

openCRX Feature Acceleration: GridDrillDown

Version 1.11.0



www.opencrx.org

09-May-2007 @ 01:01:26 PM

License

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

Copyright 2007 © CRIXP Corp. All rights reserved.



Table of Contents

| | | |
|----------|-------------------------------|----------|
| 1 | Introduction..... | 3 |
| 2 | About the Feature..... | 4 |
| 3 | Implementation..... | 8 |

List of Figures

| | |
|--|---|
| Figure 1: openCRX Grid Accounts..... | 3 |
| Figure 2: Grid Drill Down Menu..... | 4 |
| Figure 3: Select Inspector Drill Down..... | 4 |
| Figure 4: Grid Accounts with Inspector Drill Down..... | 5 |
| Figure 5: Close Button..... | 5 |
| Figure 6: Select Address Drill Down..... | 6 |
| Figure 7: Grid Accounts with Address Drill Down..... | 6 |
| Figure 8: Multi-Level Drill Down..... | 7 |

List of Listings

1 Introduction

Today's main purpose of **Grids** is the **compact visualization of similar types of objects in list form**, e.g. a list of addresses or a list of accounts. Various filtering and sorting options are available that enable users – in combination with advanced paging features – to quickly navigate to objects of interest. The figure below shows the grid Accounts (openCRX v1.10.0):

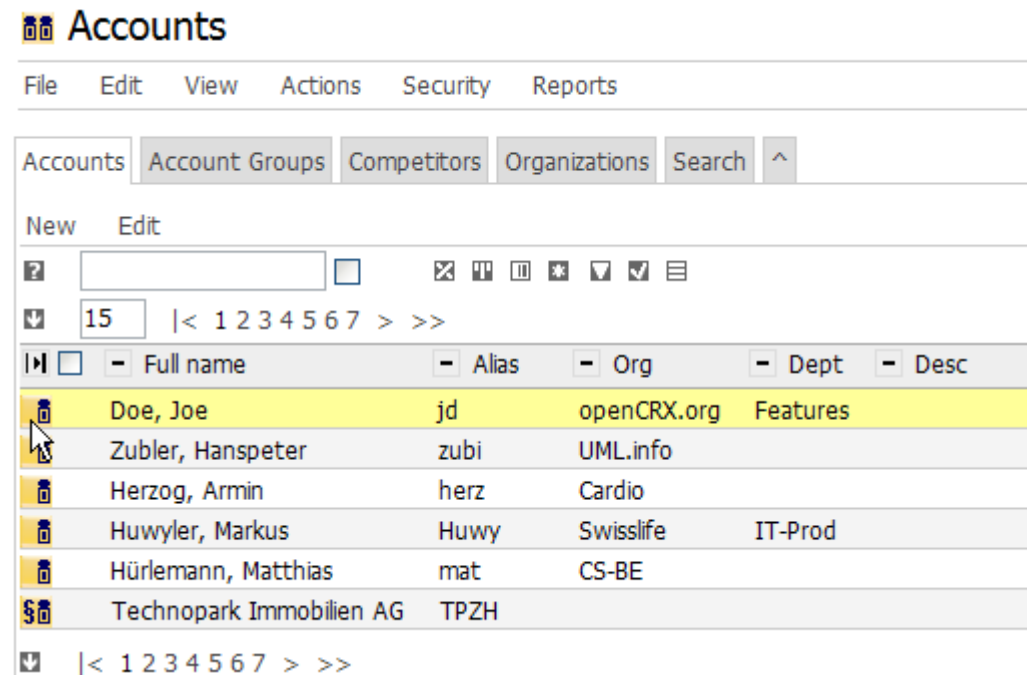


Figure 1: openCRX Grid Accounts

In contrast to the **Inspector**, which allows users to view/inspect all attributes of a particular object, a **Grid typically shows a limited selection of an object's attributes only** (to limit the width of Grids, which would otherwise be very wide for objects with many attributes) . Customization of the Grid ensures that the most relevant attributes of an object are visible. The number one user complaint about Grids has been that if a user needs more information about an object – e.g. inspect attributes not shown in the grid or retrieve information contained in Grids attached to a particular object – one must navigate to the respective object:

