

# openCRX Admin Guide

Version 1.9.1



[www.opencrx.org](http://www.opencrx.org)

# DRAFT

24-Aug-2006 @ 07:59:23 AM

---

## **License**

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

**Copyright** 2006 © CRIXP Corp. All rights reserved.



## Table of Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>About this Book.....</b>                           | <b>7</b>  |
| 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         |
| <b>2</b> | <b>Prerequisites.....</b>                             | <b>8</b>  |
| <b>3</b> | <b>Security.....</b>                                  | <b>9</b>  |
| 3.1      | Introduction.....                                     | 9         |
| 3.1.1    | Basic Concepts and Conventions.....                   | 9         |
| 3.1.2    | Permissions / Access Control.....                     | 12        |
| 3.2      | Default Settings.....                                 | 14        |
| 3.3      | Activating Security.....                              | 16        |
| 3.4      | Security Settings of New Objects.....                 | 16        |
| 3.5      | Login Procedure.....                                  | 17        |
| 3.5.1    | Application Server Login.....                         | 17        |
| 3.5.2    | Segment Login.....                                    | 17        |
| 3.5.3    | Disabling Login.....                                  | 17        |
| 3.6      | Resetting Security.....                               | 18        |
| <b>4</b> | <b>Managing Users.....</b>                            | <b>19</b> |
| 4.1      | Creating Users.....                                   | 19        |
| 4.1.1    | Create Users Manually.....                            | 19        |
| 4.1.2    | Import Subjects and Application Login Principals..... | 20        |
| 4.1.3    | Import Users.....                                     | 21        |
| 4.2      | Disable/Deactivate Users.....                         | 22        |
| 4.3      | Disabling Users at the level Application Server.....  | 22        |
| 4.4      | Disabling Users at the level openCRX.....             | 22        |
| <b>5</b> | <b>Deployment Scenarios.....</b>                      | <b>23</b> |
| 5.1      | Typical Deployment Scenarios.....                     | 23        |
| 5.2      | Multi Entity Deployment Scenarios.....                | 24        |
| 5.2.1    | Multiple Data Segments in a single DB.....            | 24        |
| 5.2.2    | Multiple DBs.....                                     | 25        |
| 5.2.3    | Multiple Applications.....                            | 25        |
| 5.3      | openCRX Custom Applications.....                      | 25        |
| <b>6</b> | <b>Workflow Controller.....</b>                       | <b>26</b> |
| 6.1      | Servlet SubscriptionHandler.....                      | 27        |
| 6.2      | Servlet WorkflowHandler.....                          | 28        |
| 6.3      | MailImporterServlet.....                              | 28        |
| 6.4      | Trouble Shooting Servlets.....                        | 29        |
| <b>7</b> | <b>Subscribe / Notify Services.....</b>               | <b>30</b> |
| 7.1      | Example Subscription – Activity Modifications.....    | 32        |

---

|           |  |           |
|-----------|--|-----------|
| 7.2       | Trouble Shooting Notification Services.....            | 34        |
| <b>8</b>  | <b>E-mail Services.....</b>                            | <b>35</b> |
| 8.1       | Import E-mails from a Mail Client into openCRX.....    | 36        |
| 8.2       | Use openCRX as an E-mail Archive/Audit Tool.....       | 37        |
| 8.3       | Send E-mail directly from openCRX.....                 | 38        |
| 8.4       | Export E-mails from openCRX to your Mail Client.....   | 39        |
| 8.5       | Installing JavaMail and opencrx-ssl.jar.....           | 41        |
| 8.6       | Configuring E-mail Services.....                       | 41        |
| 8.6.1     | Outbound E-mail / Users.....                           | 42        |
| 8.6.2     | Outbound E-mail / Users relying on SYSTEM Account..... | 44        |
| 8.6.3     | Outbound E-mail (SYSTEM) / Segment Administrator.....  | 45        |
| 8.6.4     | Inbound E-mail / Segment Administrator.....            | 46        |
| 8.6.5     | Trouble Shooting E-mail Services.....                  | 54        |
| <b>9</b>  | <b>Data Import/Export.....</b>                         | <b>56</b> |
| 9.1       | Importing Data into openCRX.....                       | 56        |
| 9.2       | Importing XML Files.....                               | 56        |
| 9.2.1     | Importing vCard Files (→ openCRX Contacts).....        | 59        |
| 9.2.2     | Importing iCalendar Files (→ openCRX Meetings).....    | 59        |
| 9.2.3     | Other Options.....                                     | 60        |
| 9.3       | Exporting Data from openCRX.....                       | 60        |
| 9.3.1     | Exporting XML Files.....                               | 60        |
| 9.3.2     | Exporting openCRX Contacts (→ vCard Files).....        | 60        |
| 9.3.3     | Exporting openCRX Contacts (→ Outlook Contacts).....   | 60        |
| 9.3.4     | Exporting openCRX Meetings (→ iCalendar Files).....    | 60        |
| 9.3.5     | Other Options.....                                     | 60        |
| <b>10</b> | <b>Customizing openCRX.....</b>                        | <b>61</b> |
| 10.1      | Managing Locales.....                                  | 61        |
| 10.2      | Managing Packages.....                                 | 62        |
| 10.3      | Role-based UI.....                                     | 62        |
| 10.3.1    | Model Permissions.....                                 | 62        |
| 10.3.2    | Multiple Applications.....                             | 62        |
| 10.3.3    | Custom JSPs.....                                       | 62        |
| <b>11</b> | <b>Reporting with BIRT.....</b>                        | <b>63</b> |
| 11.1      | Restrictions/Pitfalls of using BIRT with openCRX.....  | 63        |
| 11.2      | How to design and deploy a BIRT report.....            | 64        |
| 11.3      | Report Debugging / BIRT Logging.....                   | 67        |
| 11.4      | Additional BIRT-Servlet Options.....                   | 68        |
| 11.5      | Outlook BIRT-Integration with openCRX.....             | 68        |
| <b>12</b> | <b>Integration with Office Suites.....</b>             | <b>69</b> |
| 12.1      | MS Office.....   | 69        |
| 12.2      | Open Office.....                                       | 69        |
| <b>13</b> | <b>Next Steps.....</b>                                 | <b>70</b> |

---

---

## List of Figures

|   |    |
|---|----|
| Figure 1: Security Realms, Principals and Subjects after Initial Setup.....   | 10 |
| Figure 2: Segment Administration.....   | 11 |
| Figure 3: Role Drop Down with list of available Segment Login Principals..... | 11 |
| Figure 4: openCRX UML Model – Class Diagram SecureObject.....                 | 12 |
| Figure 5: System attributes of an openCRX object as shown in the GUI.....     | 13 |
| Figure 6: Table kernel_Segment after default installation (QuickStart).....   | 15 |
| Figure 7: Table kernel_Segment after modification.....                        | 15 |
| Figure 8: Role Drop Down with list of available Segment Login Principals..... | 17 |
| Figure 9: Operation Actions > Import Login Principals (admin-Root).....       | 20 |
| Figure 10: Operation Actions > Import Users (admin-Standard).....             | 21 |
| Figure 11: Disabling of Segment Login Principal guest by admin-Standard....   | 22 |
| Figure 12: 3-Tier with Application Server.....                                | 23 |
| Figure 13: 4-Tier with Application Server.....                                | 23 |
| Figure 14: 4-Tier with Clustered Application Servers.....                     | 23 |
| Figure 15: 4-Tier with Servlet Engine.....                                    | 23 |
| Figure 16: Multiple Data Segments in a single DB.....                         | 24 |
| Figure 17: Dedicated DB for each Entity.....                                  | 25 |
| Figure 18: Accessing the openCRX Workflow Controller.....                     | 26 |
| Figure 19: openCRX v1.9.1 Workflow Controller – connect with http only.....   | 26 |
| Figure 20: openCRX MailImporterServlet.....                                   | 28 |
| Figure 21: Event and Notification Service.....                                | 30 |
| Figure 22: Standard Topics included in the openCRX distribution.....          | 31 |
| Figure 23: Create a new Subscription – step 1.....                            | 32 |
| Figure 24: Create a new Subscription – step 2.....                            | 32 |
| Figure 25: Create a new Subscription – step 3.....                            | 33 |
| Figure 26: Create a new Subscription – step 3.....                            | 33 |
| Figure 27: Flow of data between openCRX, mail server and mail client.....     | 35 |
| Figure 28: Import E-Mails from Mail Client.....                               | 36 |
| Figure 29: E-Mail Audit – import all inbound/outbound e-mail messages.....    | 37 |
| Figure 30: Send E-Mail from openCRX – Overview.....                           | 38 |
| Figure 31: Send E-Mail from openCRX with Actions > Follow Up.....             | 38 |
| Figure 32: Export E-Mail from openCRX – Overview.....                         | 39 |
| Figure 33: Export E-Mail from openCRX with Actions > Follow Up.....           | 39 |
| Figure 34: Envelope E-mail with exported E-Mail Activity as attachment.....   | 40 |
| Figure 35: MS Outlook – Resend This Message to edit/send e-mail.....          | 40 |
| Figure 36: MS Outlook – Send message.....                                     | 41 |
| Figure 37: Create a new E-Mail Account – step 1.....                          | 42 |
| Figure 38: Create a new E-Mail Account – step 2.....                          | 42 |
| Figure 39: Create a new E-Mail Account – step 3.....                          | 43 |

|   |    |
|---|----|
| Figure 40: E-mail subject prefix and Web access URL.....                    | 43 |
| Figure 41: Create a new E-Mail Account – use SMTP-info from SYSTEM.....     | 44 |
| Figure 42: Example of outbound E-mail Action Log Entries.....               | 45 |
| Figure 43: Create a new SYSTEM E-Mail Account – step 1.....                 | 45 |
| Figure 44: Create a new SYSTEM E-Mail Account – step 2.....                 | 46 |
| Figure 45: Grid Workflow Processes with entry MailImporterServlet.....      | 46 |
| Figure 46: MailImporterServlet Default Settings.....                        | 47 |
| Figure 47: Configuration of MailImporterServlet – Default Server.....       | 48 |
| Figure 48: Configuration of MailImporterServlet – Mail Server.....          | 48 |
| Figure 49: Configuration of MailImporterServlet – Account and Password..... | 49 |
| Figure 50: Envelope E-Mail with attached E-Mail to be imported.....         | 50 |
| Figure 51: Activity Tracker E-Mail is created automatically.....            | 51 |
| Figure 52: Activity Tracker E-Mail with newly imported e-mail.....          | 51 |
| Figure 53: Newly imported e-mail.....                                       | 52 |
| Figure 54: Activity Creator Default E-mail Creator.....                     | 53 |
| Figure 55: MailImporterServlet Entry.....                                   | 54 |
| Figure 56: XML import from 3rd party system – overview.....                 | 57 |
| Figure 57: Interactive import of XML Files.....                             | 57 |
| Figure 58: Interactive import of XML Files.....                             | 58 |
| Figure 59: Operation vCard Import.....                                      | 59 |
| Figure 60: BIRT Report Designer – Data Source Definition mysql-crx.....     | 64 |
| Figure 61: BIRT Report Designer – Data Set Query.....                       | 65 |
| Figure 62: BIRT Report in openCRX GUI.....                                  | 65 |
| Figure 63: BIRT Report Designer – Title and Description of openCRX Report.. | 66 |
| Figure 64: Integration of BIRT with BIRT ODA Implementation.....            | 68 |

## List of Listings

|   |    |
|---|----|
| Listing 1: File Format Subjects and Application Login Principals.....     | 20 |
| Listing 2: Example File Subjects and Application Login Principals.....    | 20 |
| Listing 3: File Format Users.....   | 21 |
| Listing 4: Example File Users.....  | 21 |
| Listing 5: Subscription Handler checking table kernel_AuditEntry.....     | 27 |
| Listing 6: Mark Audit Entries as visited by Subscription Handler.....     | 27 |
| Listing 7: Servlets managed by Workflow Controller log to server.log..... | 29 |
| Listing 8: Debug Output of MailImporterServlet.....                       | 50 |
| Listing 9: Importing Certificate.....                                     | 55 |
| Listing 10: Locales in web.xml.....                                       | 61 |
| Listing 11: Activating/Deactivating Locales in web.xml.....               | 61 |
| Listing 12: Sample Data Source Definition for MySQL.....                  | 64 |

Listing 13: Sample Data Set for Activity Group Report from MySQL..... 65  
Listing 14: openCRX Report Title and Description..... 66  
Listing 15: Setting BIRT Logging Level in web.xml..... 67  
Listing 16: Setting BIRT Log Directory in web.xml..... 68  
Listing 17: Setting Additional Options of BIRT-Servlet in web.xml..... 68

# 1 About this Book

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

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

## 1.1 Who this book is for

The intended audience are openCRX administrators.

## 1.2 What do you need to understand this book

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

## 1.3 Tips, Warnings, etc.

We make use the following pictograms:



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

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



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



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

## **2 Prerequisites**

openCRX installed (please refer to the QuickStart guide available from <http://www.opencrx.org/documents.htm> for installation instructions).

## 3 Security

In this chapter we will present a high-level overview of openCRX security and discuss a few select issues. Additional information is available in the openCRX Security Guide available from <http://www.opencrx.org/documents.htm>.



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

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

### 3.1 Introduction

#### 3.1.1 Basic Concepts and Conventions

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

- Each segment has a segment administrator principal (*admin-  
<segment name>*) (e.g. **admin-Standard** for the segment **Standard**).

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

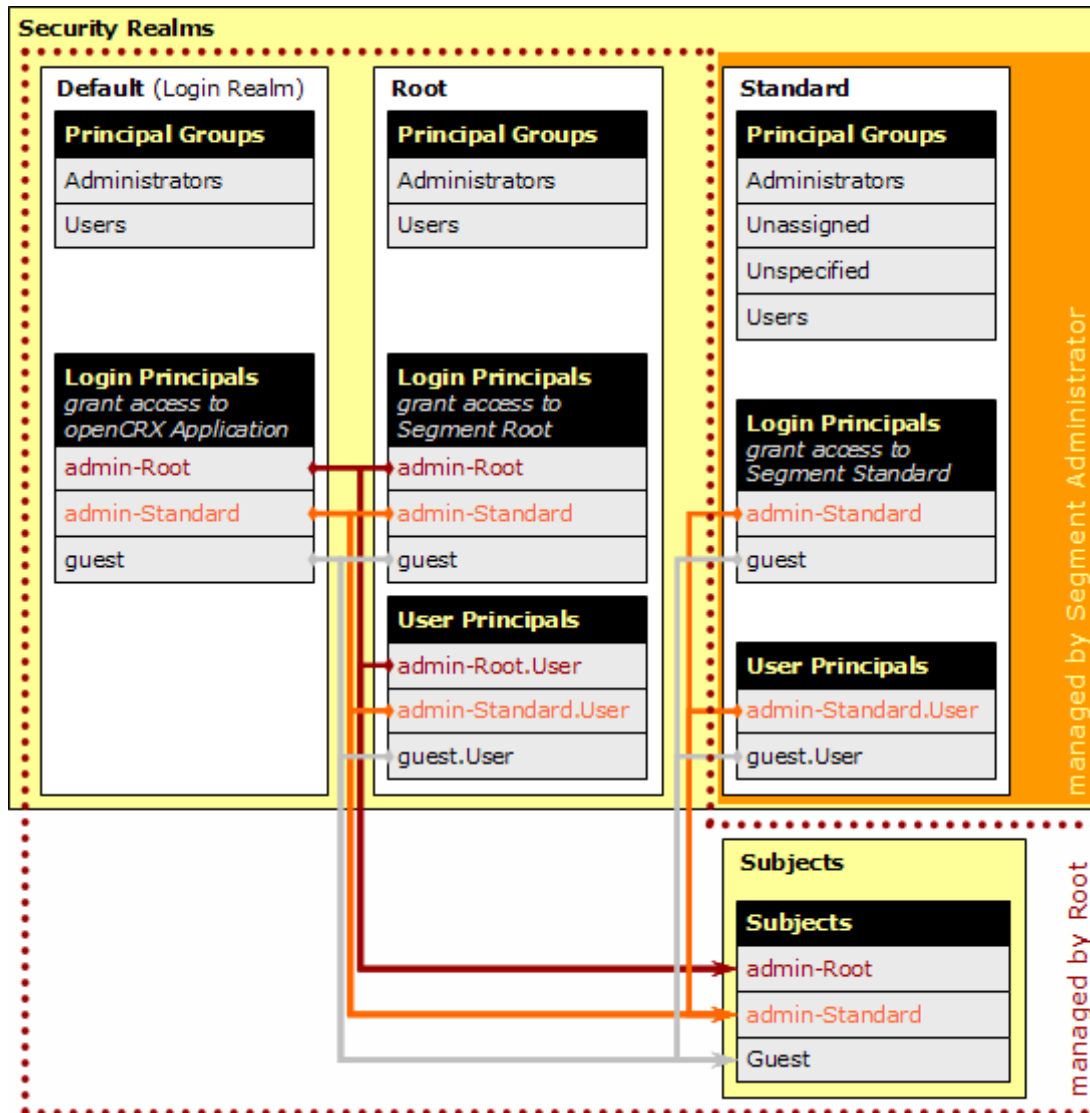


Figure 1: Security Realms, Principals and Subjects after Initial Setup

Summarizing the above:

- there is a realm for each segment (e.g. if you followed the QuickStart guide there is a realm **Standard** corresponding to the segment Standard)
- the realm **Default** acts as login realm, i.e. it contains all principals who are allowed to login to the openCRX application
- there is a subject for each "real user" and all principals of a user are assigned to the same subject; this allows openCRX to find all principals of a user (--> role selection drop down)

