

Apparo Fast Edit

How to setup the History feature

Version 3.1.1



1	Basics	3
2	Prerequisites	3
3	Enabling the feature	3
4	Settings of the History feature	5
4.1	Tab Main	5
4.1.1	Use columns as row group	5
4.1.2	Decision strategy	5
4.1.3	'Valid from' date column	6
4.1.3.1	Validate valid from	7
4.1.4	'Valid to' date column	8
4.1.5	Use for current flag	8
4.2	Tab Advanced	8
5	Hands on	9
5.1	Step 1 - Setting up the History feature	9
5.2	Step 2 - Preparing the Business Case	10
5.2.1	Primary key and hidden widgets	10
5.2.2	Setting up TIME_FROM & TIME_TO	10
5.2.3	Setting up Business Case output filter	11
5.2.4	Setting up the history	12
5.2.5	Result	12
5.3	Optional features	13
5.3.1	Using simple auditing	13
5.3.2	Using delete current data row	13

1 Basics

The history feature is based on [slowly changing dimensions type 2](#) and is intended to track data changes. Instead of overwriting a row when the user changes the value of one or several columns, the history feature creates a new row and marks it as current.

The example:

A car sharing club owns different cars and each car can be used by one member at a time only. To track who is driving the car right now they are using Fast Edit and a simple database table with the columns ID (which is auto generated when inserting a new row), CAR_NAME, DRIVER_NAME.

Now they also want to track who had the car recently, they decided to use the History feature, because of its capability to track data changes on selected dimensions.

2 Prerequisites

The target table of the Business Case requires 2 additional columns of type 'date' and one optional column for the current flag of type e.g. varchar or number.

Using an additional column 'current flag' makes it later far easier to filter the Business Case for the 'current rows' only.

The feature 'Inserting of new data rows' must be enabled, including the checkbox 'Insert area is able to update an already existing data row'.

