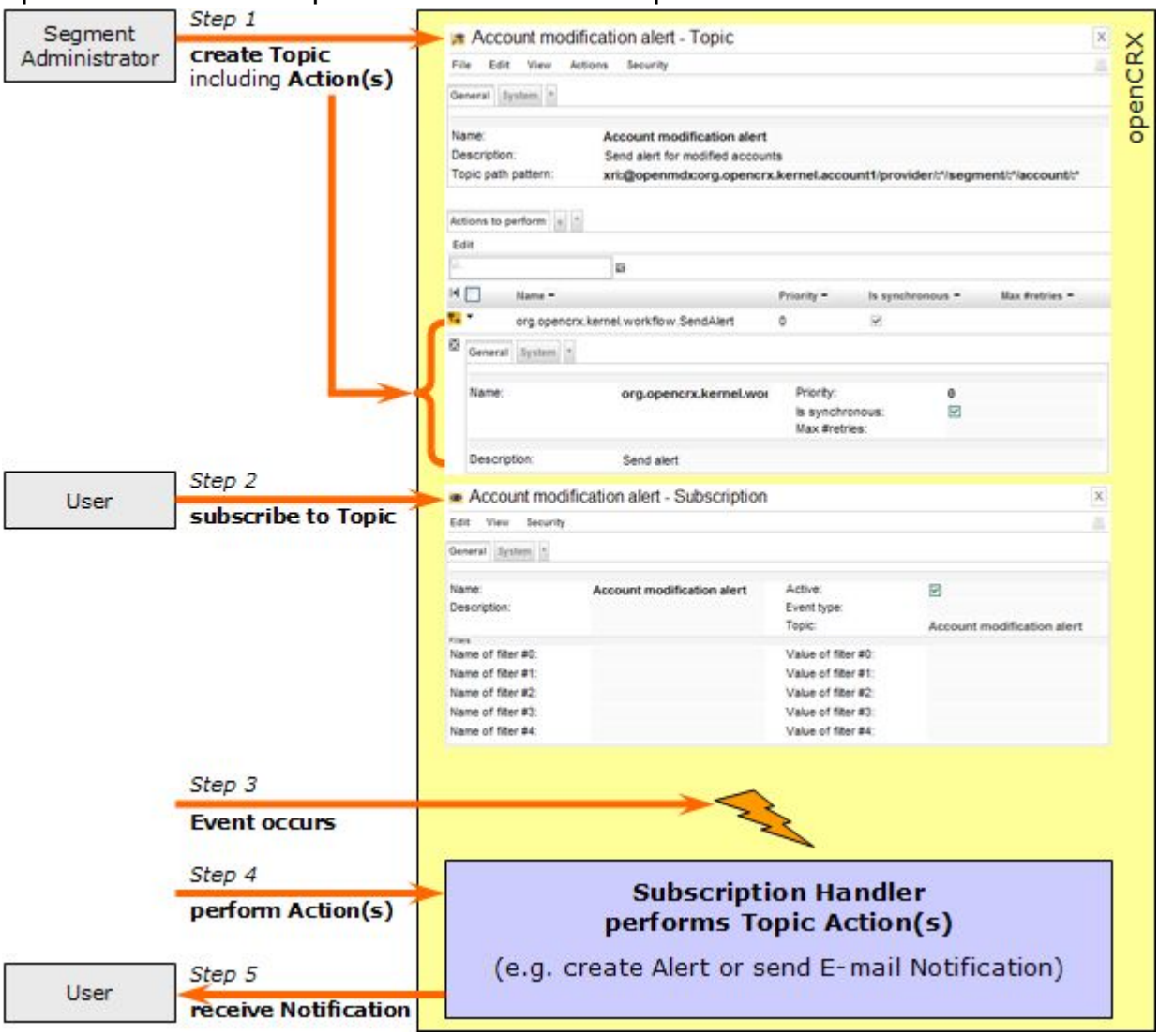


Figure 25: Event and Notification Service

Once a topic is created, openCRX users can subscribe to it. Users manage their subscriptions individually on their UserHomes (with the Wizard UserSettings or by editing their subscriptions manually). 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 the 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 will be created and E-mail Notifications will not be delivered.

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 26: Standard Topics included in the openCRX distribution*) to get you started:

- Topic **Account modification alert** sends an alert to the UserHome of subscribed users whenever an account is modified.
- Topic **Activity follow-up modification alert** sends an alert to the UserHome of subscribed users whenever a Follow Up of an Activity is modified.
- Topic **Mail notification for new alerts** 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 can be created by the segment administrator with the wizard **Segment Setup**.

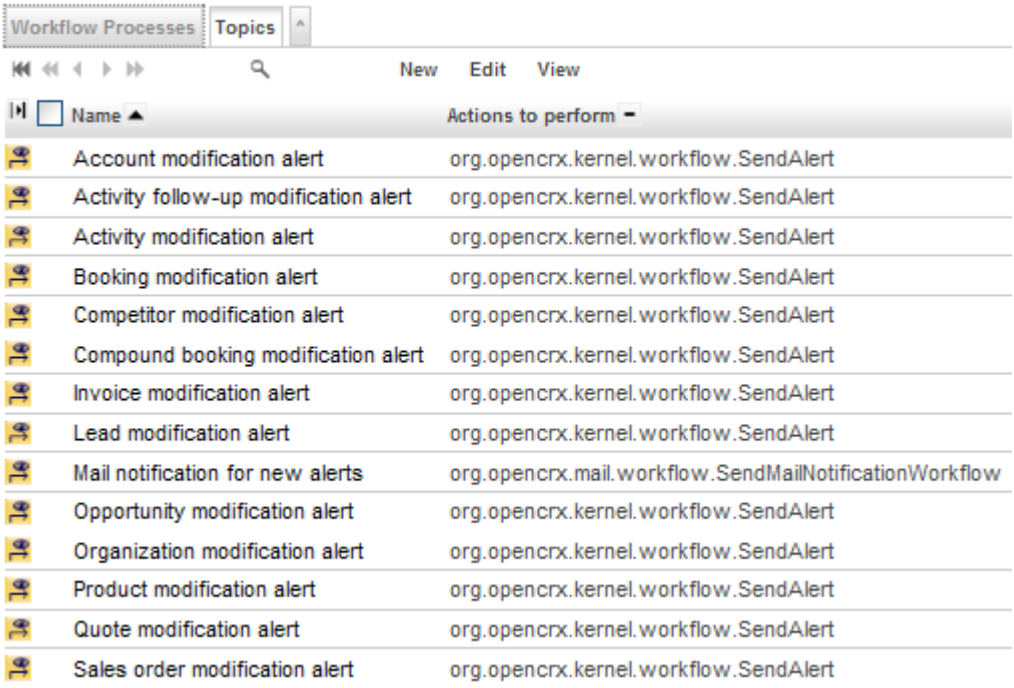


Figure 26: Standard Topics included in the openCRX distribution

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 – Account Modifications

In this example we will create a subscription to the standard Topic Account Modifications for the user "guest".

- Login as guest, and execute the operation Edit > User Settings to start the respective wizard. Check both "Is Active" and "Creation" as shown below:

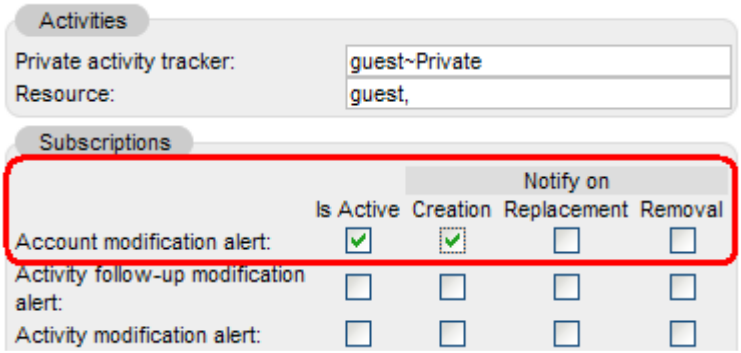


Figure 27: Create a new Subscription

- Click Apply to store your settings.

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 Activity Trackers **DivisionA:ProjectX** and **DivisionA:ProjectY** and some of your users are interested in receiving notifications related to activities of ProjectX only, whereas some users want to receive notifications related to activities of ProjectY only. A third group of users wants 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:

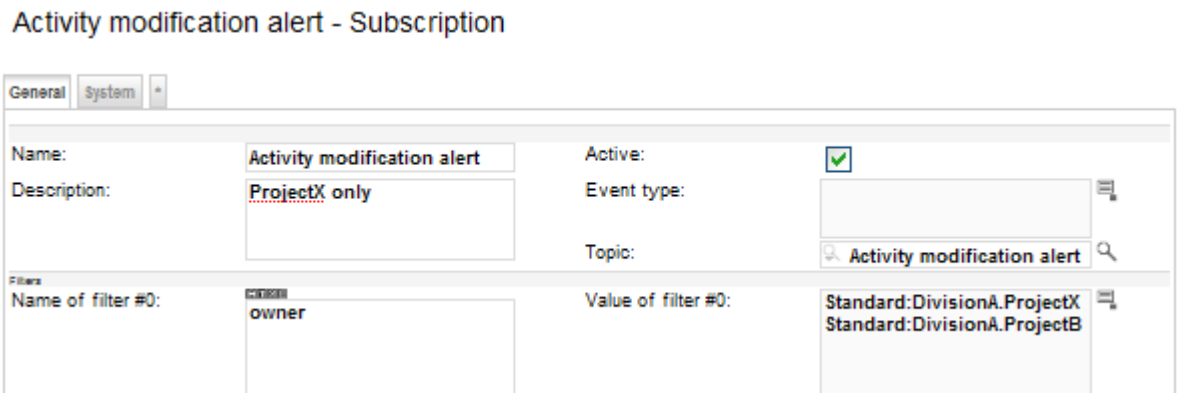


Figure 28: Create a Subscription with Filters

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 (in our case Standard:DivisionA.ProjectX and Standard:DivisionA.ProjectY)

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.	▪ verify that the Segment Administrator created Workflow Processes and Topics with the wizard Segment Setup ▪ 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	▪ 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, recreate Workflow Processes and Topics with the wizard Segment Setup, 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. NullPointer Exception during Workflow execution)
I receive Alerts triggered by my Subscriptions but no Notification E-mails	▪ verify that JavaMail is properly installed and the mail service properly configured (see chapter 8.1 Install and Configure Mail Resource and E-Mail Services for more information) ▪ 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 we have **no intention to duplicate mail server (MTA) or mail client functionality in openCRX** as there are lots of excellent products available (Open Source and commercial). It is our goal, however, that **openCRX integrates with all the major products that adhere to the major standards and support standard protocols** like SMTP, POP3, IMAP, etc. This ensures that you can continue to use your favorite mail server (Postfix, qmail, etc.) and your favorite mail client (Thunderbird, Outlook, etc.).

Installation of JavaMail is required if you want to make use of E-mail Services. Deployment of openCRX Groupware Services is required if you want to use the IMAP Adapter.

The following figure shows the flow of mail messages between openCRX, mail server, and mail client as it is supported with openCRX 2.3:

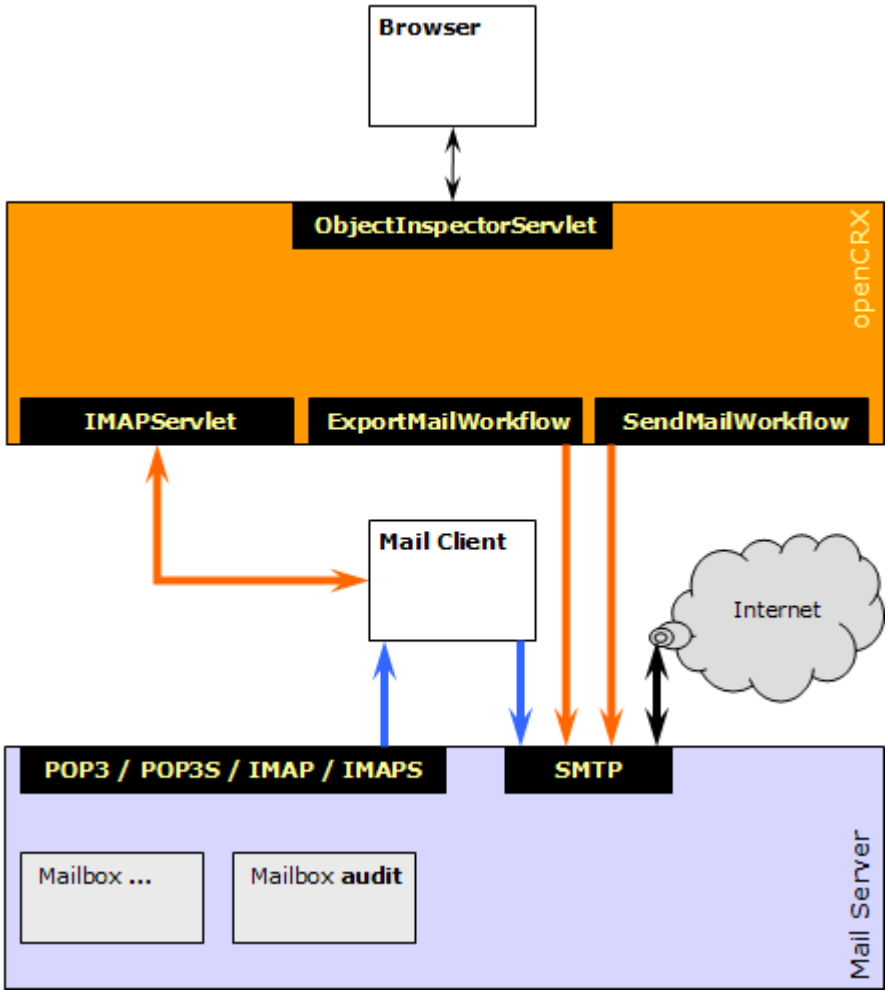


Figure 29: Flow of e-mail messages between openCRX, MTA and mail client

In this chapter we will first guide you through the required installation and configuration steps before we discuss various important use cases.

8.1 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.

8.1.1 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!

The following instructions are dependent on your deployment scenario as the steps for creating mail resources vary from application server to application server. We provide instructions for Apache Tomcat and JBoss.