- The most intrusive way of retrieving additional information is to click on the object's icon. This will cause a context switch and the object is loaded in a newly generated view (i.e. a new Inspector with all the attached Grids). While this new context is perfectly suited for retrieving any information about the respective object, often times users are interested in a particular subset only (e.g. an account's addresses). Furthermore, the previous (Grid) context is lost. And, while users typically don't care about the "burden placed on the server", it should still be noted that retrieving an object and all its visible grids can put a significant load on the DB server (and this in turn could have an adverse impact on the overall response time; and users do care about response time...).
- There are less intrusive alternatives available; for example, one could open up a new browser window (usually with **<SHIFT><CLICK>** on the object's icon) to inspect the respective object. With modern browsers, which support tabbed browsing (Firefox, Opera, IE7, ...), the respective object could also be inspected in a new tab (usually with **<CTRL><CLICK>** on the object's icon). This is somewhat better an option from a user's perspective as the (Grid's) context is not entirely lost, but the burden on the server for creating this new view is identical to the first approach, i.e. still significant.

2 About the Feature

Based on user feedback and a UseLab analysis we are currently thinking about a new feature to enhance the usability of Grids: **Grid Drill Down**

The core idea of this feature is the following one: if we offer a context sensitive (and model driven) menu for each Grid line to the user, he will be able to select the relevant additional information he is interested in. This in turn will enable us to fetch the required information with an AJAX request and display it directly in the grid without losing the user's context. The following figures illustrate our ideas:

- With any Grid and any object, move the mouse pointer next to the object's icon in the Grid to see the context sensitive menu:

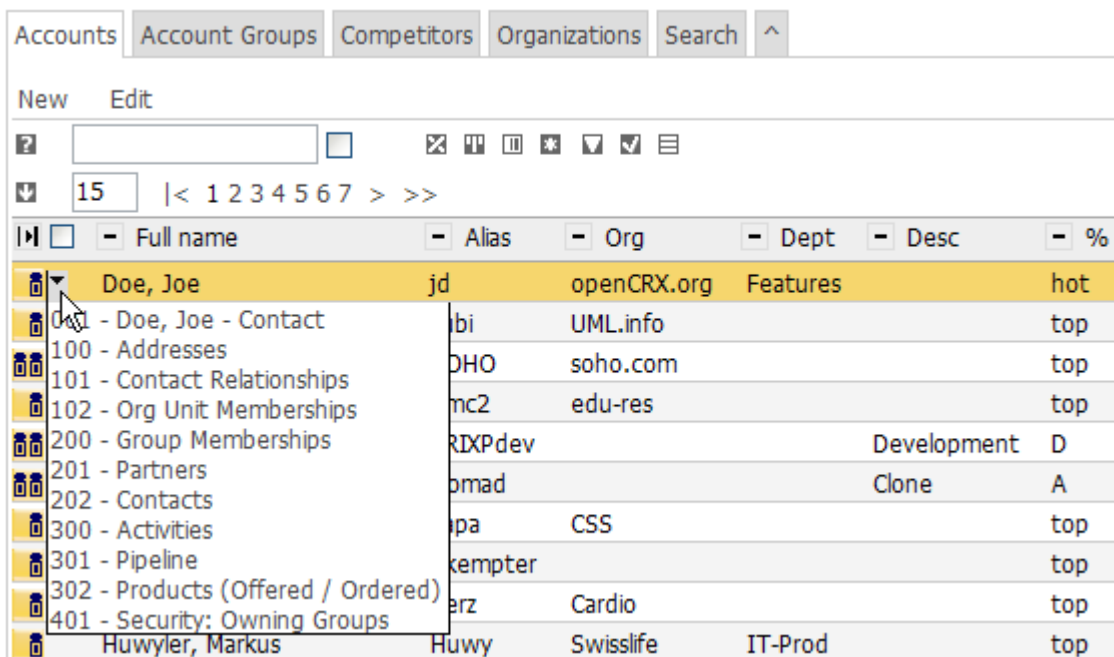


Figure 2: Grid Drill Down Menu

This menu allows the user to select the desired "drill down". The first entry of the above menu allows the user to drill down to the details of the **Contact Doe, Joe**. The remainder of the menu represents Grids of this contact, one Grid for each menu entry. Please note that both visible and hidden Grids are included in the menu.