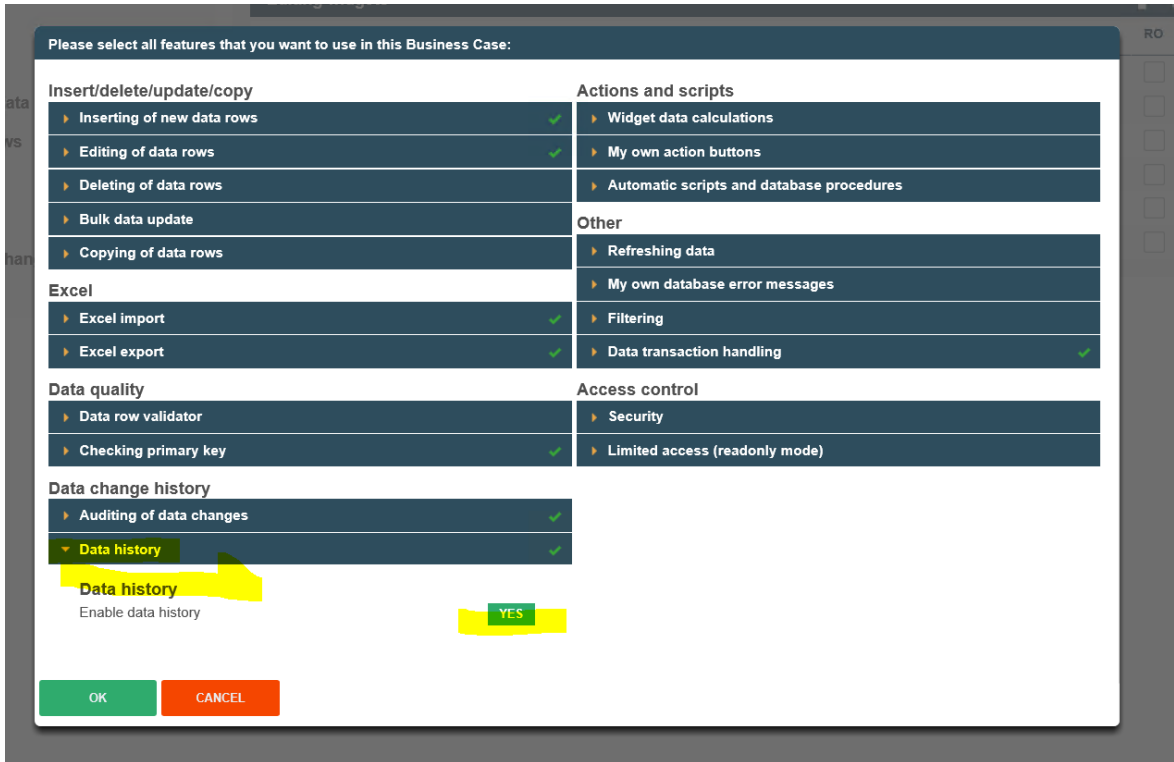
3 Enabling the feature

All features can be enabled/disabled using the 'Features Button' in the top right corner.

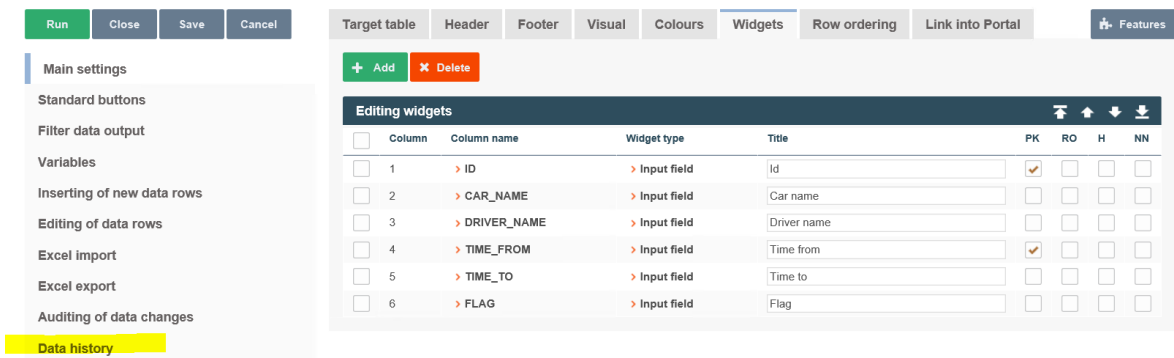
The screenshot shows the APPARO interface with the 'Features' button highlighted in yellow in the top right corner. Below the main navigation tabs, the 'Editing widgets' table is visible, listing columns and their widget types.

Column	Column name	Widget type	Title	PK	RO	H	NN
<input type="checkbox"/>	1 > ID	> Input field	Id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	2 > CAR_NAME	> Input field	Car name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	3 > DRIVER_NAME	> Input field	Driver name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	4 > TIME_FROM	> Input field	Time from	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	5 > TIME_TO	> Input field	Time to	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	6 > FLAG	> Input field	Flag	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Now click on 'Data history' and enable the feature by clicking on the 'No' button. It turns to 'Yes' and shows that the feature is enabled now.



After clicking the 'OK button' the widget edit view refreshes and the menu 'Data history' appears.



4 Settings of the History feature

4.1 Tab Main

4.1.1 Use columns as row group

The row group can be imagined as the 'primary key' group of the History group. It is used to identify the data rows belonging to a history group and can contain one or several columns.

Hints:

- The row group should build a unique identifier (like a primary key).
- Take sure that users cannot alter the values of the row group. If values of the history group are changing, this will create new history group (a new entry).

In our example the CAR_NAME is unique and can be used for the history group. Changing any other value will create a new row for this history group and mark it as current, the old values are historized and in the most cases hidden with filters.

Hint:

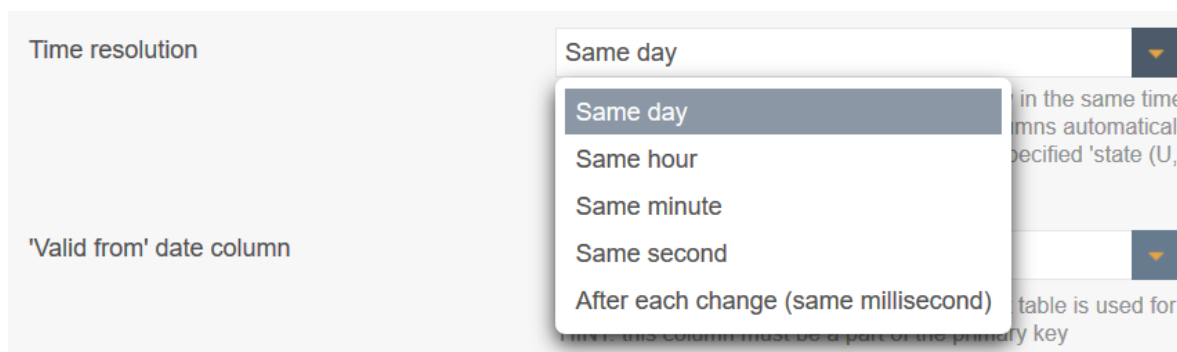
Changing CAR_NAME will create a new history group, since the unique identifier has changed.