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

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

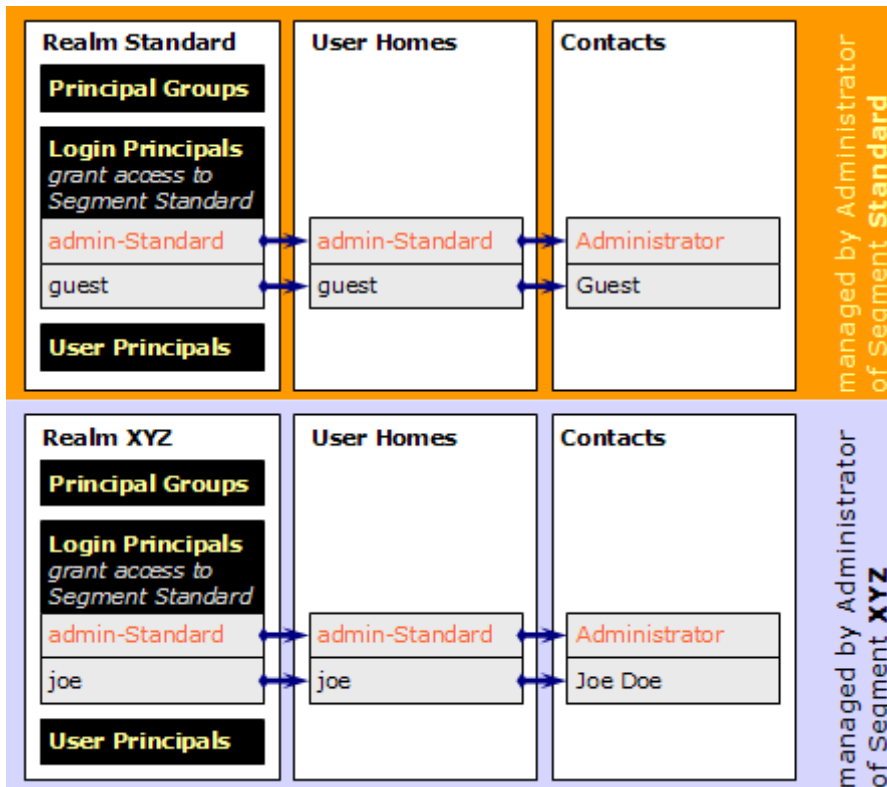


Figure 2: Segment Administration

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

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



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

### 3.1.2 Permissions / Access Control

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

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

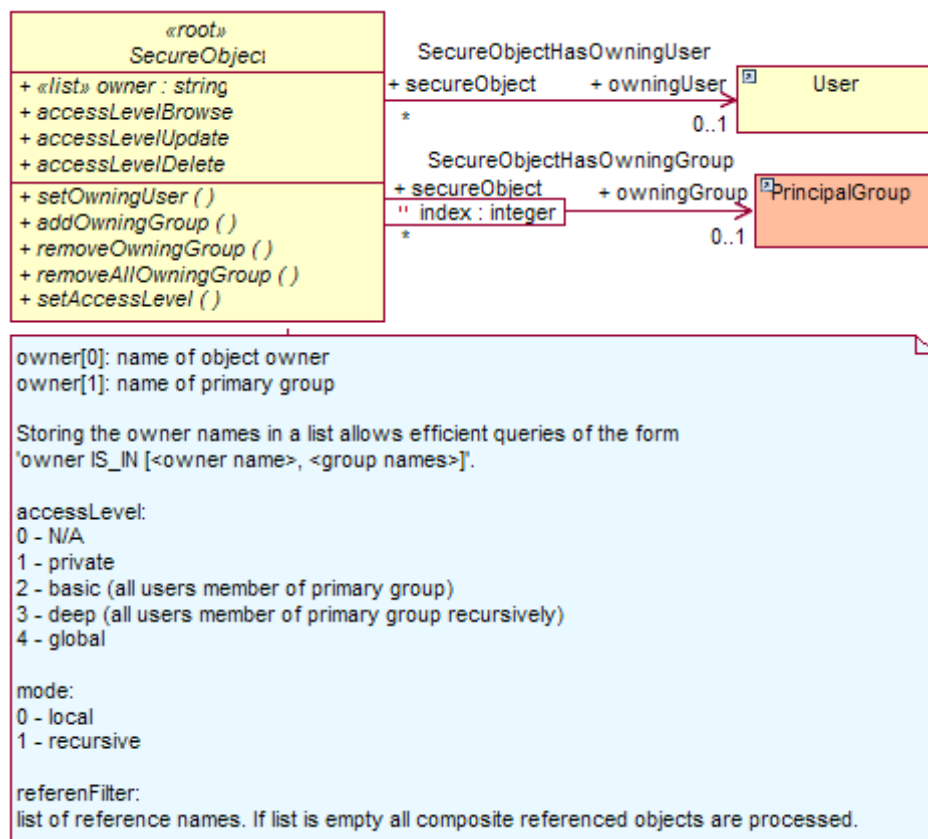


Figure 4: openCRX UML Model – Class Diagram SecureObject



Tip

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)



Tip

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

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.
- **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 | Details | Account | **System** ⓘ

Identity  
Identity: **xri:@openmdx:org.opencrx.kernel.account1/provider/CRX/segment/SI**

Object  
Created at: **3/12/06 3:48 PM** Modified at: **3/12/06 3:48 PM**  
Created by: **admin-Standard** Modified by: **admin-Standard**  
Reason: Disabled:

Security  
Owning user: **admin-Standard.User** Update access level: **[2] basic (group memb**  
Browse access level: **[3] deep (group mem** Delete access level: **[2] basic (group memb**

Common  
Categories: External links:

Addresses | Partners | Contacts | Contact Relationships | **Memberships** > ^

Search: [ ] [Q] [+] [-] **Owning Groups**

| [-] [ ] | - Name         | - User Group | - Disabled | - Description       | - Modified at   | - Modified by |
|---------|----------------|--------------|------------|---------------------|-----------------|---------------|
| [-] [ ] | Administrators |              |            | Root\Administrators | 3/12/06 1:40 PM | admin-Root    |

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"> <li>(a) the user is owning user of object X</li> <li>(b) the user is member of any of the owning groups of object X</li> <li>(c) any of the owning groups of object X is a subgroup** of any group the user is member of</li> </ul>   |
| 3 – deep     | access is granted if at least one of the following conditions is true: <ul style="list-style-type: none"> <li>(a) the user is owning user of object X</li> <li>(b) the user is member of any of the owning groups of object X</li> <li>(c) any of the owning groups of object X is a subgroup** of any group the user is member of</li> <li>(d) any of the owning groups of object X is a subgroup** of any supergroup* of any group the user is member of</li> </ul> |
| 4 – global   | all users are granted access  |
| *            | a supergroup of an owning group G is a group of which G is a member of (recursively)  |
| **           | a subgroup of an owning group G is a group which is member of G (recursively)   |

## 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_rid     | object_oid | object_class                           | owner                        | ...  | ...  | ...  |
|----------------|------------|--|------------------------------|------|------|------|
| admins/CRX     | Root       | 0 org.opencrx:kernel:admin1:Segment    | NULL                         | NULL | NULL | NULL |
| codes/CRX      | Root       | 0 org.opencrx:kernel:code1:Segment     | Root:admin-Root.User         | 4    | 2    | 2    |
| codes/CRX      | Root       | 1 NULL                                 | Root:admin-Root.User         | NULL | NULL | NULL |
| uoms/CRX       | Root       | 0 org.opencrx:kernel:uom1:Segment      | Root:admin-Root.User         | 4    | 2    | 2    |
| uoms/CRX       | Root       | 1 NULL                                 | Root:admin-Root.User         | NULL | NULL | NULL |
| accounts/CRX   | Standard   | 0 org.opencrx:kernel:account1:Segment  | Standard:admin-Standard.User | 4    | 3    | 1    |
| accounts/CRX   | Standard   | 1 NULL                                 | Standard:Administrators      | NULL | NULL | NULL |
| activities/CRX | Standard   | 0 org.opencrx:kernel:activity1:Segment | Standard:admin-Standard.User | 4    | 3    | 1    |
| activities/CRX | Standard   | 1 NULL                                 | Standard:Administrators      | NULL | NULL | NULL |

Figure 6: Table kernel\_Segment after default installation (QuickStart)

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

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

| object_rid     | object_oid | object_class                           | owner                        | access_level_browse |
|----------------|------------|--|------------------------------|---------------------|
| admins/CRX     | Root       | 0 org.opencrx:kernel:admin1:Segment    | NULL                         | NULL                |
| codes/CRX      | Root       | 0 org.opencrx:kernel:code1:Segment     | Root:admin-Root.User         | 4                   |
| codes/CRX      | Root       | 1 NULL                                 | Root:admin-Root.User         | NULL                |
| uoms/CRX       | Root       | 0 org.opencrx:kernel:uom1:Segment      | Root:admin-Root.User         | 4                   |
| uoms/CRX       | Root       | 1 NULL                                 | Root:admin-Root.User         | NULL                |
| accounts/CRX   | Standard   | 0 org.opencrx:kernel:account1:Segment  | Standard:admin-Standard.User | 3                   |
| accounts/CRX   | Standard   | 1 NULL                                 | Standard:Administrators      | NULL                |
| activities/CRX | Standard   | 0 org.opencrx:kernel:activity1:Segment | Standard:admin-Standard.User | 3                   |
| activities/CRX | Standard   | 1 NULL                                 | Standard:Administrators      | NULL                |

Figure 7: Table kernel\_Segment after modification

You may also decide to change the browse access level for some non-Root segments.



We do **not** recommend changing the settings of the segment Root.

### 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 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 access level browse 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**  
Owning User: **User who is creating the object**  
Owning Groups: **Primary User Group of the user who is creating the object and (meaning as well as ) Owning Group(s) of the parent object of the new object.**



Please note that the User Group **Users** (e.g. Standard\\Users) is **not added** to the list of *Owning Groups* of newly created top-level objects (i.e. objects attached to a segment, even if **Users** is an *Owning Group* of the segment).



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



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

## 3.5 Login Procedure

The openCRX login procedure consists of 2 levels:

### 3.5.1 Application Server Login

The application server login procedure depends on various parameters:

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

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

### 3.5.2 Segment Login

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

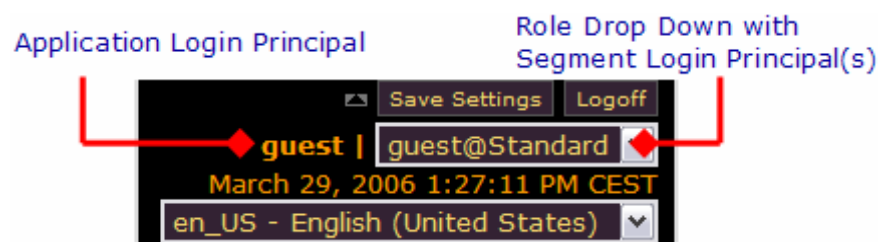


Figure 8: Role Drop Down with list of available Segment Login Principals

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

### 3.5.3 Disabling Login

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

## 3.6 Resetting Security



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

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

## 4 Managing Users



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

### 4.1 Creating Users

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

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

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

#### 4.1.1 Create Users Manually

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

## 4.1.2 Import Subjects and Application Login Principals

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

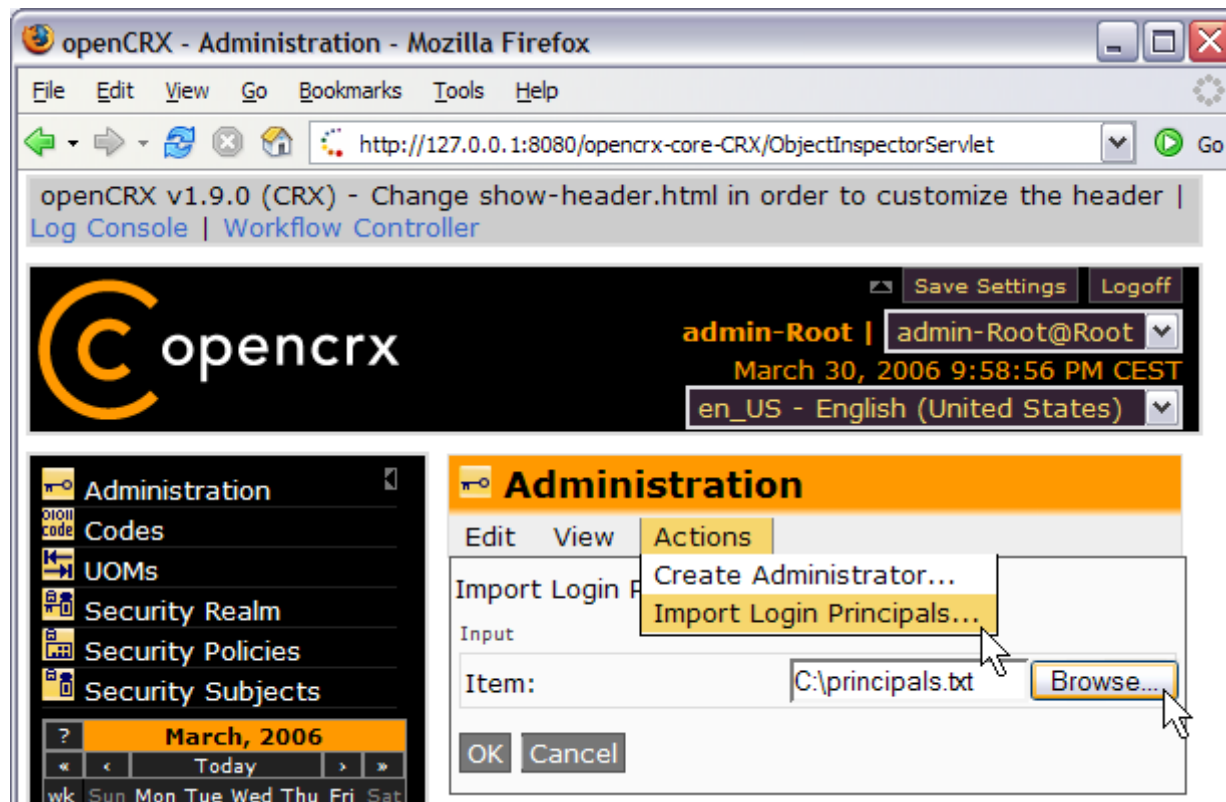


Figure 9: Operation **Actions > Import Login Principals** (admin-Root)

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

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

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

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

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

### 4.1.3 Import Users

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



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

#### Listing 3: File Format **Users**

```
Users;<principal>;<account alias>;<account full name>;<primary group>;<password>
```

Please note that <password> is a clear-text value.

#### 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 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.

## 4.3 Disabling Users at the level Application Server

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

## 4.4 Disabling Users at the level openCRX

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

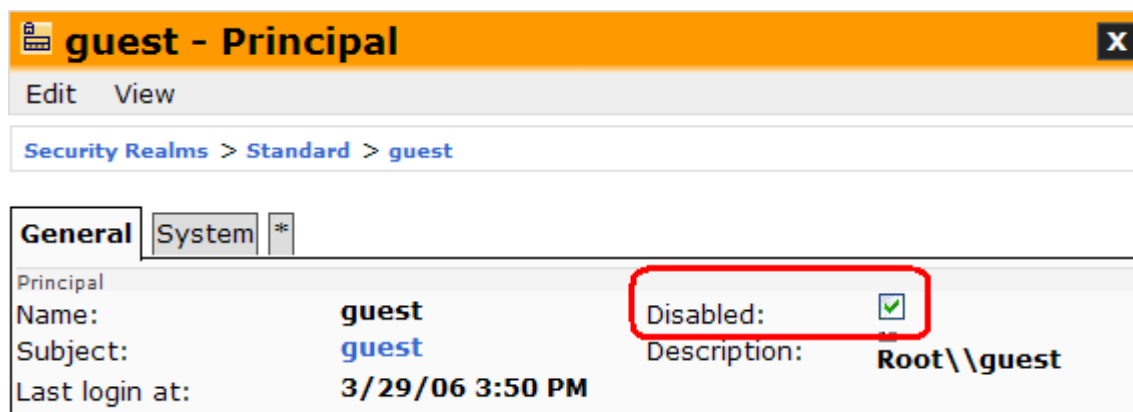


Figure 11: Disabling of Segment Login Principal **guest** by **admin-Standard**

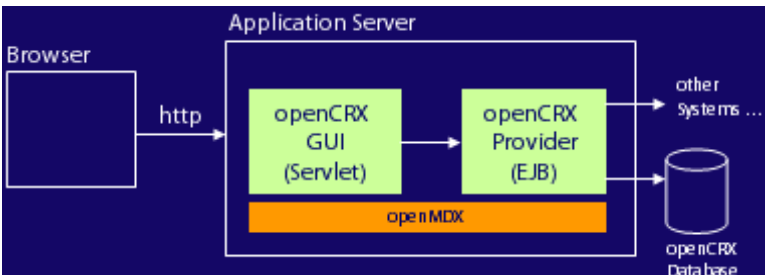
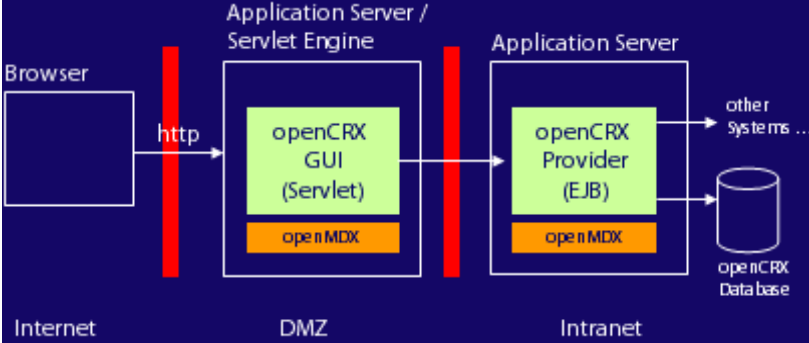
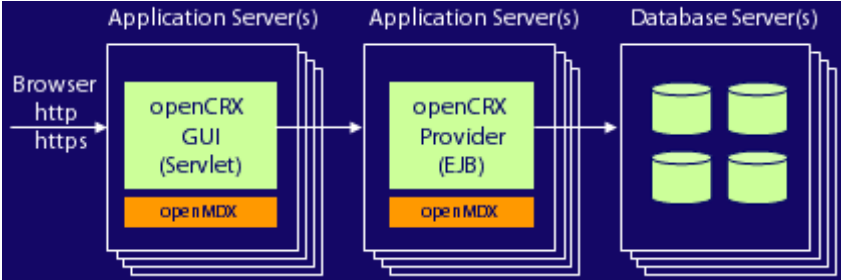
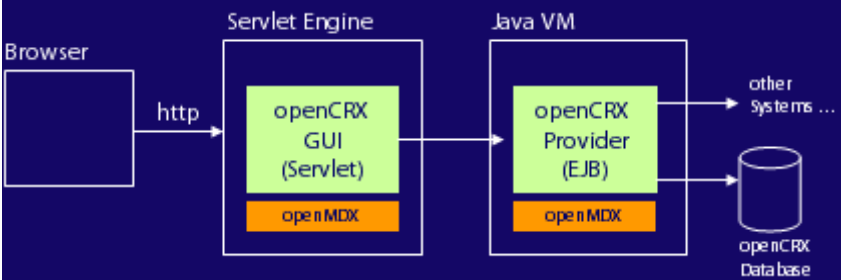


You should **not** delete a particular Subject as long as it is referenced by any Principal. Otherwise you'll end up with "dangling" Subject references.

