




ix APPLYING CHANGES
BACKDATED

UNITED PLANET INTREXX XTREME
RELEASE 4.5








Contents


1. Introduction	3
2. Approach	6

Writing Conventions

In this document, text passages will be displayed in *Italics* when they refer to settings in the displayed dialogs. Menu items that are available in context menus are also always reachable from the main menu. Main menu items will not be described, unless they are not available from the context menu. A description of the general main menu items can be found in the  *Center* handbook. Programming code in the text will be displayed in the Courier font.

<xtreme> refers in the following to your Intrex installation path, for example in Windows *C:\xtreme*, or in Linux */opt/xtreme/*. The following symbols will be used to designate special kinds of information:

-  Important notes
-  Tips and background information
-  References to additional information in an Intrex Xtreme handbook
-  Directories
-  URLs
-  Buttons in dialogs or assistants
-  Context menus are designated with this symbol and can be opened by clicking with the right mouse button on the element described.

Passages with the  **[Copy-Paste]** tip can be directly used from this document in order to avoid typing errors.

1. Introduction

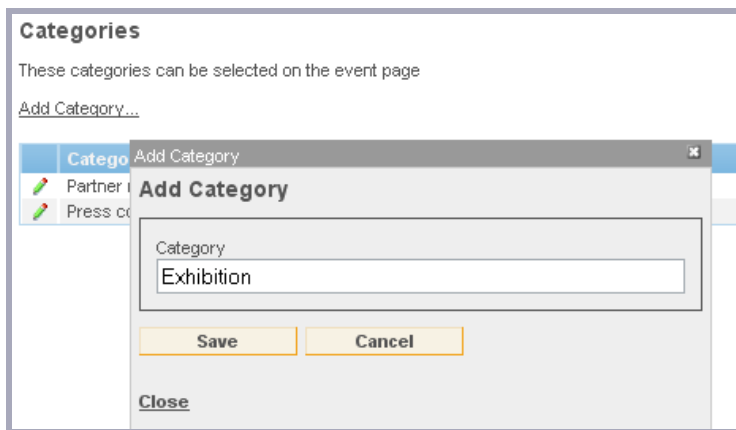
During the creation of a new record we sometimes need to select a predefined value from a selection list, e.g. a category. In a case like this, we often use a reference on another datagroup *categories* that stores all available categories. If you edit the name of a category now, the changing will affect backdated all existing data records referencing that value. This result is not always wanted.

This workshop shows you how to edit the names of predefined values and to transfer these changes backdated to already existing data records. The single steps are illustrated with the help of a descriptive model case.

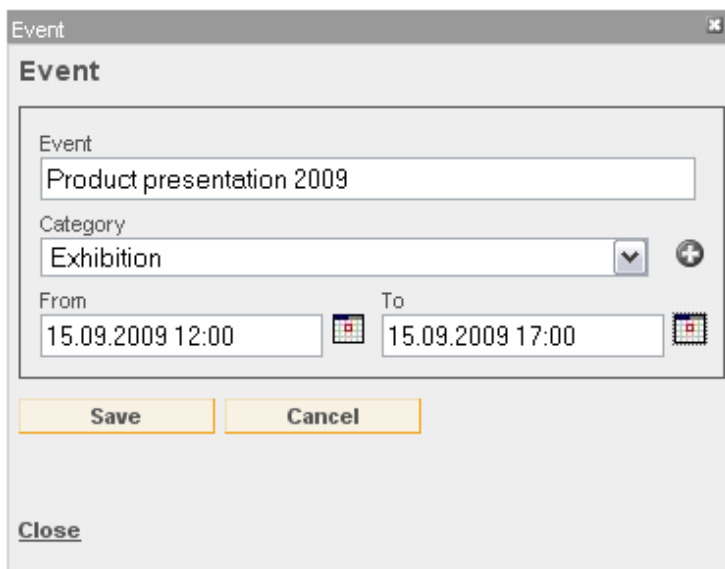
The following scenarios are possible:

- The changing of the predefined value will not affect the existing data records
- The changing of the predefined value will affect all existing records
- The changing of the predefined value will affect all existing records from a specific date on