In our simple example is 'any other value' the column DRIVER_NAME only, since all other dimensions cannot be used for the history group.

You must not use for the history group:

- Auto computed columns (as the ID column, which is generated by the DB)
- All columns mapped to the history feature (as TIME_FROM, TIME_TO, FLAG)

4.1.2 Time resolution



Sets the timeframe, within a correction (= row update) without historizing is allowed.

Let's say this property is set to 'Same minute', which means if the user is changing the history row within one minute, then no new row is created for the history feature. 'One hour' allows corrections withing one hour and 'After each change (same millisecond)' will always create a history row.

Double entries comparing with current data rows are ignored automatically (no update, no insert).

4.1.3 'Valid from' date column

This property needs to be mapped to the 'TIME_FROM' widget. It is storing the valid from value of the history feature.

Hint:

It is important that the widget for the valid from property of the history feature (TIME_FROM in our example) is part of the primary key of the Business Case.

In the most cases the **valid from** date will be generated automatically by the Business Case, using the variable <%CURRENT_DATE%> as constant value in insert and update case in the widget settings of the TIME_FROM widget (tab Mapping & Other) or this date is read from Excel document or manual entry.

Widget settings of database column TIME_FROM

Widget type	Mapping & Other	Flags	Visual	Visual help texts	Data output format
Column name	TIME_FROM ▼ *				
Default value	<input type="text"/>				
Constant value	Use constant value in insert and update case ▼				
	<input type="text" value="<%CURRENT_DATE%>"/>				
Variable for using content in detail BC	<input type="text"/>				
OK		CANCEL			

Also important is the correct output format of this widget. If the history feature is set to 'Same second' and the widget output format is storing only date & hours + minutes, it cannot work properly due the missing seconds.

We recommend to set the widgets output format to at least 'Date with Time Medium'
Please note 'Same day' requires an output format of date without time.

Widget settings of database column TIME_FROM

Widget type	Mapping & Other	Flags	Visual	Visual help texts	Data output format
Output type					<div style="border: 1px solid #ccc; padding: 5px;"> Number Currency Percentage Date / Time Text Use type of output column </div>
Show date picker					<input checked="" type="checkbox"/>
Date and time part					<div style="border: 1px solid #ccc; padding: 5px;"> Date Time Date with time </div>
Format					<div style="border: 1px solid #ccc; padding: 5px;"> Short Medium Long Full Custom </div>
Data quality check					
Custom validator Java 7 class					<input type="text" value="----"/>
Interval					Minimum allowed: <input type="text"/> Maximum allowed: <input type="text"/>
		<div style="display: flex; justify-content: space-around;"> OK CANCEL </div>			

This setting returns a date in the format 'Oct 25, 2016 9:17:13 AM'

Hint:

For milliseconds use the custom output type e.g. 'dd.MM.yyyy:hh.mm.ss.SSS'

4.1.3.1 Validate valid from

'Valid from' date column	TIME_FROM	<input type="checkbox"/>	Validate that the new 'valid from' value precede the latest existing 'valid from' value of this row group.
'Valid to' date column	TIME_TO		
Use for current flag	FLAG		

The valid from property also contains the checkbox 'Validate that the new 'valid from' value precede the latest existing 'valid from' value of this row group'.

Example:

The user is selecting the valid from date manually by a date picker (and not using the <%CURRENT_DATE%> variable as constant value), in this case we recommend to activate the checkbox, to make sure that a new entry will be valid (which requires a newer valid from date).

4.1.4 'Valid to' date column

In our example we map here the widget 'TIME_TO'.