8.1.2 Mail Resource for openCRX on Apache Tomcat

8.1.2.1 Add resource definition to opencrx-core-CRX.xml

Add the following resource definition to the file **opencrx-core-CRX.xml** in the directory

<Tomcat Install Dir>\conf\Catalina\localhost (on Windows)

<Tomcat Install Dir/conf/Catalina/localhost (on Linux)

In the sample file below you must at least adapt the highlighted strings to your own environment:

*Listing 7: File **opencrx-core-CRX.xml***

```
...
  <Resource
    name="mail/provider/CRX/segment/Standard"
    auth="Container"
    type="javax.mail.Session"
    user="test"
    password="test"
    mail.transport.protocol="smtp"
    mail.smtp.starttls.enable="true"
    mail.smtp.auth="true"
    mail.smtp.host="mail.dot.org"
    mail.smtp.port="25"
    mail.smtp.user="test"
    mail.from="noreply@dot.org"
    mail.store.protocol="pop3s"
    mail.pop3s.host="mail.dot.org"
    mail.pop3s.port="995"
    mail.pop3s.auth="true"
    mail.pop3s.user="test"
    mail.pop3s.password="test"
  />
...
```

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

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

8.1.2.2 Mail Resource in web.xml

In the file **web.xml** in the directory <Tomcat Install Dir>\webapps\opencrx-core-CRX\WEB-INF you must uncomment the following section:

*Listing 8: 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_Standard">
    <res-ref-name>mail/provider/CRX/segment/Standard</res-ref-name>
    <res-type>javax.mail.Session</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>
...
```

Restart Tomcat 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 **opencrx-core-CRX.xml**) also import the mail server's public key into the file cacerts of your JRE:

*Listing 9: Importing certificate into keystore **cacerts***

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

8.1.3 Mail Resource for openCRX on JBoss

8.1.3.1 Create mail-service.xml

Next you need to create the file **mail-service.xml** in the directory

d:\pgm\jboss-4.2.1.GA\server\default\deploy (JBoss on Windows)

/opt/jboss/server/default/deploy (JBoss on Linux)

In the sample file below you must at least adapt the highlighted strings to your own environment:

*Listing 10: 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/segment/Standard</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 is available from the JavaMail home:

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

If you leave the debug option set to true 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.

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

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

d:\pgm\jboss-4.2.1.GA\server\default\deploy\opencrx-core-CRX.ear\opencrx-core-CRX.war\WEB-INF\

or (JBoss on Linux)

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

you must uncomment the following section:

*Listing 11: 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_Standard">
    <res-ref-name>mail/provider/CRX/segment/Standard</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-4.2.1.GA\server\default\deploy\opencrx-core-CRX.ear\opencrx-core-CRX.war\WEB-INF\

or (JBoss on Linux)

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

you must uncomment the following section:

*Listing 12: Uncomment mail resource definition in **jboss-web.xml***

```
...
<!-- Configure a resource-ref for each mail resoure configured in web.xml -->
  <resource-ref>
    <res-ref-name>mail/provider/CRX/segment/Standard</res-ref-name>
    <jndi-name>java:/mail/provider/CRX/segment/Standard</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 13: Importing certificate into keystore **cacerts***

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

8.2 Outbound E-mail

8.2.1 Outbound E-mail Configuration

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:

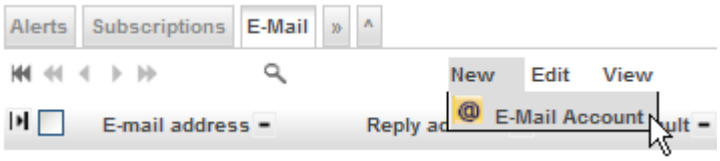


Figure 30: Create a new E-Mail Account – step 1

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

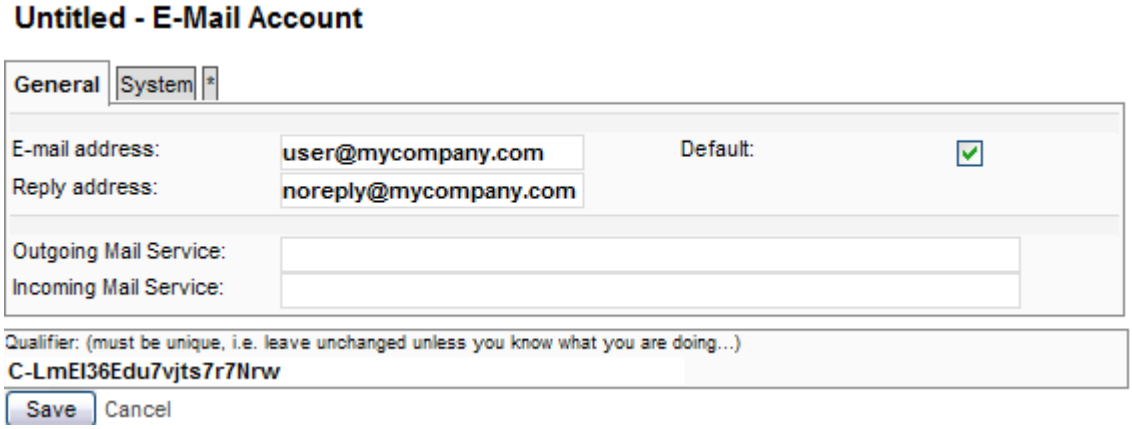


Figure 31: Create a new E-Mail Account – step 2

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**: 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/<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/<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/<provider>/segment/ is used; if there is no resource with this name the fallback name /mail/provider/<provider> is used (and if this name does not exist either, then an error is logged).

- 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:

Figure 32: Create a new E-Mail Account – step 3

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

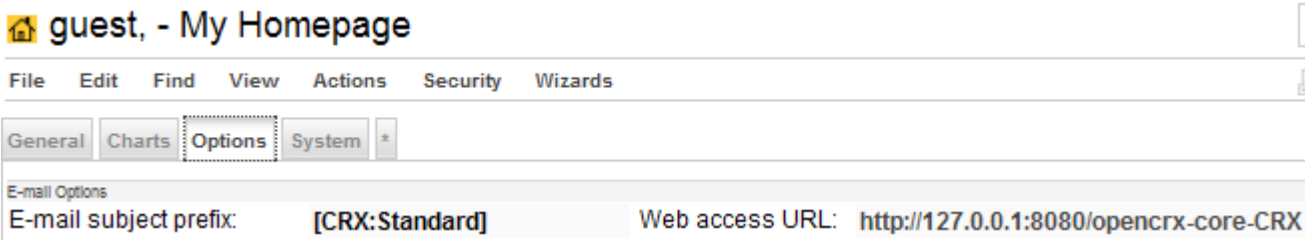


Figure 33: E-mail subject prefix and Web access URL

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 Account Modifications (see Example Subscription – Account 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).

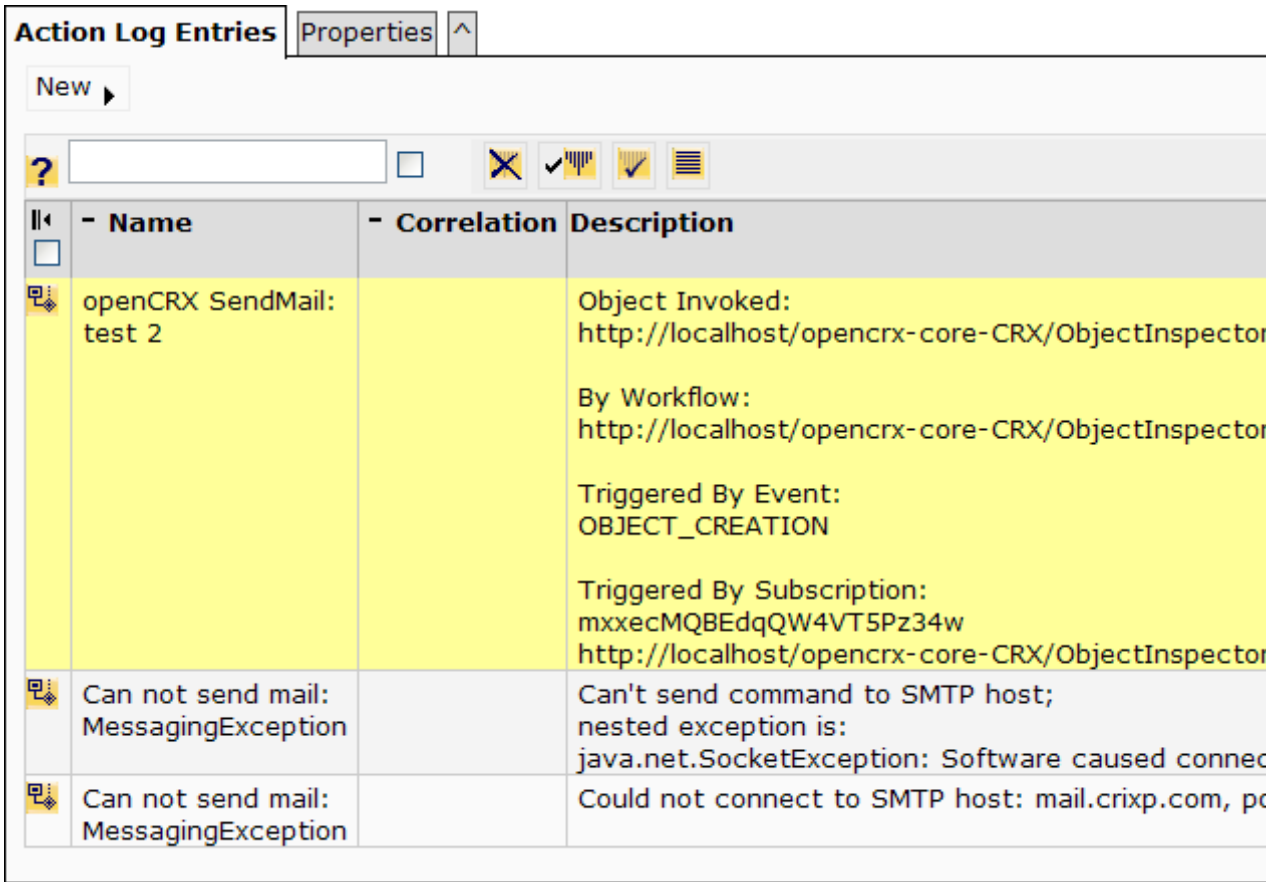


Figure 34: Example of outbound E-mail Action Log Entries

8.2.2 Export E-mails

Please refer to chapter 9.4 Mailstore / IMAP.

8.2.3 Send E-mails directly from openCRX

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

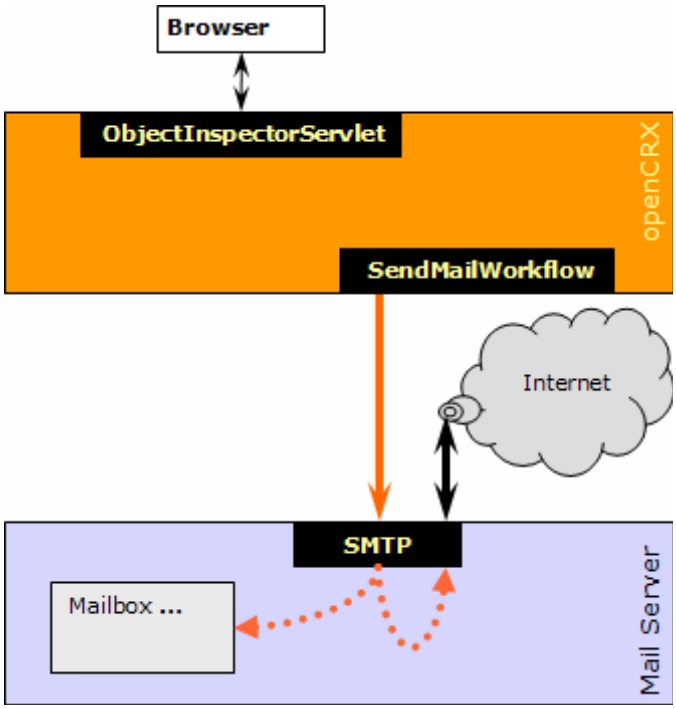


Figure 35: Send E-Mail from openCRX – Overview

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:

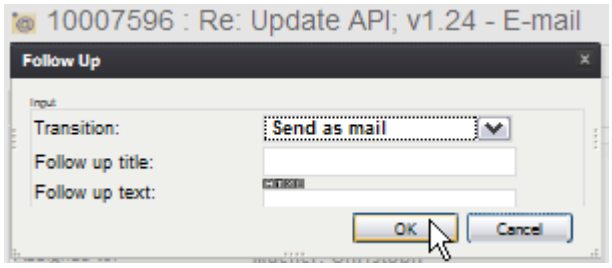


Figure 36: Send E-Mail from openCRX with Actions > Follow Up

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

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

8.2.4 Send E-mails as Attachments to your Mail Client

Any openCRX E-Mail Activity can be sent to your mail client as an attachment. 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:

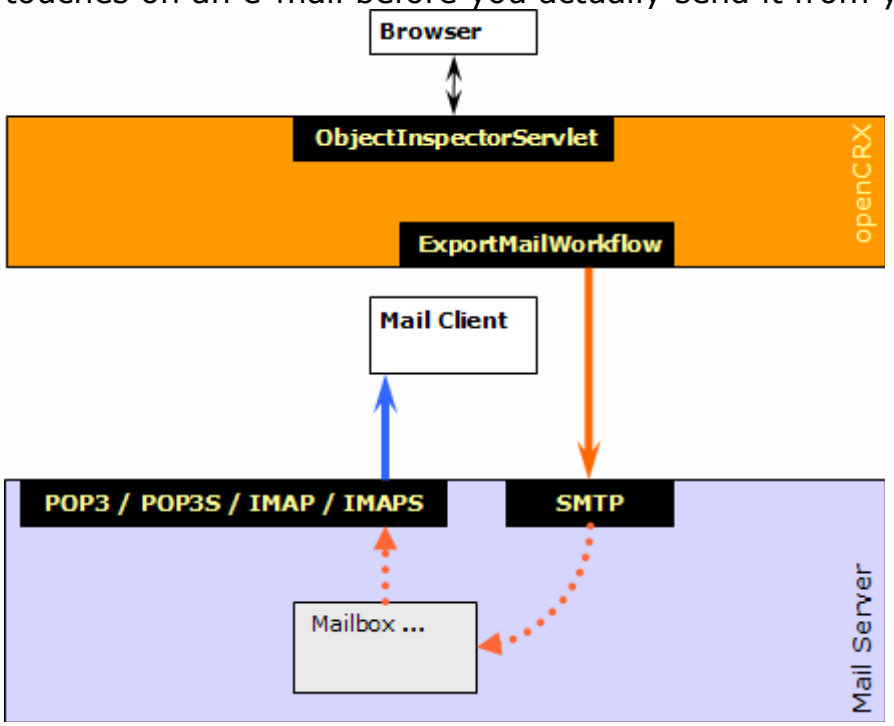


Figure 37: Send E-Mail as Attachment from openCRX – Overview

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**:

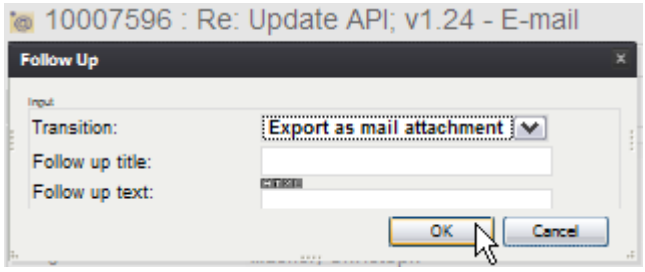


Figure 38: Export E-Mail from openCRX with Actions > Follow Up

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

Exported messages are sent as attachments to the user's default mail address. See Outbound E-mail Configuration for details.