In our example application we create a simple input page that allows us to add new categories which we can use as predefined values:



Through another input page we can add an event and assign a fitting category to it:



With a table including some new events our example could be displayed as follows:

Events Categories

Events

[Add a new Event](#)

	From	To	Event	Category
	09.10.2009 09:00	09.10.2009 17:00	Product Presentation 2009	Partner meeting
	15.09.2009 12:00	15.09.2009 17:00	Product presentation 2009	Exhibition
	10.10.2008 09:00	10.10.2008 17:00	Product Presentation 2008	Partner meeting
	15.09.2008 12:00	16.09.2008 12:00	Product presentation 2008	Exhibition
	05.10.2007 09:00	05.10.2007 17:00	Product Presentation 2007	Partner meeting
	15.09.2007 12:00	16.09.2007 12:00	Product presentation 2007	Exhibition

Now we would like to change the name of a category in all existing events from *Exhibition* to *Intrex Expo*. Because the changing should affect the events backdated, editing the categories we need to mark the checkbox **Yes** as selected:

Events **Categories**

Categories

These categories can be selected on the event page

[Add Category...](#)

Category
Exhibition
Partner meeting
Press conference

Edit Category

Add Category

Category

Apply changes for events backdated

Yes For events after

[Close](#)

After saving the new name the overview with all events should look like this:

Events		Categories		
Events				
Add a new Event				
	From	To	Event	Category
	09.10.2009 09:00	09.10.2009 17:00	Product Presentation 2009	Partner meeting
	15.09.2009 12:00	15.09.2009 17:00	Product presentation 2009	Intrex Expo
	10.10.2008 09:00	10.10.2008 17:00	Product Presentation 2008	Partner meeting
	15.09.2008 12:00	16.09.2008 12:00	Product presentation 2008	Intrex Expo
	05.10.2007 09:00	05.10.2007 17:00	Product Presentation 2007	Partner meeting
	15.09.2007 12:00	16.09.2007 12:00	Product presentation 2007	Intrex Expo

If we additionally add a date, the changes will only be transferred to events lying at or after this date. In our example we want to rename all the events *Partner meeting* that took place after the 1st October 2009 into *Partner conference*:

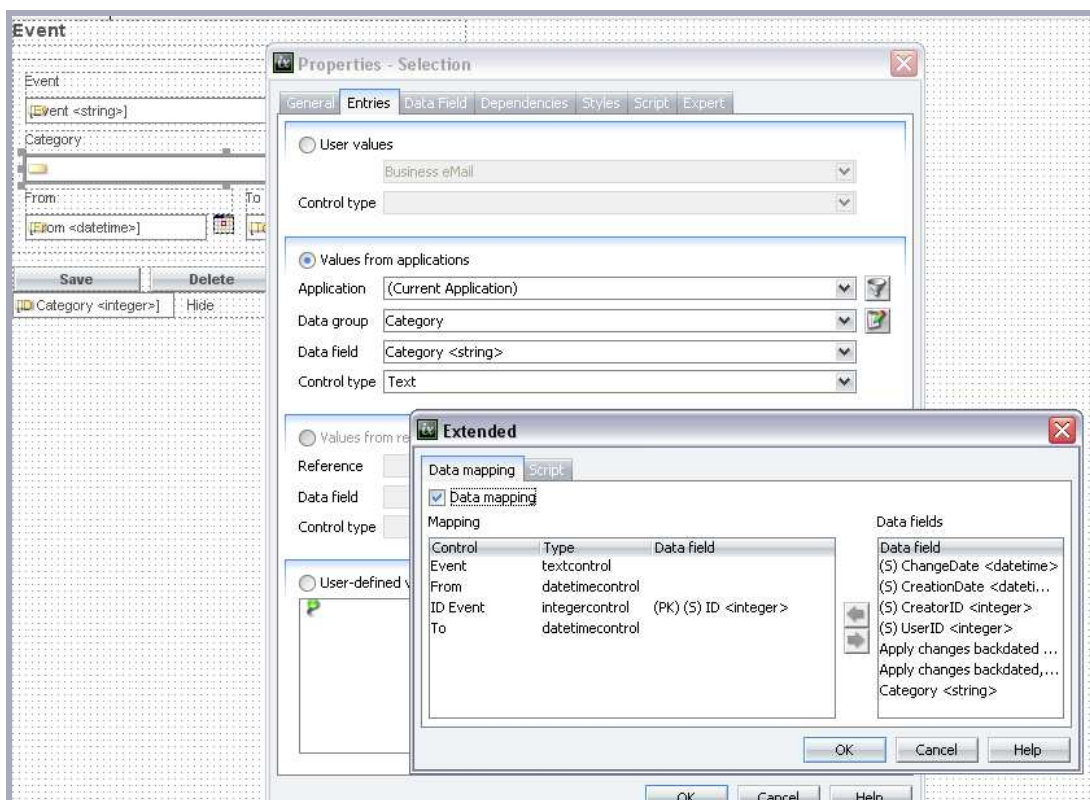
Events		Categories		
Categories				
These categories can be selected on the event page				
Add Category...				
Category				
	Intrex Expo			
	Edit Category			
Add Category				
Category				
<input type="text" value="Partner conference"/>				
Apply changes for events backdated				
<input checked="" type="checkbox"/>	Yes	For events after	<input type="text" value="01.01.2009"/>	
<input type="button" value="Save"/>		<input type="button" value="Delete"/>		<input type="button" value="Cancel"/>
Close				