Fast Edit is automatically inserting the values for this widget.

A new history row is set to the 'infinity date' (which we will explain later) and the old historized row get's a timestamp, which shows that the history row is not longer the 'current' row.

Id	Car name	Driver name	Time from	Time to	Flag
2	Porsche-Cayenne	Silvio Speed	Oct 25, 2016 9:17:13 AM	Oct 25, 2016 9:17:35 AM	0
3	Porsche-Cayenne	Rich Racer	Oct 25, 2016 9:17:36 AM	Dec 31, 2999 12:00:00 AM	1

We changed the dimension 'Driver name' to Rich Racer at 9:17:36 AM (row with ID=3), the system created a new row with infinity date (row with ID=3) and sets the valid to date (time to) of the old row to 9:17:35 AM and marks it that way as not current. Reading time from and time to we know this row was the current history row from 9:17:13 AM to 9:17:35 AM.

Hint:

We recommend to set the data output format of the valid to widget (TIME_TO) the same way as the valid from widget (TIME_FROM).

In our example to 'Date with Time (medium)'.

4.1.5 Use for current flag

This optional property can be mapped to the current flag widget.

In the most cases the current history row will be set to 1 (when using 0 and 1 for current flag).

It allows to easier filter the Business Case.

Example:

When we use this filter: FLAG = 1, then all historized rows will not be shown to the end user in the Business Case.

4.2 Tab Advanced

It contains the settings for the infinity date and for the current flag.

Main **Advanced**

Data history advanced

Date for infinity:

History flag for current row: *

History flag for not current row: *

The infinity date is always used automatically to mark the current row.

5 Hands on

We will now continue with our example and setup the Business Case step by step. As preparation we already created the Business Case, including the additional columns for the history feature.

5.1 Step 1 - Setting up the History feature

Main
Advanced

Using history functionality means that Apparo Fast Edit is automatically copying a data row if the user is changing a value. Depending on the decision strategy (time resolution) it is updating the row or copying including time frame update. The row group normally contains all used primary key widgets. It must not contain date_from & date_to columns. For auto-handling the date from widget, it requires e.g. the <%CURRENT_DATE%> variable set as constant value in ins. Please set the output type of this widget to at least date with time (medium). For milliseconds use the custom output type. Please read this [document](#) for more details.

Use columns as row group

ID

CAR_NAME

DRIVER_NAME

TIME_FROM

TIME_TO

FLAG

Time resolution

After each change (same millisecond)

If there are 2 or more data changes into a row in the same time frame and changing the 'date from' and 'date to' columns automatically to Special case: if simple auditing is used with specified 'state (U,I,D)' strategy)

'Valid from' date column

TIME_FROM

The 'valid from' database column of the target table is used for storing. HINT: this column must be a part of the primary key

'Valid to' date column

TIME_TO

The 'valid to' database column of the target table is used for storing

Use for current flag

FLAG

The 'current' database column of the target table is used for marking

With CAR_NAME as history group, we are historizing any changes in the widget DRIVER_NAME. Time resolution same second is storing any change when using manual input.

Hint:

ID cannot be part of the history group; the database is computing the value.

5.2 Step 2 - Preparing the Business Case

5.2.1 Primary key and hidden widgets

A widget is a flexible input/output component that is connected to a database column of the target table. In this tab you can place widgets into this Business Case, into different areas like filtering area, bulk update area, editing area, input area and calculation area. Each area has a different functionality:

- Filtering area: The place for filtering widgets for comfortable filtering
- Bulk update area: Widgets for updating many rows with one mouse click
- Editing area: Widgets for editing existing data rows
- Inserting area: Widgets for inserting data - The inserting area is part of the editing area
- Calculation area: Area for output of statistic data

Editing widgets

Column	Column name	Widget type	Title	PK	RO	H	NN
<input type="checkbox"/>	1 > ID	> Input field	Id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	2 > CAR_NAME	> Input field	Car name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	3 > DRIVER_NAME	> Input field	Driver name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	4 > TIME_FROM	> Input field	Time from	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	5 > TIME_TO	> Input field	Time to	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	6 > FLAG	> Input field	Flag	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The primary key is the ID and TIME_FROM.