## 5 Deployment Scenarios

### 5.1 Typical Deployment Scenarios

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

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

## 5.2 Multi Entity Deployment Scenarios

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

### 5.2.1 Multiple Data Segments in a single DB

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

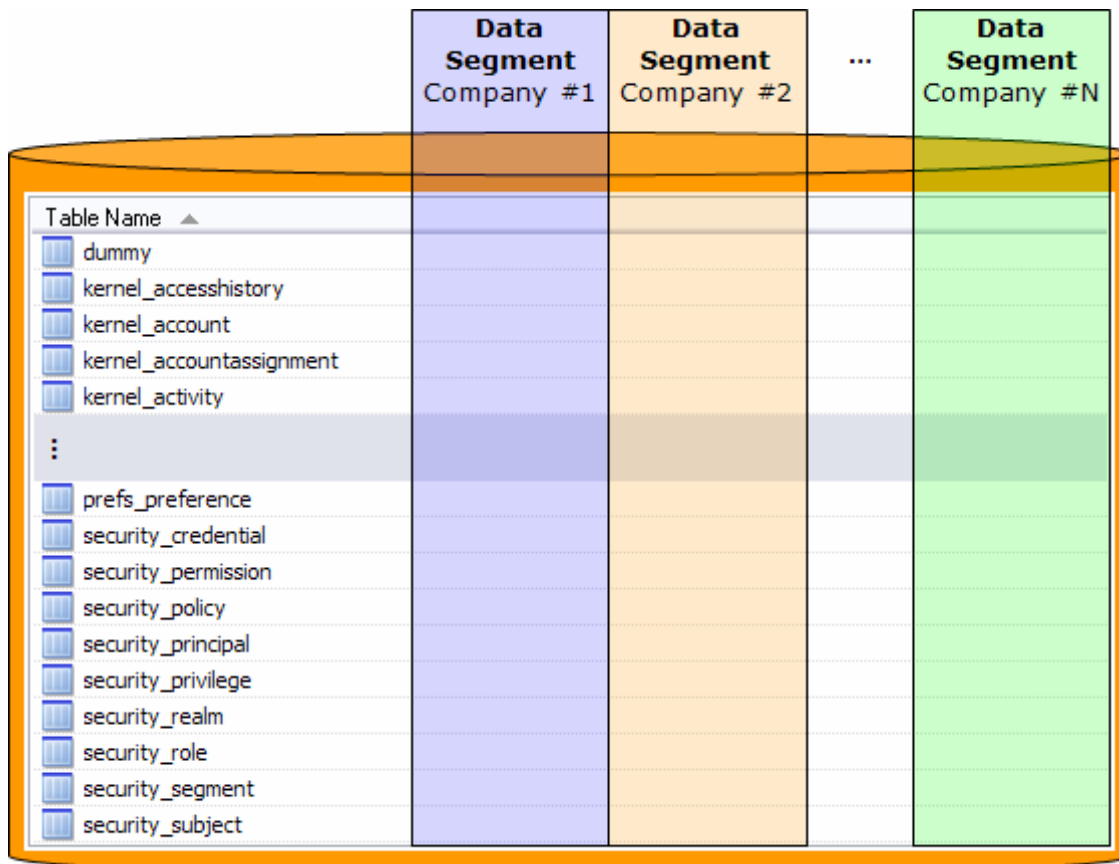


Figure 16: Multiple Data Segments in a single DB

### 5.2.2 Multiple DBs

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

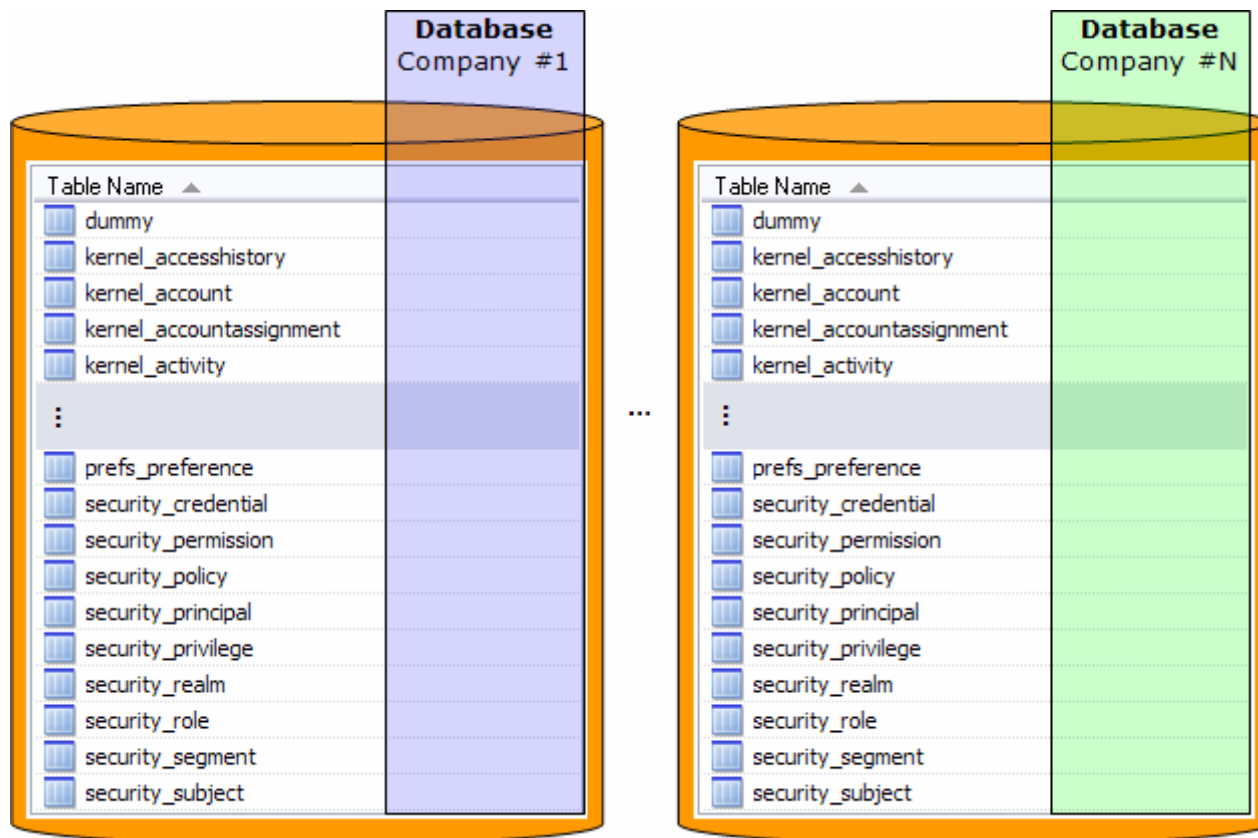


Figure 17: Dedicated DB for each Entity

### 5.2.3 Multiple Applications

<documentation pending>

### 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 must **connect to the Workflow Controller with http** (and NOT with https.). If you use SSL-secured connections to start/stop servlets they will not work properly.

The following figure shows the Workflow Controller of openCRX v1.9.1 (the look and feel may be different for other openCRX versions):

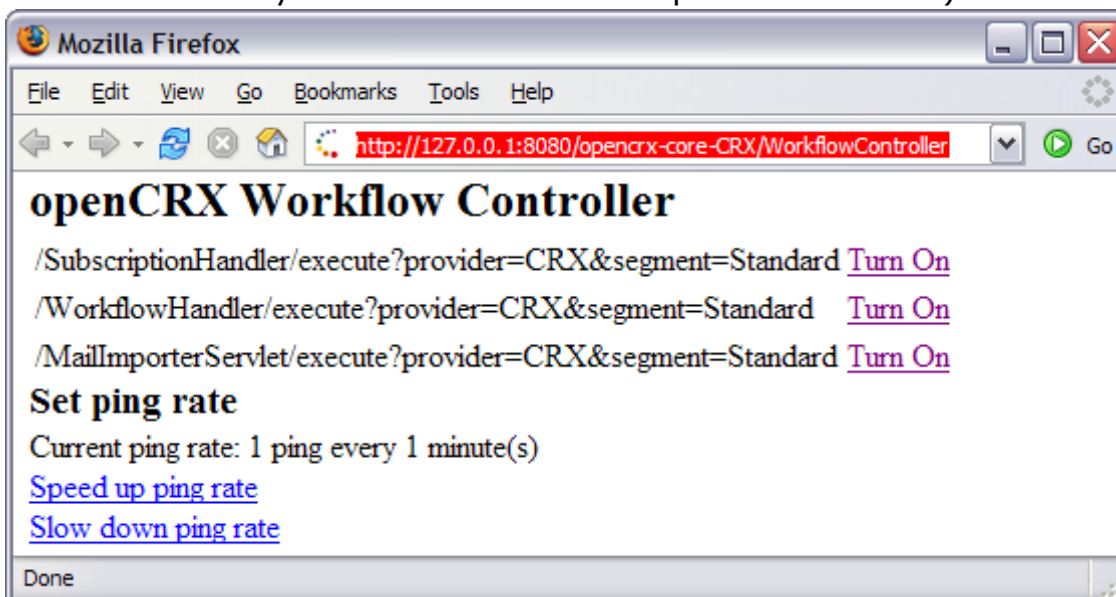


Figure 19: openCRX v1.9.1 Workflow Controller – connect with http only



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.

You can start (stop) servlets 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 "MyCompany" in addition to the segment "Standard" you can start/stop servlets of the segment "MyCompany" without interfering with the servlets of the segment "Standard".

By adjusting the ping rate you can control the time interval between individual runs of Workflow Controllers. For example, a ping rate setting of "**1 ping every 5 minute(s)**" will have the **SubscriptionHandler** check for new events once every 5 minutes. At the same interval all the other servlets will be triggered.

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

*Listing 5: Subscription Handler checking table kernel\_AuditEntry*

```
SELECT * FROM kernel_AuditEntry
WHERE 'SubscriptionHandler' NOT IN (visited_by)
```

The polling frequency can be set by the Root administrator (see *Figure 19: openCRX v1.9.1 Workflow Controller – connect with http only*).



Please note that activating the Subscription Handler on a DB with lots of (unvisited) entries in the table kernel\_AuditEntry (e.g. after an upgrade from openCRX v1.8.1 to openCRX v1.9.1) can put some heavy load on your system until all the Notifications have been generated.

You might want to use the following statement to mark all the existing entries as visited:

*Listing 6: Mark Audit Entries as visited by Subscription Handler*

```
UPDATE kernel_AuditEntry
set visited_by = 'SubscriptionHandler'
where (object_idx = 0) and (visited_by is null)
```

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

## 6.2 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 19: openCRX v1.9.1 Workflow Controller – connect with http only*). Please note that the `WorkflowHandler` is required for outbound E-Mail Services.

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



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

## 6.3 MailImporterServlet

The `MailImporterServlet` provides generic E-mail Services. The servlet regularly connects to e-mail boxes and fetches messages for import.

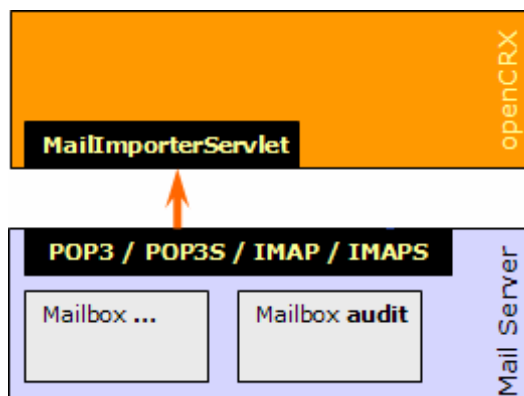


Figure 20: openCRX MailImporterServlet

The polling frequency can be set by the Root administrator (see *Figure 19: openCRX v1.9.1 Workflow Controller – connect with http only*).

## 6.4 Trouble Shooting Servlets

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

### Listing 7: Servlets managed by Workflow Controller log to server.log

```

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

```

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

It is always worth checking whether the Workflow Handlers actually are active; they must be started by the Root administrator. You can find out by connecting to the Workflow Controller (see *Figure 19: openCRX v1.9.1 Workflow Controller – connect with http only*).



After restarting the application server all servlets managed by the Workflow Controller are turned off, i.e. the Root Administrator must explicitly turn them on again (if desired). 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:

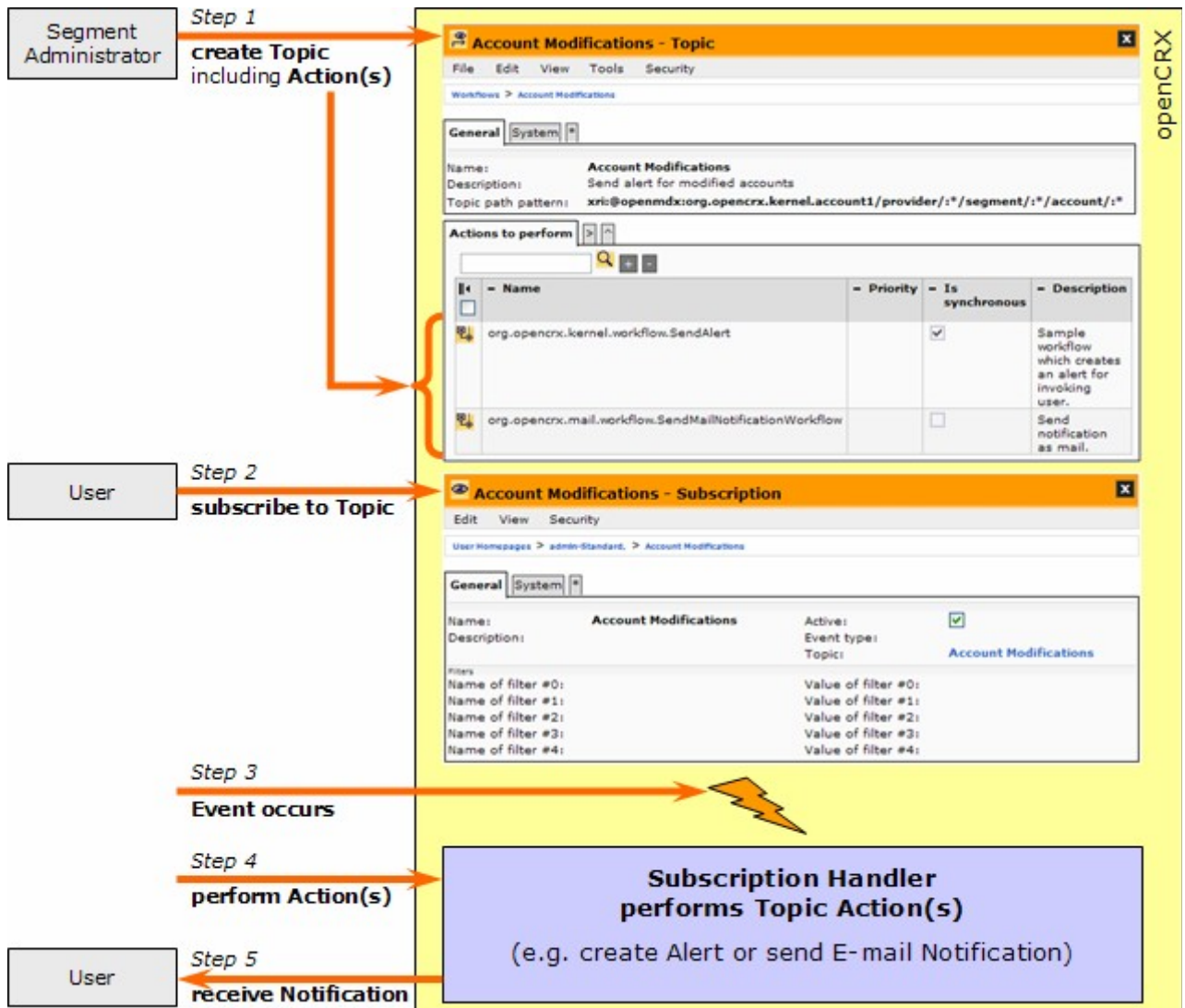


Figure 21: Event and Notification Service

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



Please note that event and notification services depend on the **Servlet SubscriptionHandler**, i.e. **you must turn on the openCRX Subscription Handler** for the respective segment with the **Workflow Controller**, otherwise **Topic Actions** are not executed, i.e. no Alerts, 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 22: Standard Topics included in the openCRX distribution*) to get you started:

- Topic **Account Modifications** sends an alert to the UserHome of subscribed users and – if e-mail services are configured - e-mail notifications to subscribed users whenever an account is modified.
- Topic **Activity Follow Up Modifications** sends an alert to the UserHome of subscribed users and – if e-mail services are configured - e-mail notifications to subscribed users whenever a Follow Up of an Activity is modified.
- etc.