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

8.2.5 Send E-mails to Fax-/SMS-Gateways

The SendMailWorkflow supports mail gateways if you set the attribute gateway of the respective e-mail activity. The gateway address is used for addresses which are not of type EmailAddress. For example, in the case of a phone number, the address is converted to an e-mail address as follows:

- take the address (e.g. phoneNumberFull in case of phone numbers)
- remove any characters other than digits and letters
- convert "+" (plus sign) to "_" (underscore)
- append domain part of gateway address

Example: if the domain address of an e-mail activity is set to the email address noreply@fax-gateway.opencrx.org, the the phone number +41 (44) 111-2233 is converted to the email address _41441112233@fax-gateway.opencrx.org.

This conversion feature allows you to mix e-mail addresses and phone numbers in member lists of address groups. Depending on the recipient's type of addresss the SendMailWorkflow will either send an e-mail to the listed e-mail address as is (e-mail address) or first convert the recipient's phone number to an e-mail address so that the resulting e-mail can be handled by your fax-/sms-gateway.

8.3 Inbound E-mail

Please refer to chapter 9.4 Mailstore / IMAP.

8.3.1 Import E-mails

Please refer to chapter 9.4 Mailstore / IMAP.

8.4 Trouble Shooting E-mail Services

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

Problem	Solution

openCRX does not initiate TLS session with mail server	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.
	
	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.
	Listing 14: Importing Certificate
	`cd $JAVA_HOME/lib/security`
	`keytool -import -alias <dom> -file <name>.cer -keystore cacerts`
	Replace <dom> with the domain of the mail server (e.g. mail.company.com) and <name> with the name of the certificate file.

I receive Alerts triggered by my Subscriptions but no Notification E-mails	▪ verify that JavaMail is properly installed and the mail service properly configured (see chapter 8.1 Install and Configure Mail Resource and E-Mail Services for more information) ▪ verify your e-mail settings (see E-mail Services for details) ▪ verify that the Servlet WorkflowHandler of the respective segment is turned on

9 Groupware Services

openCRX features the following groupware services:

Type of Service	Standard	Server Provider
Directory Service	LDAP	OpenDS (requires opencrx-groupware-CRX.ear)
vCard Server	VCF	openCRX/core (native provider)
Calendaring	FreeBusy	openCRX/core (native provider)
	iCalendar (ICS)	openCRX/core (native provider)
Mailstore	IMAP	openCRX/core (native provider)

Please note that LDAP services are only available if openCRX Groupware (**opencrx-groupware-CRX.ear**) is deployed. It is not sufficient to deploy openCRX Core only.

9.1 Directory Service / LDAP

Based on Sun's OpenDS, an Open Source Directory Service, openCRX provides LDAP Server functionality (get more information about LDAP or read what Wikipedia has to say about LDAP). In a nutshell this means that you can use any LDAP client to connect to openCRX and view openCRX accounts. Furthermore, openCRX LDAP service supports SSL. The following information is required to connect to openCRX with an LDAP client:

Host	IP address or host name of openCRX Server Examples: localhost, 127.0.0.1, myCrxServer.myCompany.com, etc.
Port	1389 (note that the LDAP standard port is 389)
BaseDN	dc=accounts,dc=,dc=<Provider>,dc=opencrx,dc=org Example: **dc=accounts,dc=Standard,dc=CRX,dc=opencrx,dc=org**
BindDN	dc=<Username>,dc=users,dc=<Provider>,dc=opencrx,dc=org Example: **dc=guest,dc=users,dc=CRX,dc=opencrx,dc=org**

Please note that LDAP services are only available if openCRX Groupware (**opencrx-groupware-CRX.ear**) is deployed.

The openCRX Directory Service / LDAP supports SSL. Hence, if your LDAP client supports SSL (Thunderbird does, MS Outlook does not), you can enable SSL for increased privacy/protection.

9.1.1 Configuring the openCRX LDAP Port

The openCRX LDAP port is by default set to 1389 (to avoid conflicts with other LDAP daemons listening on the LDAP standard port 389). You can change this configuration in the file **config.ldif** located in opencrx-groupware-CRX.ear\opencrx-ldap-CRX.war\WEB-INF\config\

Look for the entry **ds-cfg-listen-port**.

If you build your own EARs you can change the openCRX LDAP port in your project's file **build.properties** (ldap.listenPort) or directly in your **build.xml**.

9.1.2 LDAP Configuration of Thunderbird

The following steps are required to configure Thunderbird 2 for LDAP:

- start Thunderbird and select the menu Tools > Options
- select Composition and select the tab Addressing
- check Directory Server and click on the button Edit Directories

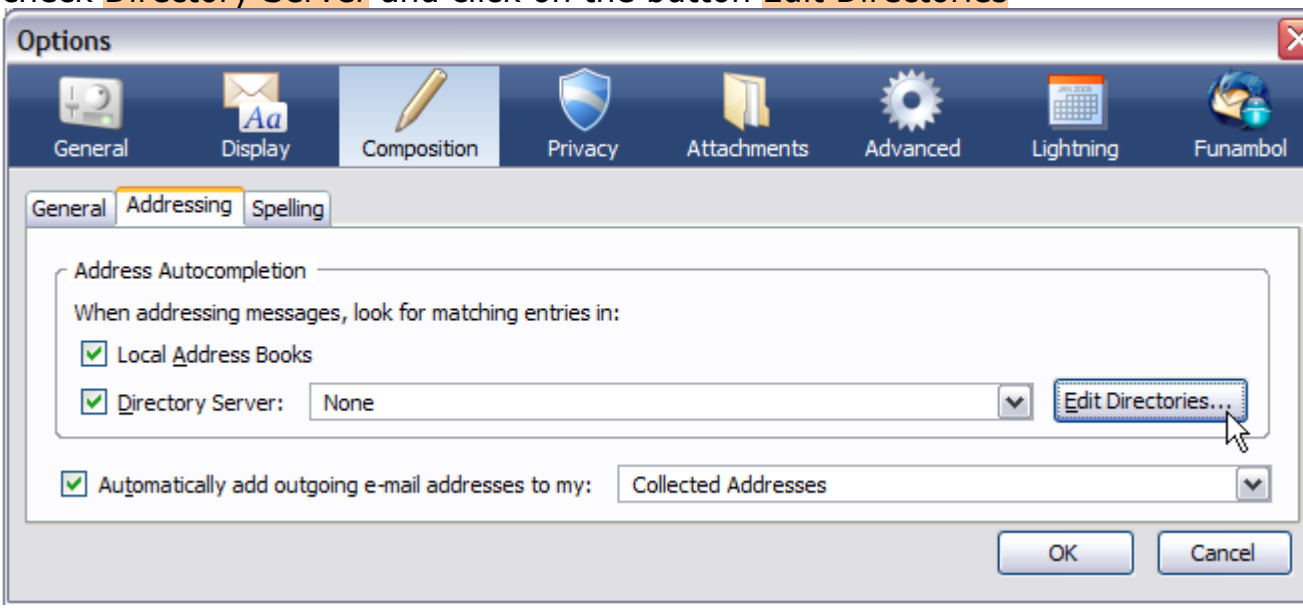


Figure 39: Thunderbird LDAP Configuration

- in the dialog window *LDAP Directory Servers* click on the button Add
- populate the *Directory Server Properties* dialog as follows (assuming the openCRX Server is at localhost, connecting with Username guest):

Name	openCRX
Hostname	localhost
Base DN	dc=accounts,dc=Standard,dc=CRX,dc=opencrx,dc=org
Port number	**1389**
Bind DN	dc=guest,dc=users,dc=CRX,dc=opencrx,dc=org

- click OK to accept

9.1.3 LDAP Configuration of MS Outlook

The following steps are required to configure MS Outlook 2007 for LDAP:

- start Outlook and select the menu Tools > Account Settings
- click on the tab Address Books
- populate the Add/*Change E-mail Account* dialog as follows (assuming the openCRX Server is at localhost, connecting with Username guest):

Server Name	localhost
This server...	check "This server requires me to log on"
User Name	dc=guest,dc=users,dc=CRX,dc=opencrx,dc=org
Password	<enter your openCRX password for user guest>

- click on the button More Settings and select the tab Connection
- enter a display name (of your choice) and verify that port is set to **1389**
- select the tab Search
- in the field group Search Base click on Custom and populate as follows:

Custom	dc=accounts,dc=Standard,dc=CRX,dc=opencrx,dc=org

Figure 40: MS Outlook LDAP Configuration

- click OK, Next, Finish, and Close to conclude the configurationion

9.2 openCRX vcard Servlet

The openCRX vcard servlet does for accounts what the openCRX ical servlet does for activities: it makes them available to third-party clients who access the openCRX server with the http protocol.
othing more and nothing less.

9.2.1 Account Selectors

openCRX can map sets of accounts to a vcard file (a sequence of vcards). The resulting vcard file can be imported and/or processed by vcard-enabled clients like Outlook, Thunderbird, etc. At this time, global account filters are the only supported account selectors. Global account filters support reading, updating, and creating of accounts (R=read, U=update, C=create)

Set of Activities	RUC	Calendar Selector
Global Filter	RUC	<provider>//filter/<filter name>

URL Composition:

http://<openCRX_Server_Name_or_IP_address>:<port>/opencrx-vcard-<provider>/vcard?id=<provider>//filter/<filter name>&resource=accounts.vcf

Example:

http://localhost:t8080/opencrx-vcard-CRX/vcard?id=CRX/Standard/filter/all&resource=accounts.vcf

9.2.1.1 Connecting MS Outlook to the openCRX vcard servlet

Detailed instructions on how to connect MS Outlook are available from
http://www.opencrx.org/opencrx/2.2/Outlook_ICS_VCF_adapter.htm#configureVCARD

9.2.1.2 Connecting Thunderbird to the openCRX vcard servlet

A Thunderbird add-on is in the works:
http://www.opencrx.org/opencrx/2.3/Thunderbird_Contacts_Add-on.htm

9.3 Calendaring

9.3.1 Calendar as a Set of Activities

openCRX supports a wide range of types of activities, including Tasks, Meetings, E-Mails, Phone Calls, etc. Even though all activities are kept in a single location (think of a box containing activities), openCRX offers a multitude of mechanisms to structure, filter, and group activities:

- **activities can be assigned to activity groups**, which enables you to group them by Tracker, Category, and Milestone
- **activities can be filtered** with predefined filters (e.g. activities filtered to a user's homepage, activities filtered by resource) and user-defined ActivityFilters (either at a global level or at the level of an activity group)

To fully understand the power of this approach, consider a large project X (e.g. building a power plant) with millions of activities. With openCRX, a project is typically mapped to an activity tracker (e.g. all activities of project X are assigned to the activity tracker Project X). As a large project is often times structured (i.e. broken down) into milestones, let us assume that the respective subset of activities related to Project X is assigned to an activity group Milestone 2. With openCRX, it's a single click and you can browse all the activities assigned to milestone 2, or you can view all these activities in a calendar application like Sunbird or MS Outlook:

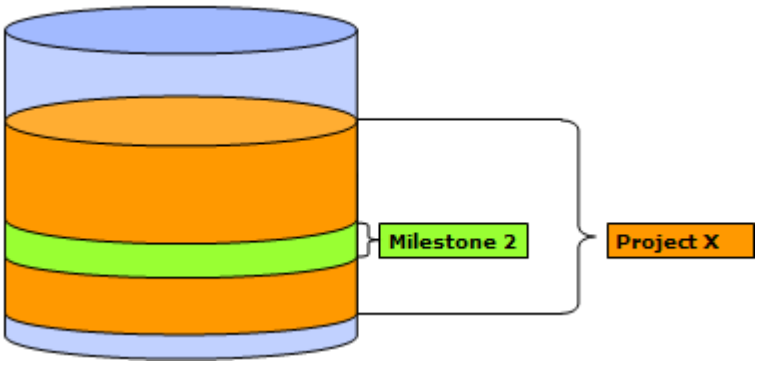


Figure 41: openCRX Activity Groups

It goes without saying that different users have different needs. It is also quite natural that the needs change over time. With openCRX, it is trivial to deliver as there are virtually unlimited possibilities to slice and dice the universe of activities. For example, instead of pulling a set of activities based on their assignment to activity groups, there are many use cases where one would like to define a filter to define a subset of activities. On the one hand, openCRX features lots of default filters, on the other hand, there are powerful tools to define custom filters virtually any way you like. For example: an auditor might be interested in all activities involving a particular subcontractor, another user could be interested in browsing through all the meetings related to Project X:

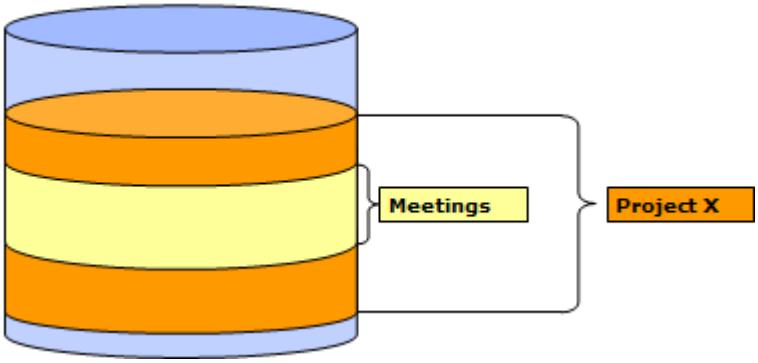


Figure 42: openCRX Activity Filters

So, in the context of calendaring it helps if you think of a calendar as a set of activities, nothing more and nothing less.

9.3.2 Calendar Selectors

openCRX can map each of the above-mentioned set of activities to a calendar. Depending on the mapping, the resulting calendar can be presented in various formats, e.g. ICS calendar, Free Busy calendar, XML file, etc. Some typical calendar selectors are listed below (R=read, U=update, C=create):

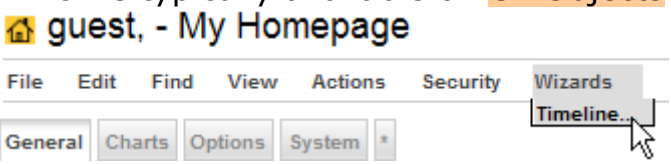
Set of Activities	RUC	Calendar Selector
Tracker	RUC	<provider>//**tracker**/<tracker name>
Category	RUC	<provider>//**category**/<category name>
Milestone	RUC	<provider>//**milestone**/<milestone name>
Filtered Tracker	RUC	<provider>//**tracker**/<tracker name>/**filter**/<filter name>
Filtered Category	RUC	<provider>//**category**/<category name>/**filter**/<filter name>
Filtered Milestone	RUC	<provider>//**milestone**/<milestone name>/**filter**/<filter name>
User's Homepage	RU	<provider>//**userhome**/<home qualifier>
Resource	RU	<provider>//**resource**/<resource name>
Global Filter	RU	<provider>//**globalfilter**/<filter name>

Selector Examples:

CRX/Standard/tracker/shared
CRX/Standard/category/presales
CRX/Standard/milestone/2.1
CRX/Standard/tracker/main/filter/open
CRX/Standard/userhome/guest
CRX/Standard/resource/jeff
CRX/Standard/globalfilter/last18months

For the purpose of this guide we will work with a tracker called “shared”.

