



Developing State-of-the-Art Table User Interfaces in Web Dynpro Java

SAP NetWeaver 7.0

Bertram Ganz, SAP NetWeaver UI Foundation OPS

- **Provide an overview of the new Web Dynpro table functions in SAP NetWeaver 7.0**
- **Layout Web Dynpro tables**
- **Use cell variants**
- **Size tables using absolute and relative widths**
- **Explain grouped table headers**
- **Talks about some of the new table functions in SAP NetWeaver CE 7.1**



Web Dynpro Table UI Elements

Designing Tables – Sizing, Layouting, Grouping

Further Notes and Topics

Outlook – Table in SAP NetWeaver CE 7.1

Summary



Web Dynpro Table UI Elements

New Feature Overview

Web Dynpro Table Metamodel

Using Table Cell Variants

Grouping Values

- Table cells can be merged on the basis of their grouping value

| Id | Connection | Airline |
|----|------------|-----------------|
| SQ | 988 | Singapore Air |
| | | Singapore Air |
| | | Singapore Air |
| UA | 233 | United Airlines |
| | 333 | United Airlines |
| | | United Airlines |
| | 941 | United Airlines |

Hierarchical Column Headers (See Exercise 2)

- Headers can be arranged in a hierarchy

| Identification | | |
|----------------|------------|-------------------|
| Id | Connection | Airline |
| A,A | 17 | American Airlines |
| A A | 17 | American Airlines |

Horizontal Scrolling

- A table can display fewer columns than nested in the table.
- Scroll to invisible columns

| Id | Connection | Airline |
|----|------------|----------------|
| AZ | 790 | Alitalia |
| AZ | 790 | Alitalia |
| DL | 106 | Delta Airlines |
| DL | 106 | Delta Airlines |

Fix Table Columns Right or Left Aligned

- You can also fix columns aligned to the right or left.
- All scrollable columns should have the same widths to avoid resizing of the table after every scrolling action

| Id | Connection | Airline |
|----|------------|----------------|
| AZ | 790 | Alitalia |
| AZ | 790 | Alitalia |
| DL | 106 | Delta Airlines |
| DL | 106 | Delta Airlines |

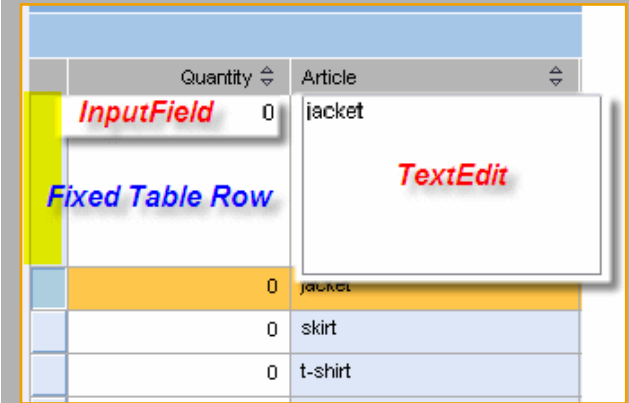
Table Cell Variants (See Exercise 1)

- Use different table cell editors in the same table column for different table rows

| CellVariants | InputField |
|-------------------------------------|------------|
| Apply | Horse |
| Text 1 | Cat |
| Bird | Bird |
| | Elephan |
| Apply | Horse |
| <input checked="" type="checkbox"/> | Bird |

Fixed Top Table Cell

- Position cell editor(s) at fixed position on top of the table rows
- Replace the generic table filter row with custom filter editors like checkboxes, dropdowns



Fixed Bottom Table Cell

- Position cell editor(s) at fixed position at bottom of displayed table rows



Fixed Table Layout

- **Fix table layout to predefined sizes**
- **Avoid *jumping* table columns based on differing text lengths in different table rows**
- **Hidden content gets displayed via tooltip**

| Column1 Header | Column2 Header | Column3 Header |
|---------------------|--------------------|---------------------|
| Column 1; Element 0 | Column 2; Eleme... | Column 3; Element 0 |
| Column 1; Element 1 | Column 2; Eleme... | Column 3; Element 1 |
| Column 1; Element 2 | Column 2; Eleme... | Column 3; Element 2 |
| Column 1; Element 3 | Column 2; Eleme... | Column 3; Element 3 |
| Column 1; Element 4 | Column 2; Eleme... | Column 3; Element 4 |

