

openCRX Admin Guide

Version 2.0



www.opencrx.org

07-Mar-2008 @ 03:59:55 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.....	7
1.1	Who this book is for.....	7
1.2	What do you need to understand this book.....	7
1.3	Tips, Warnings, etc.....	7
2	Prerequisites.....	8
3	Security.....	9
3.1	Introduction.....	9
3.1.1	Basic Concepts and Conventions.....	9
3.1.2	Permissions / Access Control.....	12
3.2	Default Settings.....	14
3.3	Activating Security.....	16
3.4	Security Settings of New Objects.....	16
3.5	Checking Permissions.....	17
3.6	Login Procedure.....	18
3.6.1	Tomcat / Application Server Login.....	18
3.6.2	Segment Login.....	18
3.6.3	Disabling Login.....	18
3.7	Resetting Security.....	19
4	Managing Users.....	20
4.1	Creating Users.....	20
4.1.1	Create Users as Segment Administrator.....	21
4.1.2	Import Subjects and Application Login Principals.....	22
4.1.3	Import Users.....	23
4.2	Disable/Deactivate Users.....	24
4.2.1	Disable Users at the level Tomcat /Application Server.....	24
4.2.2	Disable Users at the level openCRX.....	24
5	Deployment Scenarios.....	25
5.1	Typical Deployment Scenarios.....	25
5.2	Multi Entity Deployment Scenarios.....	26
5.2.1	Multiple Data Segments in a single DB.....	26
5.2.2	Multiple DBs.....	27
5.3	openCRX Custom Applications.....	27
6	Workflow Controller.....	28
6.1	Workflow Controller Configuration.....	30
6.1.1	Startup Configuration in web.xml.....	30
6.1.2	ServerURL.....	31
6.1.3	Handler pingrate and autostart.....	31
6.2	Servlet IndexerServlet.....	32
6.3	Servlet SubscriptionHandler.....	32
6.4	Servlet WorkflowHandler.....	33
6.5	Trouble Shooting Servlets.....	34

7	Subscribe / Notify Services	35
7.1	Example Subscription – Account Modifications.....	37
7.2	Example Subscription with Filtering.....	38
7.3	Trouble Shooting Notification Services.....	39
8	E-mail Services	40
8.1	Install and Configure Mail Resource and E-Mail Services.....	41
8.1.1	Installation of JavaMail and JAF.....	41
8.1.2	Mail Resource for openCRX on Apache Tomcat.....	41
8.1.2.1	Add resource definition to opencrx-core-CRX.xml.....	41
8.1.2.2	Mail Resource in web.xml.....	42
8.1.3	Mail Resource for openCRX on JBoss.....	43
8.1.3.1	Create mail-service.xml.....	43
8.1.3.2	Mail Resource in web.xml and jboss-web.xml.....	44
8.2	Outbound E-mail.....	45
8.2.1	Outbound E-mail Configuration.....	45
8.2.2	Export E-mails.....	47
8.2.3	Send E-mails directly from openCRX.....	48
8.2.4	Send E-mails as Attachments to your Mail Client.....	49
8.3	Inbound E-mail.....	50
8.3.1	Import E-mails.....	50
8.4	Trouble Shooting E-mail Services.....	50
9	Groupware Services	51
9.1	Directory Service / LDAP.....	51
9.1.1	Configuring the openCRX LDAP Port.....	52
9.1.2	LDAP Configuration of Thunderbird.....	52
9.1.3	LDAP Configuration of MS Outlook.....	53
9.2	Calendaring.....	54
9.2.1	Calendar as a Set of Activities.....	54
9.2.2	Calendar Selectors.....	54
9.2.3	openCRX Activities mapped to Calendar Events.....	55
9.2.4	Calendaring / Free Busy.....	57
9.2.4.1	Free Busy Configuration of Thunderbird.....	57
9.2.4.2	Free Busy Configuration of MS Outlook.....	57
9.2.5	Calendaring / iCalendar (ICS).....	58
9.2.5.1	ICS Configuration of Thunderbird/Lightning and Sunbird.....	58
9.2.5.2	ICS Configuration of MS Outlook.....	59
9.2.6	Calendaring / CalDAV.....	59
9.2.6.1	CalDAV Configuration of Thunderbird/Lightning and Sunbird....	60
9.2.6.2	CalDAV Configuration of MS Outlook.....	60
9.2.6.3	CalDAV Configuration of Chandler Desktop.....	61
9.2.7	Calendaring / Timeline.....	62
9.3	Mailstore / IMAP.....	63
9.3.1	Configuring the openCRX IMAP Port.....	64
9.3.2	Configuring the IMAP Maildir Cache.....	64
9.3.2.1	Maildir Configuration with Apache Tomcat.....	64

9.3.2.2	Maildir Configuration with JBoss.....	64
9.3.3	IMAP Configuration of Thunderbird.....	65
9.3.4	IMAP Configuration of MS Outlook.....	66
9.4	Collaboration / Wiki.....	67
10	Data Import/Export.....	68
10.1	Importing Data into openCRX.....	68
10.2	Importing XML Files.....	68
10.2.1	Importing vCard Files (→ openCRX Contacts).....	71
10.2.2	Importing E-Mails.....	71
10.2.3	Other Options.....	71
10.3	Exporting Data from openCRX.....	72
10.3.1	Exporting XML Files.....	72
10.3.2	Exporting openCRX Contacts (→ vCard Files).....	73
10.3.3	Exporting openCRX Contacts (→ Outlook Contacts).....	73
10.3.4	Exporting openCRX Meetings (→ iCalendar Files).....	74
10.3.5	Exporting E-Mails.....	74
10.3.6	Other Options.....	74
11	Customizing openCRX.....	75
11.1	Managing Locales.....	75
11.2	Managing Packages.....	76
11.3	Role-based UI.....	77
11.3.1	Model Permissions.....	77
11.3.2	Custom Layout JSPs.....	77
12	Integration with Office Suites.....	78
12.1	MS Office.....	78
12.1.1	MS Word.....	78
12.1.2	MS Excel.....	78
12.1.3	MS Outlook.....	78
12.2	Open Office.....	78
12.2.1	OpenOffice Writer.....	78
12.2.2	OpenOffice Calc.....	78
12.3	Mozilla Thunderbird / Sunbird.....	78
13	Next Steps.....	79

List of Figures

Figure 1:	Security Realms, Principals and Subjects after Initial Setup.....	10
Figure 2:	Segment Administration.....	11
Figure 3:	Role Drop Down with list of available Segment Login Principals.....	11
Figure 4:	openCRX UML Model – Class Diagram SecureObject.....	12
Figure 5:	System attributes of an openCRX object as shown in the GUI.....	13
Figure 6:	Table OOCKE1_SEGMENT after default installation (QuickStart).....	15

Figure 7: Table OOCKE1_SEGMENT after modification.....	15
Figure 8: Result of Check Permissions.....	17
Figure 9: Role Drop Down with list of available Segment Login Principals.....	18
Figure 10: Operation Actions > Import Login Principals (admin-Root).....	22
Figure 11: Operation Actions > Import Users (admin-Standard).....	23
Figure 12: Disabling of Segment Login Principal guest by admin-Standard....	24
Figure 13: 3-Tier with Apache Tomcat / LWC.....	25
Figure 14: 3-Tier with J2EE-compliant Application Server.....	25
Figure 15: 4-Tier with Clustered Application and DB Servers.....	25
Figure 16: Multiple Data Segments in a single DB.....	26
Figure 17: Dedicated DB for each Entity.....	27
Figure 18: Accessing the openCRX Workflow Controller.....	28
Figure 19: openCRX 2.0 Workflow Controller.....	28
Figure 20: Default Configuration of WorkflowController.....	29
Figure 21: openCRX Administration – WorkflowController.....	30
Figure 22: Workflow Controller Configuration – serverURL.....	31
Figure 23: Workflow Controller Configuration – pingrate and autostart.....	31
Figure 24: Default Workflow Processes created by WorkflowHandler.....	33
Figure 25: Event and Notification Service.....	35
Figure 26: Standard Topics included in the openCRX distribution.....	36
Figure 27: Create a new Subscription.....	37
Figure 28: Create a Subscription with Filters.....	38
Figure 29: Flow of e-mail messages between openCRX, MTA and mail client..	40
Figure 30: Create a new E-Mail Account – step 1.....	45
Figure 31: Create a new E-Mail Account – step 2.....	45
Figure 32: Create a new E-Mail Account – step 3.....	46
Figure 33: E-mail subject prefix and Web access URL.....	46
Figure 34: Example of outbound E-mail Action Log Entries.....	47
Figure 35: Send E-Mail from openCRX – Overview.....	48
Figure 36: Send E-Mail from openCRX with Actions > Follow Up.....	48
Figure 37: Send E-Mail as Attachment from openCRX – Overview.....	49
Figure 38: Export E-Mail from openCRX with Actions > Follow Up.....	49
Figure 39: Thunderbird LDAP Configuration.....	52
Figure 40: MS Outlook LDAP Configuration.....	53
Figure 41: An openCRX activity's iCal representation.....	55
Figure 42: An openCRX activity in the standard GUI.....	56
Figure 43: An openCRX activity filtered to a user's homepage.....	56
Figure 44: Timeline visualizes time-based events.....	62
Figure 45: Thunderbird IMAP Configuration.....	65
Figure 46: Thunderbird IMAP Configuration.....	66
Figure 47: XML import from 3rd party system – overview.....	69

Figure 48: Interactive import of XML Files.....	69
Figure 49: Interactive import of XML Files.....	70
Figure 50: Operation vCard Import.....	71
Figure 51: Exporting SalesOrder as XML File.....	72
Figure 52: XML Exporter provides XML data file and code tables as ZIP file....	72
Figure 53: Export Contact as vCard.....	73
Figure 54: Export Contact to MS Outlook.....	73
Figure 55: Exporting Meeting / Sales Visit as iCalendar File.....	74

List of Listings

Listing 1: File Format Subjects and Application Login Principals.....	22
Listing 2: Example File Subjects and Application Login Principals.....	22
Listing 3: File Format Users.....	23
Listing 4: Example File Users.....	23
Listing 5: web.xml – auto startup of the Workflow Controller.....	30
Listing 6: Servlets managed by Workflow Controller log to server.log.....	34
Listing 7: File opencrx-core-CRX.xml.....	42
Listing 8: Uncomment mail resource definition in web.xml.....	42
Listing 9: Importing certificate into keystore cacerts.....	42
Listing 10: File mail-service.xml.....	43
Listing 11: Uncomment mail resource definition in web.xml.....	44
Listing 12: Uncomment mail resource definition in jboss-web.xml.....	44
Listing 13: Importing certificate into keystore cacerts.....	44
Listing 14: Importing Certificate.....	50
Listing 15: Set org.opencrx.maildir for Apache Tomcat.....	64
Listing 16: Set org.opencrx.maildir for Apache Tomcat.....	64
Listing 17: Locales in web.xml.....	75
Listing 18: Activating/Deactivating Locales in web.xml.....	75
Listing 19: Locales in web.xml.....	76
Listing 20: Activating/Deactivating Locales in web.xml.....	76

1 About this Book

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

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

1.1 Who this book is for

The intended audience are openCRX administrators.

1.2 What do you need to understand this book

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

1.3 Tips, Warnings, etc.

We make use the following pictograms:



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

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



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



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

2 Prerequisites

This guide assumes that you have access to a properly installed instance of openCRX 2.0 (for installation instructions, please refer to the guides available from <http://www.opencrx.org/documents.htm>).

3 Security

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



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

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

3.1 Introduction

3.1.1 Basic Concepts and Conventions

- Each “real user” is represented by a **Subject** (e.g. “Guest”). Subjects are managed by the openCRX Root administrator (**admin-Root**).
- Each subject has an **Application Login Principal** (also called **login id**). Application login principals are managed by the openCRX Root administrator (**admin-Root**).
- Each application login principal is assigned to a subject (e.g. principal “guest” is assigned to subject “Guest”) and allows a “real user” to login.
- A “real user” can have one or more additional segment login principals. 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 worked through the QuickStart guide):

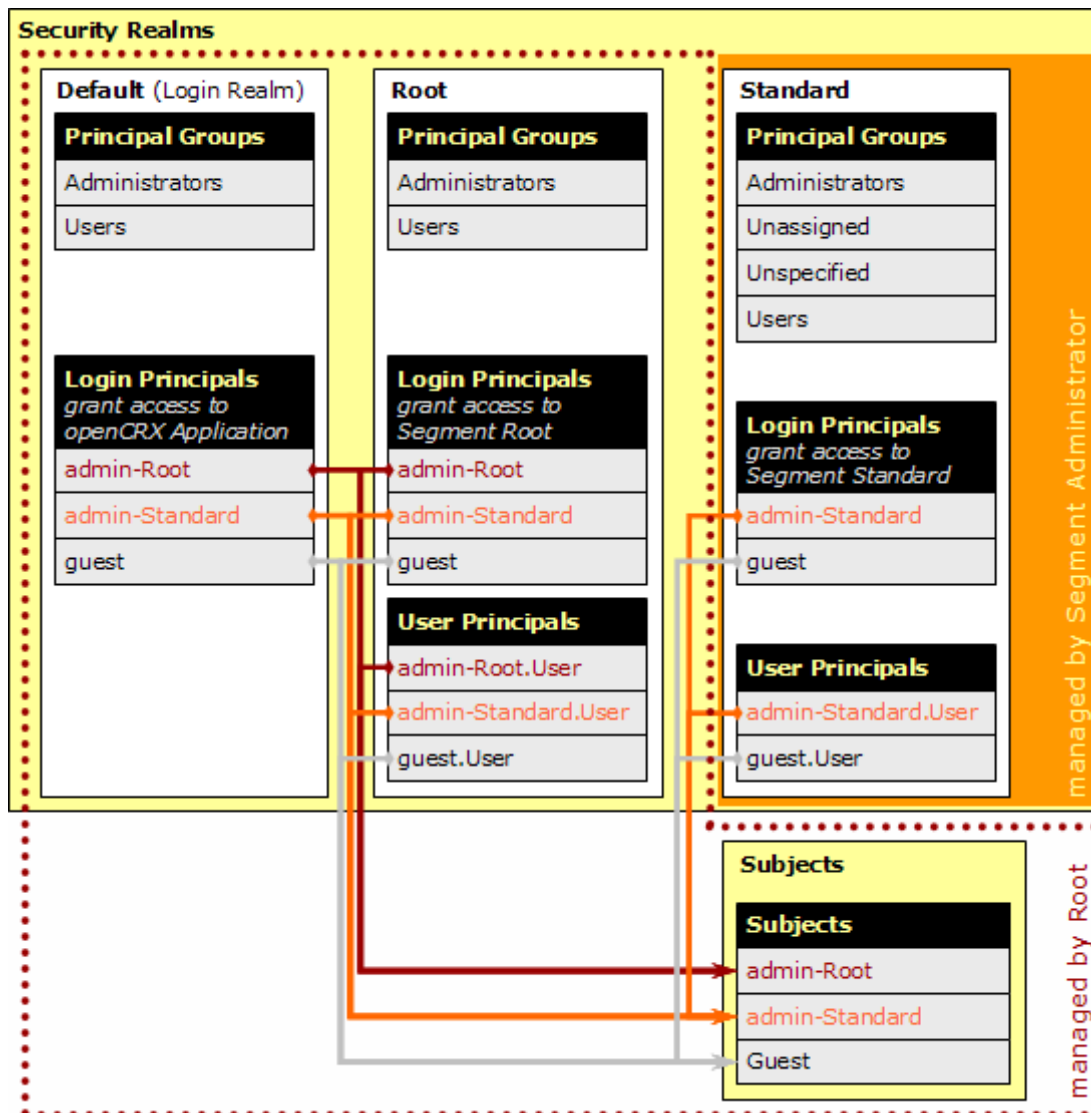


Figure 1: Security Realms, Principals and Subjects after Initial Setup

Summarizing the above:

