

Table of Contents

Introduction

The User Profile AddOn saves your screen layout if you close a screen and restores everything if you open the same screen again. This works per user and global. It's possible to create default screen layouts for every user, and a specific user will use its own layout based on the global one. The AddOn also adds a new functionality to your tables:



A new button (? icon) will be added to the table. The dialog will show the table name, the number of records/rows, and all available columns. It's possible to hide or show specific columns with this dialog.

The filter is a module and is bundled with user profiles. It allows the user to create custom filter conditions directly in the screen. Here's how this might look:



The following conditions are available:

- contains
- not contains
- starts with
- not starts with
- equals
- not equals
- between
- not between
- greater
- less
- greater or equals
- less or equals

It's possible to add as many conditions as you want and single conditions can be enabled or disabled.

The customized filter can also be saved and loaded:



Another feature is and/or grouping. Decide which option is combined with “and” or “or”.

Customizing

The AddOn is designed to be customizable. The default settings and implementations can be modified. The AddOn contains the class ProfileUtil which offers a lot of useful methods, such as:

```
setDefaultTableInformationDialogClass(IApplication app, Class clazz)
```

This method set the class which should be used to show the table information dialog. It's possible to extend the existing class **TableInformationDialog**. It's also possible to change the table information dialog per Navigation table:

```
setTableInformationDialogClass(NavigationTable table, Class clazz)
```

It's also possible to hide the table information button:

```
setDefaultShowTableInformationButton(IApplication app, boolean show)
```

Check the available methods of **ProfileUtil** to find out more.

Another feature is the definition of available column names in the table information dialog:

```
databook.getRowDefinition().setColumnView(ITableInformationControl.class,  
new ColumnView(new String[] {...}));
```

This statement defines a list of column names which will be shown in the table information dialog.

Full API documentation

[User Filter](#)
[Profiles](#)

From:

<http://doc.sibvisions.com/> - **Documentation**

Permanent link:

http://doc.sibvisions.com/visionx/userprofiles_filters



Last update: **2021/10/01 09:43**