After saving the new value the overview would now look like this:

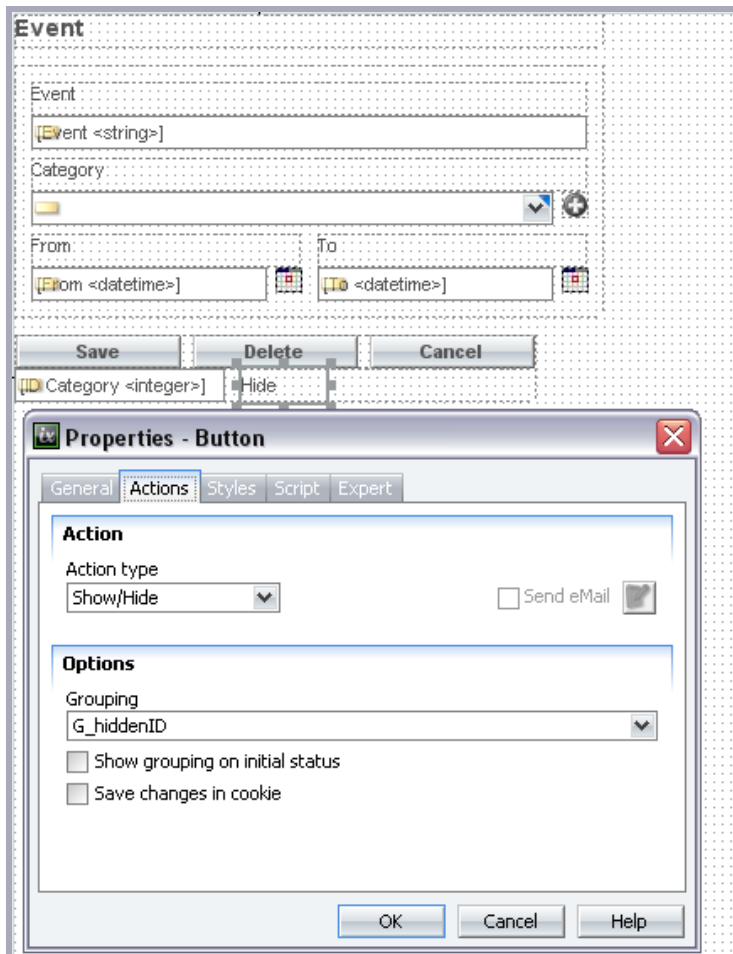
Events		Categories		
Events				
<u>Add a new Event</u>				
	From	To	Event	Category
	09.10.2009 09:00	09.10.2009 17:00	Product Presentation 2009	Partner conference
	15.09.2009 12:00	15.09.2009 17:00	Product presentation 2009	Intrex Expo
	10.10.2008 09:00	10.10.2008 17:00	Product Presentation 2008	Partner meeting
	15.09.2008 12:00	16.09.2008 12:00	Product presentation 2008	Intrex Expo
	05.10.2007 09:00	05.10.2007 17:00	Product Presentation 2007	Partner meeting
	15.09.2007 12:00	16.09.2007 12:00	Product presentation 2007	Intrex Expo