- Let us now drill down to the details of the **Contact Doe, Joe** by selecting the first menu entry as shown below:

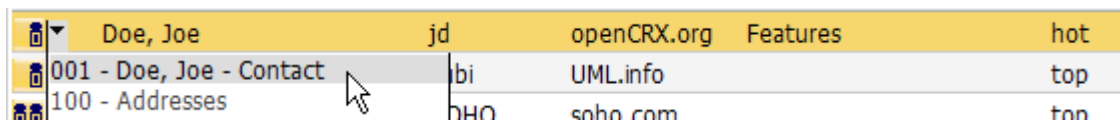


Figure 3: Select Inspector Drill Down

The result of this Inspector Drill Down is shown in the following figure:

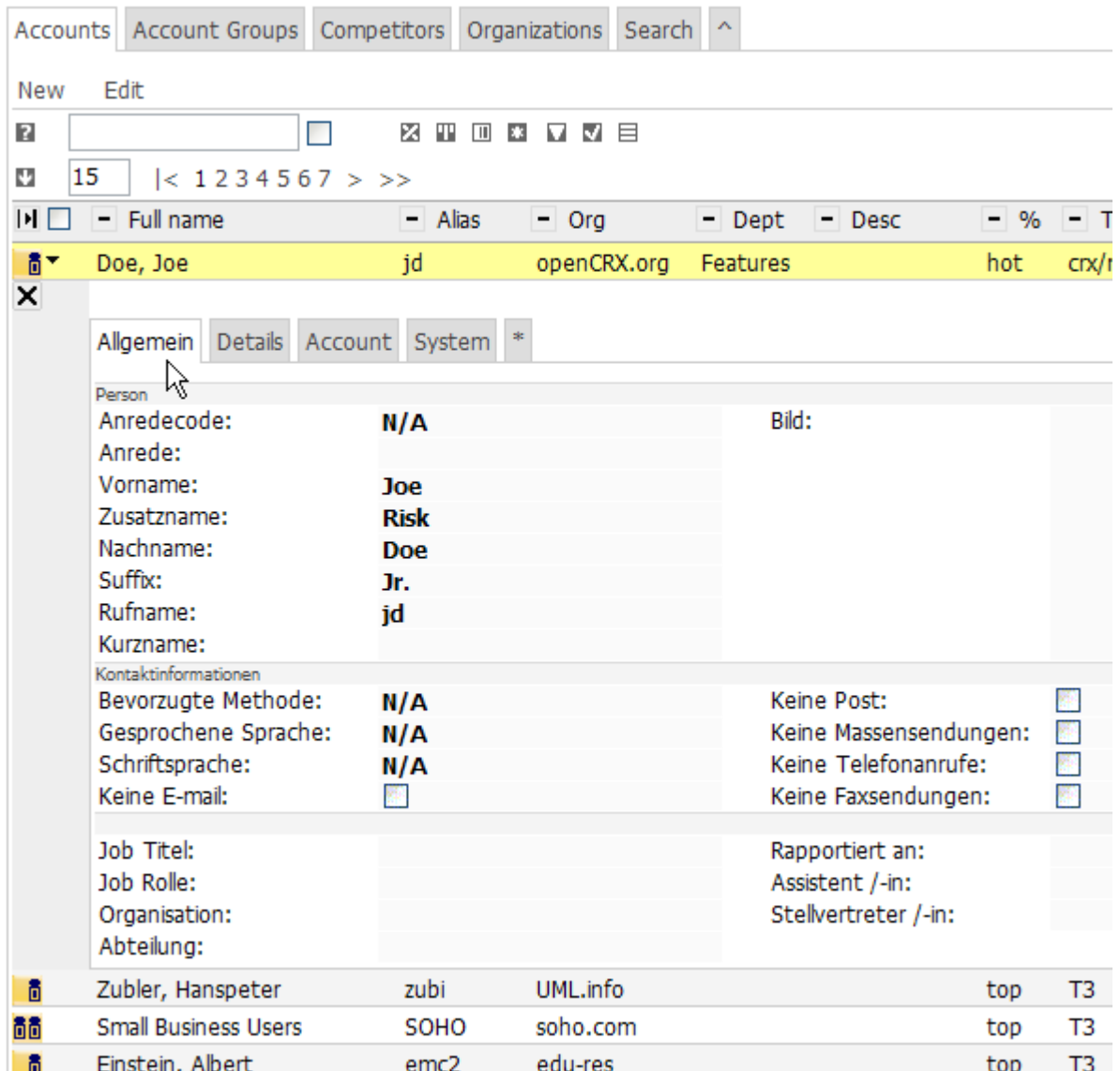


Figure 4: Grid Accounts with Inspector Drill Down

A fully functioning Contact Inspector is embedded in the Grid Accounts, right below the respective line in the Grid. The corresponding AJAX request is extremely light-weight as all the information required to build this Inspector is already available in the Cache.

From a user's perspective, the advantages are obvious: easy, fast, no loss of context.

Furthermore, a simple click on the close button will remove the Contact Inspector from the Grid and fully restore the previous view.