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

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

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

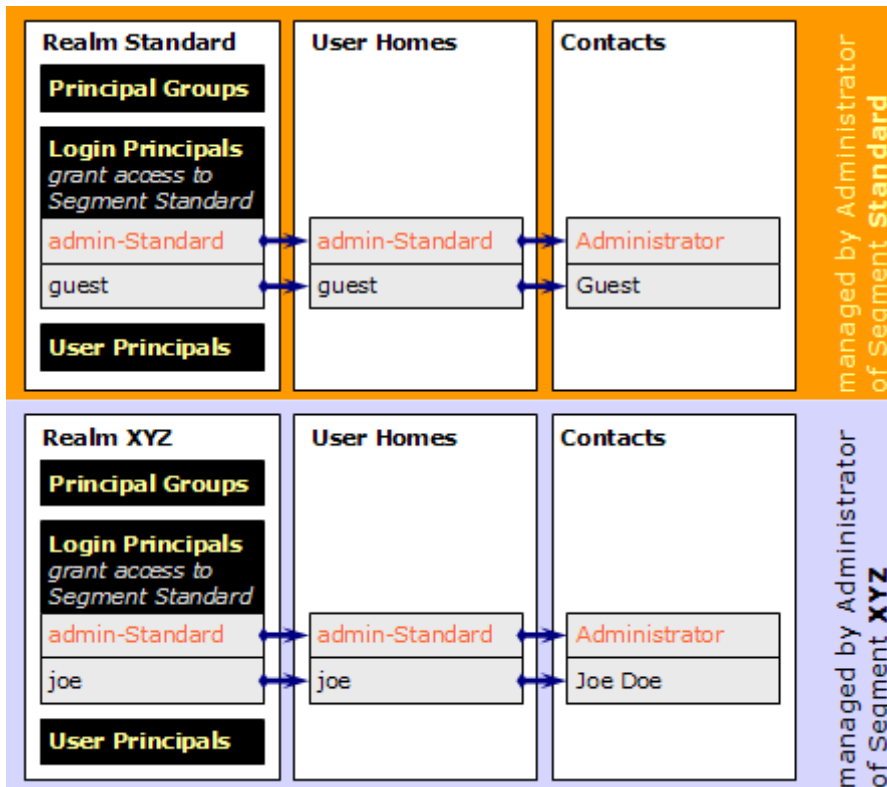


Figure 2: Segment Administration

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

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

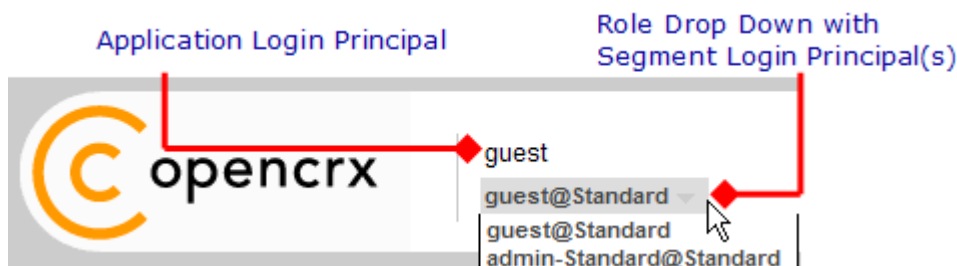


Figure 3: Role Drop Down with list of available Segment Login Principals

3.1.2 Permissions / Access Control

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

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

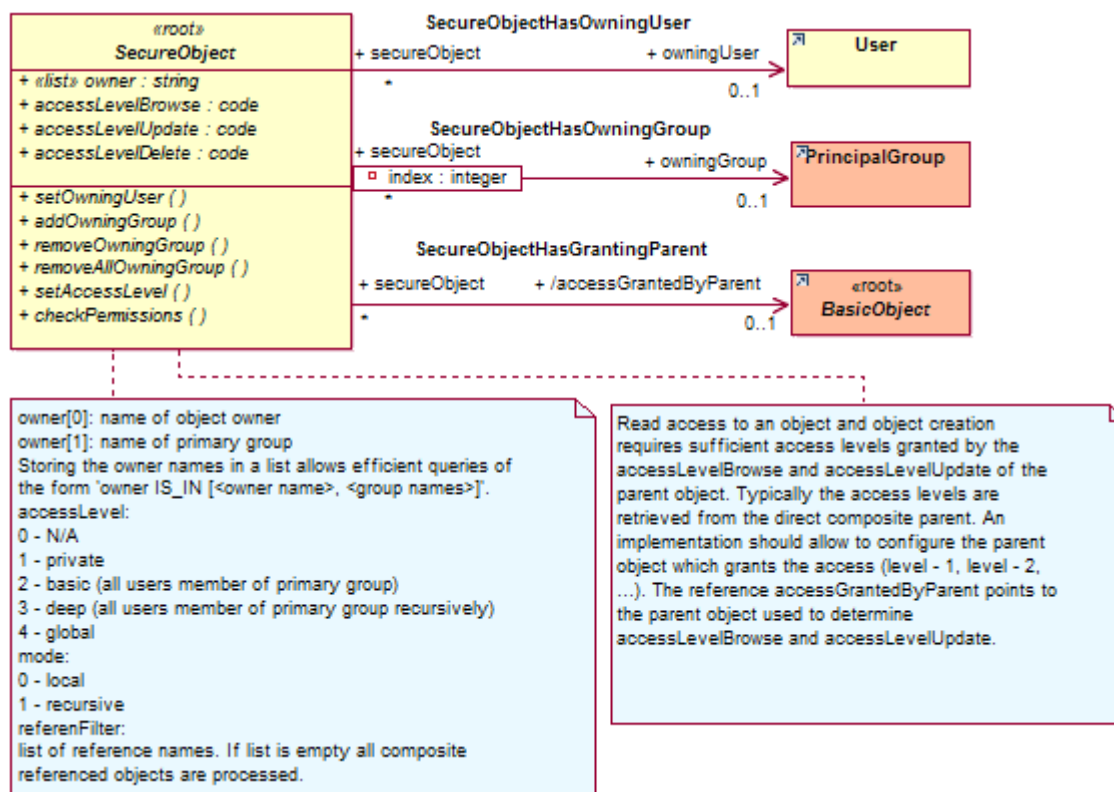


Figure 4: openCRX UML Model – Class Diagram SecureObject



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



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

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

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



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

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

General | More Charts | Options | System ▾

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

Object

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

Security

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

Alerts | Subscriptions | E-Mail | > ^

+

Owning Groups

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

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

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

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

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

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

3.2 Default Settings

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

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

OBJECT_ID	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. browse all accounts, browse all activities, browse all documents, etc.).

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

OBJECT_ID	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.

3.3 Activating Security

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

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



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

3.4 Security Settings of New Objects

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

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



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



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



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

3.5 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:

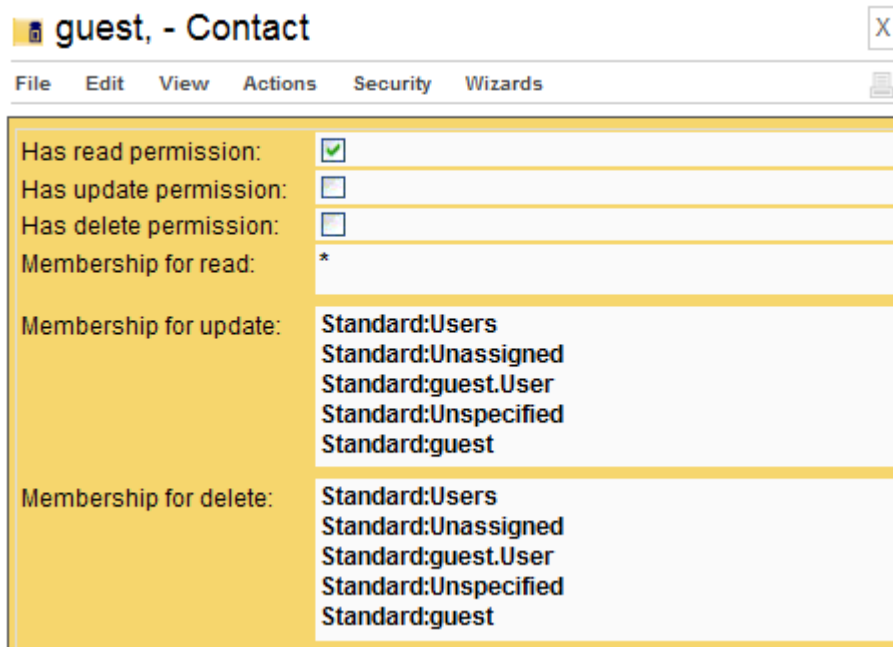


Figure 8: Result of Check Permissions

The meaning of the above result is as follows:

- | | |
|------------------------|---|
| Has read permission: | <input checked="" type="checkbox"/> principal can browse this object
<input type="checkbox"/> principal cannot browse this object |
| Has update permission: | <input checked="" type="checkbox"/> principal can modify/update this object
<input type="checkbox"/> principal cannot modify/update this object |
| Has delete permission: | <input checked="" type="checkbox"/> principal can delete this object
<input type="checkbox"/> 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 Tomcat / Application Server Login

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

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

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

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:

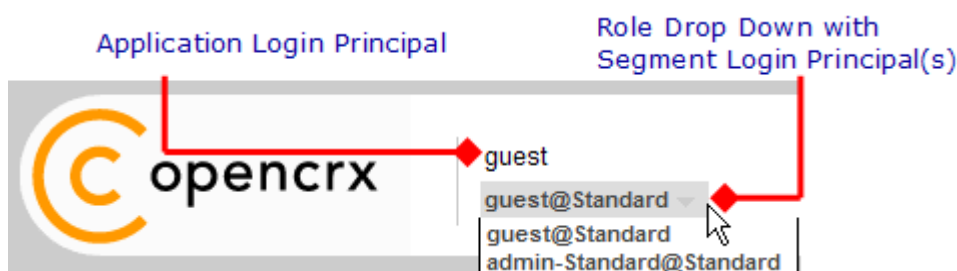


Figure 9: Role Drop Down with list of available Segment Login Principals

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!

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

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 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:

Is Active	Creation	Notify on Replacement

Item	Is Active
Accounts:	<input checked="" type="checkbox"/>
Activities:	<input checked="" type="checkbox"/>
Pipeline:	<input checked="" type="checkbox"/>
Products:	<input checked="" type="checkbox"/>

- 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.

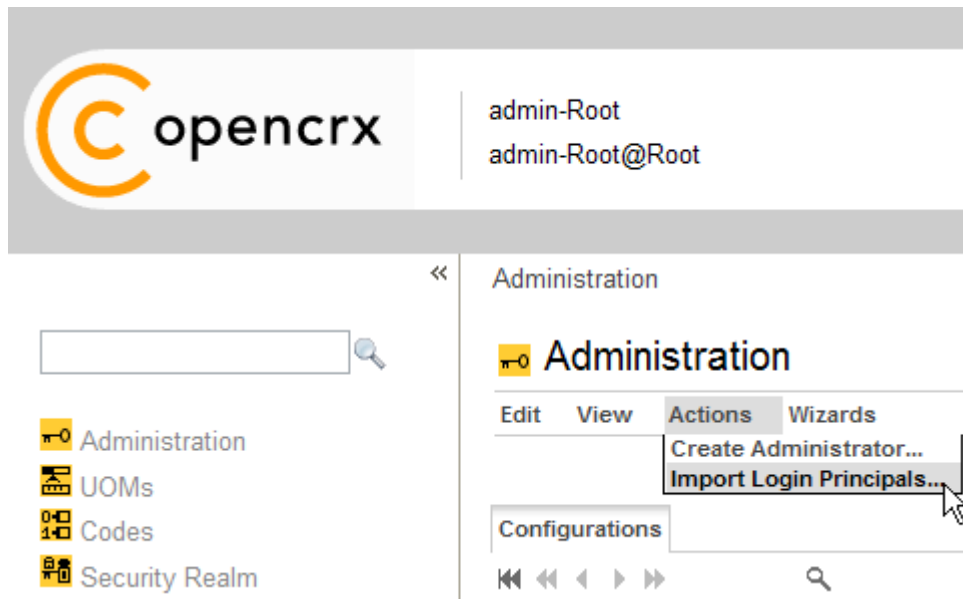


Figure 10: Operation Actions > Import Login Principals (admin-Root)

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.