Row 1 of 10

Table Grid Modes

- **Explicitly show or hide borders of table columns and table rows**

| Column1 Header | Column2 Header | Column3 Header |
|---------------------|---------------------|---------------------|
| Column 1; Element 0 | Column 2; Element 0 | Column 3; Element 0 |
| Column 1; Element 1 | Column 2; Element 1 | Column 3; Element 1 |
| Column 1; Element 2 | Column 2; Element 2 | Column 3; Element 2 |
| Column 1; Element 3 | Column 2; Element 3 | Column 3; Element 3 |
| Column 1; Element 4 | Column 2; Element 4 | Column 3; Element 4 |

Zeile 1 von 10

VERTICAL

Table *Row* Popin

- Insert popin container displayed between table rows
- Popins refers to the whole table row

| Flight Data | | | |
|---------------------|------|------------|---------|
| | No. | Date | Airfare |
| AA | 0017 | 12.01.2004 | 1 |
| AA | 0017 | 28.04.2004 | 422 |
| Max. capacity econ. | | 385 | Occupie |
| AA | 0017 | 26.05.2004 | 422 |
| AA | 0017 | 23.06.2004 | 422 |
| AA | 0017 | 21.07.2004 | 422 |

Row 1 of 100

Table *Column* Popin (also named *Cell* Popin)

- Insert popin container displayed between table rows
- Popin refers to a specific cell in a table column
- Associated cell is assigned the background color of the popin

| Flight Data | | | | | |
|-------------------------|------|------------|---------|-------|---------|
| | | Date | Airfare | Curr. | Plane |
| AA | 0017 | 12.01.2004 | 1,00 | USD | ppp |
| AA | 0017 | 28.04.2004 | 422,94 | USD | 747-400 |
| Carrier CONNID Popin | | | | | |
| AA | 0017 | 26.05.2004 | 422,94 | USD | 747-400 |
| AA | 0017 | 23.06.2004 | 422,94 | USD | 747-400 |
| AA | 0017 | 21.07.2004 | 422,94 | USD | 747-400 |

Row 1 of 100

Enhanced Table Selection Behavior

- Support multiple selection without changing the lead selection with property `selectionMode=multiNoLead`
- Define selectable and non-selectable table rows with *Table* UI Element property `rowSelectable`