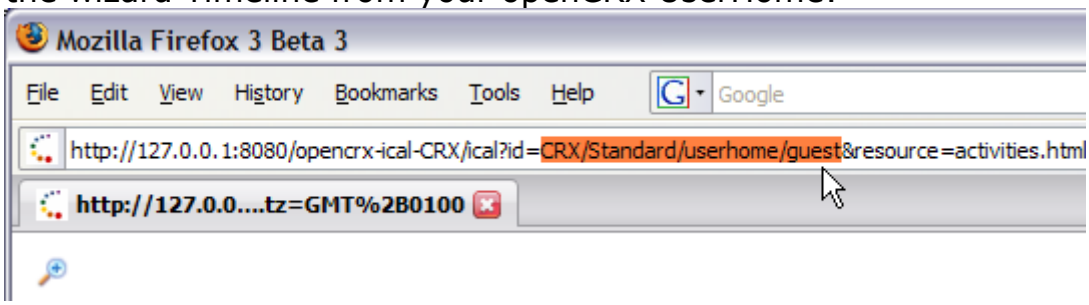
You can easily construct calendar selectors with the wizard Timeline, which is typically available on CrxObjects with sets of activities:



This wizard constructs a complete URL including a calendar selector. Hence, this wizard can be very useful whenever you need to construct a calendar selector. For example, the calendar selector for activities shown on the homepage of the user guest, i.e.

CRX/Standard/userhome/guest

can easily be retrieved from your browser's address bar after starting the wizard Timeline from your openCRX UserHome:



9.3.3 openCRX Activities mapped to Calendar Events

openCRX activities correspond to calendar events (or tasks). An event's iCal representation is stored in the corresponding activity's iCal attribute:

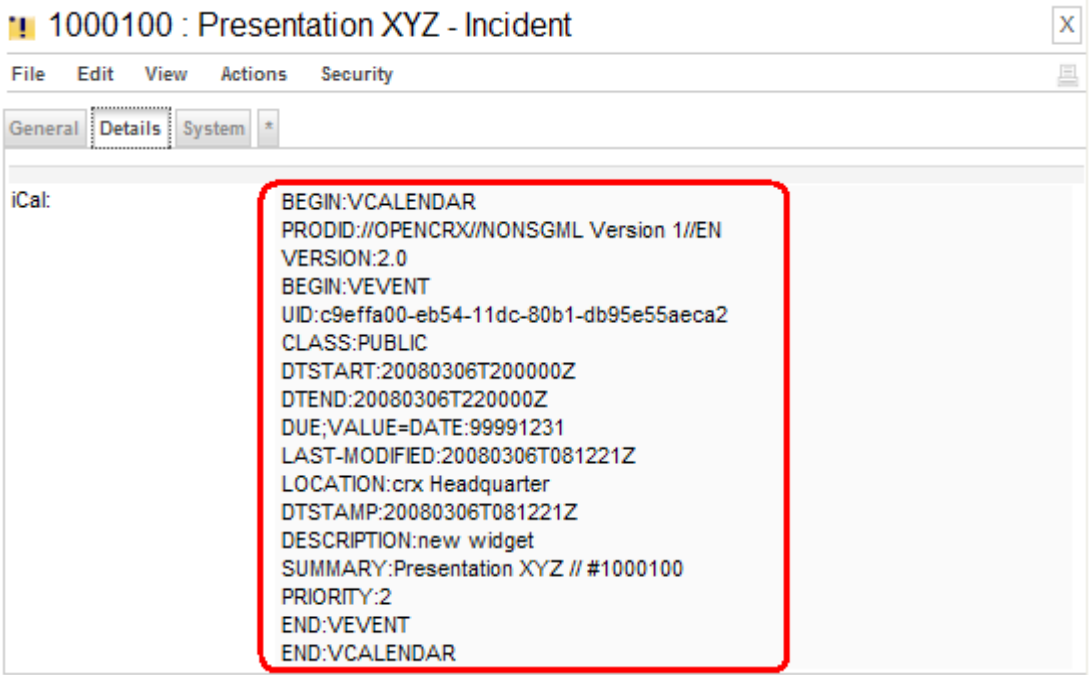


Figure 43: An openCRX activity's iCal representation

In the openCRX standard GUI the same activity is presented as follows:

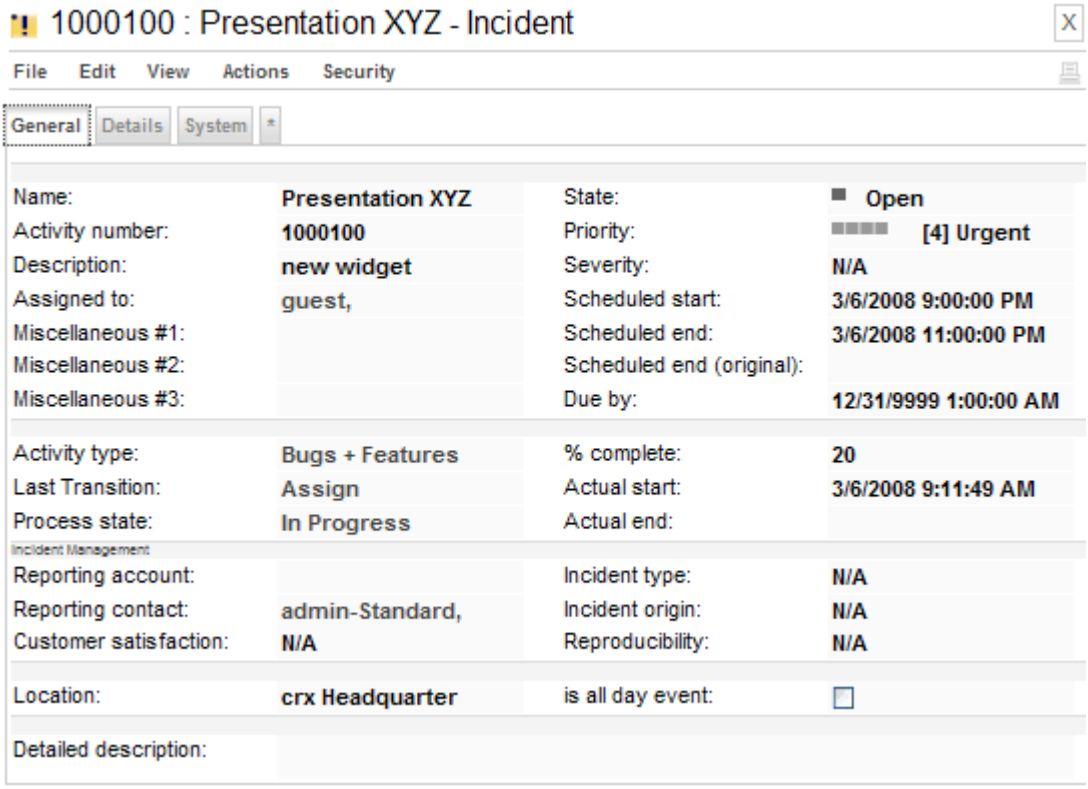


Figure 44: An openCRX activity in the standard GUI

As the above activity is assigned to "guest" it is automatically filtered to the homepage of the user guest:

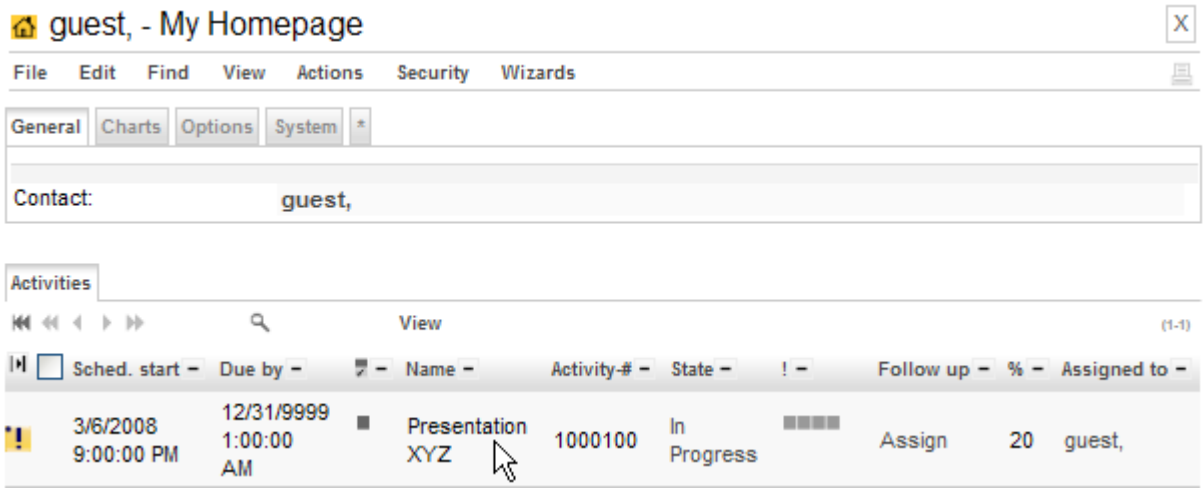


Figure 45: An openCRX activity filtered to a user's homepage

For the purpose of this guide this activity is also assigned to the tracker "shared". In various of this guide's examples we will make use of these facts by using the Calendar Selectors CRX/Standard/userhome/guest and CRX/Standard/tracker/shared.

9.3.4 Calendaring / Free Busy

Free Busy is part of the **iCalendar standard** (RFC 2445) for calendar data exchange (see also Wikipedia). Many calendar clients rely on this information for group scheduling. openCRX can derive free busy information on-the-fly from the respective activities; two methods are supported, one without authentication (supported by most clients) and one with authentication:

Free Busy URL (without authentication, requires openCRX principal guest):
http://<crxServer>:<Port>/opencrx-ical-<Provider>/**freebusy**
?id=<Provider>//<Calendar Selector>&resource=freebusy.ics
Example:
http://localhost:8080/opencrx-ical-CRX/freebusy?id=CRX/Standard/tracker/main&resource=freebusy.ics

Free Busy URL (with authentication):
http://<crxServer>:<Port>/opencrx-ical-<Provider>/**ical**
?id=<Provider>//<Calendar Selector>&resource=freebusy.ics
Example:
http://localhost:8080/opencrx-ical-CRX/ical?id=CRX/Standard/tracker/main&resource=freebusy.ics

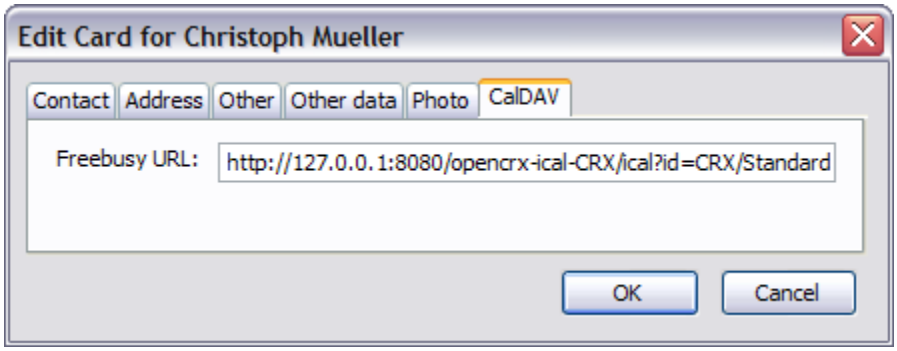
Please note that free busy information is provided by the openCRX server in a **read-only** fashion (i.e. free busy clients cannot update such information).

9.3.4.1 Free Busy Configuration of Thunderbird

Thunderbird 2.0 supports free busy if the following add-ons are installed:

- Lightning (at least v0.7)
- SOGo Connector (at least v0.67)

Once the SOGo Connector is installed, Contact Cards of the Thunderbird Address Book will feature a tab CalDAV. This is where you can enter the URL:



9.3.4.2 Free Busy Configuration of MS Outlook

See http://support.microsoft.com/kb/291621. Please note that Outlook does not support SSL with free busy.

9.3.5 Calendaring / iCalendar (ICS)

iCalendar is implemented/supported by a large number of products (see RFC 2445 or Wikipedia for information about the **iCalendar standard**, sometimes referred to as "**iCal**"). openCRX can derive iCalendar information on-the-fly from the respective activities. iCal clients must authenticate to read and/or write iCalendar data.

ICS URL (with authentication):

http://<crxServer>:<Port>/opencrx-ical-<Provider>/ical
?id=<Provider>//<Calendar Selector>&resource=activities.ics

Example:
http://localhost:8080/opencrx-ical-CRX/ical?id=CRX/Standard/tracker/main&resource=activities.ics

See chapter 9.3.2 Calendar Selectors for information on how to construct calendaring URLs.

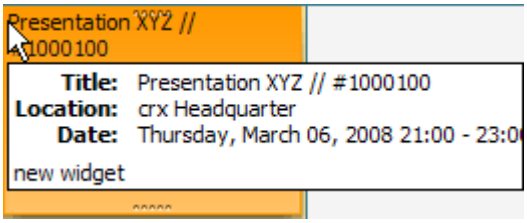
Please note that calendars marked as "R" (read only) in the table provided in chapter 9.3.2 Calendar Selectors are indeed read only, i.e. it is not possible to create new events/tasks with external ICS clients. Without a well-defined ActivityCreator being associated with the respective calendar selector it is simply not possible to create an activity and assign it the the appropriate activity groups.