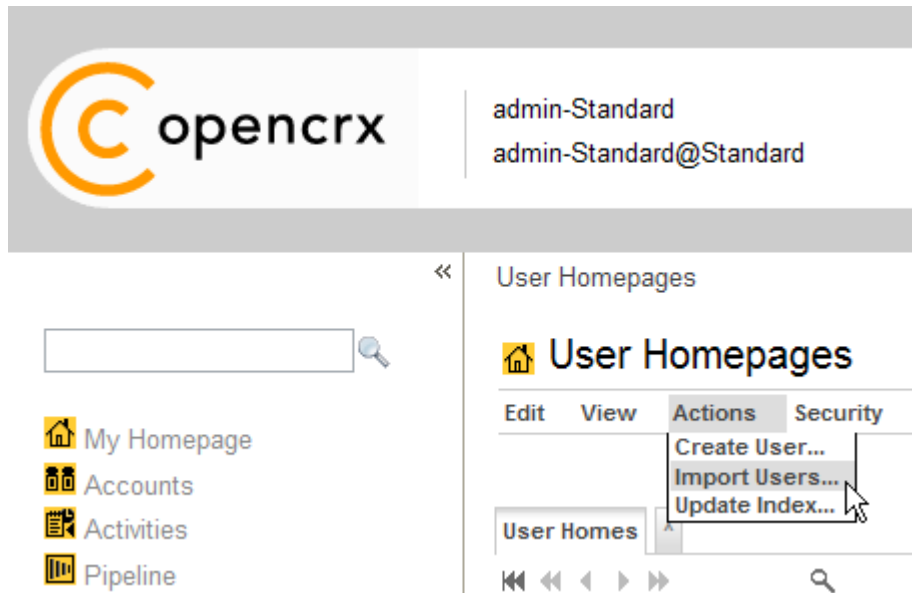


Figure 11: Operation Actions > Import Users (admin-Standard)

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 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.

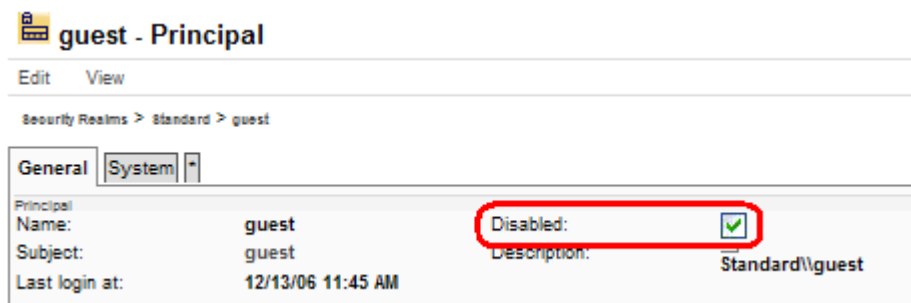


Figure 12: Disabling of Segment Login Principal **guest** by **admin-Standard**



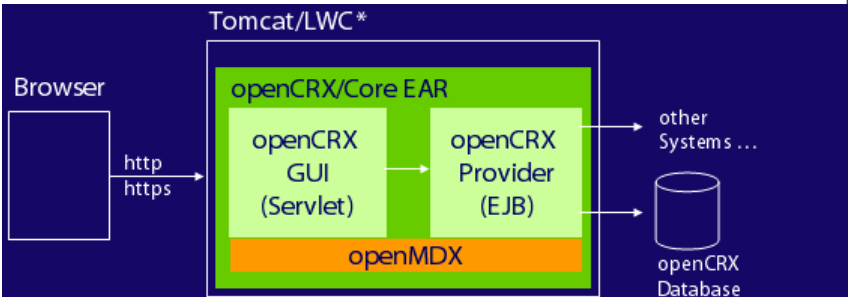
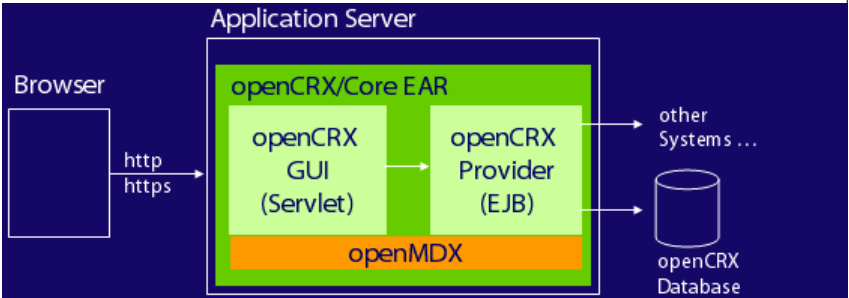
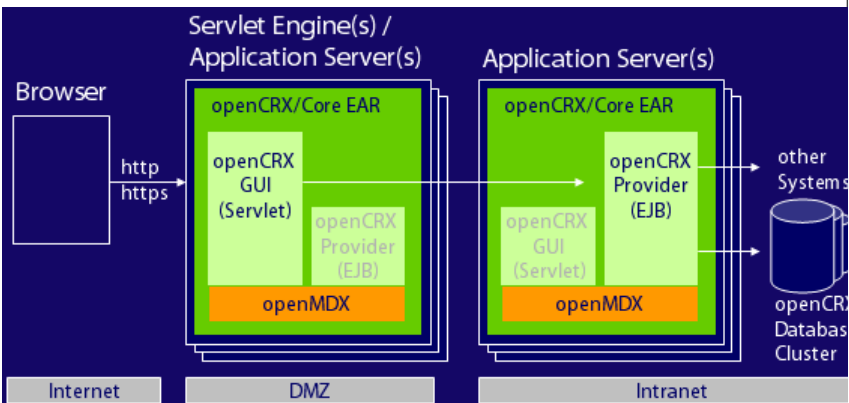
You should **not** delete a particular Subject as long as it is referenced by any Principal. Otherwise you'll end up with "dangling" Subject references.

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:

<p>3-Tier with Tomcat/LWC</p> <ul style="list-style-type: none"> ▪ 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), but highest performing 3-Tier deployment with full transaction service 	 <p><i>Figure 13: 3-Tier with Apache Tomcat / LWC</i></p>
<p>3-Tier with AppServer</p> <ul style="list-style-type: none"> ▪ 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 	 <p><i>Figure 14: 3-Tier with J2EE-compliant Application Server</i></p>
<p>4-Tier with Clusters</p> <ul style="list-style-type: none"> ▪ 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 	 <p><i>Figure 15: 4-Tier with Clustered Application and DB Servers</i></p>

5.2 Multi Entity Deployment Scenarios

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

5.2.1 Multiple Data Segments in a single DB

The setup "Multiple Data Segments in a single DB" provides adequate security for many use cases and is relatively easy to manage. As all the data is stored in a single database, however, security configuration mistakes (e.g. principals linked to the wrong subject, etc.) 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 root cause here, not a malfunction of openCRX). Furthermore, this setup is not recommended if users can get direct access to the database, e.g. with third party reporting tools as those tools typically bypass the openCRX API.

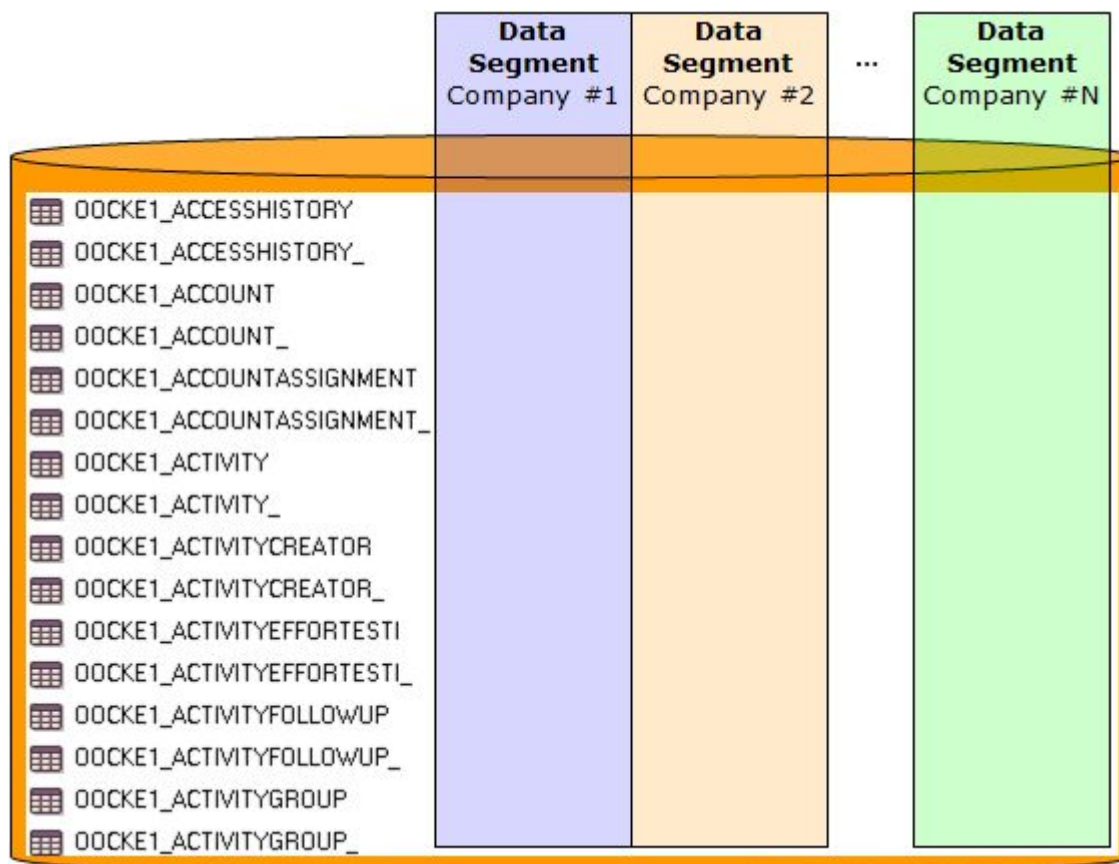


Figure 16: Multiple Data Segments in a single DB

5.2.2 Multiple DBs

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

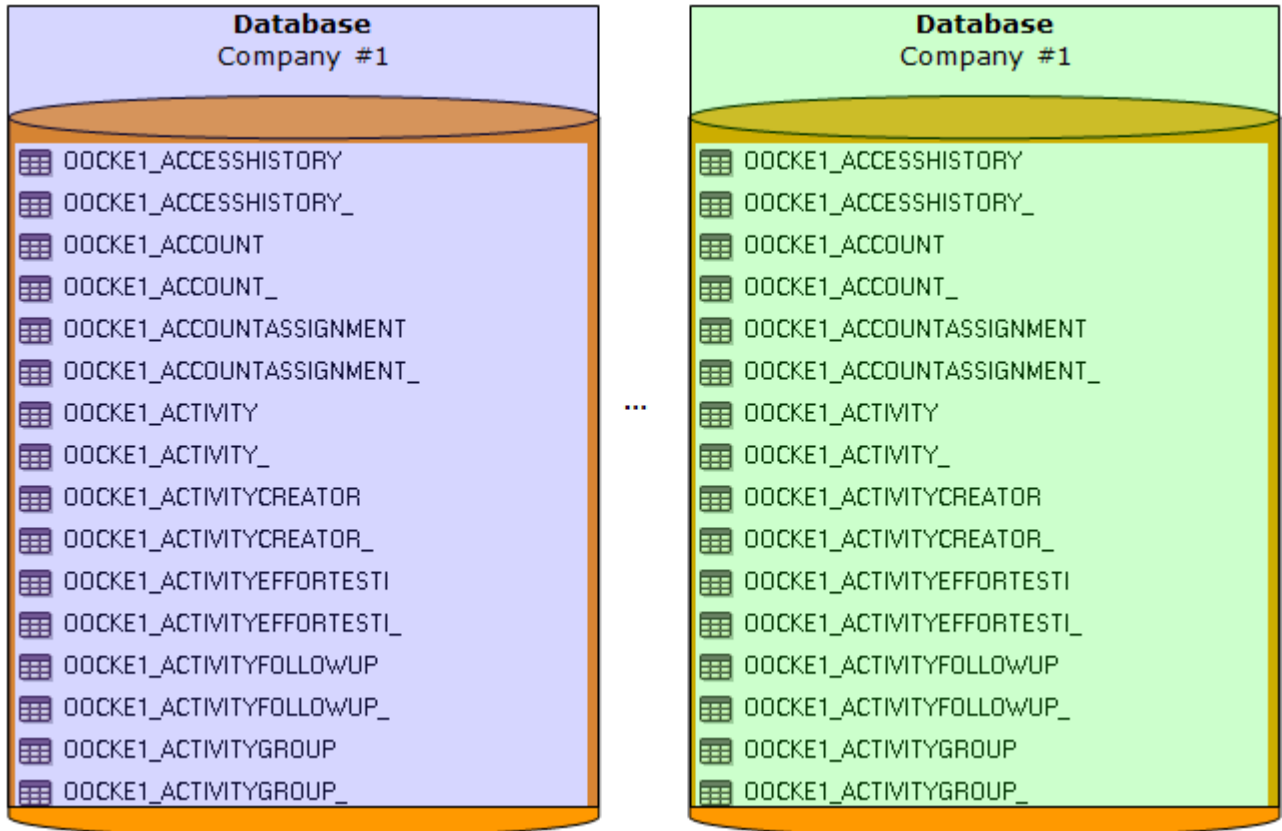


Figure 17: Dedicated DB for each Entity

5.3 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:

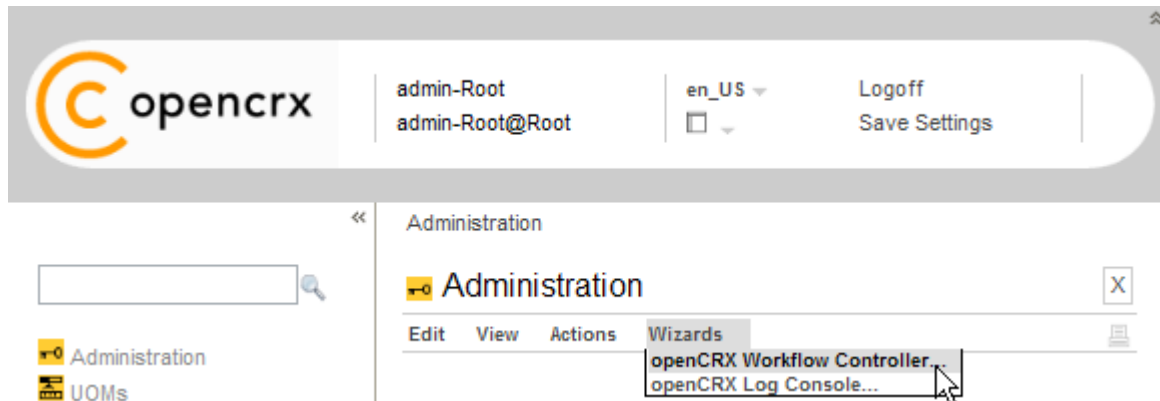


Figure 18: Accessing the openCRX Workflow Controller



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 2.0:

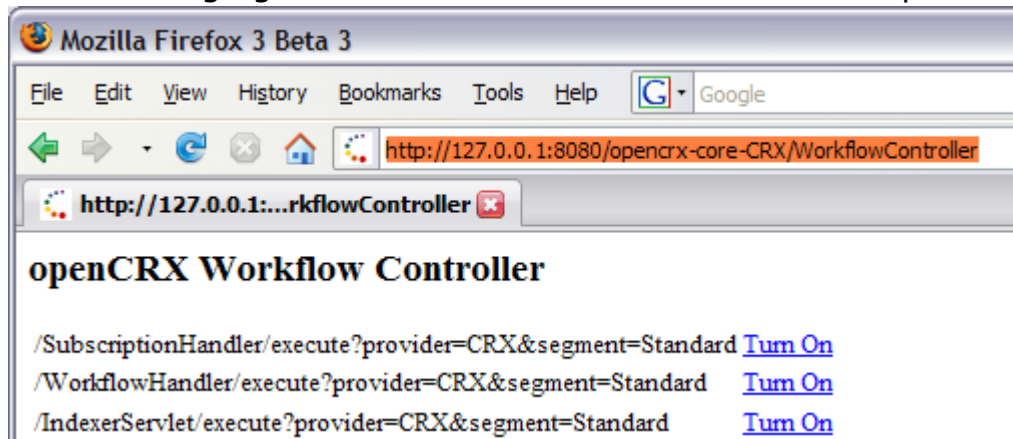


Figure 19: openCRX 2.0 Workflow Controller



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



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

Name	Description	Domain	Date value	String value
serverURL	Server URL			http://127.0.0.1/openctx-core
SubscriptionHandler.CRX.Standard.autostart	SubscriptionHandler autostart			false
SubscriptionHandler.CRX.Standard.pingrate	SubscriptionHandler pingrate			2
IndexerServlet.CRX.Standard.autostart	IndexerServlet autostart			false
IndexerServlet.CRX.Standard.pingrate	IndexerServlet pingrate			2
WorkflowHandler.CRX.Standard.autostart	WorkflowHandler autostart			false
WorkflowHandler.CRX.Standard.pingrate	WorkflowHandler pingrate			2

Figure 20: Default Configuration of WorkflowController

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

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

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

6.1 Workflow Controller Configuration

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

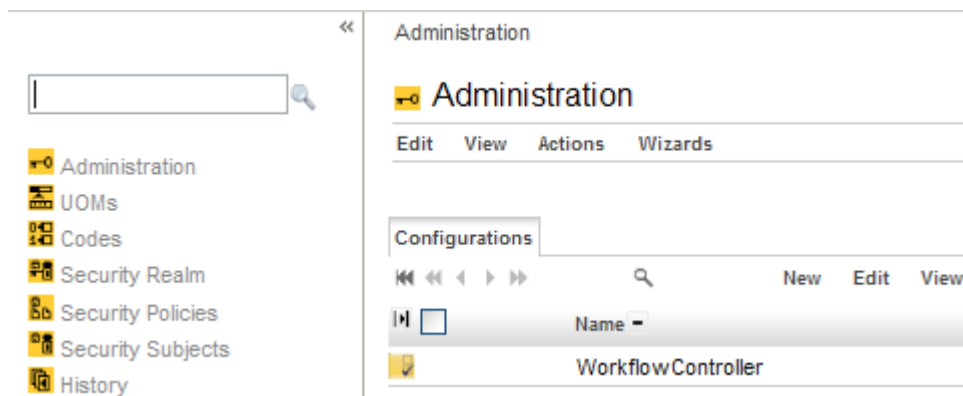


Figure 21: openCRX Administration – WorkflowController

6.1.1 Startup Configuration in web.xml

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

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

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

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

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



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

6.1.2 ServerURL

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

The screenshot shows the 'WorkflowController - Component Configuration' window. The 'General' tab is active, and the 'System' sub-tab is selected. The 'Name' field is 'WorkflowController'. The 'Properties' section is expanded, showing a table of properties. The 'serverURL' property is highlighted with a red box, with its value set to 'http://127.0.0.1:8080/opencrx-core-CRX'.

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

Figure 22: Workflow Controller Configuration – serverURL

6.1.3 Handler pingrate and autostart

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

The screenshot shows the 'WorkflowController - Component Configuration' window. The 'General' tab is active, and the 'System' sub-tab is selected. The 'Name' field is 'WorkflowController'. The 'Properties' section is expanded, showing a table of properties. The 'SubscriptionHandler.CRX.Standard.pingrate' and 'SubscriptionHandler.CRX.Standard.autostart' properties are highlighted with a red box, with values '1' and 'true' respectively.

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

Figure 23: Workflow Controller Configuration – pingrate and autostart

6.2 Servlet 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.



As indexing can put some heavy load on your database server you might consider turning off (or at least lowering the frequency of calling) the IndexerServlet during busy hours.

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:

Workflows

Edit View Actions Security

Workflow Processes Topics ^

⏪ ⏩ ⏴ ⏵ 🔍 New Edit View

<input type="checkbox"/>	Name ▲	Priority -	Is synchronous -
	org.opencrx.kernel.workflow.PrintConsole	0	<input checked="" type="checkbox"/>
	org.opencrx.kernel.workflow.SendAlert	0	<input checked="" type="checkbox"/>
	org.opencrx.mail.workflow.ExportMailWorkflow	0	<input type="checkbox"/>
	org.opencrx.mail.workflow.SendMailNotificationWorkflow	0	<input type="checkbox"/>
	org.opencrx.mail.workflow.SendMailWorkflow	0	<input type="checkbox"/>

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). 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 `NullPointerException`, 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 2.0 Workflow Controller](#)).



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

7 Subscribe / Notify Services

openCRX features a powerful event subscription and notification service:

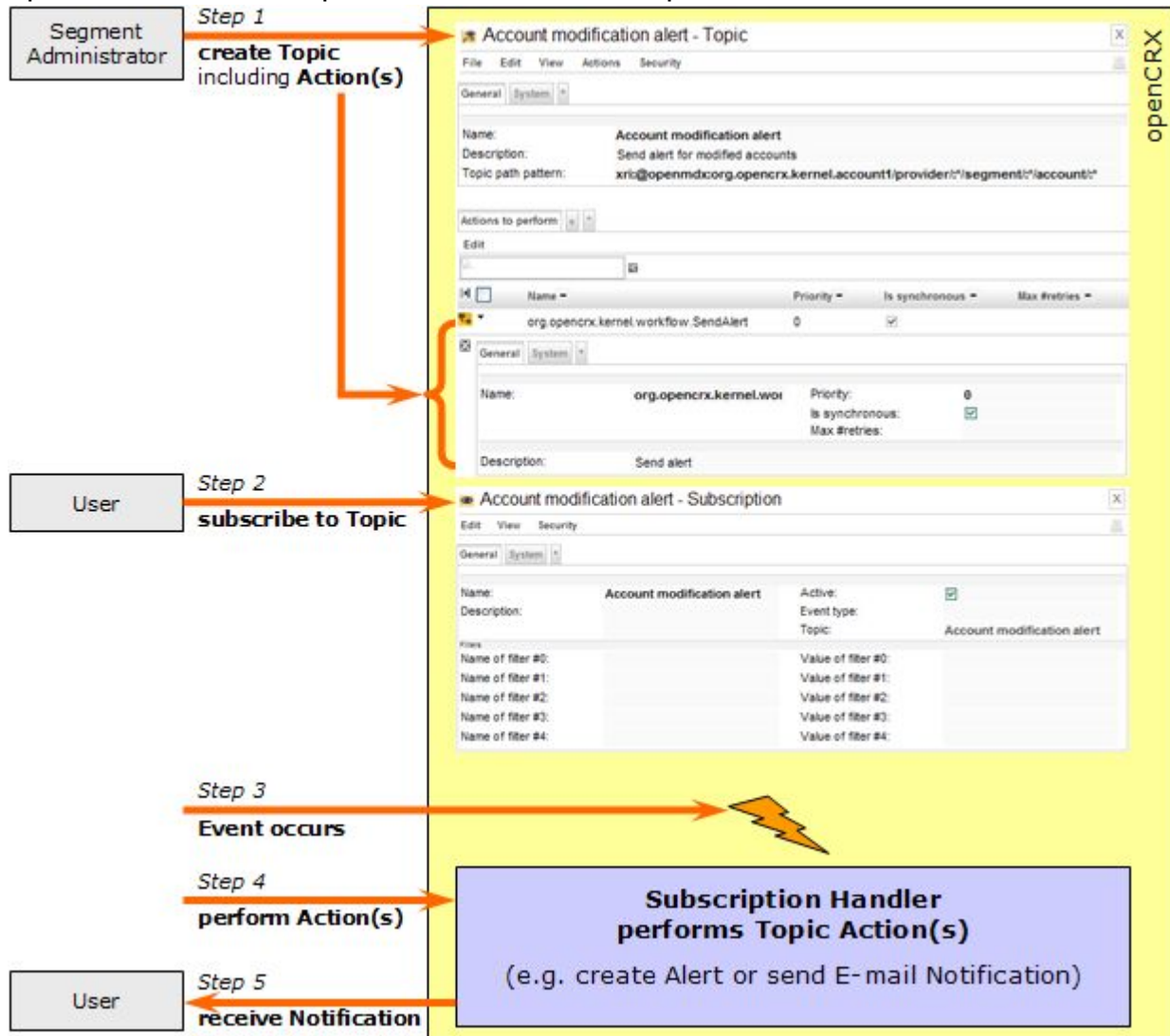


Figure 25: Event and Notification Service

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 and E-mail Notifications.



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

The openCRX distribution includes quite a few default topics (see *Figure 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 are created by the Subscription Handler of the respective segment when it is started for the first time.

Workflow Processes		Topics
		^
		⏪ ⏩ 🔍 New Edit View
<input type="checkbox"/> Name ▲	Actions to perform -	
	Account modification alert	org.opencrx.kernel.workflow.SendAlert
	Activity follow-up modification alert	org.opencrx.kernel.workflow.SendAlert
	Activity modification alert	org.opencrx.kernel.workflow.SendAlert
	Booking modification alert	org.opencrx.kernel.workflow.SendAlert
	Competitor modification alert	org.opencrx.kernel.workflow.SendAlert
	Compound booking modification alert	org.opencrx.kernel.workflow.SendAlert
	Invoice modification alert	org.opencrx.kernel.workflow.SendAlert
	Lead modification alert	org.opencrx.kernel.workflow.SendAlert
	Mail notification for new alerts	org.opencrx.mail.workflow.SendMailNotificationWorkflow
	Opportunity modification alert	org.opencrx.kernel.workflow.SendAlert
	Organization modification alert	org.opencrx.kernel.workflow.SendAlert
	Product modification alert	org.opencrx.kernel.workflow.SendAlert
	Quote modification alert	org.opencrx.kernel.workflow.SendAlert
	Sales order modification alert	org.opencrx.kernel.workflow.SendAlert

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:

Activities

Private activity tracker:

Resource:

Subscriptions

	Is Active	Notify on		
		Creation	Replacement	Removal
Account modification alert:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Activity follow-up modification alert:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Activity modification alert:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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, some users want to receive notifications related to activities of ProjectY only, and some users want to receive notifications from both projects. Such a situation could be handled as follows:

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

Activity modification alert - Subscription

General		System	
Name:	Activity modification alert	Active:	<input checked="" type="checkbox"/>
Description:	ProjectX only	Event type:	
Filters		Topic:	Activity modification alert
Name of filter #0:	owner	Value of filter #0:	Standard:DivisionA.ProjectX Standard:DivisionA.ProjectB

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.	<ul style="list-style-type: none"> ▪ verify that the Subscription Handler of the respective segment was started at least once (Workflow Processes and Topics are created automatically by the Subscription Handler if they are not present) ▪ click on the filter button <input checked="" type="checkbox"/> to see all rows without filtering (maybe you defined a default filter in the past?)
I started the Subscription Handler but I never receive any Alerts / Notifications	<ul style="list-style-type: none"> ▪ verify that you started the correct Subscription Handler (each segment has its own Subscription Handler) ▪ in case you upgraded to a new version of openCRX and forgot to delete Workflows and Topics provided by openCRX, stop the Subscription Handler, delete Workflow Processes and Topics, and then start the Subscription Handler again ▪ check the openCRX log files to find out whether bad/corrupt data might be causing problems (e.g. NullPointerException during Workflow execution)
I receive Alerts triggered by my Subscriptions but no Notification E-mails	<ul style="list-style-type: none"> ▪ verify that JavaMail is properly installed and the mail service properly configured (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 2.0:

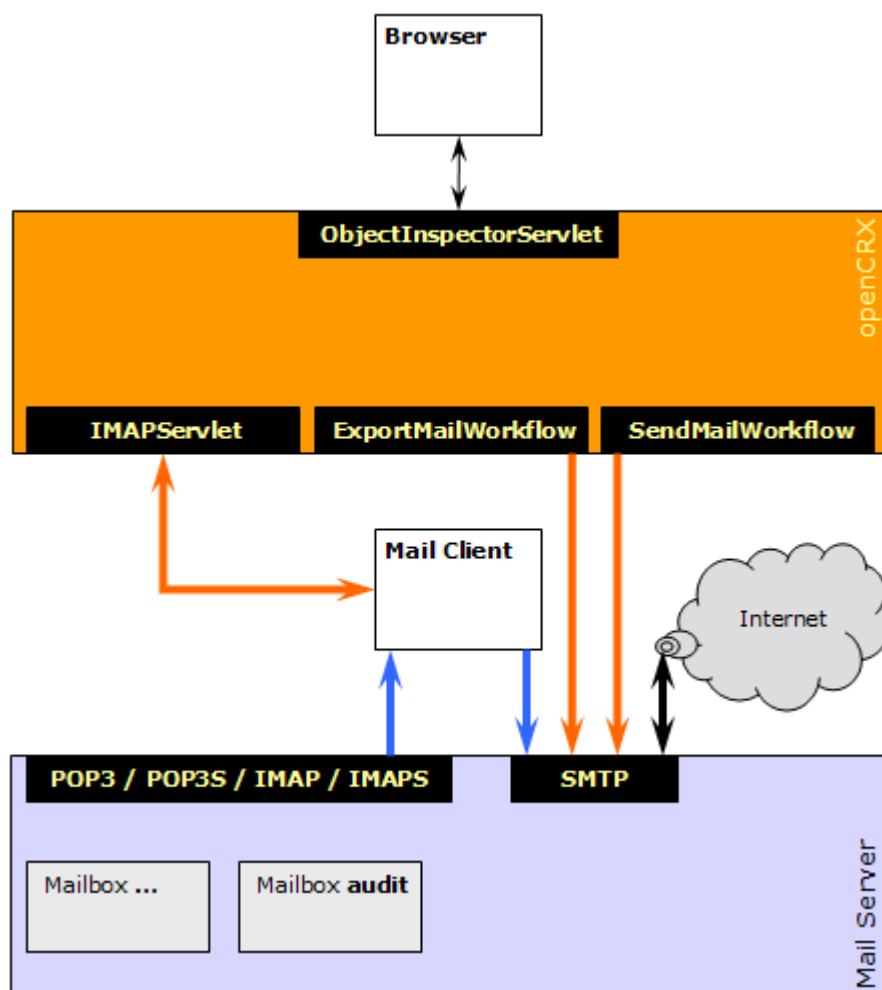


Figure 29: Flow of e-mail messages between openCRX, MTA and mail client

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>
    </attribute>
    <depends>jboss:service=Naming</depends>
  </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 resource 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:

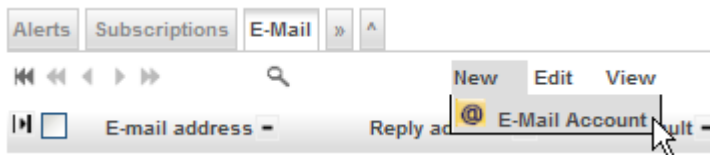


Figure 30: Create a new E-Mail Account – step 1

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

 A screenshot of the 'Untitled - E-Mail Account' configuration form. The form has a 'General' tab and a 'System' tab. The 'E-mail address' field contains 'user@mycompany.com' and has a 'Default' checkbox checked. The 'Reply address' field contains 'noreply@mycompany.com'. There are empty fields for 'Outgoing Mail Service' and 'Incoming Mail Service'. Below these fields is a 'Qualifier' field with the value 'C-LmEI36Edu7vjts7r7Nrw'. At the bottom, there are 'Save' and 'Cancel' buttons.

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/<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/<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/<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:

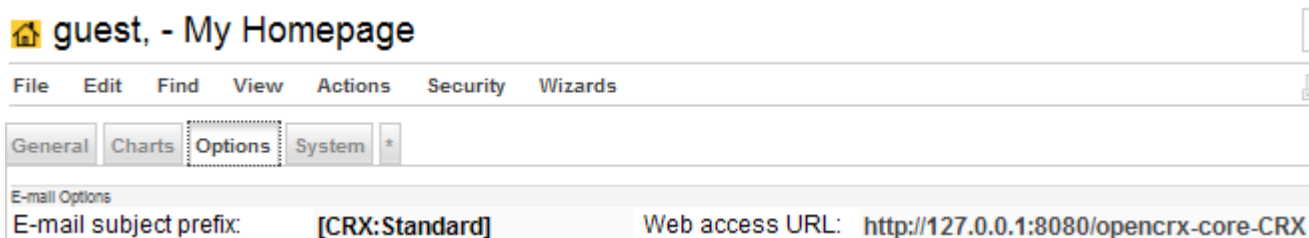


Figure 33: E-mail subject prefix and Web access URL

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).

Action Log Entries		
Properties ^		
New ▶		
<input type="text"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		
Name	Correlation	Description
openCRX SendMail: test 2		Object Invoked: http://localhost/opencrx-core-CRX/ObjectInspector By Workflow: http://localhost/opencrx-core-CRX/ObjectInspector Triggered By Event: OBJECT_CREATION Triggered By Subscription: mxxecMQBEdqQW4VT5Pz34w http://localhost/opencrx-core-CRX/ObjectInspector
Can not send mail: MessagingException		Can't send command to SMTP host; nested exception is: java.net.SocketException: Software caused connect
Can not send mail: MessagingException		Could not connect to SMTP host: mail.crixp.com, p

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

8.2.2 Export E-mails

Please refer to chapter 9.3 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:

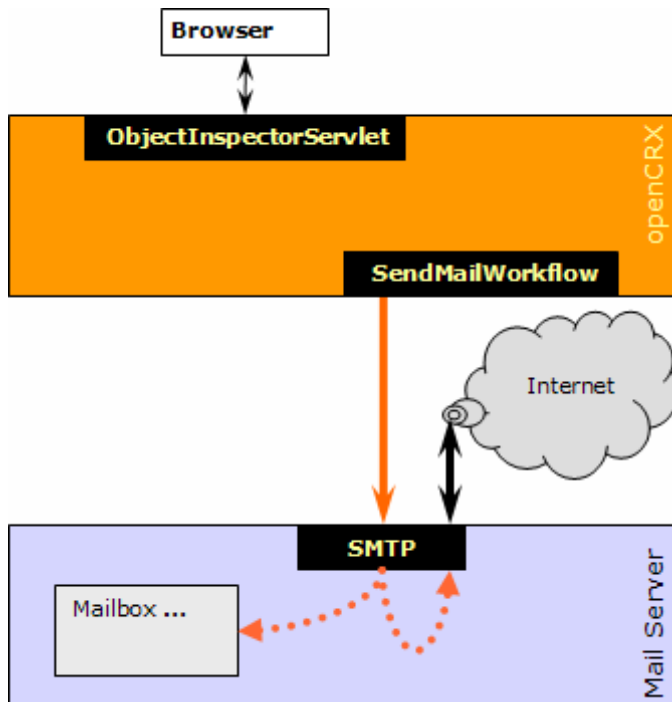


Figure 35: Send E-Mail from openCRX – Overview

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:

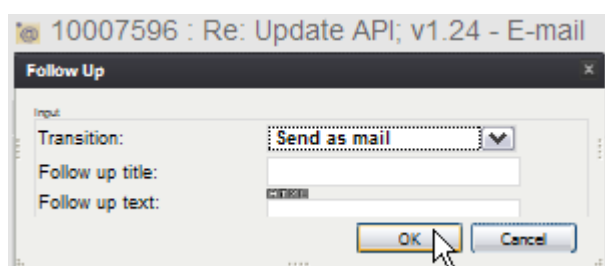


Figure 36: Send E-Mail from openCRX with Actions > Follow Up



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:

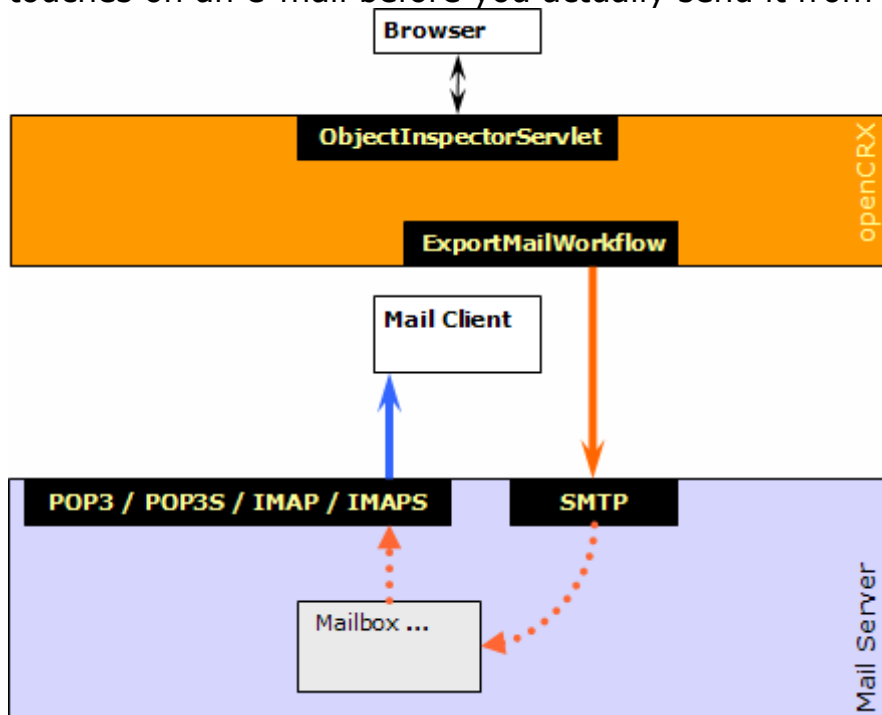


Figure 37: Send E-Mail as Attachment from openCRX – Overview

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**:

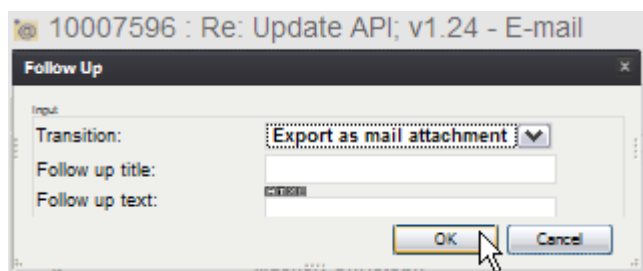


Figure 38: Export E-Mail from openCRX with Actions > Follow Up



Important

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.



Tip

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

8.3 Inbound E-mail


Please refer to chapter 9.3 Mailstore / IMAP.

8.3.1 Import E-mails

Please refer to chapter 9.3 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	<p>It seems that JavaMail sometimes does not (even try to) establish a TLS session when connecting to a mail server (smtp) if the certificate of the mail server has not been imported into the keystore. If the mail server requires TLS for authentication (e.g. SASL) and authentication is required to relay messages such failure to establish a TLS session will prevent openCRX from properly sending outbound mail.</p> <div data-bbox="561 1070 703 1256" style="border: 1px solid black; padding: 5px;">  <p>Tip</p> </div> <p>If you intend to use TLS/SSL to secure the connection to the outbound e-mail server (smtp) we recommend you import the mail server certificate into the keystore.</p> <p><i>Listing 14: Importing Certificate</i></p> <pre data-bbox="561 1323 1479 1373">cd \$JAVA_HOME/lib/security keytool -import -alias <dom> -file <name>.cer -keystore cacerts</pre> <p>Replace <dom> with the domain of the mail server (e.g. mail.company.com) and <name> with the name of the certificate file.</p>
I receive Alerts triggered by my Subscriptions but no Notification E-mails	<ul style="list-style-type: none"> ▪ verify that JavaMail is properly installed and the mail service properly configured (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
Calendaring	FreeBusy	openCRX native
	iCalendar (ICS)	openCRX native
	CalDAV	Bedework
Mailstore	IMAP	openCRX native
Collaboration	Wiki	XWiki



Please note that the above 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 is saying 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 <u>Examples:</u> localhost, 127.0.0.1, myCrxServer.myCompany.com, etc.
Port	1389 (note that the LDAP standard port is 389)
BaseDN	dc=accounts,dc=<Segment>,dc=<Provider>,dc=opencrx,dc=org <u>Example:</u> dc=accounts,dc=Standard,dc=CRX,dc=opencrx,dc=org
BindDN	dc=<Username>,dc=users,dc=<Provider>,dc=opencrx,dc=org <u>Example:</u> dc=guest,dc=users,dc=CRX,dc=opencrx,dc=org



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**

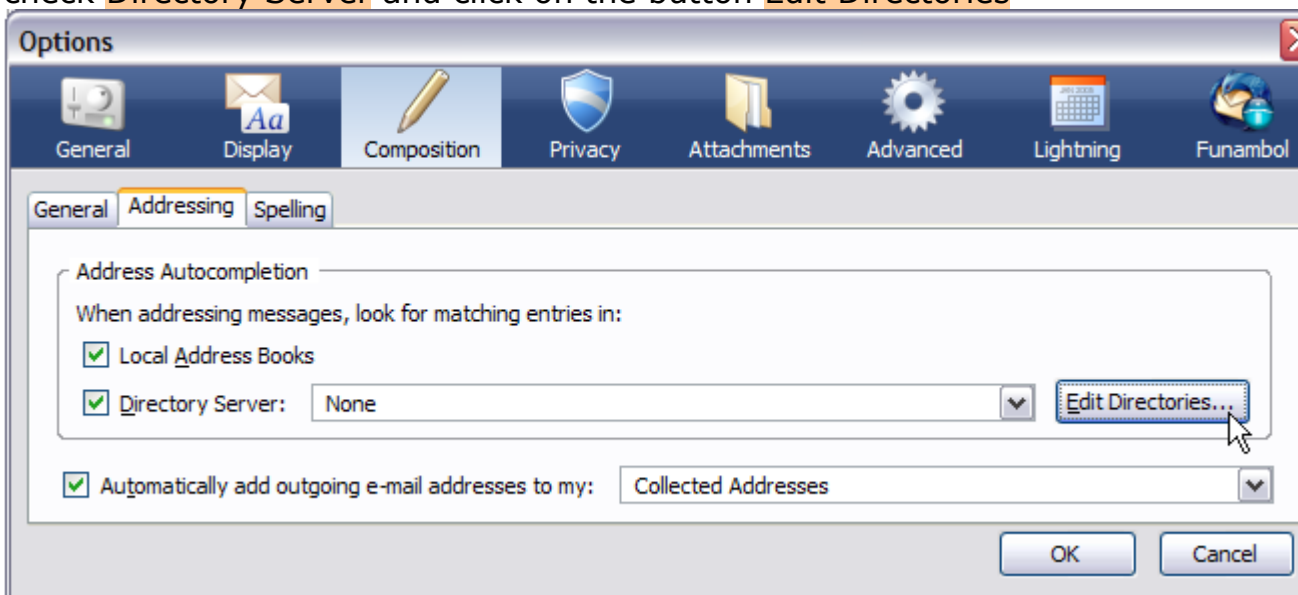


Figure 39: Thunderbird LDAP Configuration

- 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=openctx,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=openctx,dc=org
--------	--