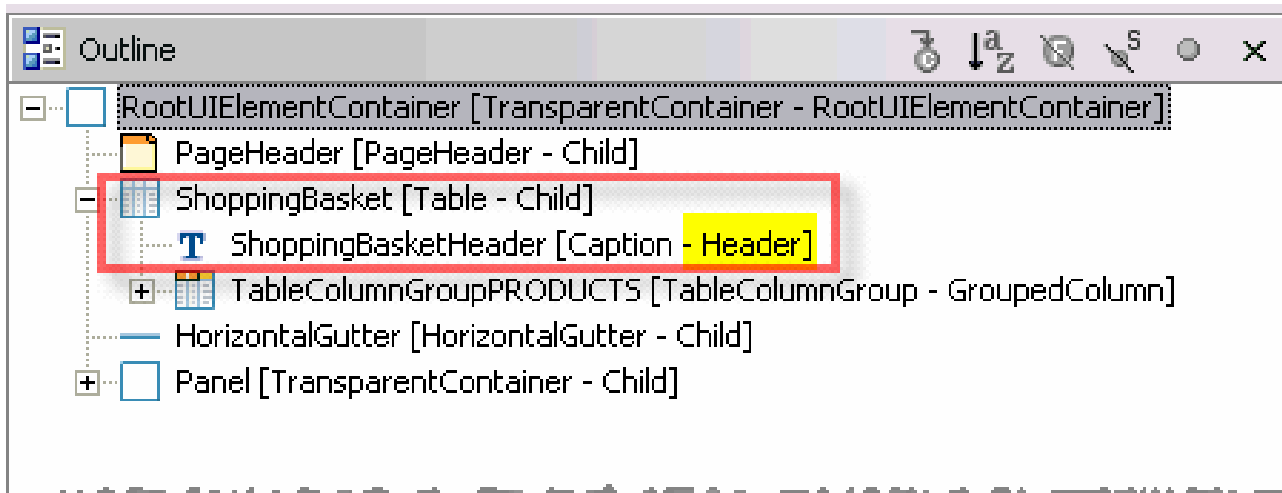
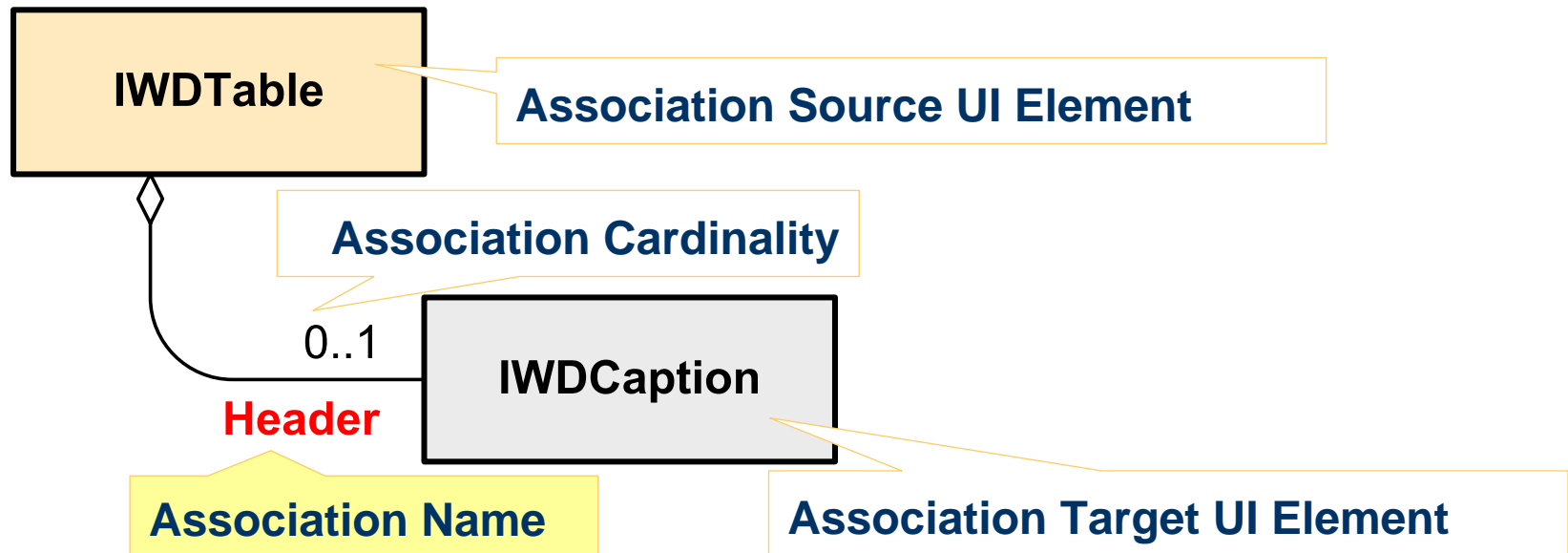
Enhanced Table Eventing

- *onFilter* Table event: `IWDTable.IWDOOnSort`
- *onSort* Table event: `IWDTable.IWDOOnFilter`
- Parameter enrichment for table's `onLeadSelectEvent`: *newRow*, *oldRow*

Further Table Enhancements

- Display Empty Table Text
- Custom extension columns for SAP NetWeaver Portal based Web Dynpro applications using an Adaptive RFC Model.

Table UI Element Associations



Adding Table UI Element Associations

The screenshot shows the 'Outline' window in SAP NetWeaver FUI Designer. The tree structure is as follows:

- RootUIElementContainer [TransparentContainer - RootUIElementContainer]
 - PageHeader [PageHeader - Child]
 - ShoppingBasket [Table - Child]
 - ShoppingBasketHeader [C...]
 - TableColumnGroupPRODU... [TableColumnGroup - Child]
 - HorizontalGutter [HorizontalG...]
 - Panel [TransparentContainer - Child]

A context menu is open over the 'TableColumnGroupPRODU...' element. The menu items are:

- Delete
- Move Up
- Move Down
- Cut
- Copy
- Paste
- Apply Template
- Create Binding
- Migrate Columns
- Translatable Texts
- Insert Column** (highlighted in yellow)
- Insert GroupedColumn
- Insert Header
- Insert LegendPopin
- Insert MasterColumn
- Insert Popin
- Insert ToolBar
- Show Help
- Properties