9.3.5.1 ICS Configuration of Thunderbird/Lightning and Sunbird

Thunderbird 2.0 with the Lightning add-on (at least version 0.8) is a fully-fledged calendar client and offers virtually the same calendaring functionality as the stand-alone calendar client Sunbird (at least version 0.8). Creating a remote calendar (hosted on your openCRX server) is rather straightforward:

- start Thunderbird/Lightning or Sunbird
- select the menu **File > New Calendar**
- in the dialog window *Create New Calendar* you select **On the Network**
- then you select **iCalendar (ICS)**
- enter an ICS URL into the field **Location**;
  example: the user guest would connect to this openCRX homepage with the URL
  http://127.0.0.1:8080/opencrx-ical-CRX/ical?id=CRX/Standard/userhome/guest&resource=activities.ics
  See chapter 9.3.2 Calendar Selectors for information on ICS URLs
- give your calendar a name and pick a color of your choice

- That's it. You care connected to openCRX:



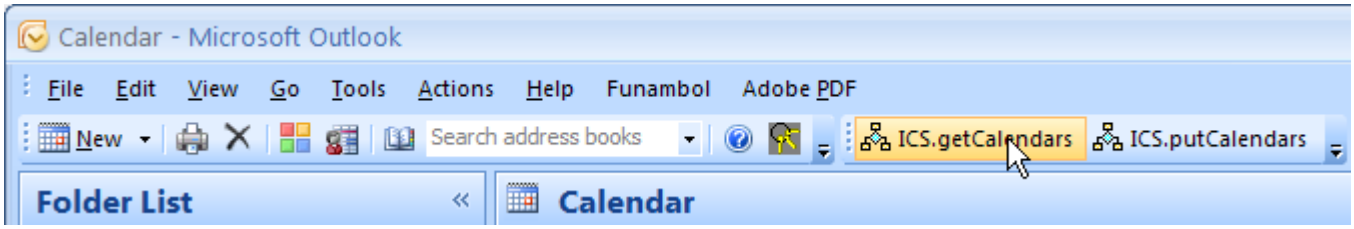
Version 0.8 of Thunderbird/Lightning and Sunbird requires a life connection to openCRX (i.e. no support for offline viewing/editing) unless you enable the experimental Cache of Ligthning/Sunbird.

9.3.5.2 ICS Configuration of MS Outlook

Out of the box Redmond's flagship MS Outlook 2007 does not offer you much choice with ICS calendars. You are stuck with one of the following 2 options:

- *Published Calendars*
  local calendar published to a remote location, but there is no sync with that remote calendar (i.e. changes to the remote calendar will never automatically make it back into your Outlook)
- *Internet Calendar Subscription*
  these calendars are stricly read-only in Outlook, i.e. not very useful...

Not to leave you out in the rain, we put together a bunch of VBA scripts that teach your Outlook a few new tricks. The scripts and detailed instructions for both MS Outlook 2003 and MS Outlook 2007 are available from http://www.opencrx.org/opencrx/2.2/Outlook_ICS_VCF_adapter.htm



9.3.5.3 Deleting Events

The openCRX ICS Adapter does not support deleting events (because deleting objects is typically not an acceptable operation in an enterprise environment). openCRX does support disabling of objects, however. If there is a need to disable events directly from your ICS Client, here is how to do it:

- Assuming you have a remote calendar CAL with the URL <U> defined in your ICS client, define a new remote calendar CAL-trash with the URL <U>**&disabled=true** (i.e. append the string "&disabled=true" to the URL); the name of the calendar is not important, we just call it "CAL-trash" to indicate that it contains disabled activities.

example:
existing calendar **CAL** (showing only events that are not disabled) with the URL
http://127.0.0.1:8080/opencrx-ical-CRX/ical?id=CRX/Standard/userhome/guest&resource=activities.ics

the URL of the calendar **CAL-trash** showing events that are disabled is
http://127.0.0.1:8080/opencrx-ical-CRX/ical?
id=CRX/Standard/userhome/guest&resource=activities.ics&disabled=true

- to delete an event of your calendar CAL, move the event from the calendar CAL to the calendar CAL-trash

Of course, you can also disable events with the standard GUI.

9.3.5.4 iCalender Guard Event

If you retrieve an iCalender from the openCRX ICS Adapter, the very first event is a so-called Guard Event:

- the Guard Event is always the first event delivered by the ICS Adapter
- the attribute SUMMARY corresponds to the Calendar-ID
- the attribute DTSTART is set to 19000101T000000Z

The openCRX ICS Adapter supports the creation of new events/tasks as long as a calendar's Guard Event is posted to the adapter together with the new event/task. The openCRX ICS Adapter also verifies the UID of the Guard Event.

9.3.5.5 VEVENT vs. VTODO

Many calendar applications differentiate between events (entries in a calendar with well-defined start and end date) and tasks or to-dos (entries in a task list with a well-defined due date). openCRX also supports this distinction. openCRX can even convert activities from VEVENT to VTODO and vice versa without loss of information. To convert an openCRX activity from one type to the other, simply edit the activity and change the iCal type in the respective drop down as shown below:

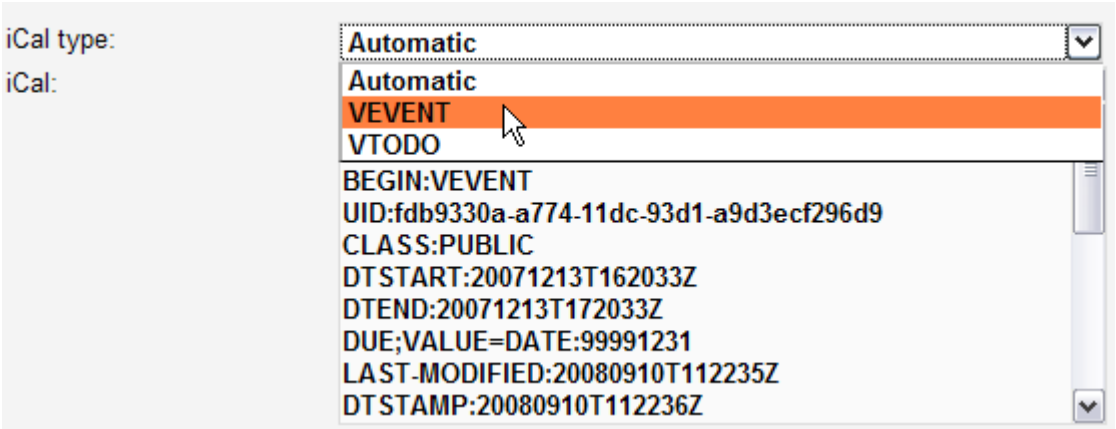


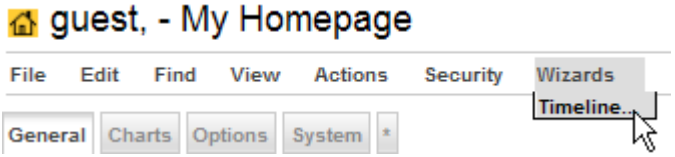
Figure 46: iCalendar conversion between VEVENT and VTODO

9.3.6 Calendaring / Timeline

Timeline is an extremely interesting DHTML-based AJAX widget for visualizing time-based events. It is like Google Maps for time-based information. A live example is available at http://simile.mit.edu/timeline/

Figure 47: Timeline visualizes time-based events

CrxObjects with sets of activities typically feature the wizard Timeline. Simply call that wizard to construct a timeline to visualize activities right in your browser:



9.4 Mailstore / IMAP

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 IMAP adapter (get more information about IMAP or read what Wikipedia is saying about IMAP). The advantages of such a standardized IMAP adapter are:

- works with any IMAP 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 references to sender and recipient(s) if the respective e-mail addresses are present in openCRX

In a nutshell this means that you can use any IMAP client to connect to openCRX and view openCRX EMailActivities. openCRX activity groups are mapped to IMAP folders. The folders contain openCRX EMailActivities.

Viewing/exporting of EmailActivities is always possible, creating/updating of EmailActivities requires that an E-Mail Activity Creator is defined for the respective Activity Group, and deleting of EmailActivities is not supported.

If you **move** an e-mail message from a non-openCRX IMAP folder to an openCRX IMAP folder and the target folder does not have a valid E-Mail Activity Creator defined, openCRX will not be able to create an EMailActivitiy in that folder. Due to the **move** operation the message is deleted from the source folder and your e-mail message is lost. Hence, it is good practice to **copy** (and not move) e-mails to openCRX IMAP folders. Only after verifying that the EMailActivitiy was actually created in the target folder should you delete (if necessary) the message from the source folder.

E-mail addresses should be unique!

If you import e-mails into openCRX with the IMAP Adapter, openCRX tries to match sender and recipients based on e-mail addresses. For obvious reasons, this will produce unexpected (if not undesired) results if e-mail addresses in openCRX are not unique.

You can test your openCRX database for duplicate e-mail addresses with the following query:

```
SELECT email_address, count(*)
FROM OOCKE1_ADDRESS
GROUP BY email_address HAVING count(*) > 1
```

E-mail address UNASSIGNED.

If the openCRX IMAP Adapter is not able to link a sender/recipient address to an existing e-mail address in openCRX, the adapter can assign such senders/recipients to an account of your choice. Simply add an e-mail address UNASSIGNED to the particular account as shown below:

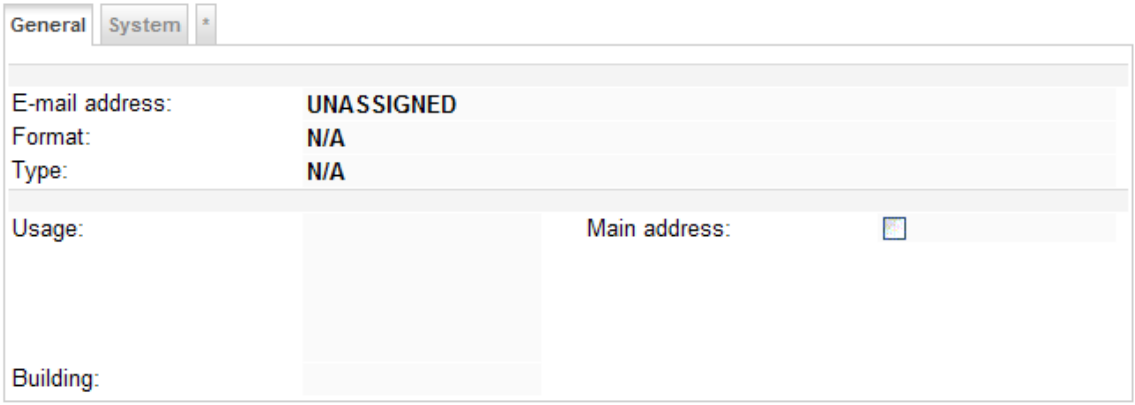


Figure 48: E-Mail Address UNASSIGNED

In order to detect missing e-mail addresses (and then enter them and reassign the respective e-mail activities) you can simply work through the activities assigned to the account with the above e-mail address.

The following information is required to connect an IMAP client to openCRX:

Host	IP address or host name of openCRX Server Examples: localhost, 127.0.0.1, myCrxServer.myCompany.com, etc.
Port	1143 (note that the IMAP standard port is 143)
User name	<login principal name>@ Example: **guest@Standard**
Password	principal's openCRX password

The openCRX IMAP adapter does not support SSL, i.e. you should use this feature on secured networks only (Intranet, VPN, ...).

9.4.1 Configuring the openCRX IMAP Port

The openCRX IMAP port is by default set to 1143 (to avoid conflicts with other IMAP daemons listening on the IMAP standard port 143). You can change this configuration in the file **web.xml** located in opencrx-groupware-CRX.ear\opencrx-imap-CRX.war\WEB-INF\

Look for the the param-name **port**.

If you build your own EARs you can change the openCRX LDAP port in your project's file **build.properties** (imap.listenPort) or directly in your **build.xml**.

9.4.2 Configuring the IMAP Maildir Cache

For increased performance the openCRX IMAP Adapter works with a cache. The location of this cache, the so-called Maildir, can be set as a JAVA_OPTS.

You can reset the cache by deleting it. The openCRX IMAP Adapter will recreate the cache automatically. Hence, you might want to delete the cache when restarting Apache Tomcat or JBoss.

9.4.2.1 Maildir Configuration with Apache Tomcat

Add the following line to your JAVA_OPTS in your Tomcat start batch file (e.g. **tomcat.bat**, **run.bat**, **run.sh**, etc.):

Listing 15: Set org.opencrx.maildir for Apache Tomcat

```
...
set JAVA_OPTS=%JAVA_OPTS% -Dorg.opencrx.maildir="%CATALINA_HOME%\maildir"
...
```

9.4.2.2 Maildir Configuration with JBoss

AAdd the following line to your JAVA_OPTS in your JBoss start batch file (e.g. **run.bat**, **run.sh**, etc.):

Listing 16: Set org.opencrx.maildir for Apache Tomcat

```
...
set JAVA_OPTS=%JAVA_OPTS% -Dorg.opencrx.maildir=D:\jboss-4.2.1.GA\server\default\maildir
...
```

Adapt D:\jboss-4.2.1.GA\server\default\maildir to your environment and make sure that there are no line breaks in the set commands. Each -D options is of the form -Dname=value and must be on a single line.

9.4.3 IMAP Configuration of Thunderbird

The following information is required to configure an IMAP account:

Email account	<login principal name>@ Example: guest@Standard
Password	openCRX password of the respective principal
Your name	<login principal name>@ Example: guest@Standard
Email Address	your e-mail address Example: guest@mycompany.com
Type of server	**IMAP**
Incoming Server Outgoing Server	name or IP address of your openCRX server Example: 127.0.0.1
Port	**1143**
Incoming User Name	<login principal name>@ Example: guest@Standard

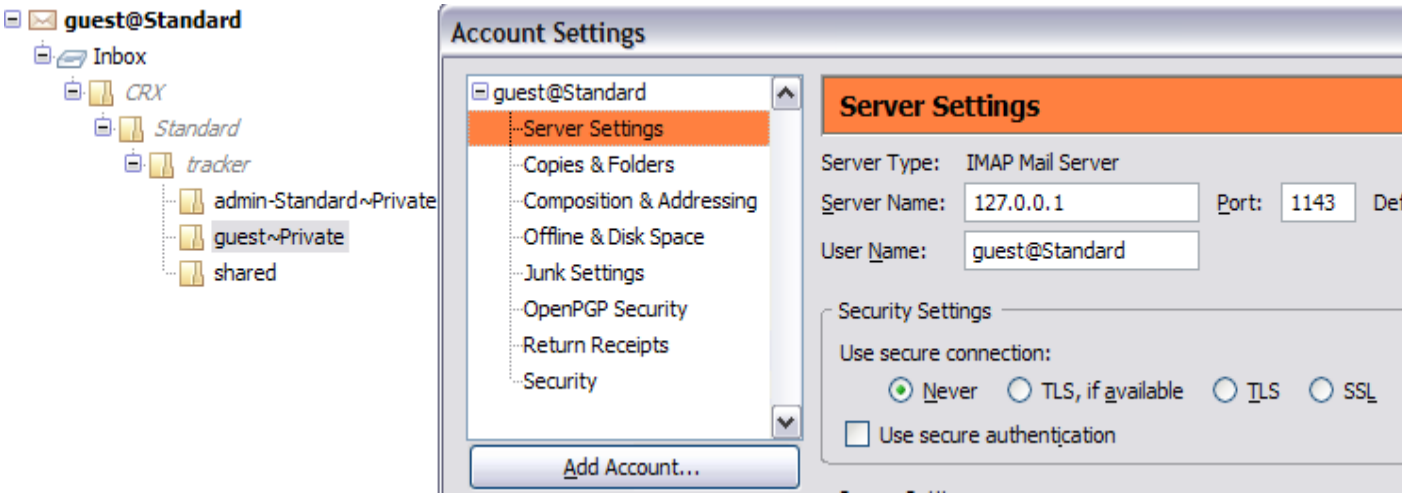


Figure 49: Thunderbird IMAP Configuration

If you **move** an e-mail message from a non-openCRX IMAP folder to an openCRX IMAP folder and the target folder does not have a valid E-Mail Activity Creator defined, openCRX will not be able to create an EMailActivitiy in that folder. Due to the **move** operation the message is deleted from the source folder in your IMAP client and your e-mail message will be lost.