ID can be part of the primary key but not part of the history group, the primary key combination of the Business Case can differ from the primary key of the database table.

ID, TIME_FROM, TIME_TO and FLAG are set to hidden, users don't need to change them.

5.2.2 Setting up TIME_FROM & TIME_TO

Widget settings of database column TIME_FROM

Widget settings of database column TIME_FROM

Widget type | **Mapping & Other** | **Flags** | **Visual** | **Visual help texts** | **Data output format**

Column name: TIME_FROM

Default value:

Constant value: Use constant value in insert and update case

<%CURRENT_DATE%>

Variable for using content in detail BC:

OK CANCEL

We insert CURRENT_DATE as constant value in insert and update case in the widget TIME_FROM. This allows Fast Edit to automatically manage the history properties 'valid from' and 'valid to'.

Widget settings of database column TIME_FROM

Widget type	Mapping & Other	Flags	Visual	Visual help texts	Data output format
Output type					<ul style="list-style-type: none"> Number Currency Percentage Date / Time Text Use type of output column
Show date picker					<input checked="" type="checkbox"/>
Date and time part					<ul style="list-style-type: none"> Date Time Date with time
Format					<ul style="list-style-type: none"> Short Medium Long Full Custom
Data quality check					
Custom validator Java 7 class					<input type="text" value="----"/>
Interval					Minimum allowed: <input type="text"/> Maximum allowed: <input type="text"/>
				<input type="button" value="OK"/> <input type="button" value="CANCEL"/>	

We set the data output format of the widgets **TIME_FROM** and **TIME_TO** to Date with Time (medium)

5.2.3 Setting up Business Case output filter

Run	Close	Save	Cancel
<ul style="list-style-type: none"> Main settings Standard buttons Filter data output Variables Inserting of new data rows Editing of data rows Excel import Excel export Auditing of data changes Data history 			
Filter data output			
The SQL filter conditions are filtering data rows of the target table for the output.			
<input type="text" value="FLAG = 1"/>			
<input type="text" value="FLAG"/>			
<input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="*"/> <input type="button" value="/"/> <input type="button" value="&"/> <input type="button" value=" "/> <input type="button" value="^"/> <input type="button" value=" "/> <input type="button" value="="/> <input type="button" value=">"/> <input type="button" value="<"/> <input type="button" value=">="/> <input type="button" value="<="/> <input type="button" value="("/> <input type="button" value=")"/> <input type="button" value="'"/>			

FLAG=1 filters all historized rows, since those have the value 0 in the column FLAG.

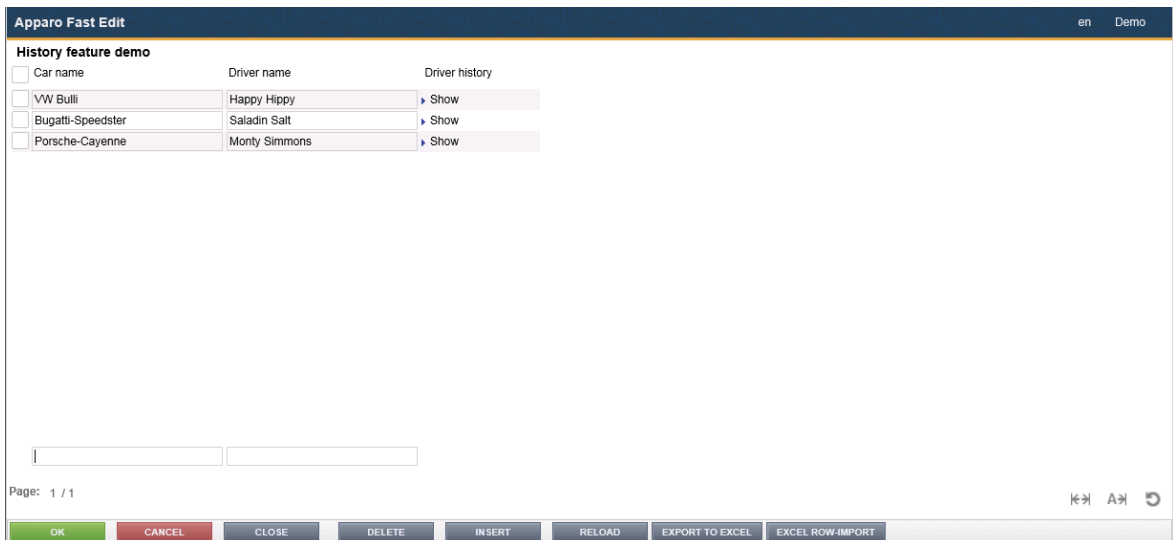
5.2.4 Setting up the history

As last part we want to add a driver history, that shows all recent drivers, which is technically seen a list of all historized rows.