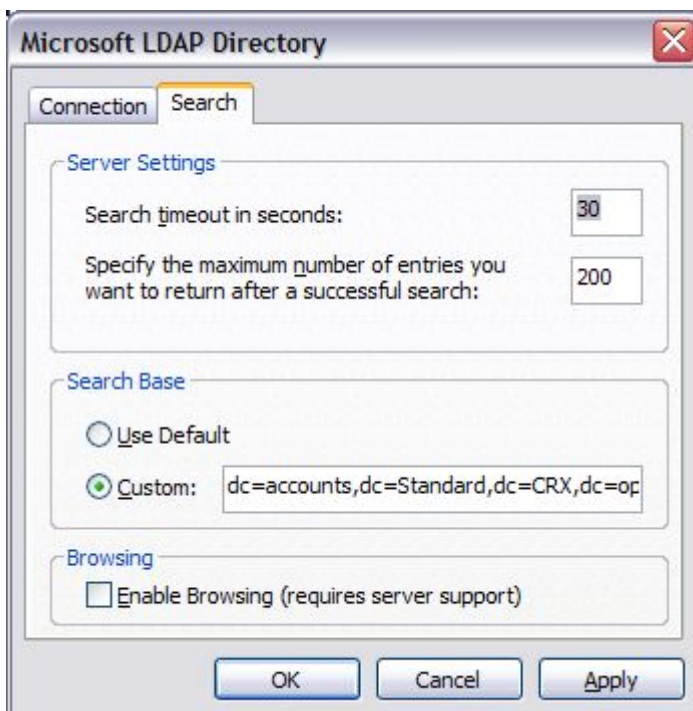


Figure 40: MS Outlook LDAP Configuration

- click **OK**, **Next**, **Finish**, and **Close** to conclude the configuration

9.2 Calendaring

9.2.1 Calendar as a Set of Activities

openCRX offers a multitude of mechanisms to structure, filter, and group activities:

- activities can be assigned to activity groups, i.e. grouped 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)

9.2.2 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. CalDAV calendar, ICS calendar, Free Busy calendar, etc. Some typical calendar selectors are listed below:

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

Selector Examples:

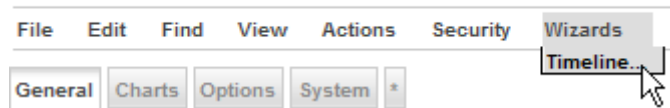
CRX/Standard/tracker/shared
 CRX/Standard/category/presales
 CRX/Standard/milestone/2.0
 CRX/Standard/tracker/main/filter/open
 CRX/Standard/userhome/guest
 CRX/Standard/resource/jeff
 CRX/Standard/globalfilter/last18months

For the purpose of this guide we will mostly work with custom tracker called "shared".



CrxObjects with sets of activities typically feature the wizard Timeline:

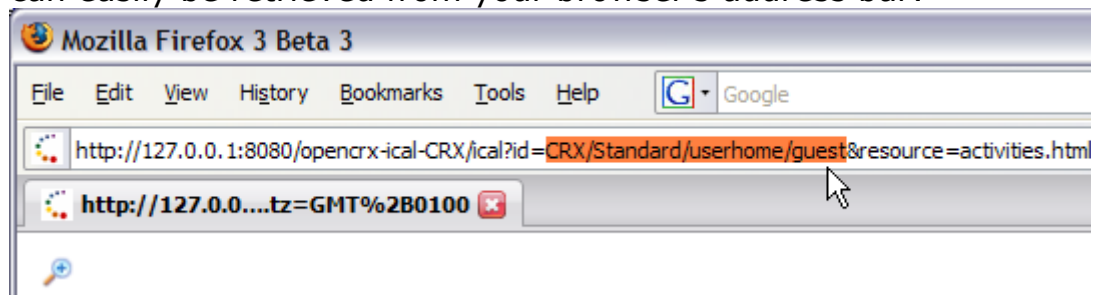
guest, - My Homepage



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:



9.2.3 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:

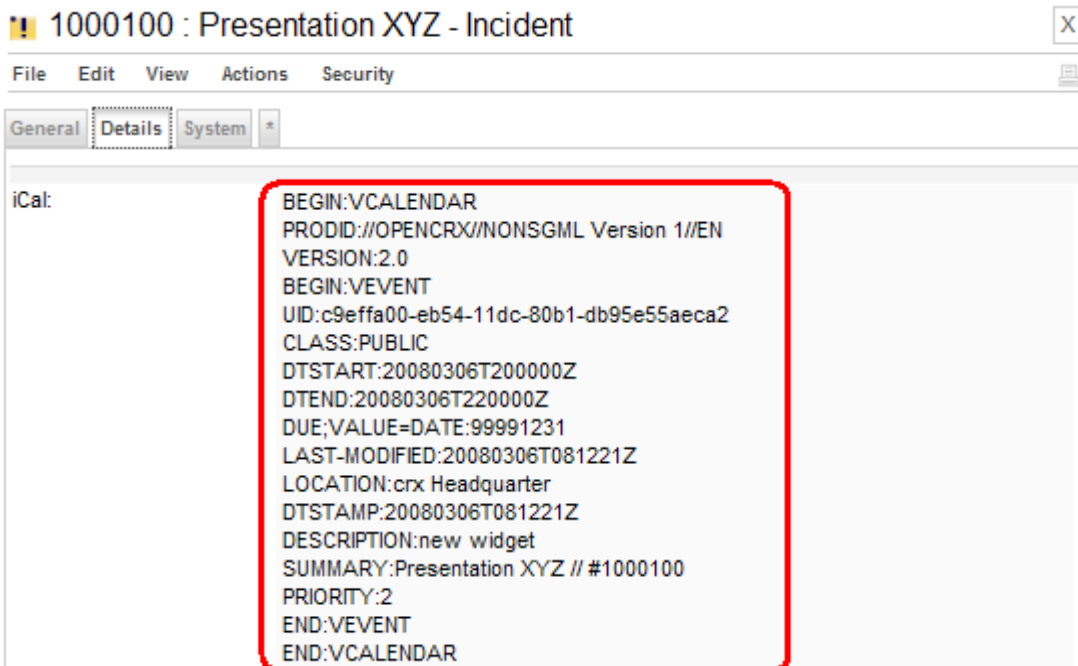


Figure 41: An openCRX activity's iCal representation

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

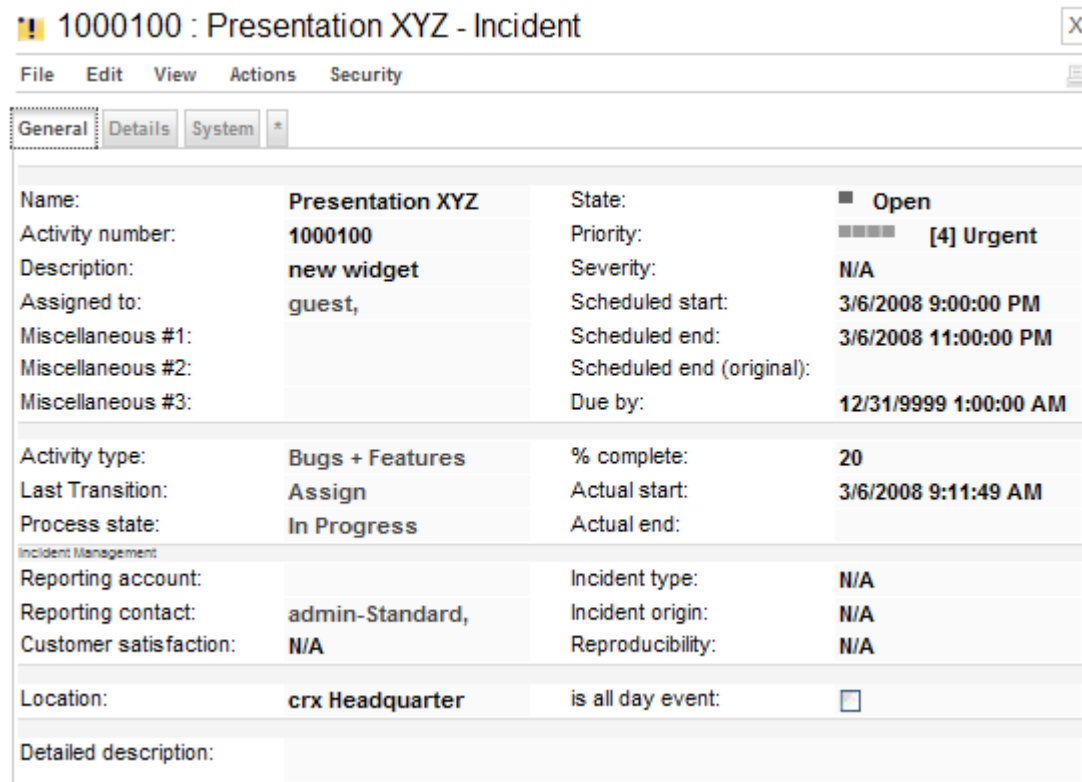


Figure 42: An openCRX activity in the standard GUI

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

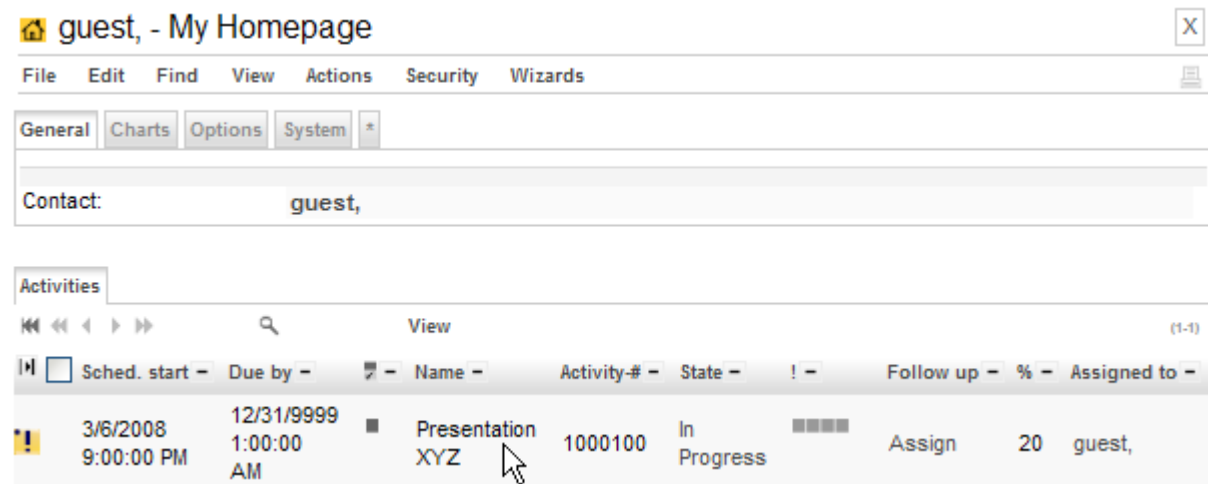


Figure 43: An openCRX activity filtered to a user's homepage

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](#).

9.2.4 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>/<Segment>/<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>/<Segment>/<Calendar Selector>&resource=freebusy.ics`

Example:

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

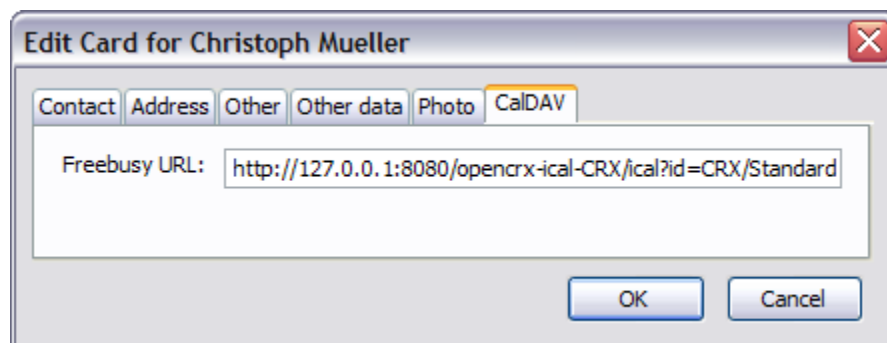
Please note that free busy information is provided by the openCRX server in a **read-only** fashion (i.e. free busy clients cannot update such information).

9.2.4.1 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 v0.66)

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:



9.2.4.2 Free Busy Configuration of MS Outlook

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

9.2.5 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>/<Segment>/<Calendar Selector>&resource=activities.ics
```

Example:

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

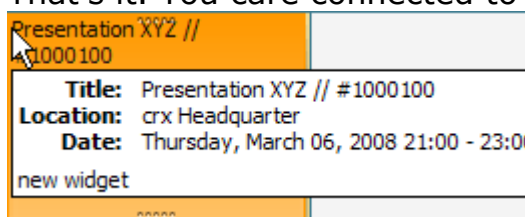


See chapter 9.2.2 Calendar Selectors for information on how to construct calendaring URLs.

9.2.5.1 ICS Configuration of Thunderbird/Lightning and Sunbird

Thunderbird 2.0 with the Lightning add-on (at least version 0.7) is a fully-fledged calendar client and offers virtually the same calendaring functionality as the stand-alone calendar client Sunbird (at least version 0.7). 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 9.2.2 Calendar Selectors for information on ICS URLs
- give your calendar a name and pick a color of your choice
- That's it. You are connected to openCRX:



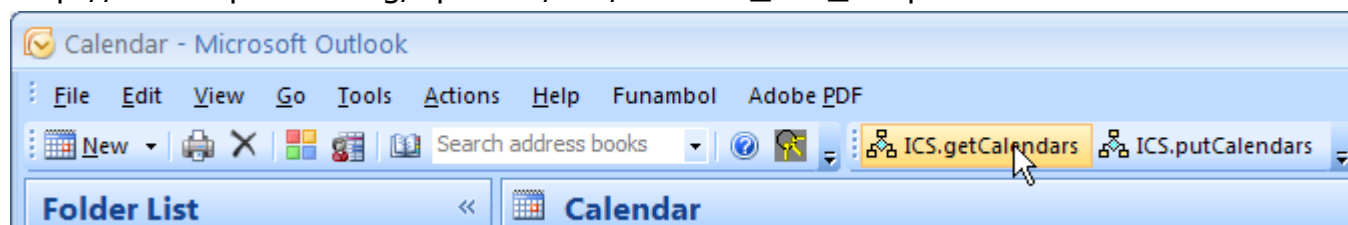
Unfortunately, the version 0.7 of Thunderbird/Lightning and Sunbird requires a life connection to openCRX (i.e. no support for offline viewing/editing).

9.2.5.2 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 strictly read-only in Outlook, i.e. about as useful as wallpaper...

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/2.0/Outlook_ICS_adapter.htm



9.2.6 Calendaring / CalDAV

CalDAV is one of the newer standards (see <http://www.caldav.org/> or Wikipedia for information about the **CalDAV standard**). openCRX can derive CalDAV information on-the-fly from the respective activities. CalDAV clients must authenticate to read and/or write CalDAV data.

CalDAV URL (with authentication):

`http://<crxServer>:<Port>/opencrx-caldav-
<Provider>/<Provider>/<Segment>/<Calendar Selector>`

Example:

`http://127.0.0.1:8080/opencrx-caldav-CRX/CRX/Standard/tracker/shared`



See chapter 9.2.2 Calendar Selectors for information on how to construct calendaring URLs.

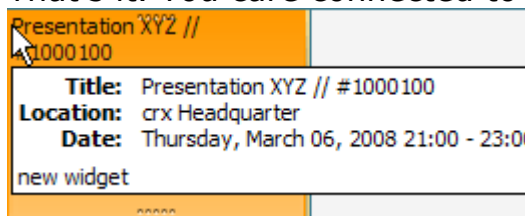


The CalDAV Servlet supports creation of new activities, and this requires a (default) ActivityCreator. As (default) ActivityCreators can only be defined for ActivityGroups, Calendar Selectors of the type homepage, resource, and global filter are not supported with CalDAV.

9.2.6.1 CalDAV Configuration of Thunderbird/Lightning and Sunbird

Thunderbird 2.0 with the Lightning add-on (at least version 0.7) is a fully-fledged CalDAV client and offers virtually the same calendaring functionality as the stand-alone calendar client Sunbird (at least version 0.7). 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 **CalDAV**
- enter a CalDAV URL into the field **Location**;
example: the user guest would connect to the tracker shared with the URL
`http://127.0.0.1:8080/opencrx-caldav-CRX/CRX/Standard/tracker/shared`
See chapter 9.2.2 Calendar Selectors for information on ICS URLs
- give your calendar a name and pick a color of your choice
- That's it. You are connected to openCRX:



Unfortunately, the version 0.7 of Thunderbird/Lightning and Sunbird requires a live connection to openCRX (i.e. no support for offline viewing/editing). Furthermore, CalDAV remote calendars are awfully slow (not because of Bedework or openCRX, it's a Lightning/Sunbird-internal issue).

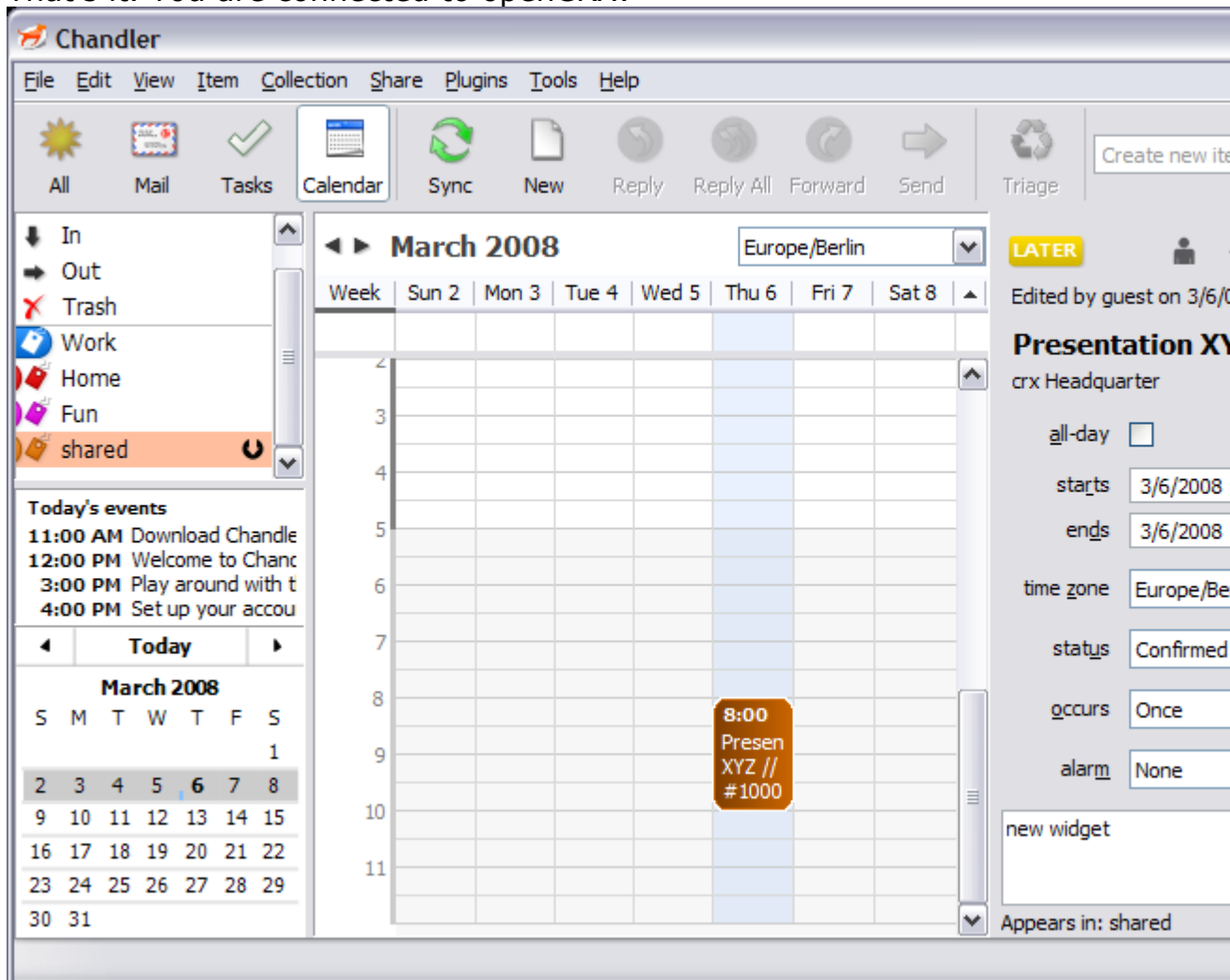
9.2.6.2 CalDAV Configuration of MS Outlook

Redmond, are you listening? How about adding CalDAV support to MS Outlook? Unfortunately, we are not aware of an affordable third-party plugin/add-on that could fill the gap. You are not out of luck, however: try our Open Source ICS Adapter for Outlook, available from http://www.opencrx.org/opencrx/2.0/Outlook_ICS_adapter.htm

9.2.6.3 CalDAV Configuration of Chandler Desktop

We connected Chandler Desktop v0.7.4.1 as follows with openCRX:

- select the menu **Share > Subscribe**
- enter the CalDAV URL into the field **URL**;
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/tracker/shared>
 See chapter 9.2.2 Calendar Selectors for information on CalDAV URLs
- you will be prompted for a user name (**guest**) and a password (**guest**)
- next you click the button **Subscribe**
- That's it. You are connected to openCRX:



The tested version of Chandler Desktop (v0.7.4.1) was very talkative and generated a lot of requests on the openCRX Server. Select the option "Suspend Syncing shared" on subscribed collections to stop the chatter. You can still sync on demand. It also seems that there are some timezone issues with Chandler. As Chandler is very much work in progress you must decide yourself whether this works for you or not.

9.2.7 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/>

» [SIMILE](#) » [Timeline](#)

Timeline

Timeline is a DHTML-based AJAXy widget for visualizing time-based events. It is like [Google Maps](#) for time-based information. Below is a live example that you can play with. Pan the timeline by dragging it horizontally.

Timeline © SIMILE

11:00 12:00 15 30 45 13:00 15

Man with seizure in front of TSBD
 Witing motorcade area
 Man with rifle seen in TSBD
 TSBD sixth floor empty
 Oswald seen having lunch
 Police radio for ambulance
 Williams left 6th floor
 Kennedy shot
 Washington D.C.'s
 Oswald leaves T
 Officers ordered to n

LINKS

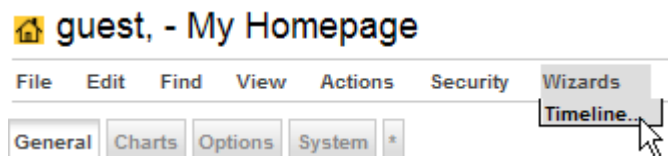
- [Live Examples](#)
- [Documentation](#)
 - [The Basics](#)
 - [How to Create Timelines](#)
 - [More...](#)
- [Browse the Code](#)
- [Issue Tracker](#)

RELATED PROJECTS

- [Timeplot](#)
- [Exhibit](#)

Figure 44: 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:



9.3 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 **E-MailActivities**. openCRX activity groups are mapped to IMAP folders. The folders contain openCRX **E-MailActivities**.

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.



Warning

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 **E-MailActivity** in that folder. Due to the **move** operation the message is deleted from the source folder 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 **E-MailActivity** was actually created in the target folder should you delete (if necessary) the message from the source folder.

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

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



Important

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

9.3.1 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**.

9.3.2 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.

9.3.2.1 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"
...
```

9.3.2.2 Maildir Configuration with JBoss

Add 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.

9.3.3 IMAP Configuration of Thunderbird

The following information is required to configure an IMAP account:

Email account	<login principal name>@<Segment> Example: guest@Standard
Password	openCRX password of the respective principal
Your name	<login principal name>@<Segment> 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>@<Segment> Example: guest@Standard

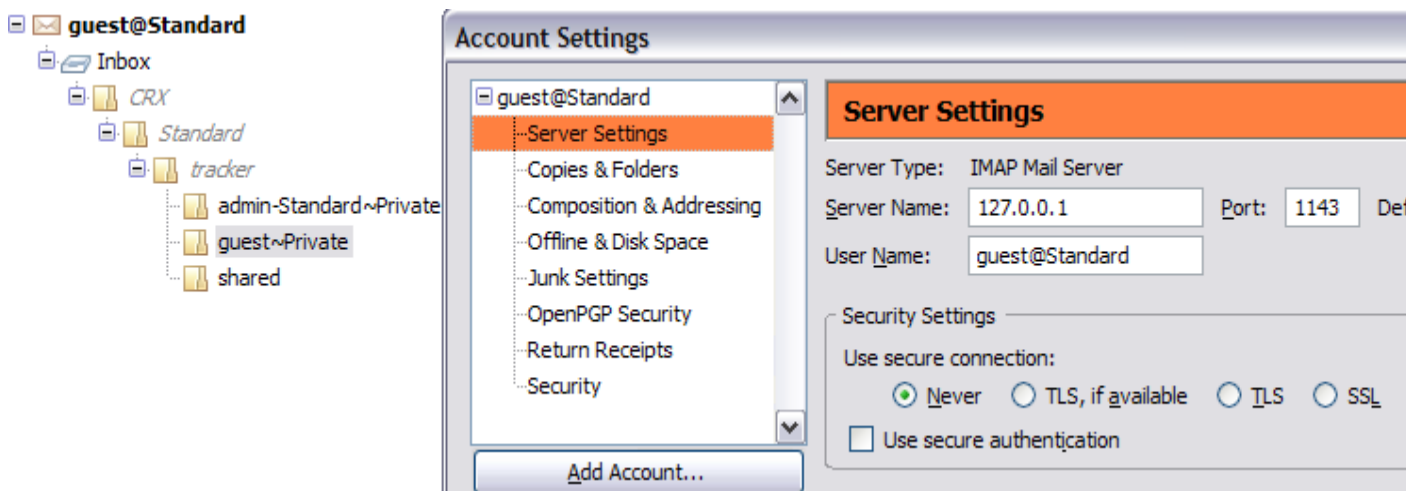


Figure 45: Thunderbird IMAP Configuration



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 **E-Mail Activity** in that folder. Due to the **move** operation the message is deleted from the source folder 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 **E-Mail Activity** was actually created in the target folder should you delete (if necessary) the message from the source folder.

9.3.4 IMAP Configuration of MS Outlook

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

Email account User Name	<login principal name>@<Segment> Example: guest@Standard
Password	openCRX password of the respective principal
Your Name	<login principal name>@<Segment> 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>@<Segment> Example: guest@Standard

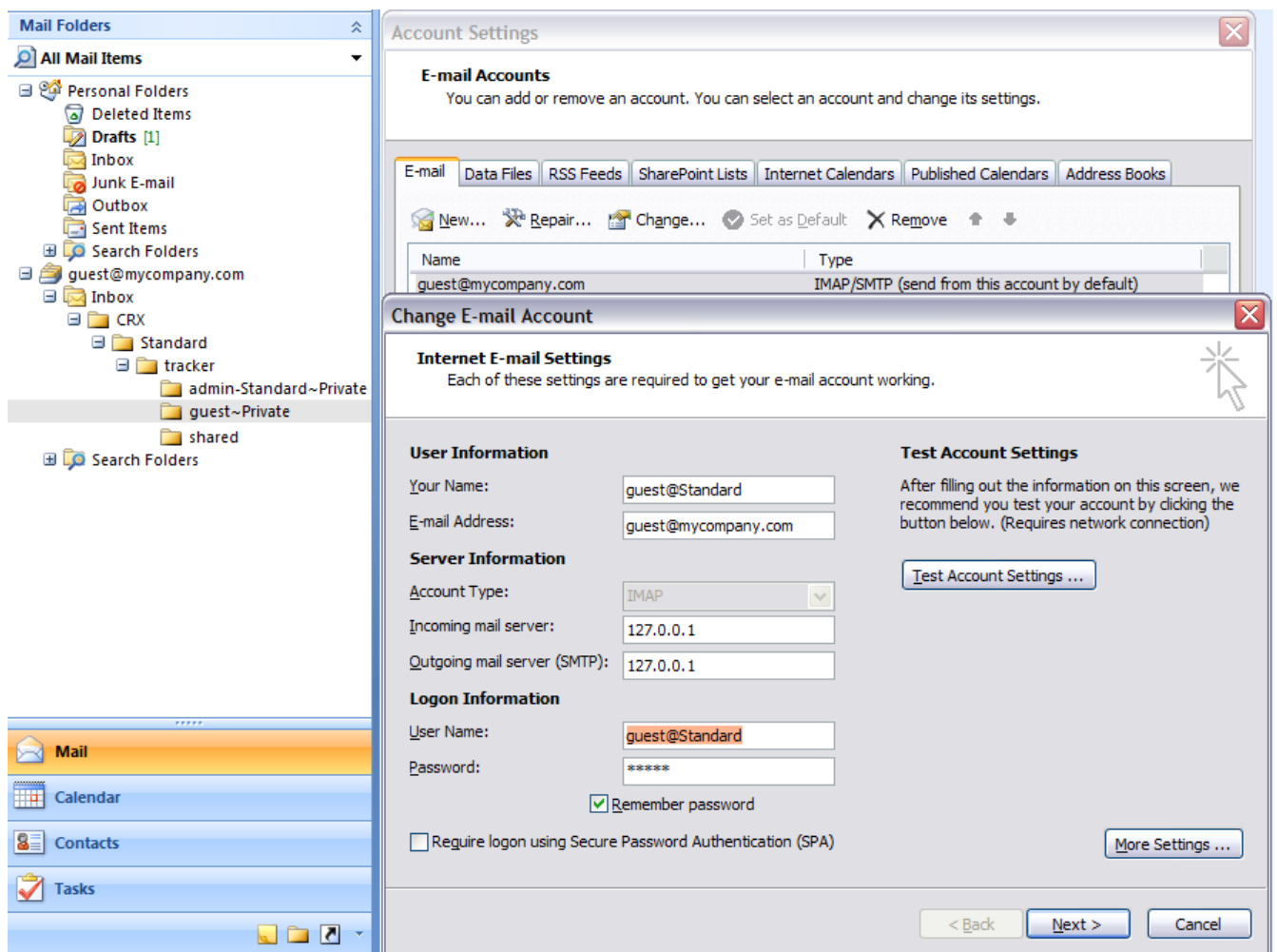


Figure 46: Thunderbird IMAP Configuration



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 **E-MailActivity** in that folder. Due to the **move** operation the message is deleted from the source folder 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 **E-MailActivity** was actually created in the target folder should you delete (if necessary) the message from the source folder.