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.

|  | - Name                           | - Description                       | - Topic path pattern          |
|--|----------------------------------|-------------------------------------|-------------------------------|
|  | Account Modifications            | Send alert for modified accounts    | xri:@openmdx:org.opencrx.kern |
|  | Activity Follow Up Modifications | Send alert for modified activity fo | xri:@openmdx:org.opencrx.kern |
|  | Activity Modifications           | Send alert for modified activities  | xri:@openmdx:org.opencrx.kern |
|  | Booking Modifications            | Send alert for modified bookings    | xri:@openmdx:org.opencrx.kern |
|  | Competitor Modifications         | Send alert for modified competitc   | xri:@openmdx:org.opencrx.kern |
|  | Competitor Modifications         | Send alert for modified competitc   | xri:@openmdx:org.opencrx.kern |
|  | Compound Booking Modifications   | Send alert for modified compouni    | xri:@openmdx:org.opencrx.kern |

*Figure 22: Standard Topics included in the openCRX distribution*

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

## 7.1 Example Subscription – Activity Modifications

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

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

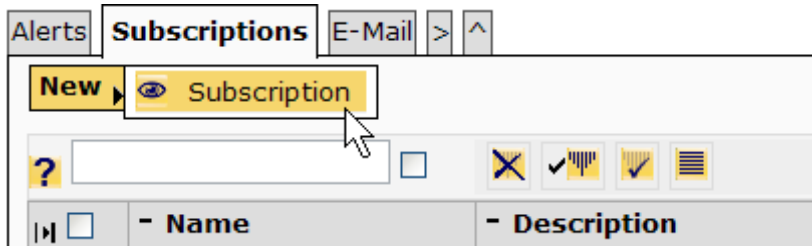


Figure 23: Create a new Subscription – step 1

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

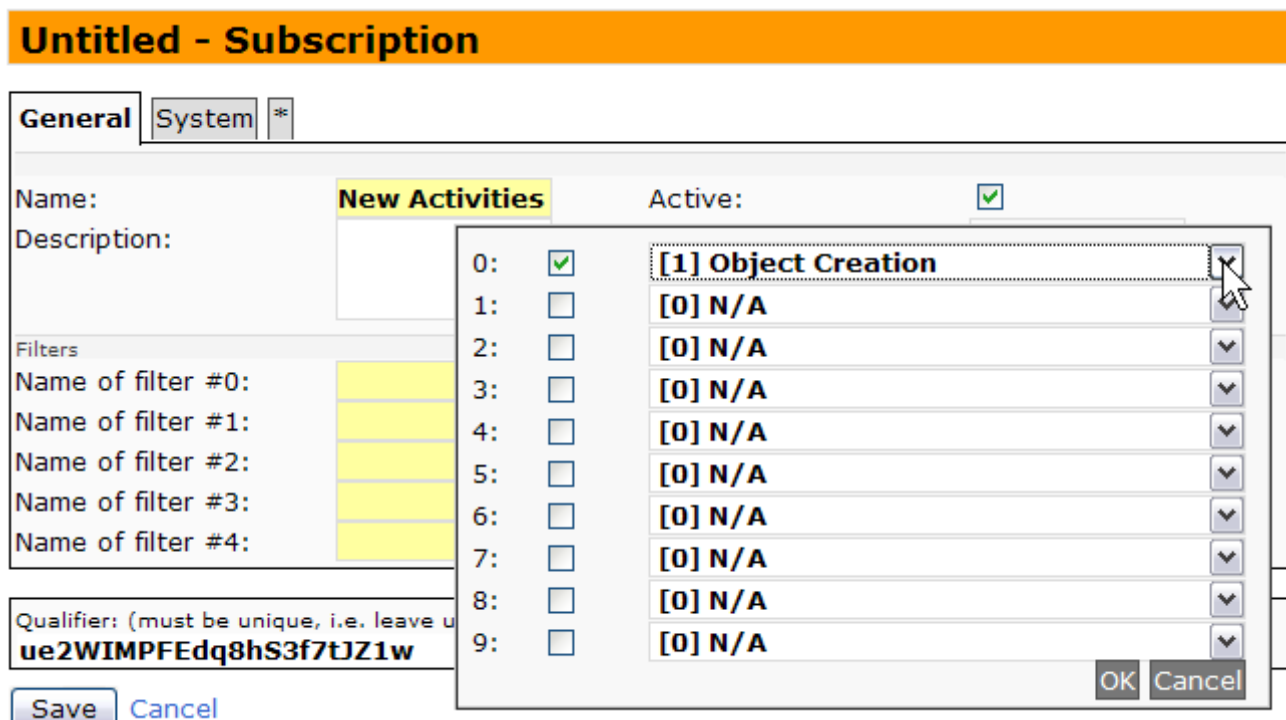


Figure 24: Create a new Subscription – step 2

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

- Then you click the lookup icon next to the field Topic to start the lookup inspector where you select the topic entry **Activity Modifications**:

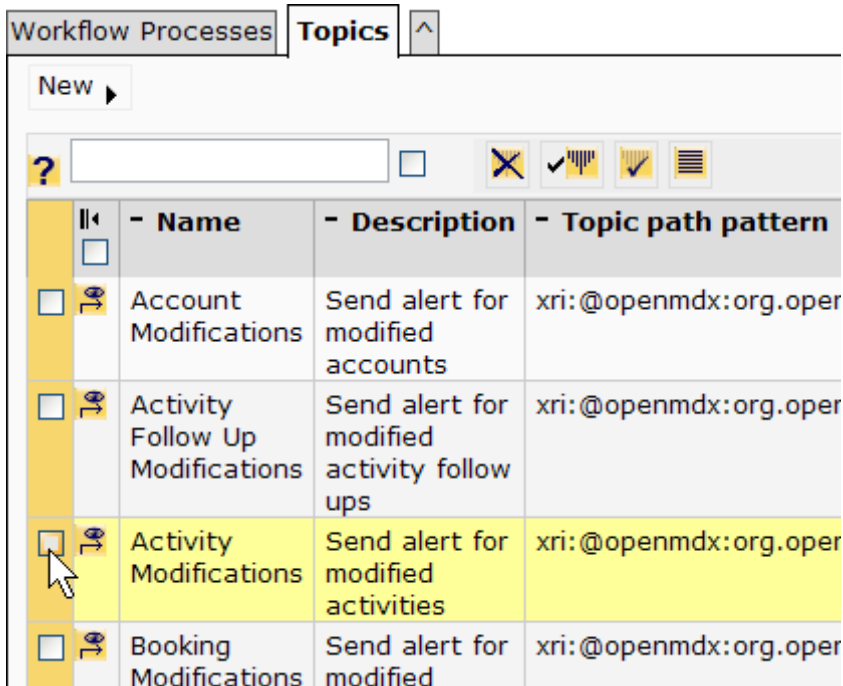


Figure 25: Create a new Subscription – step 3

- Clicking into the checkbox of the Topic Activity Modification takes you back to the Subscription. Click the button **Save** to commit the new Subscription. The grid **Subscriptions** should now contain an entry for the new Subscription:

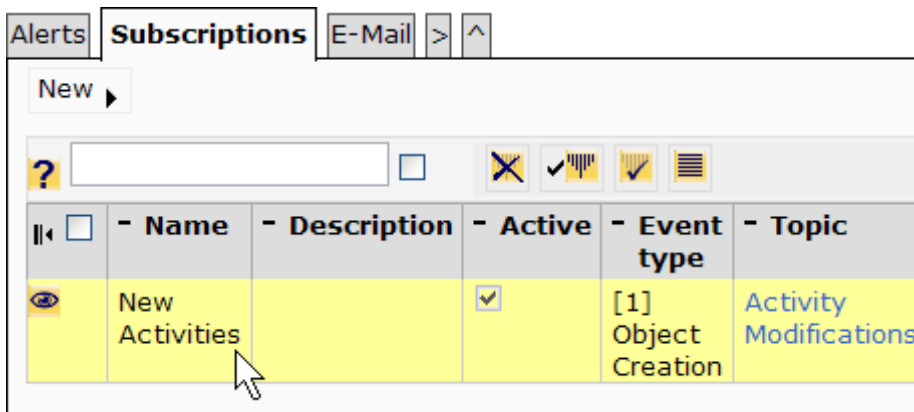


Figure 26: Create a new Subscription – step 3


- This completes the creation of this new Subscription.



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

## 7.2 Trouble Shooting Notification Services

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

| Problem   | Solution  |
|---|---|
| The grids <b>Workflow Processes</b> and/or <b>Topics</b> are empty.               | <ul style="list-style-type: none"> <li>▪ 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)</li> <li>▪ click on the filter button  to see all rows without filtering (maybe you defined a default filter in the past?)</li> </ul>  |
| I started the Subscription Handler but I never receive any Alerts / Notifications | <ul style="list-style-type: none"> <li>▪ verify that started the correct Subscription Handler (each segment has its own Subscription Handler)</li> <li>▪ in case you upgrade 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</li> <li>▪ check the openCRX log files to find out whether bad/corrupt data might be causing problems (e.g. NullPointerException during Workflow execution)</li> </ul> |
| I receive Alerts on my Subscriptions but no Notification E-mails                  | <ul style="list-style-type: none"> <li>▪ verify that JavaMail and opencrx-ssl.jar are properly installed</li> <li>▪ verify your e-mail settings (see E-mail Services for details)</li> <li>▪ verify that the Servlet WorkflowHandler of the respective segment is turned on</li> </ul>  |

## 8 E-mail Services

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

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

The following figure shows the potential data flows between openCRX, mail server, and mail client:

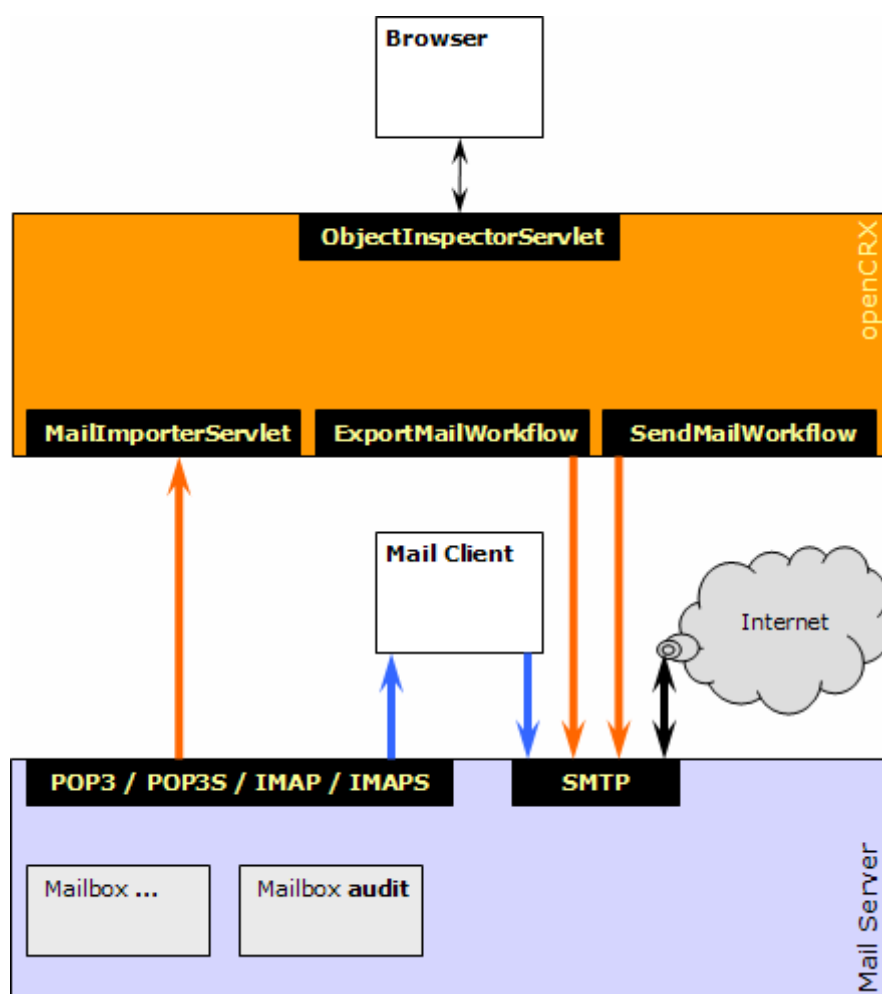


Figure 27: Flow of data between openCRX, mail server and mail client

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

## 8.1 Import E-mails from a Mail Client into openCRX

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

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

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

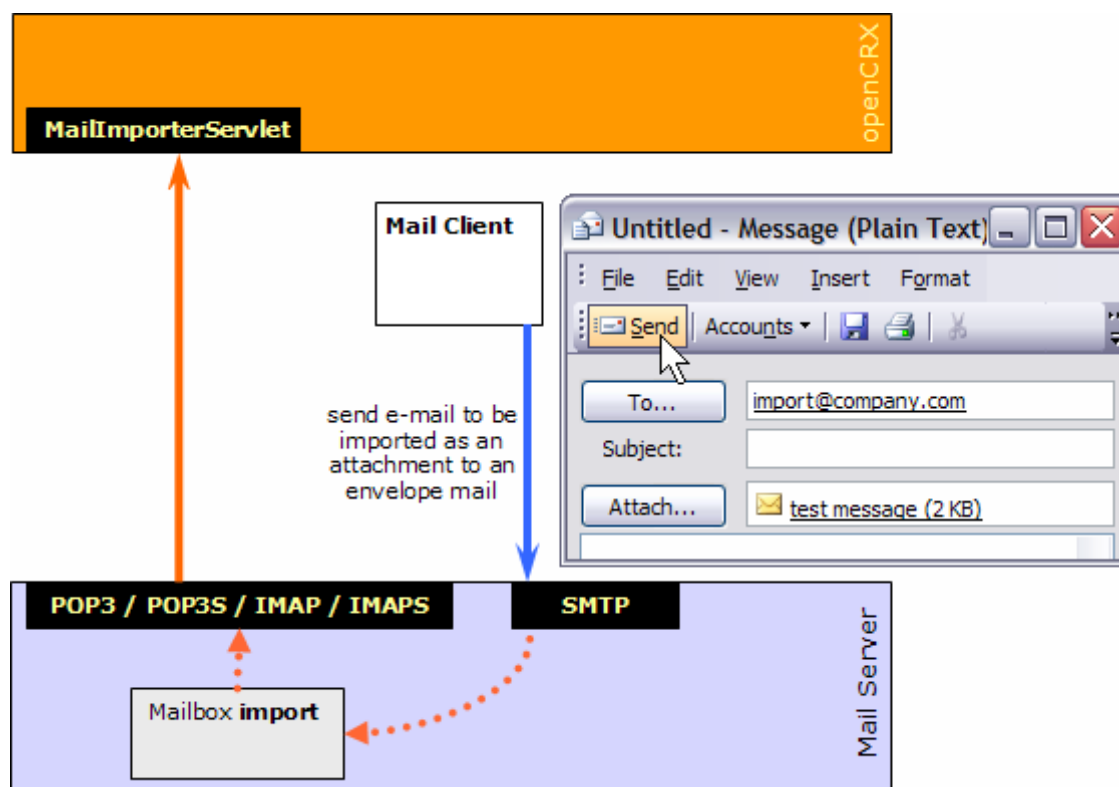


Figure 28: Import E-Mails from Mail Client

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

to be sent to your importer mailbox, e.g. by entering `import@company.com` into the **TO** field of the new message. Optionally you can specify the name of the Activity Creator in the **Subject** of the new message. Next you **attach** the message(s) to be imported to that new message (yes, you can attach multiple e-mail messages and if those messages contain attachments themselves they will also be imported) and send it off. Once delivered to the appropriate mailbox (called "import" in our example) the `MailImporterServlet` will fetch it from there and then import the messages attached to that envelope message.

This process works for messages in your Inbox, Outbox or any other folder.