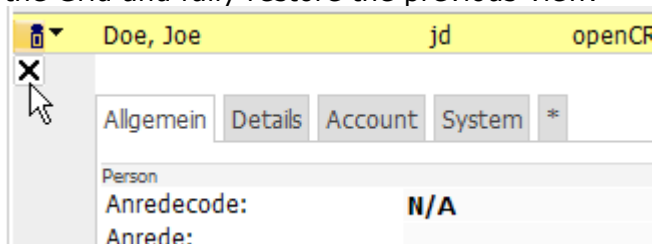


Figure 5: Close Button

- Similarly, we can drill down to the addresses of the **Contact Doe, Joe** by selecting the respective menu entry as shown below:

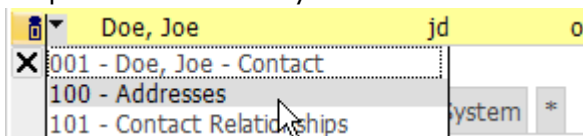


Figure 6: Select Address Drill Down

The result of this address Drill Down is shown below:

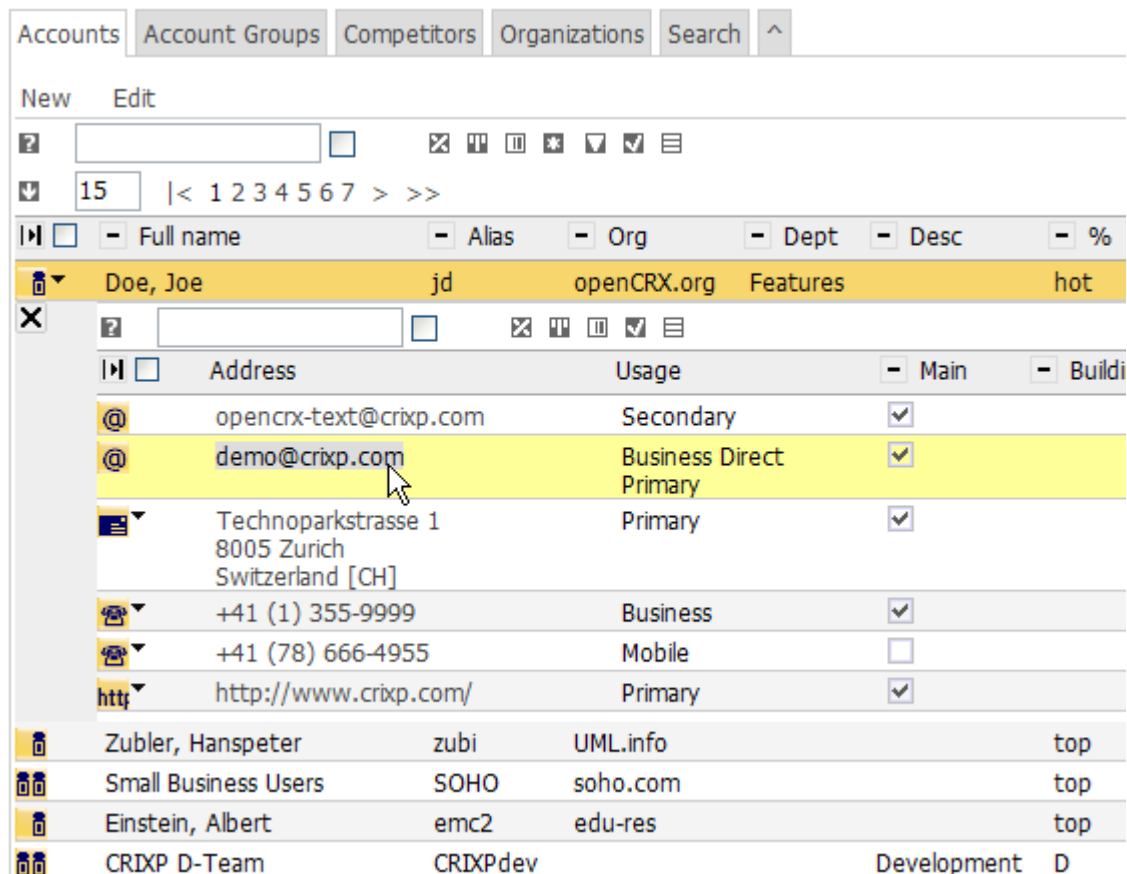


Figure 7: Grid Accounts with Address Drill Down

A fully functioning Address Grid is embedded in the Grid Accounts, right beneath the respective line in the Grid. The corresponding AJAX request is very light-weight.