**UI Element
Associations**

**UI element association “Column”
is deprecated. Use
“GroupedColumn” instead**

Web Dynpro Table APIs

Abstract APIs

`IWDAbstractMasterTableColumn`

`IWDAbstractTableCellVariant`

`IWDAbstractTableColumn`

`IWDAbstractTreeTableColumn`

`IWDTable`

`IWDTableColumn`

`IWDTableCellEditor`

`IWDTableColumnGroup`

`IWDTableMarkableCellEditor`

`IWDTablePopin`

`IWDTablePopinToggleCell`

`IWDTableSingleMarkableCell`

`IWDTableStandardCell`

`IWDTreeByNestingTableColumn`

`WDTTableMethods`

`WDTTableMethods.IWDTableMethods`

View Element APIs

`WDTTableCellDesign`

`WDTTableColumnFixedPosition`

`WDTTableColumnHAlign`

`WDTTableColumnSortDirection`

`WDTTableCompatibilityMode`

`WDTTableDesign`

`WDTTableGridMode`

`WDTTablePopinTitleDesign`

`WDTTableSelectionChangeBehaviour`

`WDTTableSelectionMode`

`IWDAbstractTreeTableColumn`

`.IWDOnLoadChildren`

`IWDAbstractTableColumn`

`.IWDOnAction`

`IWDTable.IWDOnSort`

`IWDTable.IWDOnLeadSelect`

`IWDTablePopinToggleCell`

`.IWDOnToggle`

Enumerations

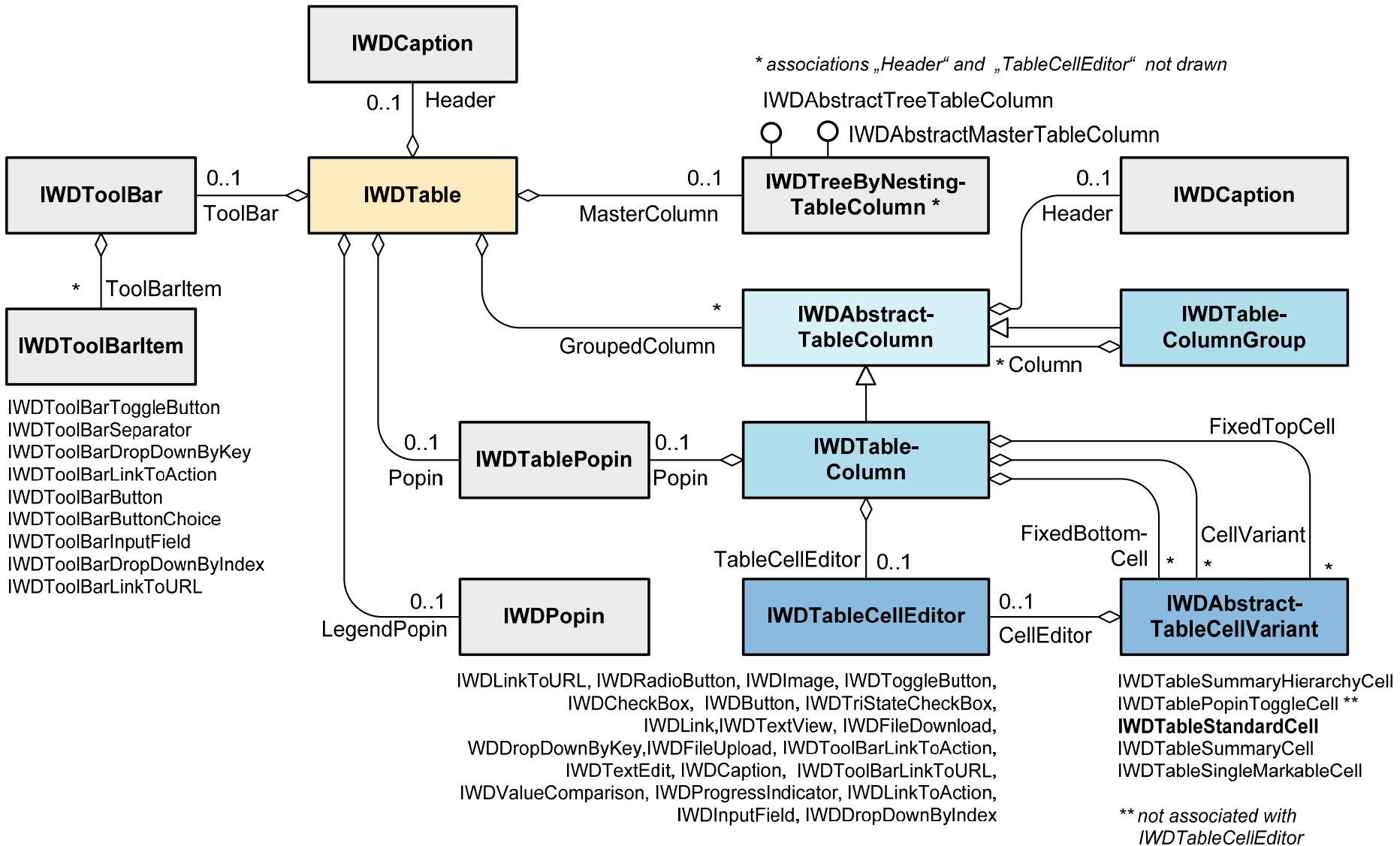
Events

Helper Class

In **Web Dynpro Java Runtime API - package**

`com.sap.tc.webdynpro.clientserver.uielib.standard.api`

Web Dynpro Table Metamodel in SAP NetWeaver 7.0



With the 0:n association **CellVariant** of the *TableColumn* UI element, you can display different cell editors in the same table column.

Table Cell Variant
= InputField

Table Cell Variant
= TextView

Table Cell Design

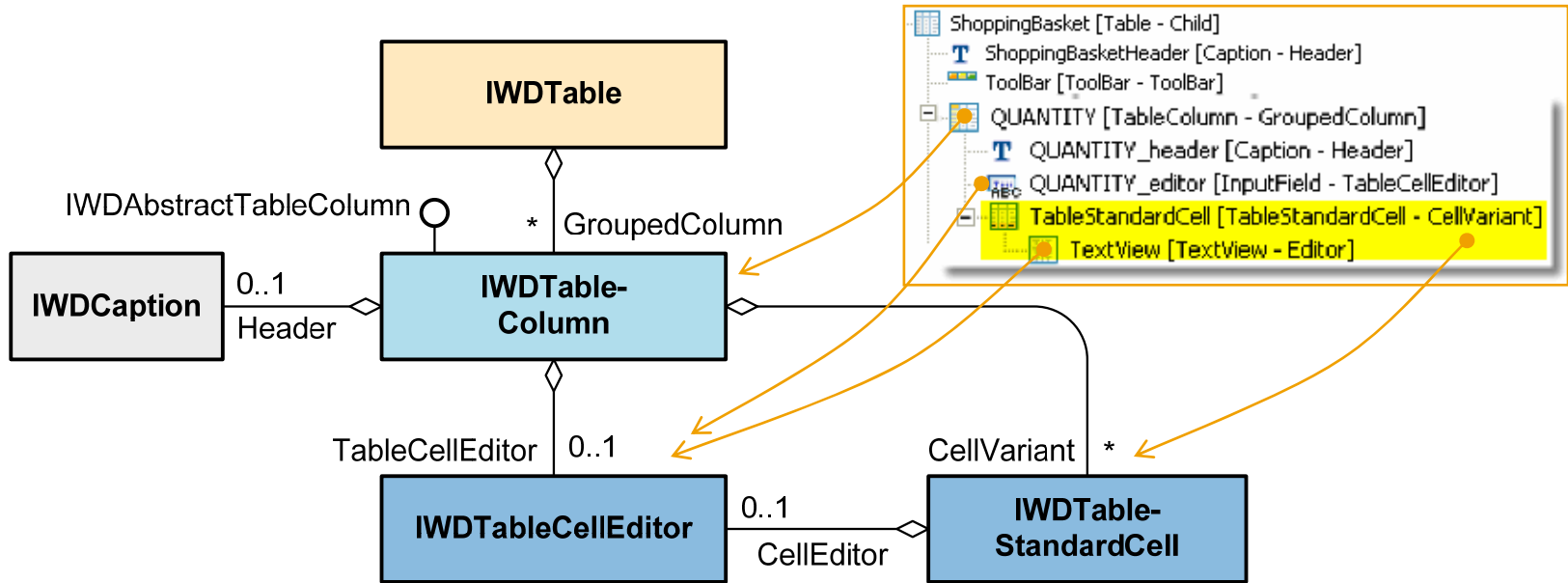
Using Cell Variants in Web Dynpro Tables