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

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

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

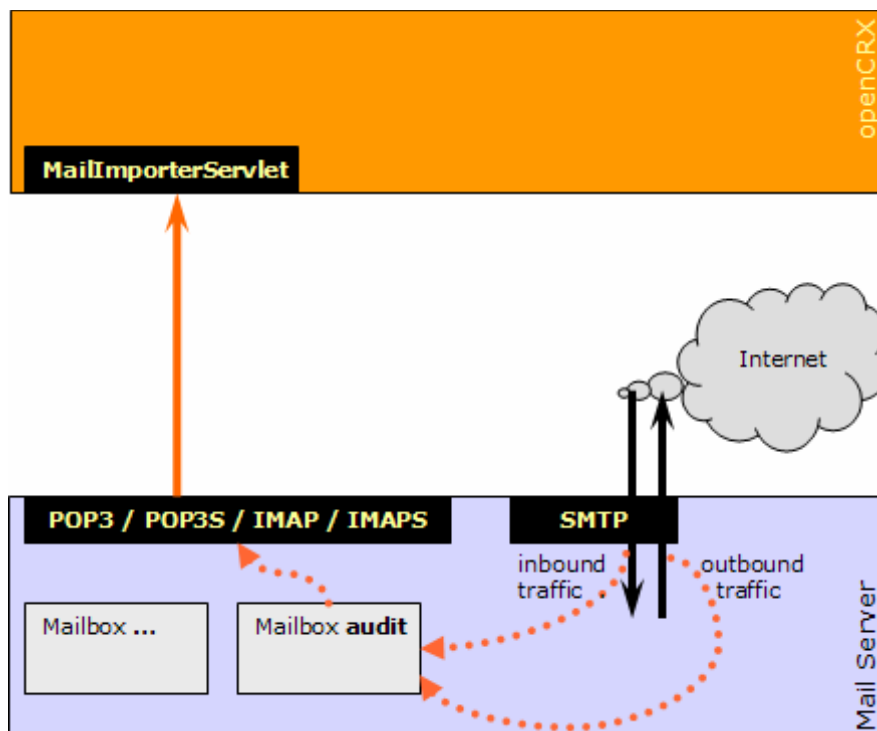


Figure 29: E-Mail Audit – import all inbound/outbound e-mail messages

### 8.3 Send E-mail directly from openCRX

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

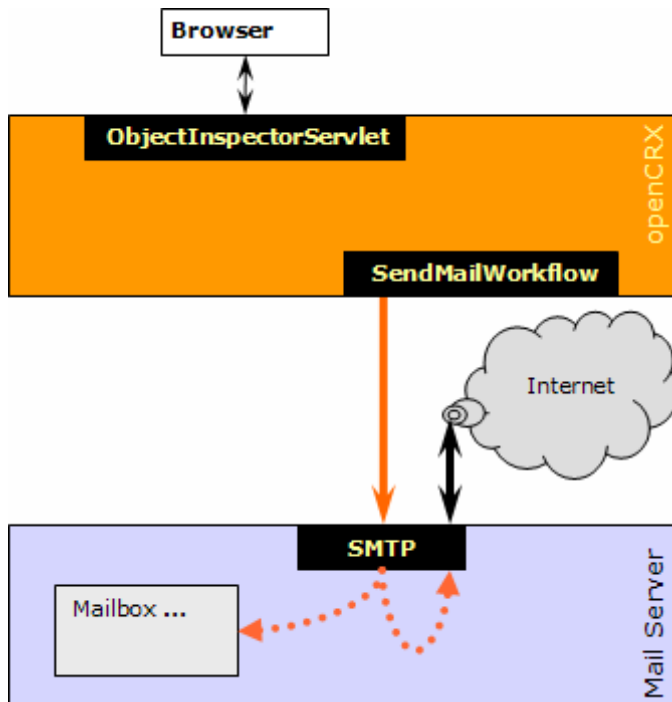


Figure 30: Send E-Mail from openCRX – Overview

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:

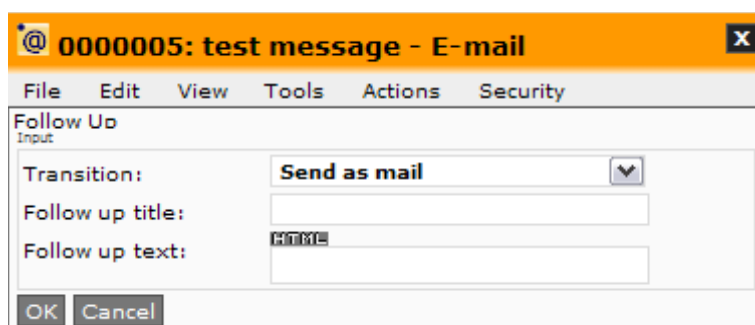


Figure 31: Send E-Mail from openCRX with Actions > Follow Up



Important

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



Tip

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

## 8.4 Export E-mails from openCRX to your Mail Client

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

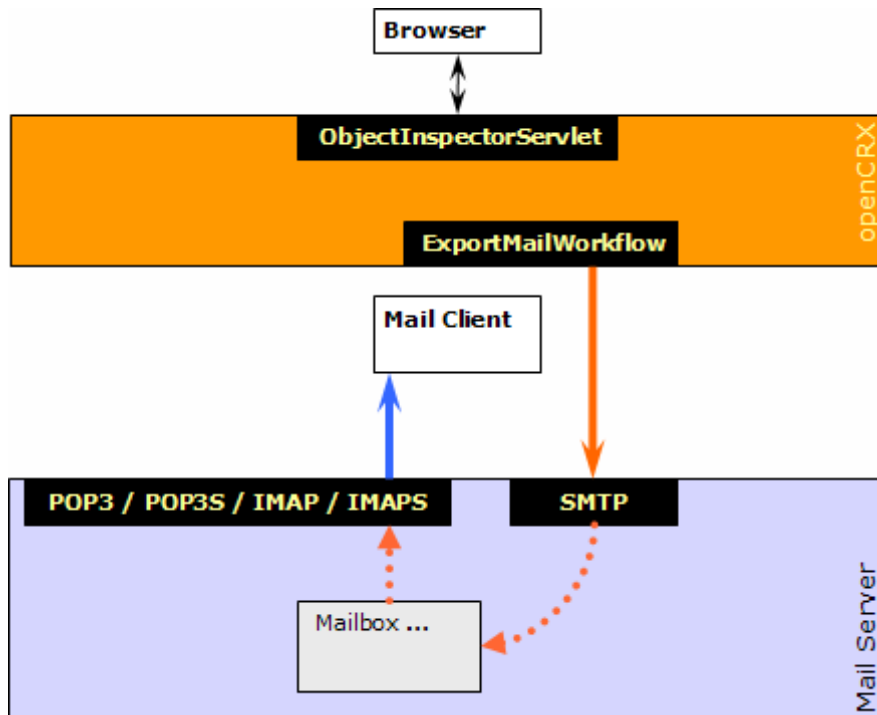


Figure 32: Export E-Mail from openCRX – Overview

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

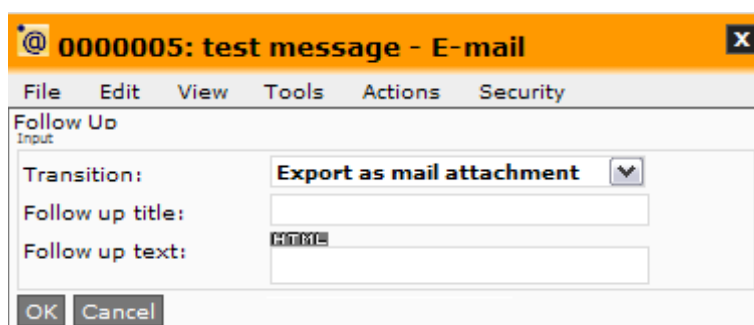


Figure 33: Export E-Mail from openCRX with Actions > Follow Up



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



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



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

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

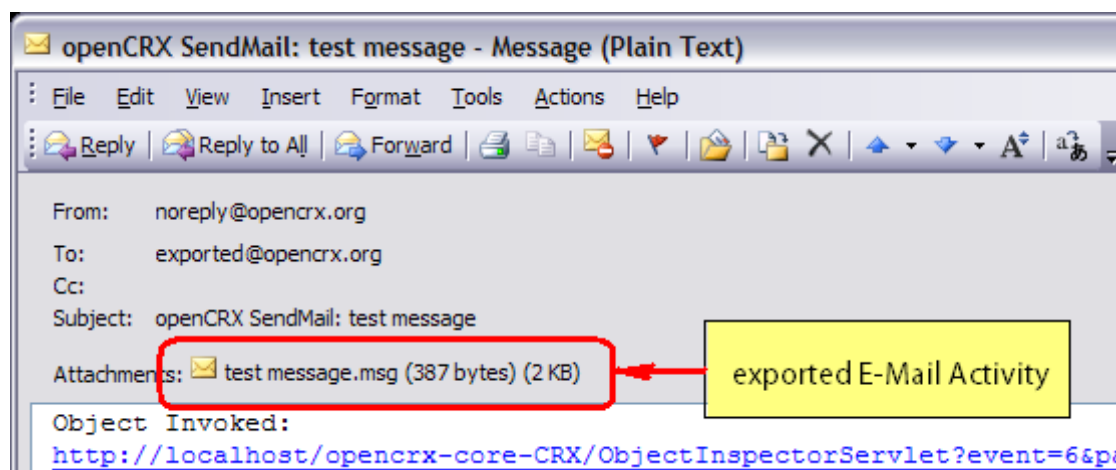


Figure 34: Envelope E-mail with exported E-Mail Activity as attachment

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

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

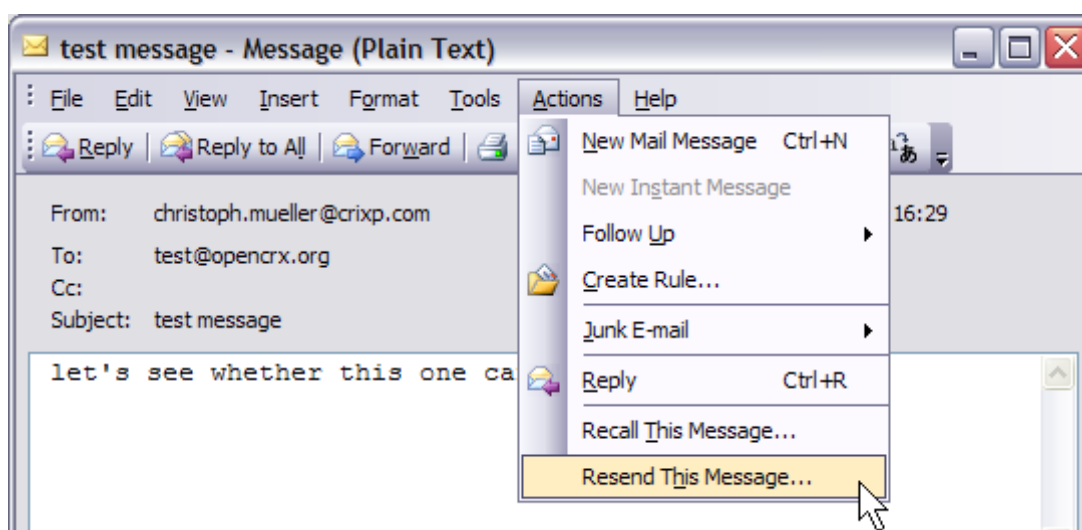


Figure 35: MS Outlook – Resend This Message to edit/send e-mail

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

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

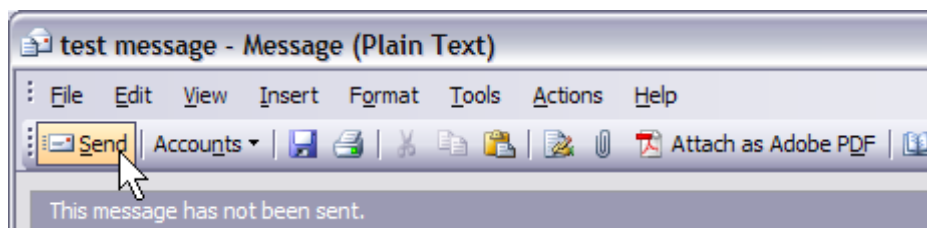


Figure 36: MS Outlook – Send message

## 8.5 Installing JavaMail and opencrx-ssl.jar

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 1.3.3) 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 1.0.2 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`

In addition to JavaMail you also must install **opencrx-ssl.jar** – included in the openCRX distribution – into the directory `JAVA_HOME\jre\lib\ext`.



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

## 8.6 Configuring E-mail Services

After Installing JavaMail and opencrx-ssl.jar you should verify that the options of the Java VM running openCRX include the following one:

`-Dmail.SSLSocketFactory.class=org.opencrx.ssl.DefaultSSLSocketFactory`



If you followed our application server installation guides, this option should already be included.



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

The following chapters explain how to configure various in- and outbound E-mail Services.

## 8.6.1 Outbound E-mail / Users

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:

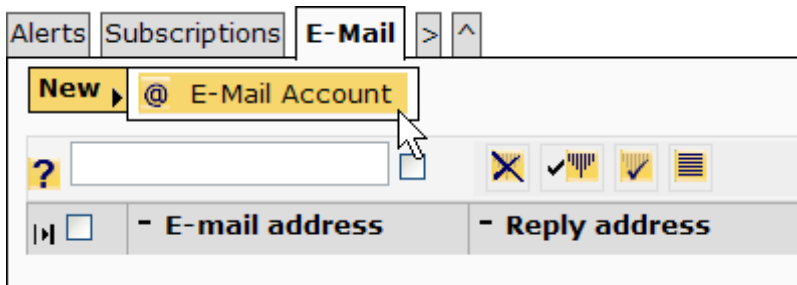


Figure 37: Create a new E-Mail Account – step 1

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

**Untitled - E-Mail Account**

**General** System

User Information

E-mail address: **user@mycompany.com** Default:

Reply address: **noreply@mycompany.com**

Outgoing Mail Server (SMTP)

Mail server (SMTP): **smtp.mycompany.com** Mail server port: **25**

User name: **guest** Password: **\*\*\*\*\***

Authentication required  SSL/TLS:

Qualifier: (must be unique, i.e. leave unchanged unless you know what you are doing...)  
**99IqcMQEEdqQW4VT5Pz34w**

Figure 38: Create a new E-Mail Account – step 2

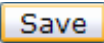
The various fields have the following meanings:

- **E-mail address:** enter your e-mail address (i.e. the address where you would like to receive e-mail notifications)
- **Reply address:** enter a reply address (is also used for the From field)
- **Default:** check if this is your default e-mail address (notifications will only be sent to your default e-mail address)

- **Mail server (SMTP):** openCRX will connect to this SMTP server to submit outbound e-mail messages
- **Mail server port:** the default port for SMTP is 25, but you can change the port if your mail server is listening on a different port
- **User name:** enter the user name to be used for authentication purposes (only used if *Authentication required* is checked)
- **Password:** enter the password to be used for authentication purposes (only used if *Authentication required* is checked)
- **Authentication required:** if checked, openCRX will authenticate with User name and Password to get access to the mail server
- **SSL/TLS:** check this option if the outgoing mail server supports TLS and you want to encrypt the communication between openCRX and the mail server



See also "openCRX does not initiate TLS session with mail server" for hints on trouble shooting TLS/SSL.

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

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

Figure 39: Create a new E-Mail Account – step 3

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

**guest, - My Homepage**

General | More Charts | **Options** | System ∞

E-mail Options

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

Figure 40: 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

## 8.6.2 Outbound E-mail / Users relying on SYSTEM Account

If the Segment Administrator (e.g. **admin-Standard**) configured a SYSTEM account – see Outbound E-mail (SYSTEM) – users do not need to provide any SMTP-related information; it is sufficient to provide values for the fields **E-mail address** and **Reply address**. Furthermore, you still need to check **Default** if you want openCRX to use this account:

**Untitled - E-Mail Account**

| General   |                              | System            | ⊗                                   |
|---|------------------------------|-------------------|-------------------------------------|
| User Information  |                              |                   |                                     |
| E-mail address:   | <b>user@mycompany.com</b>    | Default:          | <input checked="" type="checkbox"/> |
| Reply address:  | <b>noreply@mycompany.com</b> |                   |                                     |
| Outgoing Mail Server (SMTP)   |                              |                   |                                     |
| Mail server (SMTP):   | <input type="text"/>         | Mail server port: | <input type="text"/>                |
| User name:  | <input type="text"/>         | Password:         | <input type="text"/>                |
| Authentication required   | <input type="checkbox"/>     | SSL/TLS:          | <input type="checkbox"/>            |
| Qualifier: (must be unique, i.e. leave unchanged unless you know what you are doing...) |                              |                   |                                     |
| <b>99IqcMQEEdqQW4VT5Pz34w</b>   |                              |                   |                                     |
| <input type="button" value="Save"/> <input type="button" value="Cancel"/>               |                              |                   |                                     |

Figure 41: Create a new E-Mail Account – use SMTP-info from SYSTEM

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

- create a new account (e.g. a new contact)
- navigate to your Homepage and check whether you actually received an alert related to the newly created account
- next click on the Grid Tab **Workflow Process Instances** 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 Entry object
- the grid **Action Log Entries** contains either the message body of the e-mail that was sent or an error message if the workflow failed (please note that it is not unusual to see a "timeout" error message even if you actually received an e-mail; this is typically caused by an e-mail server with high latency – try sending out notifications through an e-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"/> |  |               |  |
| ?   | - Name                                   | - Correlation | Description  |
| <input type="checkbox"/>  | openCRX SendMail:<br>test 2              |               | Object Invoked:<br><a href="http://localhost/opencrx-core-CRX/ObjectInspector">http://localhost/opencrx-core-CRX/ObjectInspector</a><br><br>By Workflow:<br><a href="http://localhost/opencrx-core-CRX/ObjectInspector">http://localhost/opencrx-core-CRX/ObjectInspector</a><br><br>Triggered By Event:<br>OBJECT_CREATION<br><br>Triggered By Subscription:<br>mxxecMQBEdqQW4VT5Pz34w<br><a href="http://localhost/opencrx-core-CRX/ObjectInspector">http://localhost/opencrx-core-CRX/ObjectInspector</a> |
| <input type="checkbox"/>  | Can not send mail:<br>MessagingException |               | Can't send command to SMTP host;<br>nested exception is:<br>java.net.SocketException: Software caused connec   |
| <input type="checkbox"/>  | Can not send mail:<br>MessagingException |               | Could not connect to SMTP host: mail.crixp.com, pc   |

Figure 42: Example of outbound E-mail Action Log Entries

### 8.6.3 Outbound E-mail (SYSTEM) / Segment Administrator

The Segment Administrator (e.g. **admin-Standard**) can define an outbound E-mail that can be used by all users of a segment without sharing any critical information (e.g. username/password required to authenticate for relaying e-mails, etc.). This is done as follows:

- Login as Segment Administrator (e.g. **admin-Standard**)
- 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:

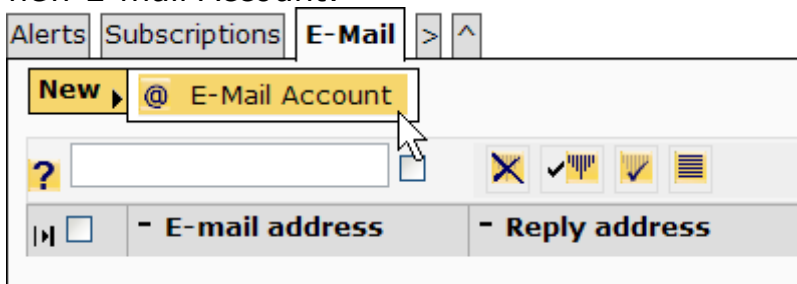


Figure 43: Create a new SYSTEM E-Mail Account – step 1

- Now you can configure your E-Mail Account for outbound e-mail service just as you would as a normal user. The only difference is that you change the Qualifier to **SYSTEM** as shown below:

**Untitled - E-Mail Account**

General System \*

User Information

E-mail address: admin@company.com Default:

Reply address: noreply@company.com

Outgoing Mail Server (SMTP)

Mail server (SMTP): smtp.company.com Mail server port: 25

User name: admin Password: \*\*\*\*\*

Authentication required:  SSL/TLS:

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

**SYSTEM**

Save Cancel

Figure 44: Create a new SYSTEM E-Mail Account – step 2

- Click the button **Save** to commit the new SYSTEM E-Mail Account.