Hence, it is good practice to **copy** (and not move) e-mails to openCRX IMAP folders. Only after verifying that the EMailActivitiy was actually created in the target folder should you delete (if necessary) the message from the source folder.

9.4.4 IMAP Configuration of MS Outlook

The following steps are required to configure MS Outlook 2007 for LDAP:

Email account User Name	<login principal name>@ Example: guest@Standard
Password	openCRX password of the respective principal
Your Name	<login principal name>@ Example: guest@Standard
E-mail Address	your e-mail address Example: guest@mycompany.com
Account type	**IMAP**
Incoming mail server Outgoing mail server	name or IP address of your openCRX server Example: 127.0.0.1
Incoming Port	**1143**
Incoming User Name	<login principal name>@ Example: guest@Standard

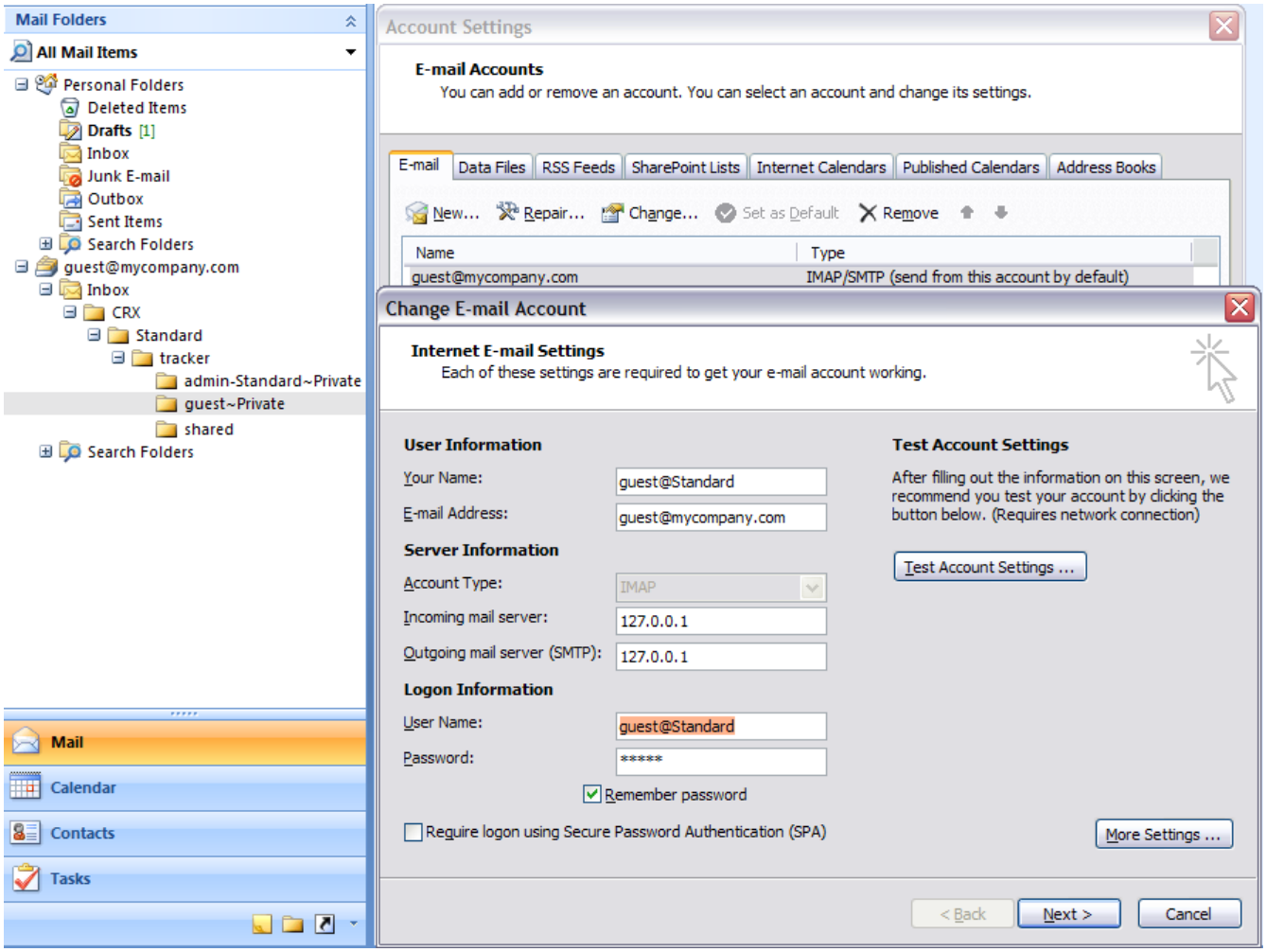


Figure 50: Thunderbird IMAP Configuration

If you **move** an e-mail message from a non-openCRX IMAP folder to an openCRX IMAP folder and the target folder does not have a valid E-Mail Activity Creator defined, openCRX will not be able to create an EMailActivitiy in that folder. Due to the **move** operation the message is deleted from the source folder of your IMAP client and your e-mail message will be lost.

Hence, it is good practice to **copy** (and not move) e-mails to openCRX IMAP folders. Only after verifying that the EMailActivitiy was actually created in the target folder should you delete (if necessary) the message from the source folder.

10 openCRX is a REST Service (Web Service)

The openCRX REST servlet allows easy 3rd party integration with openCRX. The full functionality of the openCRX API can be accessed via REST requests, i.e. you can use openCRX as a REST Service.

See http://code.google.com/p/rest-client/ for a REST client.

Sample REST requests are available from http://www.opencrx.org/opencrx/2.3/new.htm#REST

11 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.

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).

11.1 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.

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 Error: Reference source not found) in addition to the XML importer.

11.2 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-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.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:

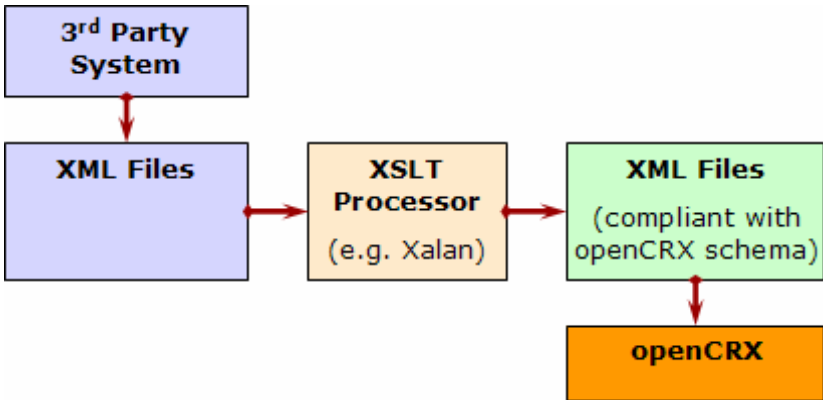


Figure 51: XML import from 3rd party system – overview

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:

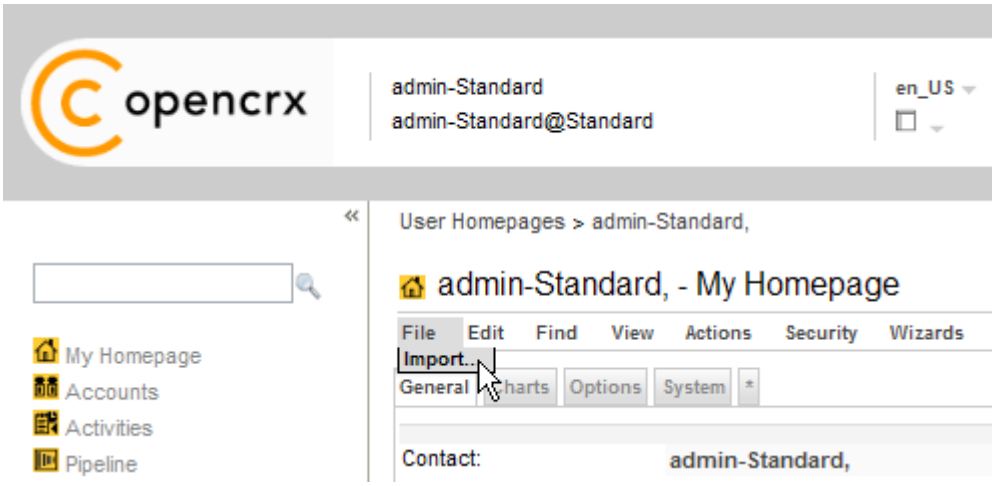


Figure 52: Interactive import of XML Files

  Click on the button Browse... and navigate to XML file to be imported. Next you click OK to 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.

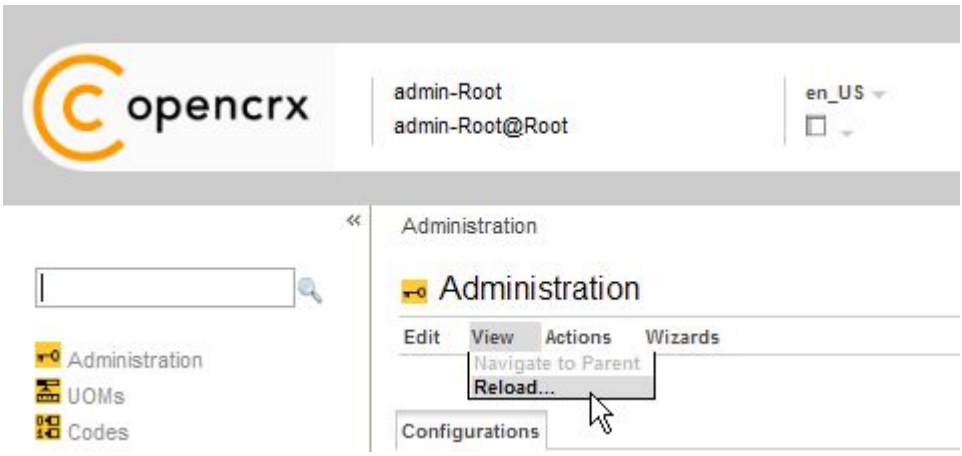
- **Bulk Import**

  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.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 53: 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.

11.2.1 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:

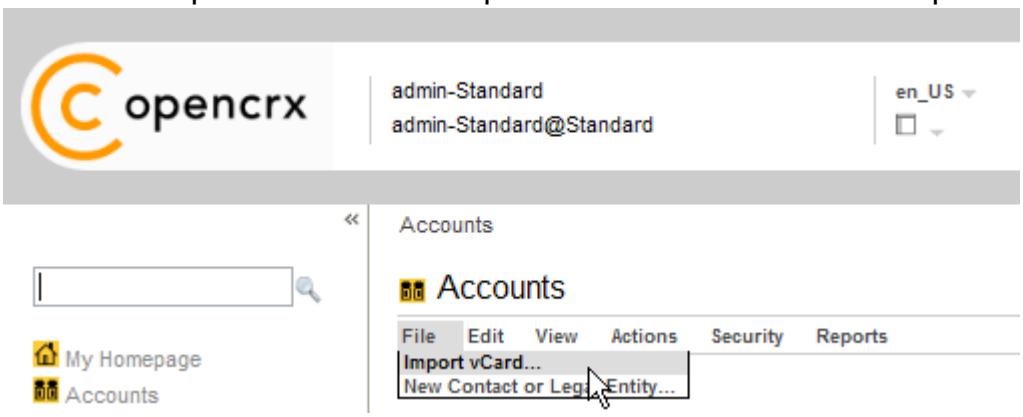


Figure 54: Operation vCard Import

- 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

11.2.2 Importing E-Mails

Please refer to the Chapter 8 E-mail Services, in particular chapter 8.3.1 Import E-mails.

11.2.3 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.

11.3 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.

11.3.1 Exporting XML Files

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

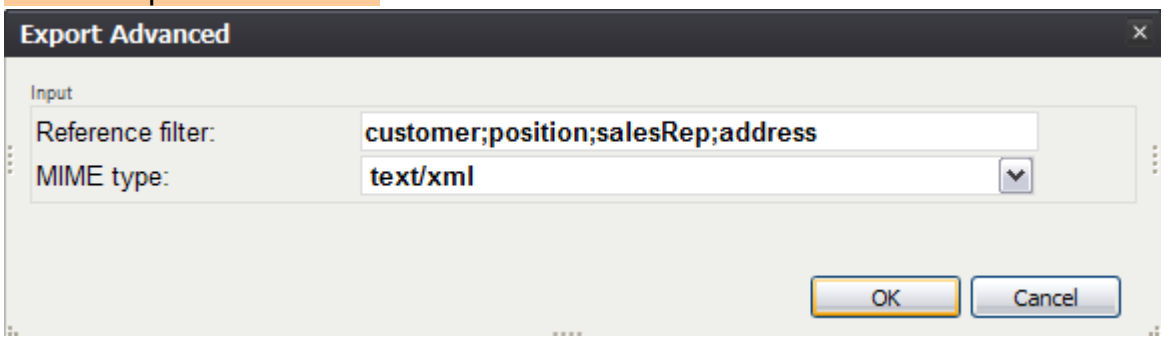


Figure 55: Exporting SalesOrder as XML File

In order to better control which additional objects (composites, referenced objects, ...) the XLM exporter should export together with the object loaded in the Inspector, you can (optionally) provide a **reference filter**. By default, only the current object will be exported. If you provide – for example when exporting a sales order – **customer;address** as a reference filter, the customer and all referenced addresses will be exported together with the main object.