2. Approach

Please open the Application Designer and go to the desired edit page that contains predefined values from another data group. Create a new selection list that gets its values from the data group containing the predefined values. If the value is a name, we allocate a *string* data field. Also add a new edit field for the ID of the assigned value. To fill out this edit field please open the properties dialogue of the selection list and go to **Advanced**. There enable data mapping as follows:



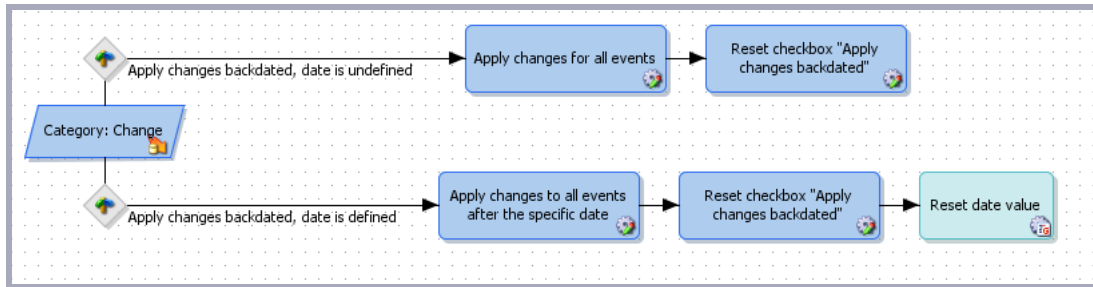
To hide the edit field, please surround it by a grouping. Next also place a new button with the action type *Show/Hide* inside that grouping. Now our edit field will be hidden.



Create a new checkbox on the edit page that allows you to change the predefined value. If the checkbox is marked the changing should be adapted to all existing records. We also add a date edit field to define a date from which on the changing should be transferred:

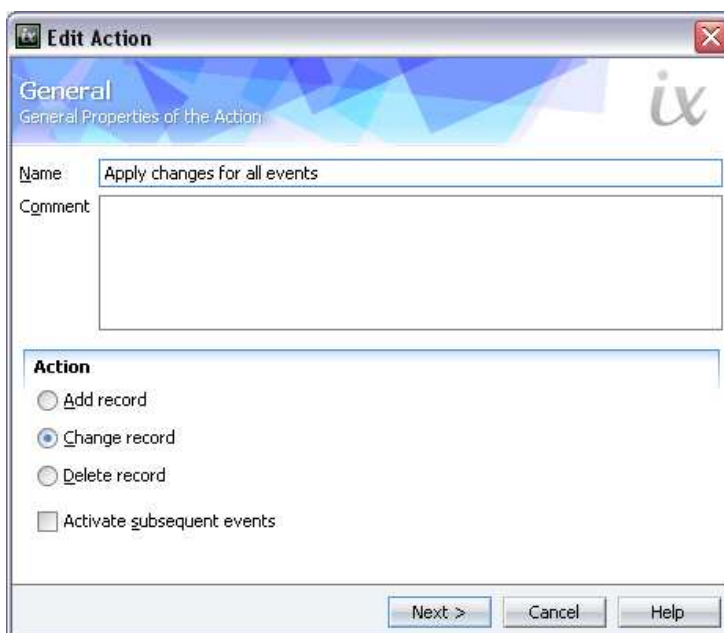


The transfer of the changed value for existing records will be managed by a process in the background:



The initializing event is the changing of a predefined value, in this example the category. In the condition we check if the checkbox has been marked.