From now on Users can define their outbound e-mail settings with a minimal set of information (see Users relying on SYSTEM Account for details).

## 8.6.4 Inbound E-mail / Segment Administrator

The Root Administrator (admin-Root) must turn on the MailImporterServlet (see Workflow Controller for information on how this is done). Segment Administrators (e.g. admin-Standard) can now configure the openCRX Mail Importer for their segment as follows:

- As Segment Administrator (e.g. admin-Standard) navigate to **Workflows > Workflow Processes**.
- Click on the icon of the MailImporterServlet to load it into the Inspector:

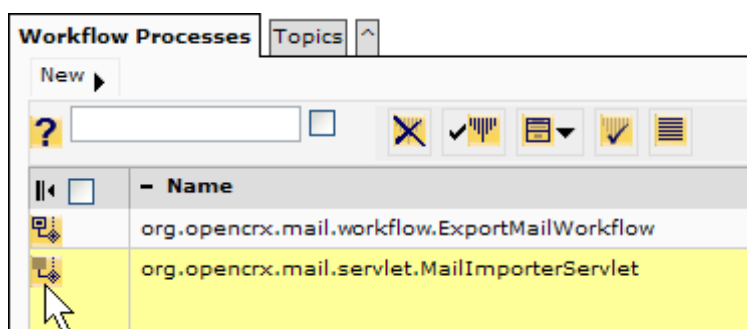


Figure 45: Grid Workflow Processes with entry MailImporterServlet

- The configuration of the MailImporterServlet is stored in properties; the following figure shows the default configuration:

The screenshot shows the configuration page for the `org.opencrx.mail.servlet.MailImporterServlet` external task. The page has a menu bar with 'File', 'Edit', 'View', 'Tools', and 'Security'. Below the menu bar, there is a breadcrumb trail: 'Workflows > org.opencrx.mail.servlet.MailImporterServlet'. The main content area is divided into two tabs: 'General' and 'System'. The 'General' tab is active, showing the following details:

- Name:** `org.opencrx.mail.servlet.MailImporterServlet`
- Priority:** (empty)
- Description:** Mail importer. To override the default property create a new property without s



Below the general information is the 'Properties' section, which includes a 'New' button and a toolbar with icons for help, search, delete, clone, and list. A table displays the default properties:

|    | - Name                         | - Description            | - String value    |
|----|--------------------------------|--------------------------|-------------------|
| 12 | port.Default                   | Server port number       |                   |
| ab | server.Default                 | Incoming mail server     | mail.changeme.com |
| ab | account.Default                | Account name             | account           |
| ab | password.Default               | Account password         | password          |
| 01 | deleteImportedMessages.Default | Delete imported messages |                   |
| ab | protocol.Default               | Mail protocol            | pop3              |
| ab | mailbox.Default                | Mailbox name             | INBOX             |
| 01 | sslRequired.Default            | Server requires SSL      |                   |
| 01 | debug.Default                  | Enable debugging output  |                   |

Figure 46: MailImporterServlet Default Settings

- As the MailImporterServlet is supposed to connect to a POP3 or IMAP account you typically need to adapt to default configuration. This can easily be done by copying an existing property and then removing the suffix **".Default"** from the name and entering the appropriate value.

To connect to a mail server with the name **"mail.company.com"**, for example, you would follow the following steps:

1. Click on the button  Edit to put the grid into edit mode.
2. Click on the clone button  to create a clone of the property with the name **server.Default**:

org.opencrx.mail.servlet.MailImporterServlet - External Task

File Edit View Tools Security

Workflows > org.opencrx.mail.servlet.MailImporterServlet

General System \*

Name: org.opencrx.mail.servlet.MailImporterServlet Priority:

Description: Mail importer. To override the default property create a new property without suffix .De

Properties > ^

New ▶

|      | Name            | Description          | String value      | URI value |
|------|-----------------|----------------------|-------------------|-----------|
| 12 = | port.Default    | Server port number   |                   |           |
| ab = | server.Default  | Incoming mail server | mail.changeme.com |           |
| ab = | account.Default | Account name         | account           |           |

Figure 47: Configuration of MailImporterServlet – Default Server

3. Still in edit mode you change the name of the cloned property from **server.Default** to **server** and the String value **mail.changeme.com** to **mail.company.com**:

|      | Name                | Description          | String value      |
|------|---------------------|----------------------|-------------------|
| 12 = | port.Default        | Server port number   |                   |
| ab = | server.Default      | Incoming mail server | mail.changeme.com |
| ab = | account.Default     | Account name         | account           |
| ab = | password.Default    | Account password     | password          |
| 01 = | deleteImportedMes:  | Delete imported me   |                   |
| ab = | protocol.Default    | Mail protocol        | pop3              |
| ab = | mailbox.Default     | Mailbox name         | INBOX             |
| 01 = | sslRequired.Default | Server requires SSL  |                   |
| 01 = | debug.Default       | Enable debugging o   |                   |
| ab = | server              | Incoming mail server | mail.company.com  |

Save Cancel

Figure 48: Configuration of MailImporterServlet – Mail Server

4. Click on the button **Save** to commit the changes.

- Similarly, you clone new properties named **account** and **password**.



You can clone/edit multiple rows without committing each change individually, i.e. you can first make all your changes and then commit the new grid in the end.

- Depending on your e-mail account settings you might have to adapt other properties as well, but **server**, **account**, and **password** must definitely be adapted. Assuming you want to connect to the mail server as account "**import**" with password "**importSecret**" your grid **Properties** should look as follows:

|    | - Name                         | - Description            | - String value    |
|----|--------------------------------|--------------------------|-------------------|
| ab | server                         | Incoming mail server     | mail.company.com  |
| ab | account                        | Account name             | import            |
| ab | password                       | Account password         | importSecret      |
| 12 | port.Default                   | Server port number       |                   |
| ab | server.Default                 | Incoming mail server     | mail.changeme.com |
| ab | account.Default                | Account name             | account           |
| ab | password.Default               | Account password         | password          |
| 01 | deleteImportedMessages.Default | Delete imported messages |                   |
| ab | protocol.Default               | Mail protocol            | pop3              |
| ab | mailbox.Default                | Mailbox name             | INBOX             |
| 01 | sslRequired.Default            | Server requires SSL      |                   |
| 01 | debug.Default                  | Enable debugging output  |                   |

Figure 49: Configuration of MailImporterServlet – Account and Password



There is no need to delete default entries as they are ignored as soon there exists an entry with the same name without the suffix **".Default"**.

Example: Because there exists an entry with name **"account"** the entry with name **"account.Default"** is ignored.



JavaMail currently supports the following values for protocol: **pop3** and **imap**. Both protocols are also supported over secure connections (SSL).

- Test your settings by sending an e-mail to the account specified in the steps above (import@company.com). Attach the message to be imported to this “envelope” e-mail (please note that the attached e-mail only is imported by the MailImporterServlet):

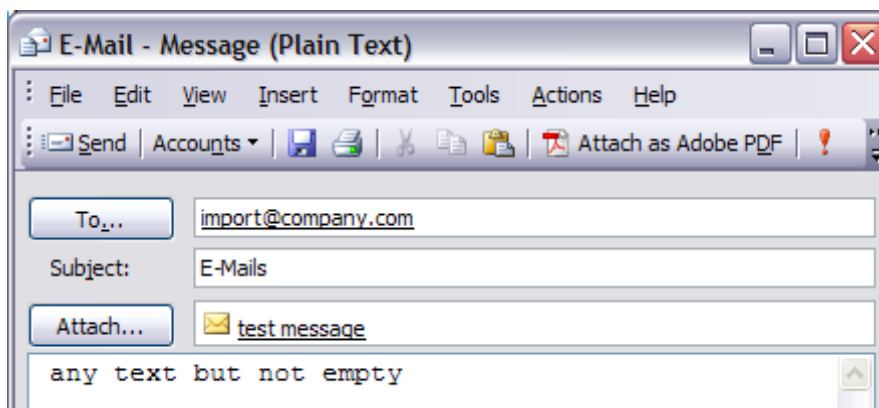


Figure 50: Envelope E-Mail with attached E-Mail to be imported



Please note that neither Subject nor message body should be empty. Use the subject to pass the name of the Activity Tracker you want the imported e-mail to be assigned to (by default imported e-mails are assigned to the Activity Tracker E-Mails).

- Once the Workflow Controller triggers the MailImporterServlet you will see the debug output of the servlet on the console (on JBoss also in the file **server.log**):

#### Listing 8: Debug Output of MailImporterServlet

```

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

```

```

18:42:58,701 INFO [STDOUT] S: +OK Logged in.
18:42:58,717 INFO [STDOUT] C: STAT
18:42:58,748 INFO [STDOUT] S: +OK 1 3204
18:42:58,748 INFO [STDOUT] C: NOOP
18:42:58,779 INFO [STDOUT] S: +OK
18:42:58,842 INFO [STDOUT] C: TOP 1 0
18:42:58,889 INFO [STDOUT] S: +OK
...
    
```

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

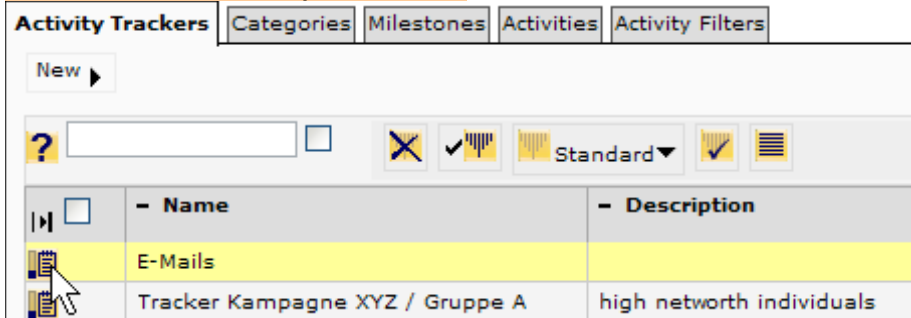


Figure 51: Activity Tracker E-Mail is created automatically

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

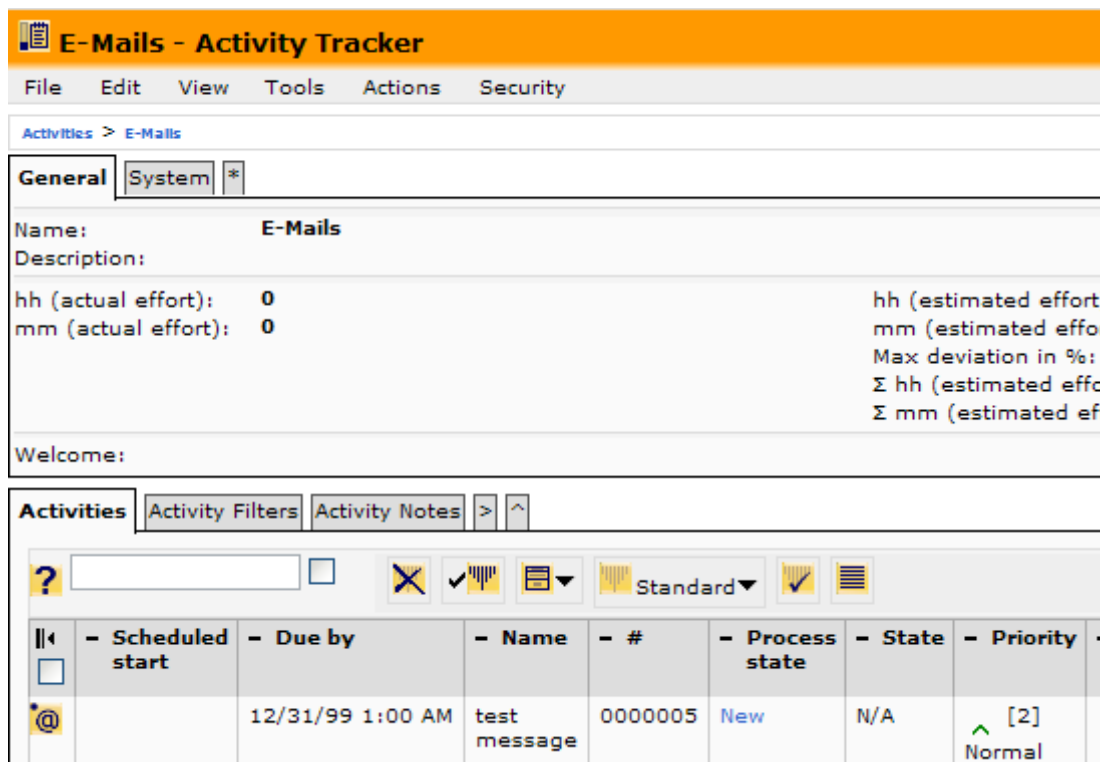


Figure 52: Activity Tracker E-Mail with newly imported e-mail

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

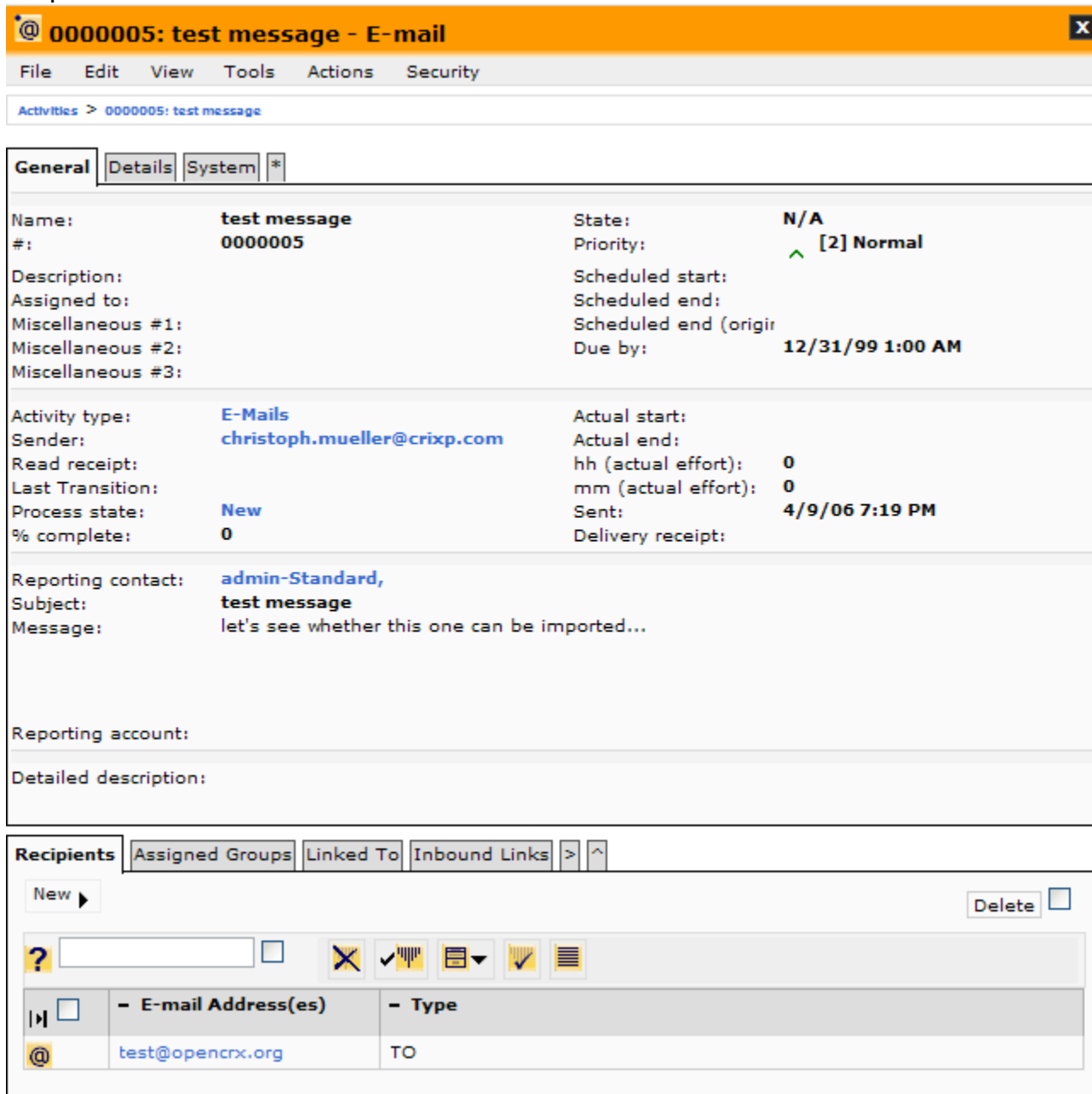


Figure 53: Newly imported e-mail

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

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



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

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

File Edit View Tools Actions Security

Activities > Default E-Mail Creator

**General** Details System \*

Name: **Default E-Mail Creator**

Activity type: **E-Mails**

Description:

**Activity Groups** > ^

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

|  | - Name  | - Scheduled date | - Desc |
|--|---------|------------------|--------|
|  | E-Mails |                  |        |

Figure 54: Activity Creator Default E-mail Creator

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


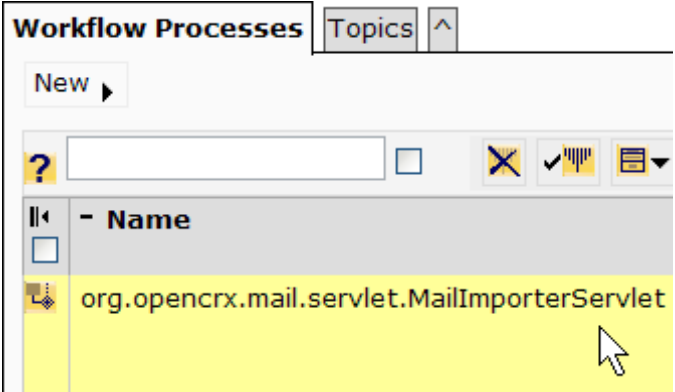
You can easily change the contents of the grid Activity Groups so that newly imported e-mails will be attached to different Activity Tracker(s). It is also possible to create additional Activity Creators with different behavior (just make sure that these Activity Creators create Activities of type **E-Mails**).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




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

## 8.6.5 Trouble Shooting E-mail Services

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

| Problem   | Solution   |
|---|--|
| the grids <b>Workflow Processes</b> and/or <b>Topics</b> are empty                              | <ul style="list-style-type: none"> <li>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)</li> <li>click on the filter button  to see all rows without filtering (maybe you defined a default filter in the past?)</li> </ul>   |
| the grid <b>Workflow Processes</b> does not contain an entry for the <b>MailImporterServlet</b> | <ul style="list-style-type: none"> <li>verify that the MailImporterServlet was started at least once (the Workflow Process "or.opencrx.mail.servlet.MailImporterServlet" is created automatically by the servlet MailImporterServlet if the Workflow Process is not present)</li> </ul>  <p>The screenshot shows a web interface with two tabs: 'Workflow Processes' and 'Topics'. Below the tabs is a 'New' button and a search bar. A table below has a header 'Name' and one row containing 'org.opencrx.mail.servlet.MailImporterServlet', which is highlighted in yellow. A mouse cursor is pointing at the entry.</p> <p><i>Figure 55: MailImporterServlet Entry</i></p> <ul style="list-style-type: none"> <li>depending on your segment access levels this entry might be visible to the Segment Administrator only (e.g. <b>admin-Standard</b>); see also <i>Figure 7: Table kernel_Segment after modification</i></li> </ul> |
| 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</p>   |

| Problem  | Solution  |
|--|---|
|  | <p>messages such failure to establish a TLS session will prevent openCRX from properly sending outbound mail.</p> <div data-bbox="742 432 847 517">  </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 9: Importing Certificate</i></p> <pre data-bbox="742 719 1474 768">cd \$JAVA_HOME/lib/security keytool -import -alias &lt;dom&gt; -file &lt;name&gt;.cer -keystore cacerts</pre> <p>replace <b>&lt;dom&gt;</b> with the domain of the mail server (e.g. mail.company.com) and <b>&lt;name&gt;</b> with the name of the certificate file</p> |
| I receive Alerts on my Subscriptions but no Notification E-mails | <ul style="list-style-type: none"> <li>▪ verify that JavaMail and opencrx-ssl.jar are properly installed</li> <li>▪ verify your e-mail settings (see E-mail Services for details); if you use the SYSTEM account configured by the Segment Administrator you might want to verify the settings of that SYSTEM account</li> <li>▪ verify that the Servlet WorkflowHandler of the respective segment is turned on</li> </ul>  |

## 9 Data Import/Export

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

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

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



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

### 9.1 Importing Data into openCRX

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



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 and iCalendar files in addition to the XML importer.