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

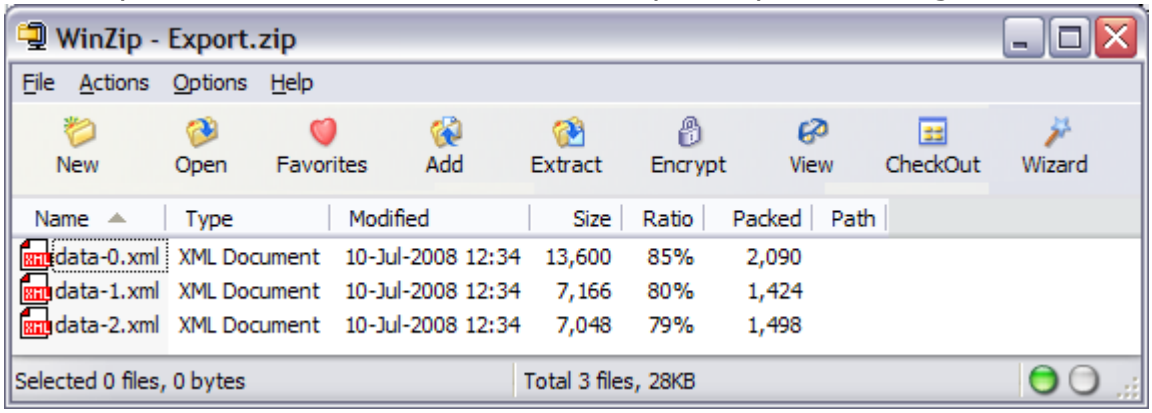


Figure 56: XML Exporter provides XML data file and code tables as ZIP file

The openCRX ICS Adapter can also export iCalendars in XML format:

ICS URL (to get XML file with authentication):

http://<crxServer>:<Port>/opencrx-ical-<Provider>/ical
?id=<Provider>//<Calendar Selector>&resource=activities.xml

Example:
http://localhost:8080/opencrx-ical-CRX/ical?id=CRX/Standard/tracker/main&resource=activities.xml

See chapter 9.3.2 Calendar Selectors for information on how to construct calendaring URLs.

11.3.1.1 ICS Configuration of Thunderbird/Lightning and Sunbird

Thunderbird 2.0 with the Lightning add-on (at least version 0.8) is a fully-fledged ca

11.3.2 Exporting Data to MS Excel / OO Calc Files

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

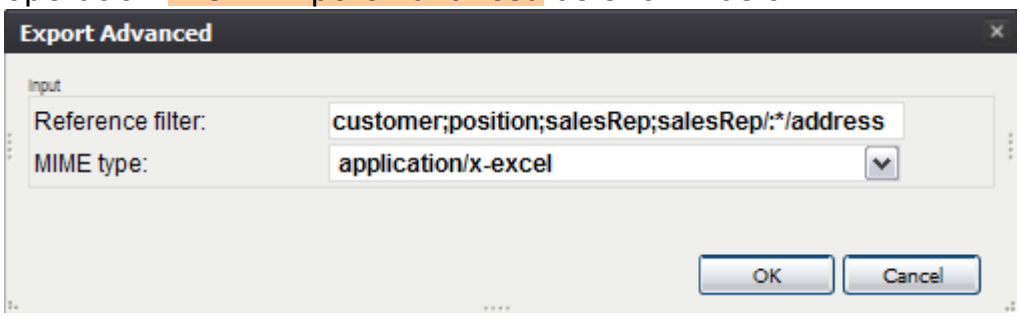


Figure 57: Exporting SalesOrder as Spreadsheet File

In order to better control which additional objects (composites, referenced objects, ...) the XLM exporter should export together with the object loaded in the Inspector, you can (optionally) provide a **reference filter**. By default, only the current object will be exported. If you provide – for example – **:*/:*** as a reference filter, all composites up to 2 levels deep will be exported together with the main object (this should be sufficient for most use cases). You can also provide a reference filter to dereference and export referenced objects like the customer or the salesRep of a sales order.

If the export is successful the exporter will terminate with status OK and you will be provided with a link to the file Export.xls containing the raw data:

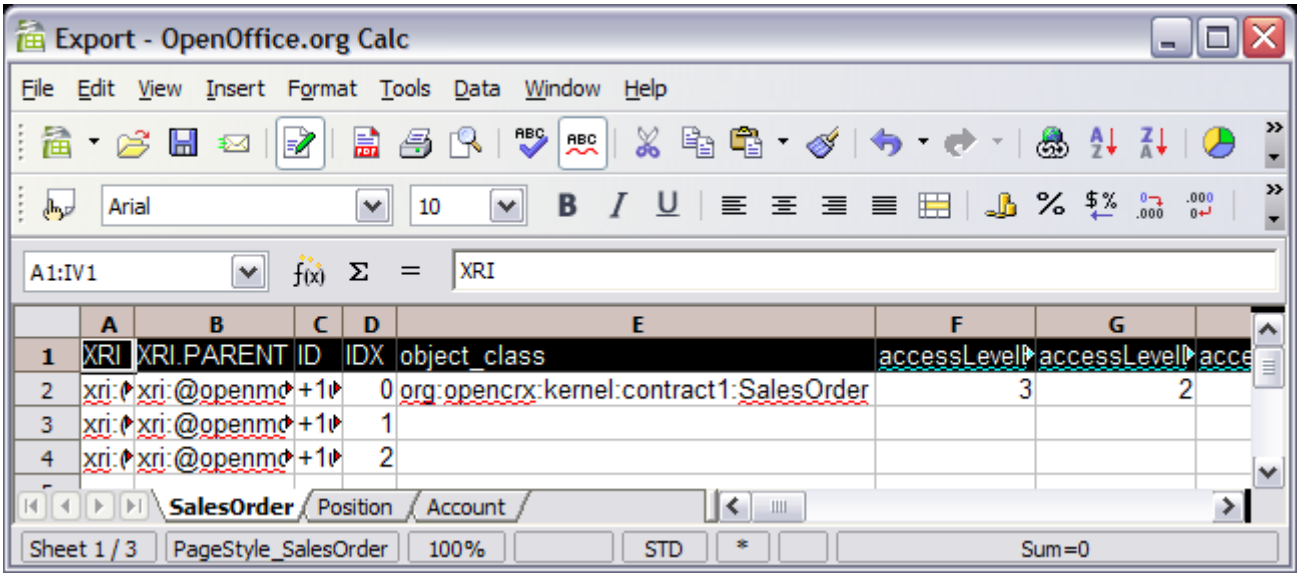


Figure 58: Exported Spreadsheet File

Based on such spreadsheet files end-users can easily create reports or do some ad-hoc data analysis without the need to know anything about Java or writing JSPs. The standard distribution of openCRX includes various example reports based on this technology.

11.3.3 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
- click on the tab Account
- select the contents of the field vCard and copy it to an empty text file
- save the text file with any name and extension "vcf", e.g. contact.vcf:

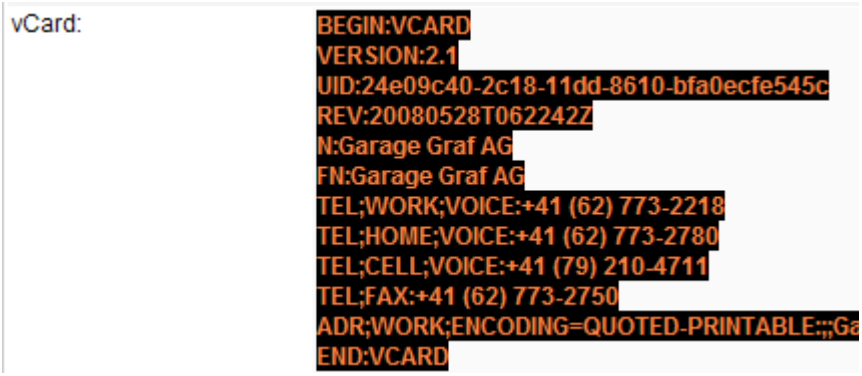


Figure 59: Export Contact as vCard

11.3.4 Exporting openCRX Contacts (→ Outlook Contacts)

Navigate to the contact you want to export to MS Outlook and execute the

Wizard Export to MS Outlook:

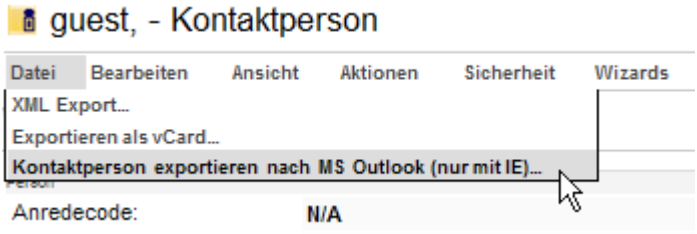


Figure 60: Export Contact to MS Outlook

Please note that this wizard requires that Internet Explorer and MS Outlook are both installed on your computer (the export invokes an ActiveX control). With other browsers 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

An animation is available at

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

11.3.5 Exporting openCRX Meetings (→ iCalendar Files)

These are the steps to export an individual activity (e.g. a meeting or a sales visit) to an iCalendar file:

- navigate to the meeting (or sales visit) you want to export
- click on the tab Details
- select the contents of the field iCal and copy it to an empty text file
- save the text file with any name and extension "ics", e.g. meeting.ics:

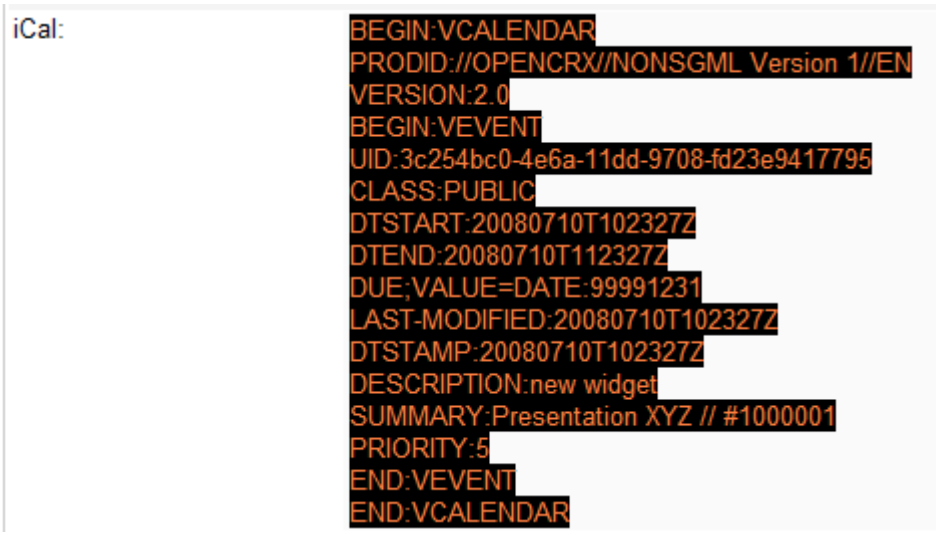


Figure 61: Exporting Meeting / Sales Visit as iCalendar File

11.3.6 Exporting E-Mails

Please refer to the Chapter 8 E-mail Services, in particular chapter 8.2.2 Export E-mails.

11.3.7 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.

12 Customizing openCRX

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

12.1 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.ear\opencrx-core-CRX.war\WEB-INF**web.xml**

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

Listing 17: 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 18: 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.

12.2 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.ear\opencrx-core-CRX.war\WEB-INF**web.xml**