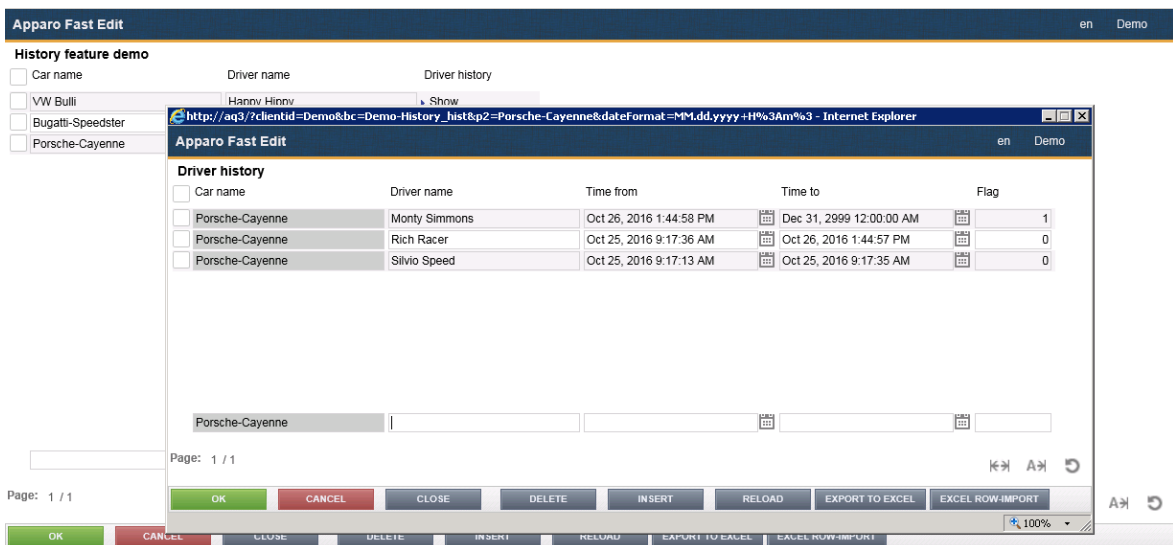
We simply use an adapted copy of the Business Case, removed the filter and the history feature and created a hyperlink widget to this Business Case.

5.2.5 Result

For the user, the history feature works totally hidden.



Clicking the Driver history link opens a popup window with all recent drivers (historized data rows)



5.3 Optional features

5.3.1 Using simple auditing

Simple auditing has a different behavior if history feature is enabled.

If a new data row is inserted and this is the first version of a history then state column contains “I”.
If a data row is changed and a new data row (new version) is created then state column contains “U”.

Auditing of data changes

Simple auditing settings		
	Inserting a new row case	Updating or deleting row case
User name column	PLANNING_CHANGE_USER	PLANNING_CHANGE_USER
Date column	PLANNING_CHANGE_DATE	PLANNING_CHANGE_DATE
State (U,I,D) column		PLANNING_CHANGE_TYPE
Row edit type column		
In delete case delete data row physically		<input type="checkbox"/>

5.3.2 Using delete current data row

It is possible to delete the current data row of a history.
In this case the “Simple Auditing” feature must be enabled and the column “State” must be used.

If the user is selecting the current data row and is pressing the “Delete” button then a copy of this data row is getting the status “D” and time-from and time-to is changed to current date.

Auditing of data changes

Simple auditing settings		
	Inserting a new row case	Updating or deleting row case
User name column	PLANNING_CHANGE_USER	PLANNING_CHANGE_USER
Date column	PLANNING_CHANGE_DATE	PLANNING_CHANGE_DATE
State (U,I,D) column		PLANNING_CHANGE_TYPE
Row edit type column		
In delete case delete data row physically		<input type="checkbox"/>