9.4 Collaboration / Wiki

openCRX v2.0 features seamless integration with XWiki thereby offering all the features that you can find in a typical wiki. XWiki is a second generation wiki (a.k.a an application wiki), ideally suited for developing collaborative web applications. Each openCRX segment can have its own wiki (multi-entity enabled). Wiki pages are stored as openCRX documents or openCRX media attachments (org.opencrx.groupware.xwiki.spi.StoreImpl implements the XWiki document backend)

The following steps prepare an openCRX segment for XWiki:

- create a document folder named **XWikiSpaces**
- create the user **guest**. This is required to allow anonymous access for XWiki users. You can disable the guest login as it is not required (see chapter 4.2 Disable/Deactivate Users)
- in the database increase the size to 4K for the following columns **OOCKE1_PROPERTY.string_value** and **OOCKE1_DOCUMENT.keywords**
- launch XWiki with the URL **http://localhost:8080/opencrx-wiki-CRX/**
- login as **superadmin**. The preconfigured password is **changeit**. It can be configured in build.xml
- go to **Administration > Import** and create the page **Import**
- go to **Administration > Upload and Installation** and import the file **xwiki-enterprise-wiki-1.2-milestone-1.xar** (can be obtained from <http://www.xwiki.org/>) to install a default XWiki
- after successful import the panel **Search** has to be modified; put the page in edit mode and set the form action to **<form action="/opencrx-wiki-CRX/bin/view/Main/WebSearch">**

That's it for the setup. It is worthwhile, however, to browse the information available from the XWiki site at <http://www.xwiki.org/>.

10 Data Import/Export

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

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

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



Tip

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

10.1 Importing Data into openCRX

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



Warning

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

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

10.2 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:

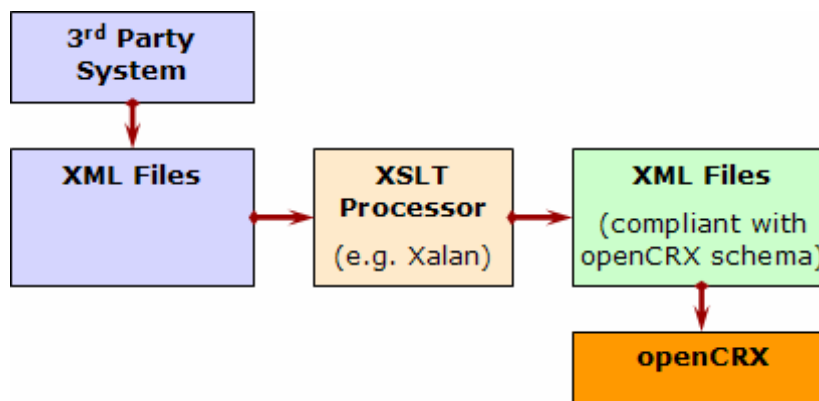


Figure 47: XML import from 3rd party system – overview

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**:

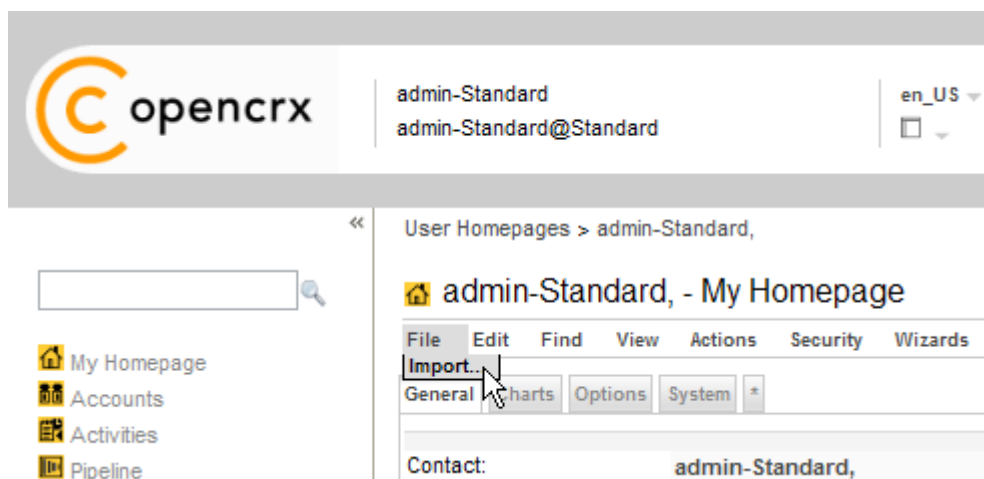


Figure 48: Interactive import of XML Files

Click on the button **Browse...** and navigate to XML file to be imported. Next you click OK to start the import. Please note that this method is

very suitable for small XML files and *on-the-fly* imports. If you are dealing with larger XML files, however, you should consider the bulk import described below.

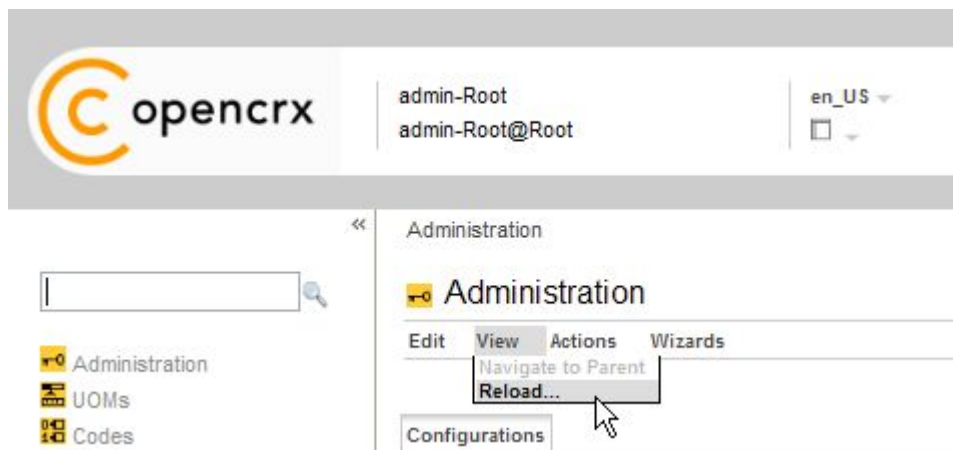
- **Bulk Import**

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

`opencrx-core-CRX.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 49: Interactive import of XML Files*



Tip

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.



Tip

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.

10.2.1 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:

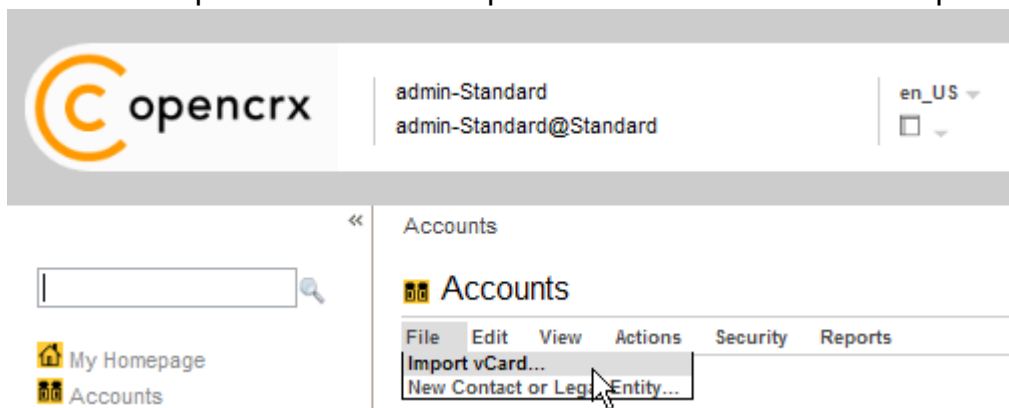


Figure 50: Operation vCard Import

- 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

10.2.2 Importing E-Mails

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

10.2.3 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.

10.3 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.

10.3.1 Exporting XML Files

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

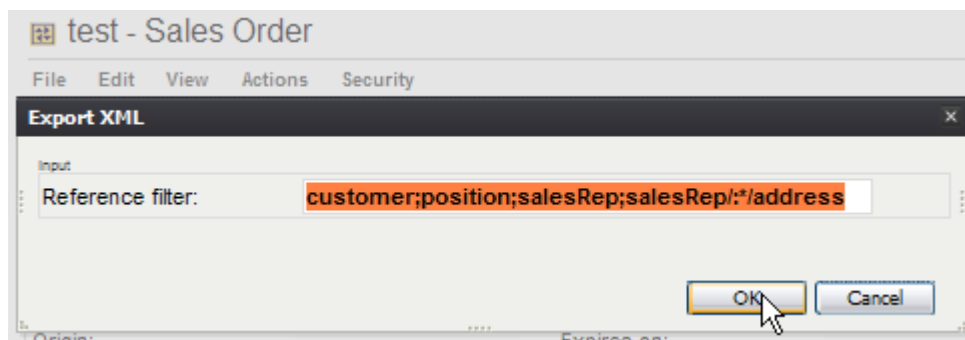


Figure 51: Exporting SalesOrder as XML File

In order to better control which additional objects (composites, referenced objects, ...) the XML exporter should export together with the object loaded in the Inspector, you can (optionally) provide a **reference filter**. The default reference filter is ***/*** meaning that all composites 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 a zip file containing the raw data and all the referenced code tables:

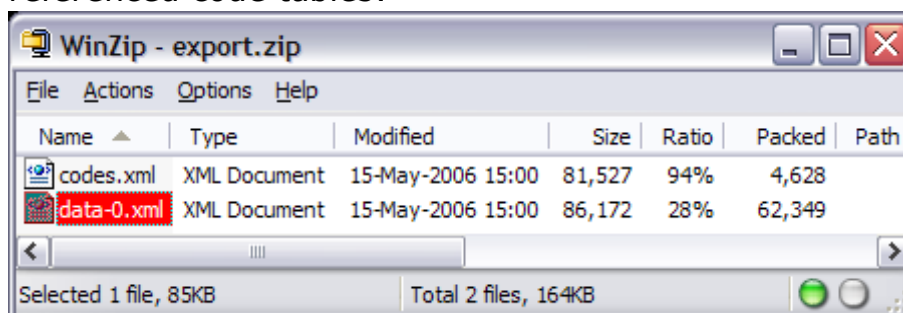


Figure 52: XML Exporter provides XML data file and code tables as ZIP file

10.3.2 Exporting openCRX Contacts (→ vCard Files)

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

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

guest, - Contact

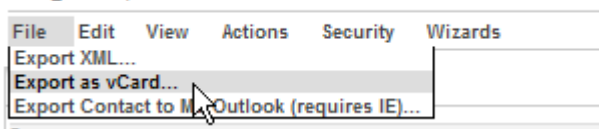


Figure 53: Export Contact as vCard

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

10.3.3 Exporting openCRX Contacts (→ Outlook Contacts)

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

guest, - Kontaktperson

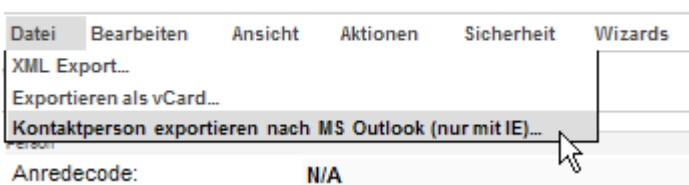


Figure 54: Export Contact to MS Outlook



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>

10.3.4 Exporting openCRX Meetings (→ iCalendar Files)

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

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

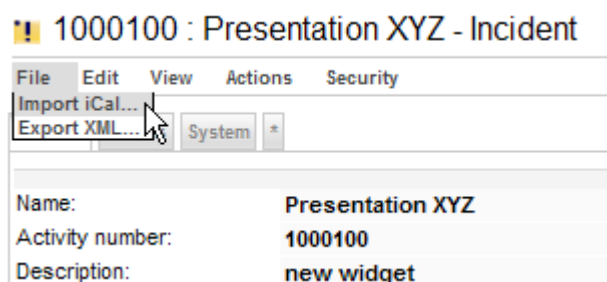


Figure 55: Exporting Meeting / Sales Visit as iCalendar File

10.3.5 Exporting E-Mails

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

10.3.6 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.

11 Customizing openCRX

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

11.1 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.

11.2 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: Locales in web.xml

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

```

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

Listing 20: Activating/Deactivating Locales in web.xml

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

```



Warning

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



Tip

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

11.3 Role-based UI

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

11.3.1 Model Permissions

Model permissions are not implemented yet.

11.3.2 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.

12 Integration with Office Suites

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

12.1 MS Office

12.1.1 MS Word

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

12.1.2 MS Excel

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

12.1.3 MS Outlook

See chapter 9 Groupware Services.

12.2 Open Office

12.2.1 OpenOffice Writer

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

12.2.2 OpenOffice Calc

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

12.3 Mozilla Thunderbird / Sunbird

See chapter 9 Groupware Services.

13 Next Steps

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