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

Listing 19: Packages 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 20: Activating/Deactivating Packages 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 -->
...
```

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**).

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.

12.2.1 Enabling/Disabling Root Menu Entries

Individual user can enable/disable root menu entries with the wizard User Settings (available on a user's Homage):

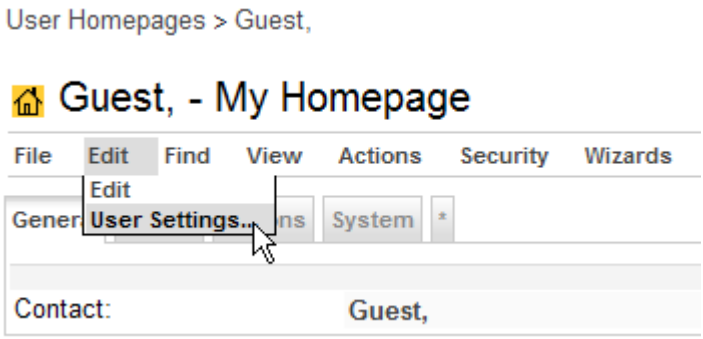


Figure 62: Launch Wizard User Settings

Once the wizard has loaded, uncheck entries you don't need:

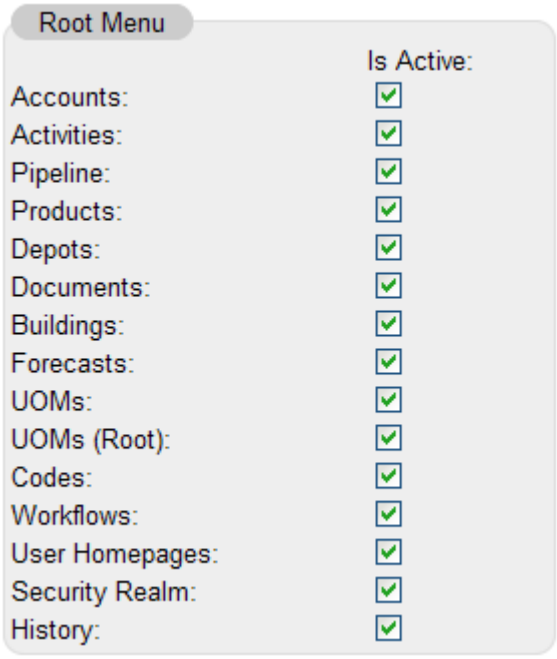


Figure 63: Wizard User Settings – enable/disable Root Menu Entries

Please note that entries corresponding to packages disabled by the openCRX administrator cannot be enabled with this wizard. Packages disabled in web.xml are not available at all!

Depending on the width of your screen you can adjust the number of items shown as tabs in the top-level navigation in the same wizard (fewer items for narrow screens, more items for wider screens).

12.3 Role-based UI

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

12.3.1 Model Permissions

Model permissions are not implemented yet.

12.3.2 Custom Layout JSPs

openCRX is distributed with 2 default layout JSPs located in the directory opencrx-core-CRX.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 Tomcat or 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.

13 Integration with Office Application

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

13.1 MS Word, OO Writer, etc.

openCRX supports the JSP-wizard-based generation of RTF documents. You can generate RTF documents from scratch or merge data with existing RTF templates. The RTF documents are generated on the fly and can be opened with any RTF-compatible word processor including OpenOffice Writer and MS Word.

You can test this feature on our demo server (or on your own installation if you installed the openCRX Server) with the following steps:

- connect and login as user **guest**
- navigate to any contact and execute the operation **Tools > Mail Merge --> RTF Document**
- the RTF document is generated on the fly and can be downloaded from the openCRX server

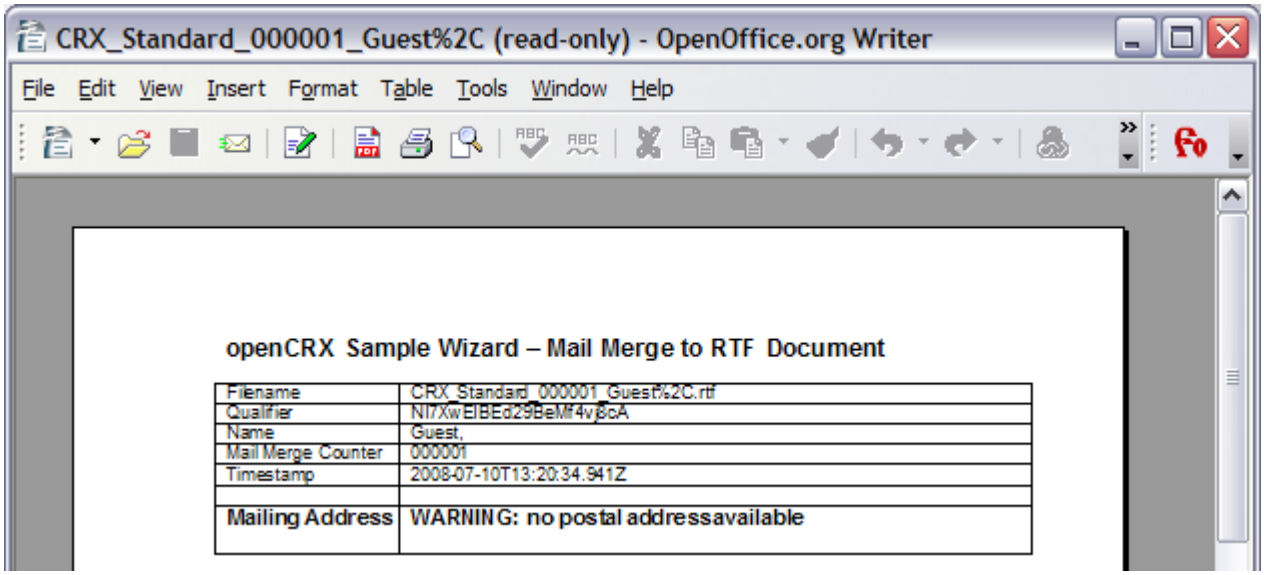


Figure 64: RTF Document generated by merging live data with template

If you installed the openCRX SDK you will find the template and the JSP wizard in the following locations:

- <SDK_Install_Dir>\opencrx-2.1.0\core\src\data\org.opencrx\documents\MailMerge_Template.rtf
- **<SDK_Install_Dir>\opencrx-2.1.0\core\src\data\org.opencrx\wizards\en_US\MailMerge.jsp**

With this approach it is quite easy to generate all kinds of documents, including letters, invoices, purchase orders, etc.

13.2 MS Excel, OO Calc, etc.

openCRX supports both a raw spreadsheet export of data and the JSP-wizard-based generation of XLS documents. You can generate XLS documents from scratch or merge data with existing XLS templates. The XLS documents are generated on the fly and can be opened with any XLS-compatible spreadsheet program including OpenOffice Calc and MS Excel.

You can test this feature on our demo server (or on your own installation if you installed the openCRX Server) with the following steps:

- connect and login as user **guest**
- navigate to **Accounts**
- execute the operation **Reports > Contacts Report** as shown below

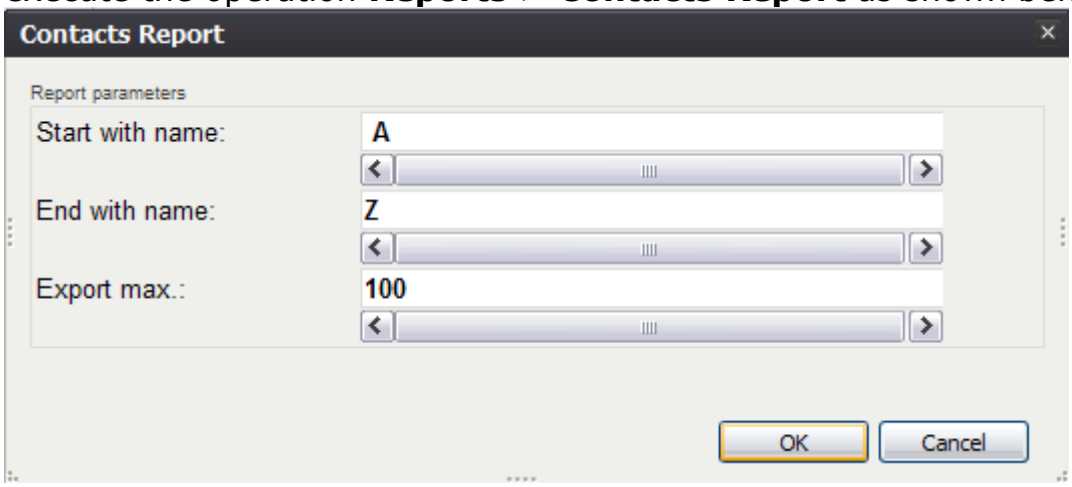


Figure 65: Contacts Export Dialog

- the Spreadsheet is generated on the fly and can be downloaded from the openCRX server by clicking on the link provided in the operation result

You might also want to look at the information provided in the chapters

- 11.3.2 Exporting Data to MS Excel / OO Calc Files
- 14.1 Standard Reports

13.3 MS Outlook, Thunderbird/Lightning, Sunbird, etc.

See chapter 9 Groupware Services.

14 Reporting

openCRX provides various technologies that enable you to create reports of a wide variety, anything from simple ad-hoc reports to large scale bulk reports.

14.1 Standard Reports

Included in the openCRX standard distribution are the following reports:

- Account List (available on Account Filters)
- Account Member List (available on Groups)
- Activity List (available on Activity Groups and Filters)
- Activity List with Follow-Ups (available on Activity Groups and Filters)
- Contract List (available on Contract Filters)
- Contract List with Positions (available on Contract Filters)

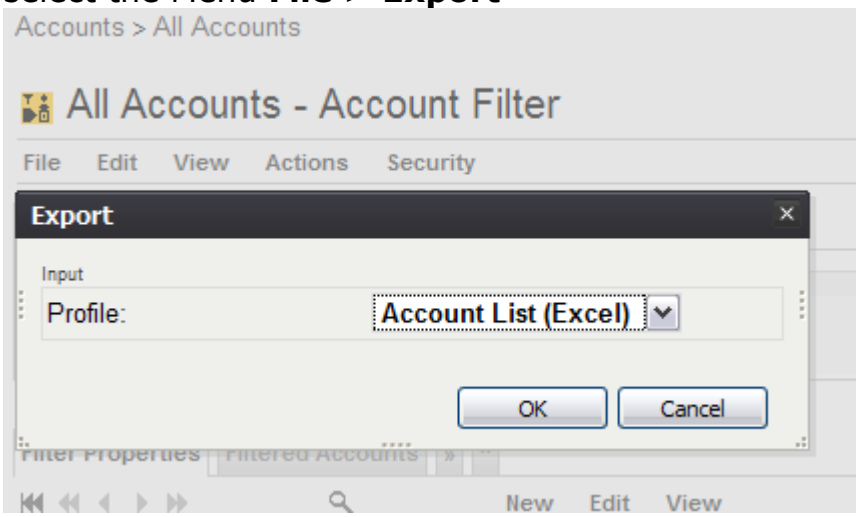
You can install these standard reports into any openCRX Segment with the wizard Segment Setup. The reports are based on spreadsheet templates. If you installed the openCRX SDK you will find the templates in the following location:

<SDK_Install_Dir>\opencrx-2.1.0\core\src\data\org.opencrx\documents

The templates contain VBA macros to "beautify" the reports. These macros will probably not work in OO Calc...

You can test – for example – the Report Account List with the following steps:

- connect and login as user **guest**
- navigate to **Accounts > Account Filters** and select All Accounts
- select the Menu **File > Export**



- click the button [OK] to generate Export.xls
- open Export.xls with Excel (enable Macros) to get an alphabeticallly sorted list of accounts complete with e-mail address and phone numbers

14.2 Ad-hoc Reporting

See chapter 11.3.2 Exporting Data to MS Excel / OO Calc Files.

Standard office suite know how (and maybe a bit of macro programming) should get you a long way. Once you're ready to "institutionalize" a report you can prepare a template and an openCRX Export Profile. Consult the templates and the Export Profiles provided by the opeCRX distribution for examples.

14.3 Large Scale Reporting

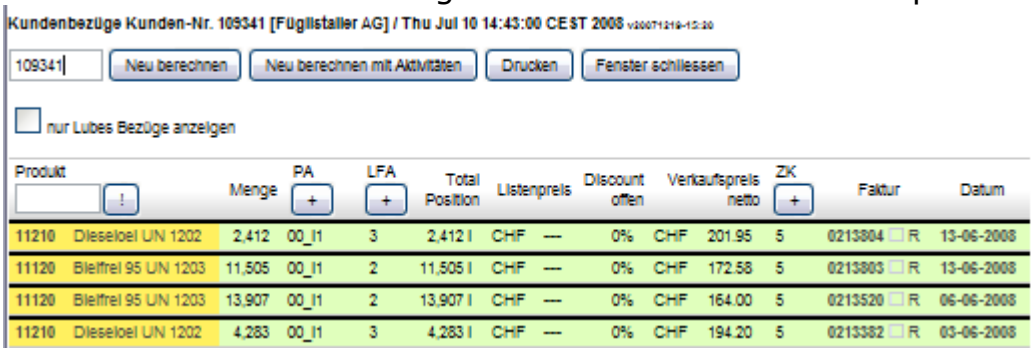
If your task is to produce a large number of reports (e.g. monthly reporting for all your clients) or reports based on large amounts of data, spreadsheet-based reporting is probably not the way to go. Maybe you want to generate reports in a format other than XLS. On the one hand, openCRX already includes libraries to generate reports and documents in various formats, on the other hand you can easily add additional libraries to openCRX.

Format	Library / Additional Information
XLS	Apache POI / http://poi.apache.org/
PDF	iText / http://www.lowagie.com/iText/
RTF	Simple RTF Writer / http://www.it-s-easy.com/downloads.htm

Obviously, there are many more possibilities, like for example exporting data in XML format and then doing some kind of fancy transformation. One such example is available from http://www.koalix.org/

In terms of how to generate your reports, there are also various options available depending on your preferences:

- JSP-Based Reporting
  This approach is typically recommended if you need on-demand reporting and the generation of the report does not put an undue burden on the server. The following screen shot shows an example HTML-report:



- Java Program
  Large-scale batch reporting can be done with a Java Program (basically an openCRX client programmed in Java that prepares the desired reports).
- BI-Reporting Suite
  If you plan to use a BI-Reporting Suite (e.g. Crystal Reports, Pentaho, BIRT, etc.), you should keep in mind that **directly accessing the openCRX database is not a very good idea**. We strongly recommend you either retrieve data through the openCRX API or set up a dedicated reporting DB (the process to populate such a reporting DB should retrieve data from the openCRX DB through the openCRX API). The reason for not accessing the openCRX database directly is the following one: while the openCRX API is stable, the OO-to-relational mapping is not and hence the schema of the openCRX DB is subject to change over time. Hence, if you access the openCRX DB directly you will have to adapt your reports if the DB schema changes, a potentially expensive proposition.

15 Miscellaneous Topics

15.1 SNMP Monitoring

The SNMP agent for the Sun JVM can be enabled as described at http://java.sun.com/j2se/1.5.0/docs/guide/management/SNMP.html and http://www.ilikespam.com/blog/monitoring-the-jvm-with-snmp.

1) Put snmp.acl in TOMCAT_LWC_HOME/bin and give read access to the file for the owner only.

```
#The communities public and private are allowed access from the local host.
acl = {
  {
    communities = public, private
    access = read-only
    managers = localhost
  }
}
# Traps are sent to localhost only
trap = {
    {
      trap-community = public
      hosts = localhost
    }
  }
```

2) Add the following options to tomcat.sh
   `-Dcom.sun.management.snmp.port=8161`
   `-Dcom.sun.management.snmp.acl.file=TOMCAT_LWC_HOME/bin/snmp.acl`

3) After startup the variables can be retrieved with
   `snmpwalk -c public -v2c localhost:8161 .1.3.6.1.4.1.42.2.145.3.163.1.1.4.1.`
   See http://support.ipmonitor.com/mibs/JVM-MANAGEMENT-MIB/oids.aspx for a list of all OIDs supported by the JVM.

A simple cron-based monitoring environment can invoke snmpwalk periodically and send mail if a monitored parameter violates a predefined constraint. Use gkrellm with the snmp extension (see http://triq.net/gkrellm_snmp.html) or OpenNMS (see http://www.opennms.org/) for more advanced monitoring solutions.

This approach only works for the Sun JVM. JRockit does not seem to support the options -Dcom.sun.management.snmp.*

15.2 Tomcat w/ openCRX and LDAP-based Authentication

You can connect Tomcat w/ openCRX with OpenLDAP as follows:

- replace the JDBC Realm entry in the context XML opencrx-core-CRX.xml with the following entry:

```
 <Realm
  className="org.apache.catalina.realm.JNDIRealm"
  debug="99"
  connectionName="cn=Manager,dc=opencrx,dc=org"
  connectionPassword="secret"
  connectionURL="ldap://localhost:389"
  userPassword="userPassword"
  userPattern="cn={0},ou=people,dc=opencrx,dc=org"
  roleBase="ou=groups,dc=opencrx,dc=org"
  roleName="cn"
  roleSearch="(uniqueMember={0})"
/>
```

- Users must be entered into your LDAP server as follows (userPattern):

```
# Users
dn: cn=guest,ou=people,dc=opencrx,dc=org
objectClass: organizationalPerson
sn: guest
cn: guest
userPassword: opencrx

dn: cn=admin-Root,ou=people,dc=opencrx,dc=org
objectClass: organizationalPerson
sn: admin-Root
cn: admin-Root
userPassword: opencrx

dn: cn=admin-Standard,ou=people,dc=opencrx,dc=org
objectClass: organizationalPerson
sn: admin-Standard
cn: admin-Standard
userPassword: opencrx
```

- Groups must be entered into your LDAP server as follows (roleBase, roleName, and roleSearch):

```
# Groups
dn: ou=groups,dc=opencrx,dc=org
objectClass: organizationalUnit
ou: groups

dn: cn=OpenCrxRoot,ou=groups,dc=opencrx,dc=org
objectClass: groupOfUniqueNames
cn: OpenCrxRoot
uniqueMember: cn=admin-Root,ou=people,dc=opencrx,dc=org

dn: cn=OpenCrxAdministrator,ou=groups,dc=opencrx,dc=org
objectClass: groupOfUniqueNames
cn: OpenCrxAdministrator
uniqueMember: cn=admin-Standard,ou=people,dc=opencrx,dc=org

dn: cn=OpenCrxUser,ou=groups,dc=opencrx,dc=org
objectClass: groupOfUniqueNames
cn: OpenCrxUser
uniqueMember: cn=guest,ou=people,dc=opencrx,dc=org
```

16 Next Steps

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