- As the proposed Grid Drill Down is completely model-driven and all the embedded objects (Inspectors, Grids) are fully functioning, it is possible to drill down through multiple levels. The following figure shows the Drill Down Menu of an Address Grid that is embedded in an Account Grid:

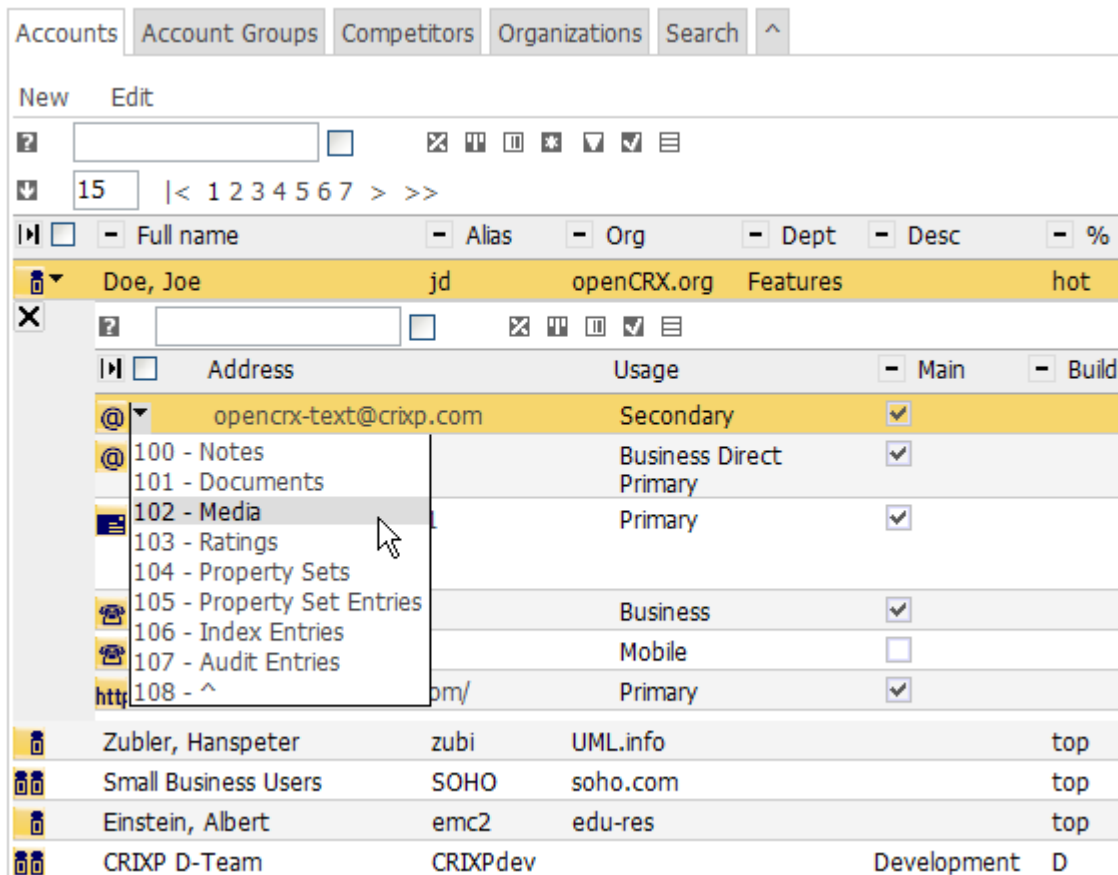


Figure 8: Multi-Level Drill Down

Advantages of this new feature:

- **Enhanced User Experience** because additional information about an object can be retrieved in an intuitive manner and viewed in-place without losing context.
- **Improved Work Flow** thanks to dramatic reduction of context switches.
- **Drastically reduced load on servers** as heavy requests for complete object views (consisting of Inspector and all visible Grids) can be replaced with light-weight AJAX requests; the reduction is best illustrated with an example: in order to generate a complete view of a Contact with all its visible grids it is necessary to retrieve several hundred objects (the exact number depends on the customizing); retrieving a Contact's address grid only typically requires a few requests only. The overall result is that the number of concurrent users can easily be doubled if not tripled.

3 Implementation

The implementation of the **Grid Drill Down** feature requires a refactoring of openMDX/portal and the design and development of new AJAX/DHTML features (e.g. embedding/removing of nodes into/from grids).

We expect to deliver this feature within 2 to 3 weeks of having reached the funding goal.

The funding status is available from <http://www.opencrx.org/community.htm#GridDrillDown>