If the box has been marked and no date has been defined, the changing will be transferred to all existing records. In this case a data group action will be executed that changes all records (here *Events*) containing the predefined value (here *Category*). The next screenshots show the configuration of the data group action:



Edit Action
Manipulation Quantity
 Restriction of Target Records

Simple restriction
 Source field: (PK) (S) ID <integer>
 Destination data field: ID Category <integer>

Restriction via filter

< Back Next > Cancel Help

Edit Action
Field Assignment
 Assignment of Target and Source Data Fields

Target data group			Source data group	
Field	Type	Assigned	Field	Type
(PK) (S) ID	integer		(PK) (S) ID <inte...	integer
(S) ChangeDate	datetime		(S) ChangeDate ...	datetime
(S) CreationDate	datetime		(S) CreationDate ...	datetime
(S) CreatorID	integer		(S) CreatorID <in...	integer
(S) UserID	integer		(S) UserID <integ...	integer
Category	string	Category <st...	Apply changes ba...	boolean
Event	string		Apply changes ba...	datetime
From	datetime		Category <string>	string
ID Category	integer			
To	datetime			

< Back OK Cancel Help

By clicking **OK** you finish the data group action. In the next step we reset the boolean value in the record by using another data group action:

Edit Action
General
 General Properties of the Action

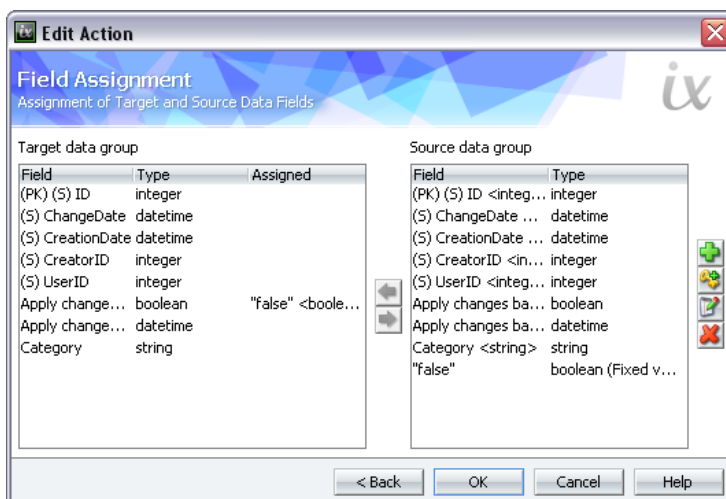
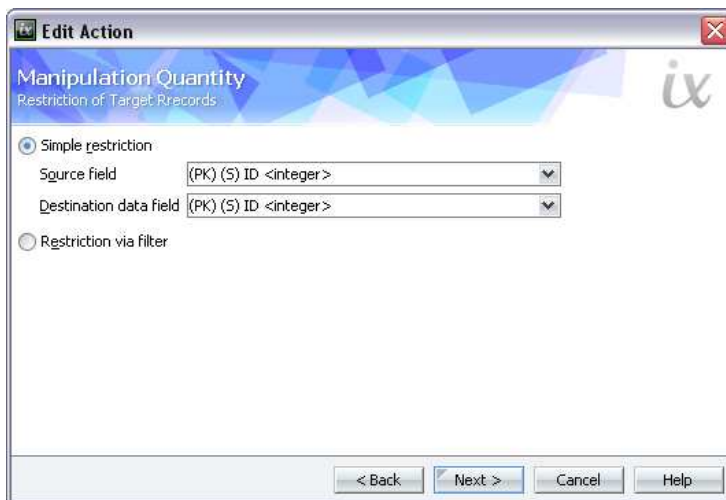
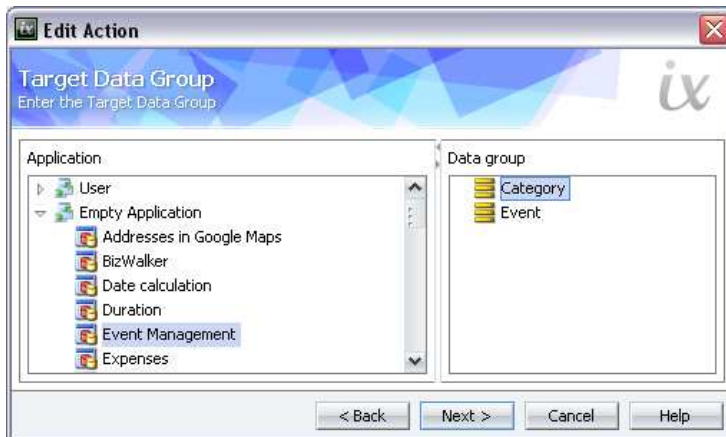
Name: Reset checkbox "Apply changes backdated"