### 9.2 Importing XML Files

You can import virtually any data into openCRX as long as you provide it in the form of schema-compliant XML files. The openCRX schema files can be found in the file `opencrx-1.9.1\jre-1.4\core\lib\opencrx-kernel.jar` (unzip and look for xmi subdirectories). Alternatively, you can export example objects as XML files and look at the produced XML files.

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

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

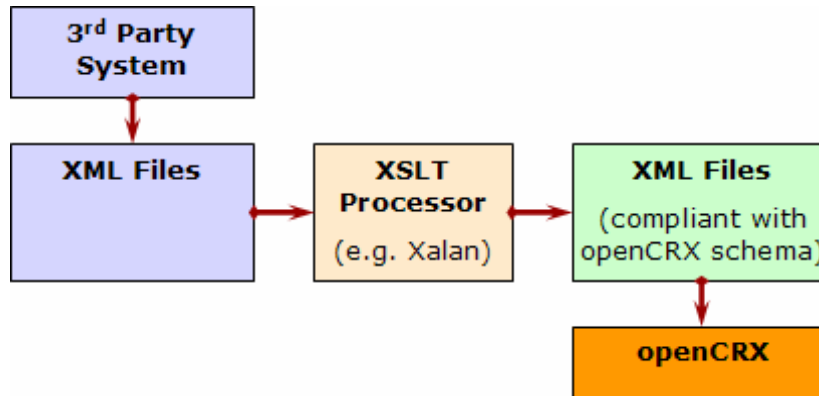


Figure 56: XML import from 3<sup>rd</sup> party system – overview

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:

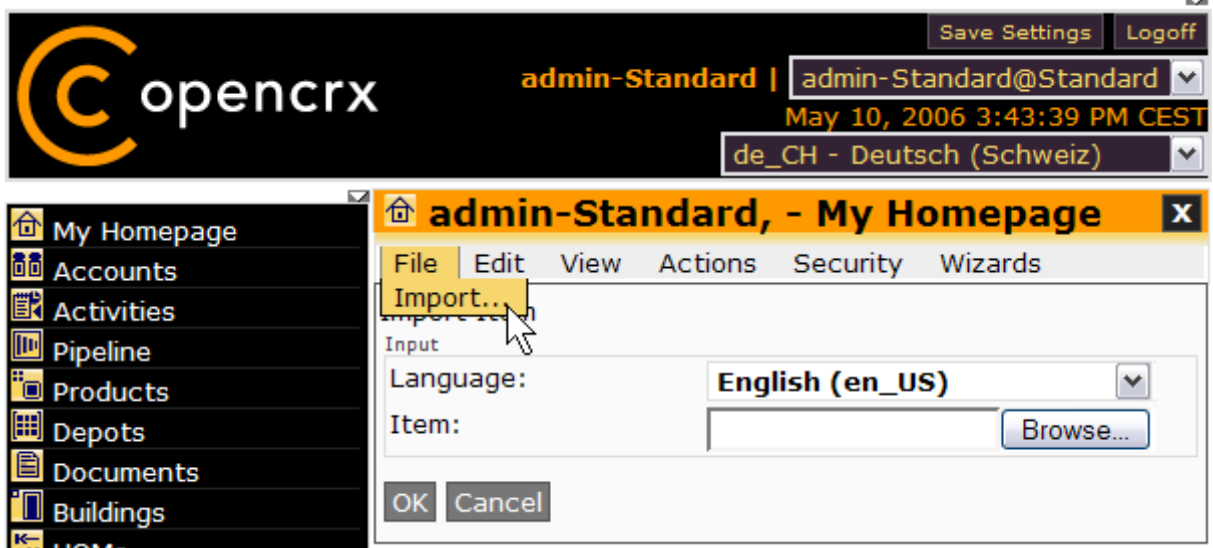


Figure 57: Interactive import of XML Files

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

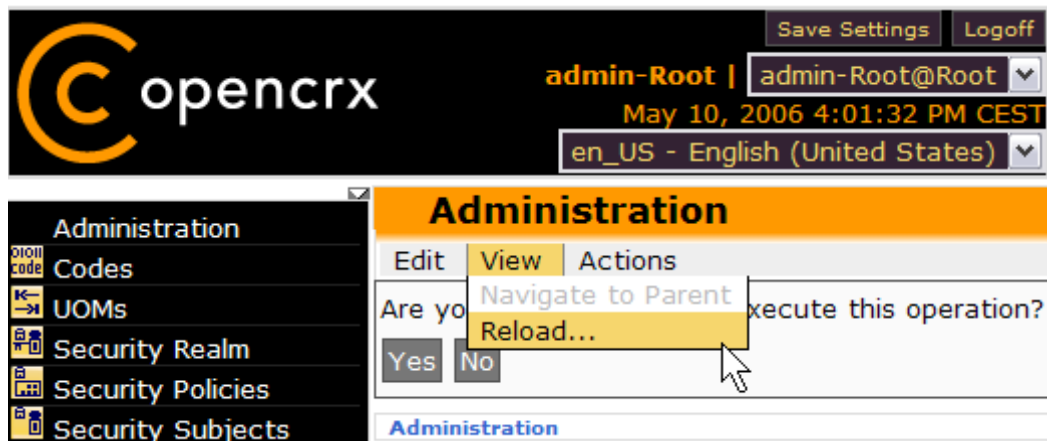
- **Bulk Import**

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

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

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

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



- *Figure 58: 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.

## 9.2.1 Importing vCard Files (→ openCRX Contacts)

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

These are the steps to import a vCard file:

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

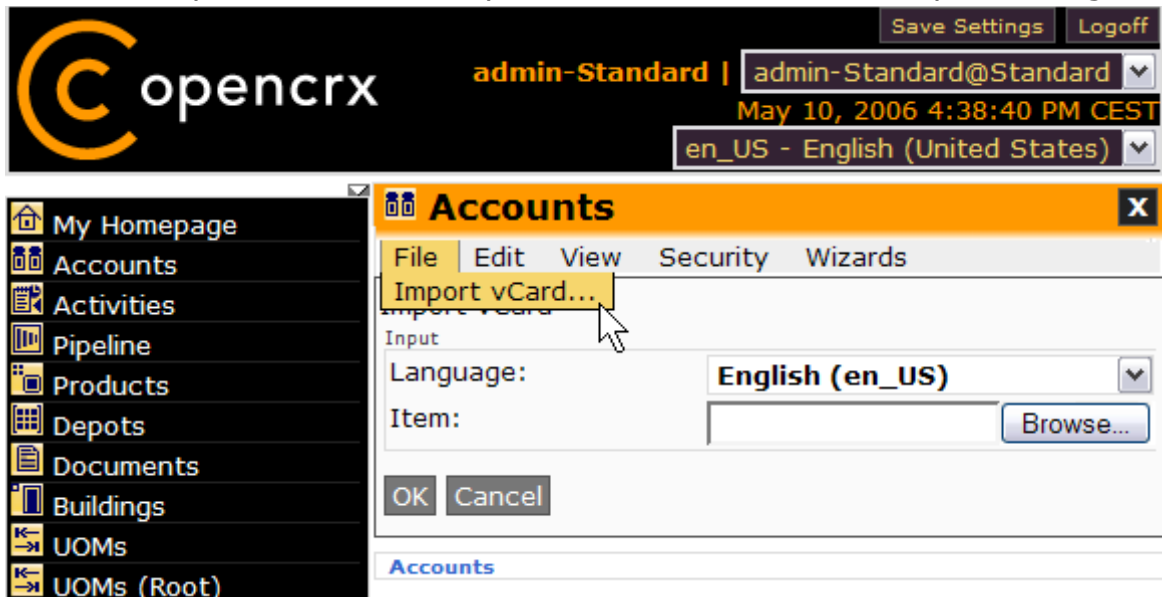


Figure 59: Operation vCard Import

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

## 9.2.2 Importing iCalendar Files (→ openCRX Meetings)

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

These are the steps to import an iCalendar file:

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

### 9.2.3 Other Options

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

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

## 9.3 Exporting Data from openCRX

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

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

### 9.3.1 Exporting XML Files

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

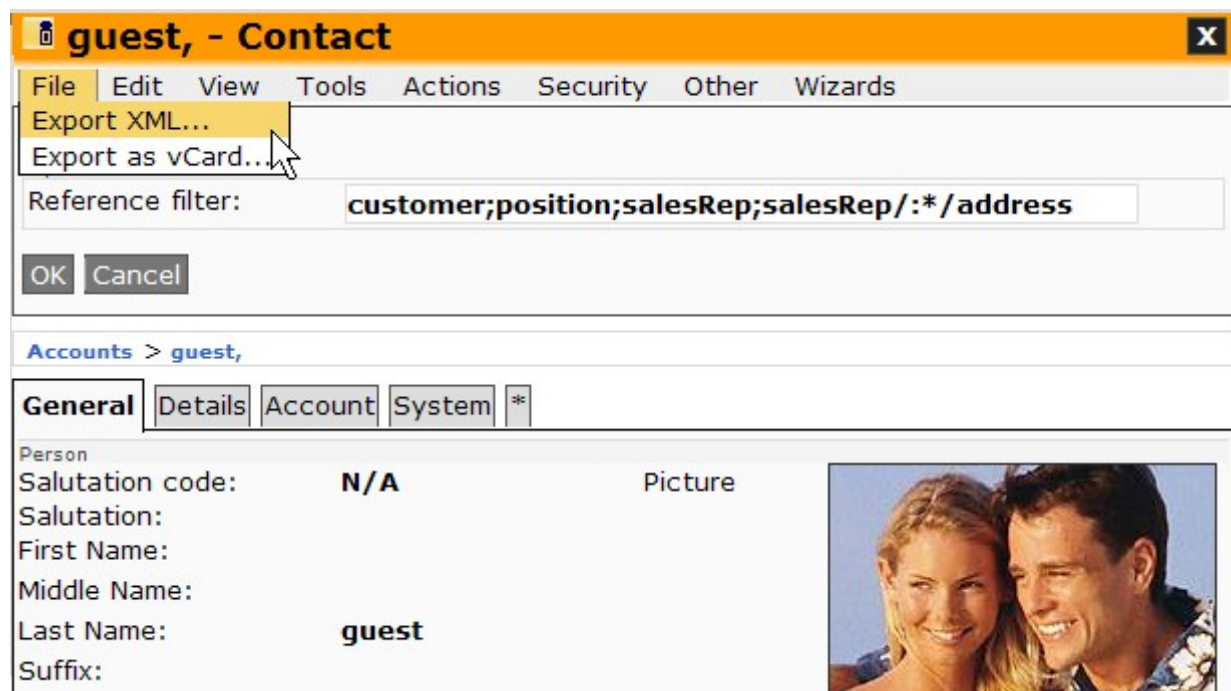


Figure 60: Exporting Contact as XML File

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

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

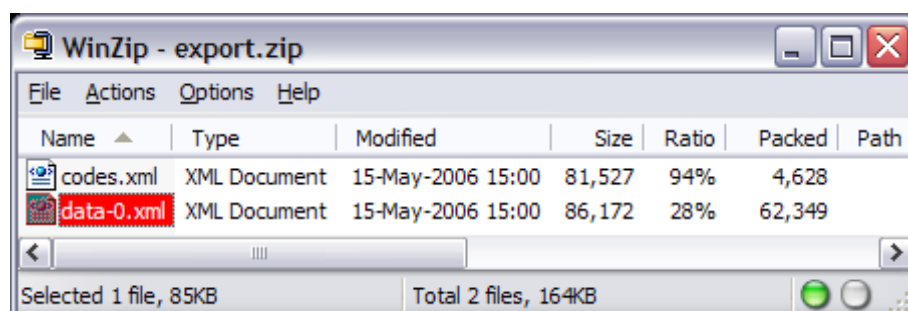


Figure 61: XML Exporter provides XML data file and code tables as ZIP file

### 9.3.2 Exporting openCRX Contacts (→ vCard Files)

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

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

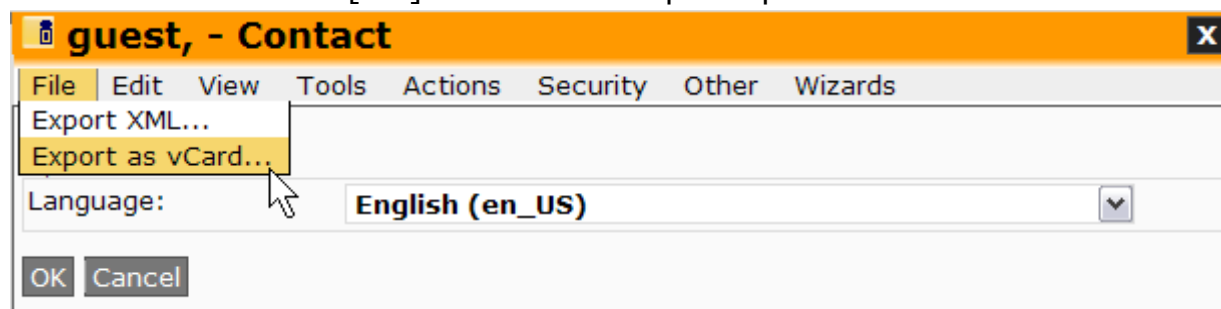


Figure 62: Export Contact as vCard

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

### 9.3.3 Exporting openCRX Contacts (→ Outlook Contacts)

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



Figure 63: Export Contact to MS Outlook



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

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

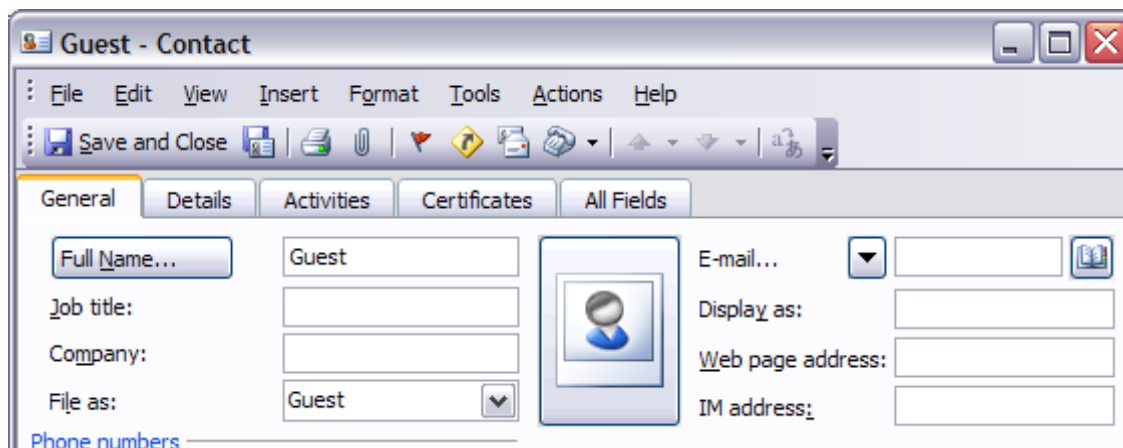


Figure 64: MS Outlook Contact created from openCRX Contact

An animation is available at

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

### 9.3.4 Exporting openCRX Meetings (→ iCalendar Files)

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

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



Figure 65: Exporting Meeting / Sales Visit as iCalendar File

### 9.3.5 Other Options

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

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

## 10 Customizing openCRX

### 10.1 Managing Locales

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

The locale list is contained in the file

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

Look for the section `<!-- locales -->` where you will find a list of all available locales:

*Listing 10: 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 11: Activating/Deactivating Locales in web.xml*

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



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

## **10.2 Managing Packages**

<documentation pending>

## **10.3 Role-based UI**

<documentation pending>

### **10.3.1 Model Permissions**

<documentation pending>

### **10.3.2 Multiple Applications**

<documentation pending>

### **10.3.3 Custom JSPs**

<documentation pending>

## 11 Reporting with BIRT

A clean integration of BIRT (Business Intelligence and Reporting Tools, available from <http://www.eclipse.org/birt/>) is on the openCRX roadmap, but there are still a few pieces missing. In the mean time, we offer **experimental support for BIRT** and the following notes should help you navigate around the current restrictions and pitfalls of using BIRT with openCRX.