Natural Clothes: Online Shop

| | Quantity | Article | Color | Price (€) | Total per article (€) |
|------|----------|------------|---------|-----------|-----------------------|
| Sold | | jacket | blue | 34,60 | 0,00 |
| | 0 | skirt | red | 24,95 | 0,00 |
| | 0 | t-shirt | orange | 29,90 | 0,00 |
| | 0 | trousers | black | 64,90 | 0,00 |
| Sold | | top | black | 44,90 | 0,00 |
| Sold | | dress | colored | 78,90 | 0,00 |
| | 0 | blouse | white | 35,50 | 0,00 |
| | 0 | jeans | blue | 89,90 | 0,00 |
| Sold | | pullover | red | 69,00 | 0,00 |
| | 0 | sweatshirt | green | 61,60 | 0,00 |

Zeile 1 von 25

In a typical scenario you define an *InputField* UI element as the **standard cell editor** and a *TextView* UI element for the associated cell variant of type *TableStandardCell*.



IWDLinkToURL, IWDRadioButton, IWDCheckBox, IWDButton, IWDTriStateCheckBox, IWDLink, IWDDownload, IWDDropDownByKey, IWDFileUpload, IWDToolBarLinkToAction, IWDTextView, IWDFileDownload, IWDTTextEdit, IWDCaption, IWDToolBarLinkToURL, IWDDropDownByIndex, IWDFileUpload, IWDToolBarLinkToAction, IWDTTextEdit, IWDCaption, IWDToolBarLinkToURL, IWDDropDownByIndex, IWDFileUpload, IWDToolBarLinkToAction, IWDTTextEdit, IWDCaption, IWDToolBarLinkToURL, IWDDropDownByIndex

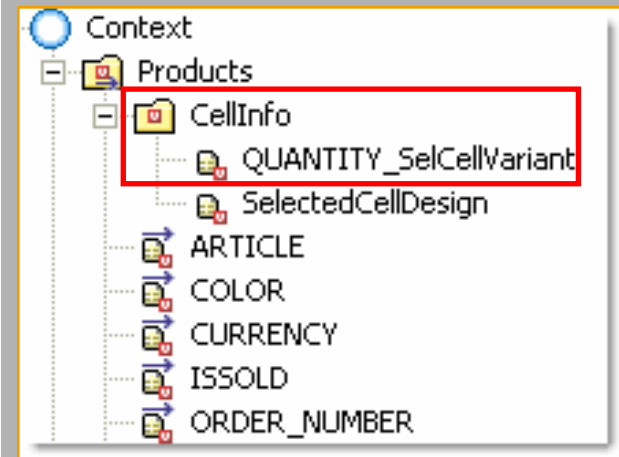
* skipped association „Column“ from IWDCaption to IWDTTableColumnGroup, skipped associations „FixedBottomCell“ and „FixedTopCell“ from IWDTTableColumn to IWDCaption, skipped association „Popin“ from IWDTTableColumn to IWDTTablePopin

Adding a Cell Variant to a Table Column (1)

1

Define Required Context Elements

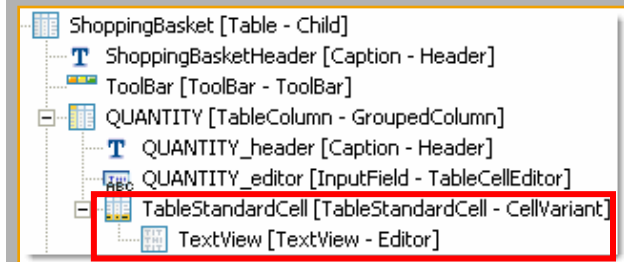
- Every table node element must be enriched with a context attribute storing the selected cell variant ID per node element.
- Add **non-singleton** node **CellInfo** with cardinality 1..1 to (mapped) table data node.
- Add **calculated context attribute** of type *String* to get variantKey per node element (row) at runtime.



2

Add UI Elements to View Layout

- Add **CellVariant** of type *TableStandardCell* to Table Column.
- Add cell editor of type *TextView* to *TableStandardCell* cell variant.