Comment:

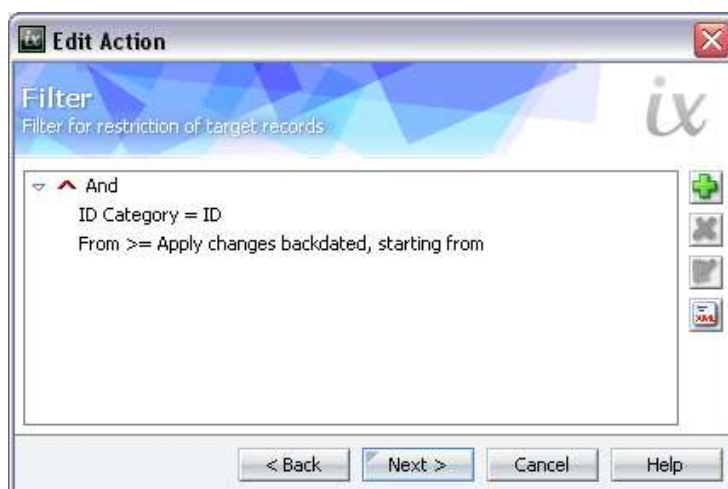
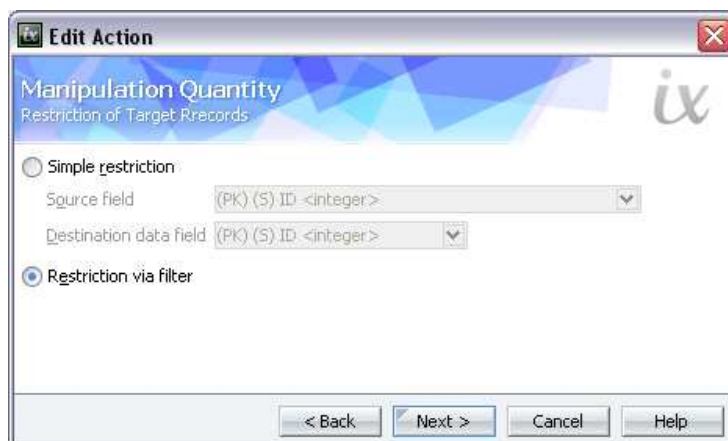
Action

Add record
 Change record
 Delete record
 Activate subsequent events

Next > Cancel Help



If the checkbox has been marked and in addition a date has been entered, another part of the process will be executed. The first step is to run a datagroup action that initiates the same action as mentioned before, but only for a certain timeframe. In order to do so we use a filter:



Next we also execute another data group action to reset the boolean value of the checkbox.
 Finally we need to reset the date, too. This can be done by using a Groovy script as follows:

```
import de.uplanet.lucy.server.dataobjects.impl.ValueHolderFactory as VH

// System connection
def l_conn = g_dbConnections.systemConnection

// ID Category, datafield (PK) (S) ID
def l_vhCategoryID = g_record["0B55BAD86F3BC46E03514894736197CFD6C512D9"]

// Constants Datagroup Category and field date from
static final String STR_TABLE_CATEGORY = "XWS08_CATEGORY"
static final String STR_FIELD_DATE_FROM = "DT_APPLY_CHANGES"

def l_stmtUpdate = g_dbQuery.prepare(l_conn, "UPDATE ${STR_TABLE_CATEGORY}
SET ${STR_FIELD_DATE_FROM} = null WHERE LID = ?")

l_stmtUpdate.setInt(1, l_vhCategoryID)
l_stmtUpdate.executeUpdate()

return true
```

You can follow every step within the attached example application.

Have a good time transferring the above to your own applications.