### 11.1 Restrictions/Pitfalls of using BIRT with openCRX

openCRX is distributed with the **BIRT Report Engine**, i.e. it is possible to develop reports with the **BIRT Report Designer** (not distributed with openCRX, but available from the BIRT website) and then use such reports with openCRX.

Before we look at the steps required to create/deploy a report it is worthwhile noting some **restrictions and pitfalls inherent to the current setup**:

- **BIRT directly accesses the openCRX database, i.e. permissions and security provided by openCRX do not apply!**  
Example: if you create a report that lists all the leads, anybody who can execute this report will see all the leads, regardless of user-roles, permissions, etc.
- **BIRT directly access the openCRX database, i.e. data are not retrieved through the openCRX API.** The nasty consequence (in addition to the above-mentioned lack of security) is that each time the openCRX project decides to change the OO-to-relational mapping or the database structure it is quite likely that reports must be adapted to the new mapping/structure.
- **Data Source Definitions are located inside the BIRT report definition file.** In addition to defining DB URLs, user names and passwords in multiple places (redundancy!) it is also not possible to define platform neutral reports (if you change the DB, for example, you also must update the data source definitions in all the report definition files).
- **BIRT logs into a directory located inside the openCRX EAR.** If this directory does not exist or is not writable (e.g. because the EAR is not exploded) you can not rely on BIRT log files to track down problems (the BIRT Report Engine will still work, though).
- **BIRT support is experimental** and there is no guarantee that BIRT support will be available in future versions of openCRX. BIRT is (just like openCRX) a project that is under heavy development and even with the best intentions the future is difficult to predict.

## 11.2 How to design and deploy a BIRT report



Please note that we have tested BIRT v2.0.1 with openCRX v1.9.1. You might have to adapt the following instructions if you work with different versions.

- Download and install the BIRT Report Designer v2.0.1 (reports created with other versions might work with the report engine distributed with openCRX – BIRT Report Engine v2.0.1 – but you are really on your own if versions don't match).
- Follow the BIRT tutorials to create your report.
- During the report creation process you will have to define a **Data Source**. The following figure shows the data source definition for a local instance of MySQL with a schema crx-crx:

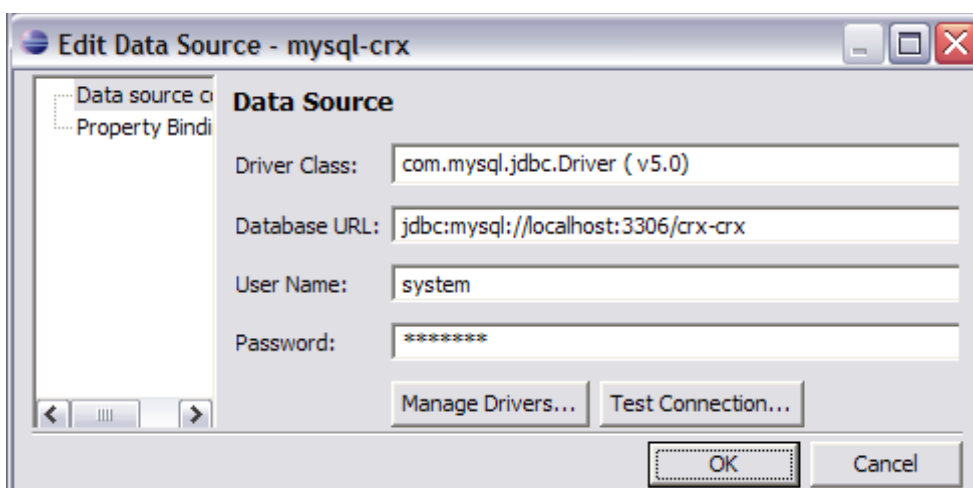


Figure 66: BIRT Report Designer – Data Source Definition mysql-crx



Please note that this data source definition is included in the BIRT report definition file (extension *rptdesign*). Hence, data source definitions must be adapted to your deployment environment (e.g. development vs. production).

### Listing 12: Sample Data Source Definition for MySQL

```
<data-sources>
  <oda-data-source extensionID="org.eclipse.birt.report.data.oda.jdbc" name="MySQL (localhost)" id="107">
    <property name="odaDriverClass">com.mysql.jdbc.Driver</property>
    <property name="odaURL">jdbc:mysql://localhost:3306/crx-CRX</property>
    <property name="odaUser">system</property>
    <encrypted-property name="odaPassword">bWFuYWdlcg==</encrypted-property>
  </oda-data-source>
</data-sources>
```



The openCRX distribution includes a sample report (located in the directory `core\src\data\org.opencrx\report\en_US`) with various default data source definitions that you might be able to use/adapt to your own environment.

- You will also have to define a **Data Set**, which contains a reference to your Data Source Definition and a **Query**. The following figure shows a simple Query to create an Activity Group report:

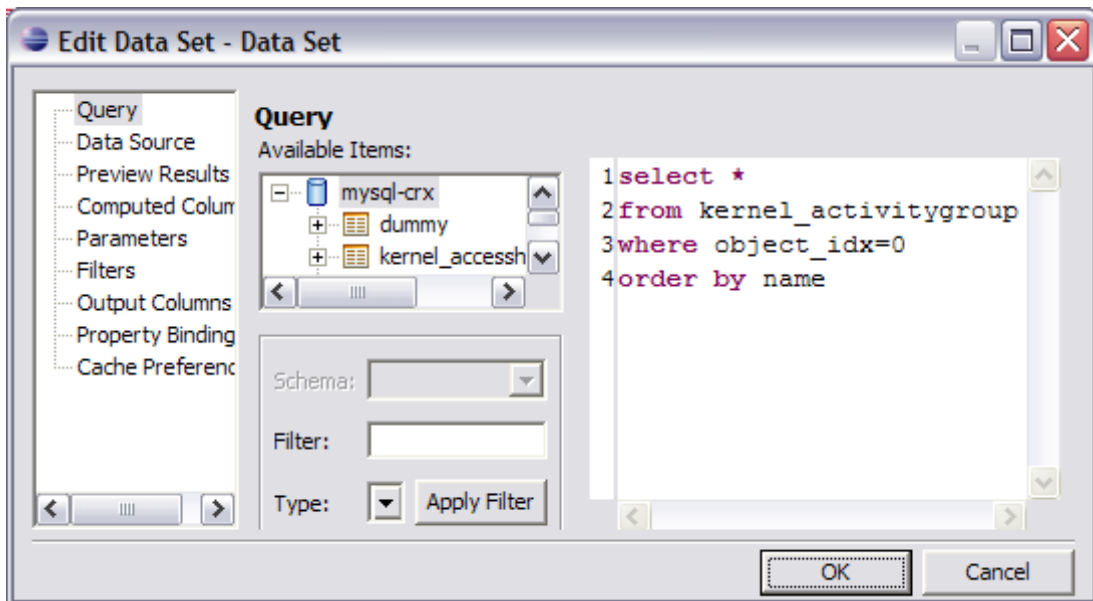


Figure 67: BIRT Report Designer – Data Set Query

The above will look as follows in the report definition file (note that you refer to the Data Source by its **name**):

Listing 13: Sample Data Set for Activity Group Report from MySQL

```
<data-sets>
  <oda-data-set extensionID="org.eclipse.birt.report.data.oda.jdbc.JdbcSelectDataSet" name="Data Set" id="5">
    <property name="dataSource">MySQL (localhost)</property>
    <property name="queryText">select *
from kernel_activitygroup
where object_idx=0
order by name</property>
  </oda-data-set>
</data-sets>
```

- You can also define an openCRX report title and a report description that will be used in the openCRX GUI as shown in the figure below:

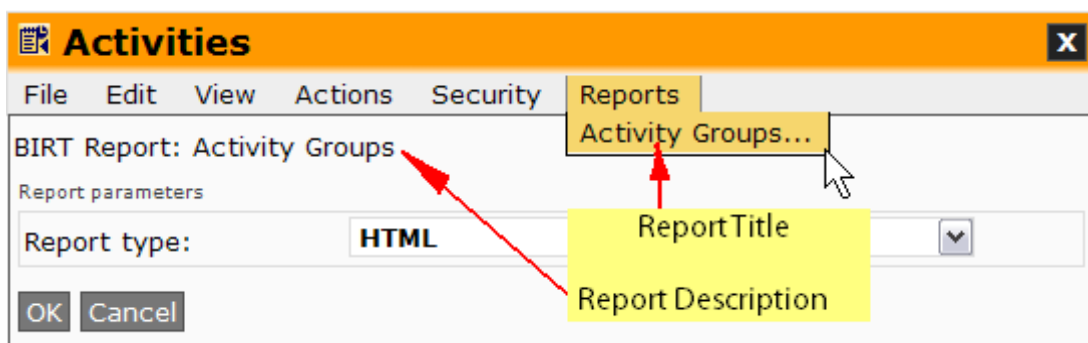


Figure 68: BIRT Report in openCRX GUI

openCRX report title and description are defined by the value of the BIRT report property **Description** as follows: `<title>-<description>`. The text in front of the first dash ("-") is interpreted as the **title of the openCRX report** (used as label for the menu entry), the remainder is interpreted as the **description of the openCRX report**.

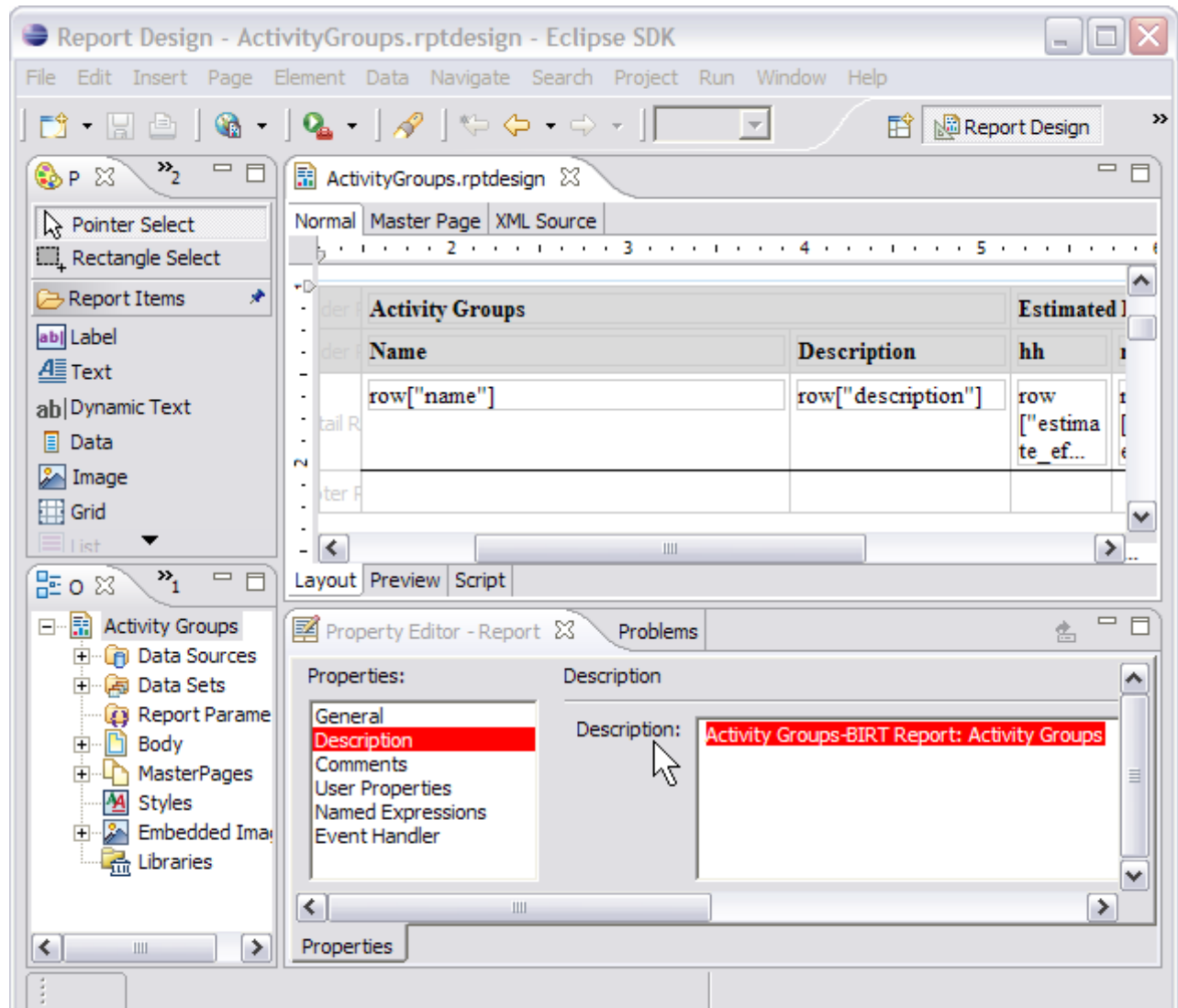


Figure 69: BIRT Report Designer – Title and Description of openCRX Report

The above will look as follows in the report definition file:

*Listing 14: openCRX Report Title and Description*

```
<html-property name="description">Activity Groups-BIRT Report: Activity Groups</html-property>
```

- You can test your reports with the preview features of the BIRT Report Designer. Once you have your report working as desired it is time to integrate it into the openCRX EAR in the directory `opencrx-core-CRX-web.ear\opencrx-core-CRX.war\WEB-INF\config\report\en_US`. Reports available in the locale directory `en_US` are available in all locales, but you can also provide localized reports by deploying them to the respective directories. For example, a report in French (locale `fr_FR`) would have to be deployed to the directory `opencrx-core-CRX-web.ear\opencrx-core-CRX.war\WEB-INF\config\report\fr_FR`.



For a report to be active, however, it must be present in the directory `...\report\en_US`.



The filename of the report determines the placement of the report in the openCRX menu structure. The part in front of the first dash (“-”) reflects the path, the remainder is not used for the purpose of menu placement. The sample report **Activity Groups** distributed with openCRX, for example, shows up in the menu **Reports** of the package `activity1`: `org.opencrx.kernel.activity1.Segment-ActivityGroups.rptdesign`

### 11.3 Report Debugging / BIRT Logging

Unfortunately, by default BIRT logs to a directory inside the openCRX EAR:

`opencrx-core-CRX-web.ear\opencrx-core-CRX.war\logs`

To debug a report causing problems you must create this directory and make sure that it is writable (on JBoss you can simply explode the EAR and the WAR files to directories with the appropriate name and then delete the archive files). The BIRT Report Viewer distributed with openCRX logs at the level “FINEST”. You can change this settings in the file `opencrx-core-CRX-web.ear\opencrx-core-CRX.war\WEB-INF\web.xml`

#### Listing 15: Setting BIRT Logging Level in web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app id="webapp">
  <description>openCRX WebGUI</description>
  <context-param>
    <param-name>BIRT_VIEWER_LOG_LEVEL</param-name>
    <param-value>FINEST</param-value>
  </context-param>
  <servlet id="ReportViewerServlet">
    <servlet-name>ReportViewerServlet</servlet-name>
    <servlet-class>org.eclipse.birt.report.viewer.servlet.ViewerServlet</servlet-class>
  </servlet>
  ...

```

Alternatively, you can set a value for `BIRT_VIEWER_LOG_DIR` in the file `opencrx-core-CRX-web.ear\opencrx-core-CRX.war\WEB-INF\web.xml` so that BIRT logs to a directory of your choice:

*Listing 16: Setting BIRT Log Directory in web.xml*

```
...
<context-param>
  <param-name>BIRT_VIEWER_LOG_DIR</param-name>
  <param-value>D:\jboss-4.0.3SP1\server\default\log</param-value>
</context-param>
...
```

## 11.4 Additional BIRT-Servlet Options

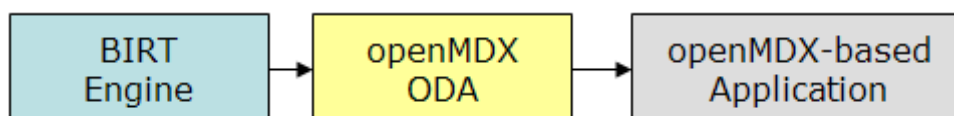
You might want to refer to the BIRT documentation for a full list of options, but the following ones might be useful in combination with openCRX:

*Listing 17: Setting Additional Options of BIRT-Servlet in web.xml*

```
...
<context-param>
  <param-name>BIRT_VIEWER_LOCALE</param-name>
  <param-value>en-US</param-value>
</context-param>
<context-param>
  <param-name>BIRT_VIEWER_REPORT_ROOT</param-name>
  <param-value></param-value>
</context-param>
<!-- Absolute downloaded image/chart directory. Default to ${birt home}/report/images -->
<context-param>
  <param-name>BIRT_VIEWER_IMAGE_DIR</param-name>
  <param-value></param-value>
</context-param>
...
```

## 11.5 Outlook BIRT-Integration with openCRX

As mentioned earlier, support for BIRT is currently experimental. A clean integration of BIRT with openCRX would make use of a BIRT ODA implementation (ODA = **Open Data Access**):



*Figure 70: Integration of BIRT with BIRT ODA Implementation*

At this time there are ODA-Implementations for **flatfile**, **jdbc**, and **xml**.

Alternatively, one could fetch data through the openCRX API and then pass it to the BIRT Engine.

## **12 Integration with Office Suites**

<documentation pending>

### **12.1 MS Office**

<documentation pending>

### **12.2 Open Office**

<documentation pending>

## 13 Next Steps

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