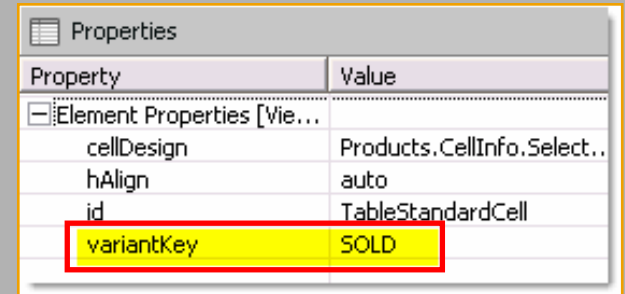


Adding a Cell Variant to a Table Column (2)

3

Assign Variant Key to Added Cell Variant

- Every cell variant added to a table column must be specified with a corresponding variant key.
- Set property **variantKey** for the added cell variant of type *TableStandardCell*.
- **Example**
variantKey = SOLD

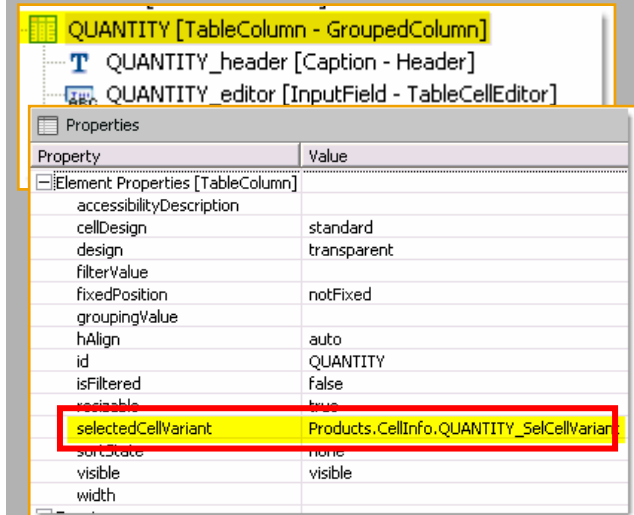


| Property | Value |
|------------------------------|-----------------------------|
| Element Properties [View...] | |
| cellDesign | Products.CellInfo.Select... |
| hAlign | auto |
| id | TableStandardCell |
| variantKey | SOLD |

4

Define Binding Relation from TableColumn UI Element to View Context

- Property **selected-CellVariant** of the *TableColumn* UI element defines the cell variant to be displayed per table row.
- Bind this property to the *calculated context attribute SelCellVariant*.



QUANTITY [TableColumn - GroupedColumn]

QUANTITY_header [Caption - Header]

QUANTITY_editor [InputField - TableCellEditor]

| Property | Value |
|----------------------------------|--|
| Element Properties [TableColumn] | |
| accessibilityDescription | |
| cellDesign | standard |
| design | transparent |
| filterValue | |
| fixedPosition | notFixed |
| groupingValue | |
| hAlign | auto |
| id | QUANTITY |
| isFiltered | false |
| isSelectable | true |
| selectedCellVariant | Products.CellInfo.QUANTITY_SelCellVariant |
| sortOrder | none |
| visible | visible |
| width | |

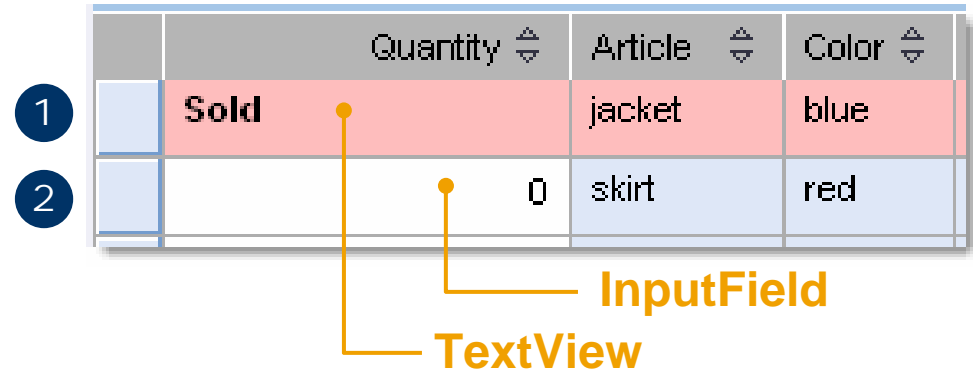
Table Cell Variants and Context Binding

Table in View Layout

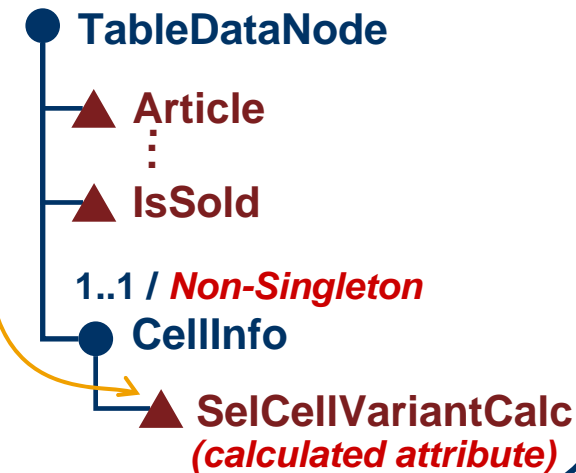


variantKey = **SOLD**
 selectedCellVariant

Table in View UI



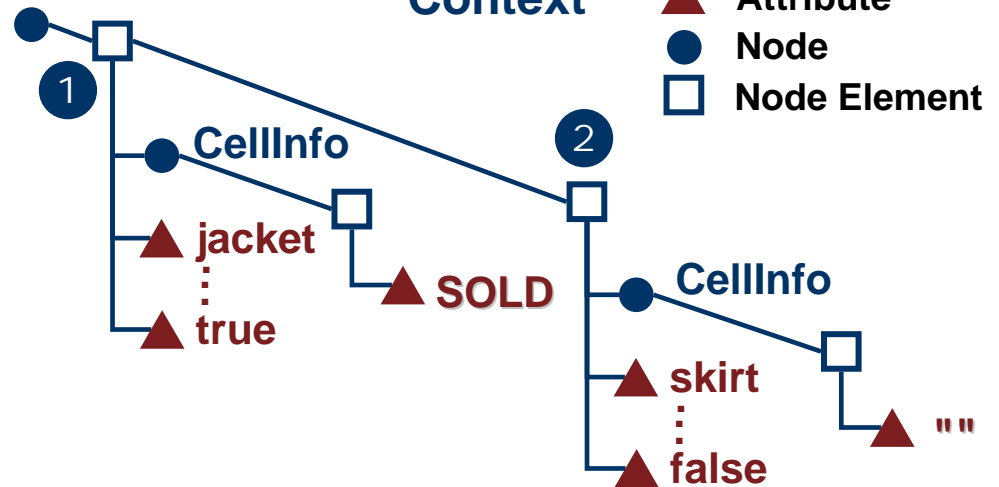
0..n Context



Design time

TableDataNode

Context

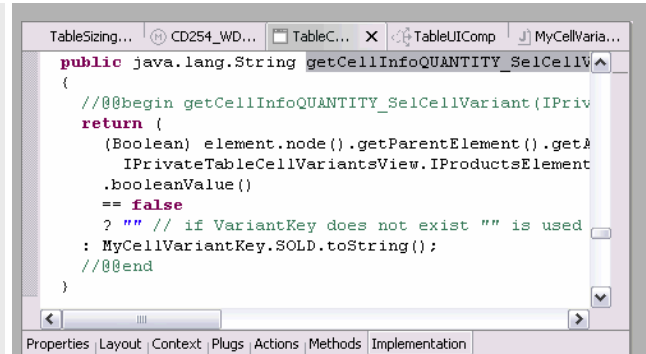


Runtime

5

Implement Calculated Context Attribute Getter Method in View Controller

- Open view controller “**Implementation**” tab.
- Implement **calculated context attribute getter** to retrieve variant keys per node element (table row).
- For a calculated context attribute **CellInfo.SelCellVariant** this getter method is named `getCellInfoSelCellVariant()`.
- Returned variant keys must match the defined cell variant keys (see step 3).
- Use empty String “” as key for standard cell editor.



```
public java.lang.String getCellInfoQUANTITY_SelCellVariant()  
{  
    //@@begin getCellInfoQUANTITY_SelCellVariant(IPriv  
    return {  
        (Boolean) element.node().getParentElement().getA  
        IPrivateTableCellVariantsView.IProductsElement  
        .booleanValue()  
        == false  
        ? "" // if VariantKey does not exist "" is used  
        : MyCellVariantKey.SOLD.toString();  
    //@@end  
}
```

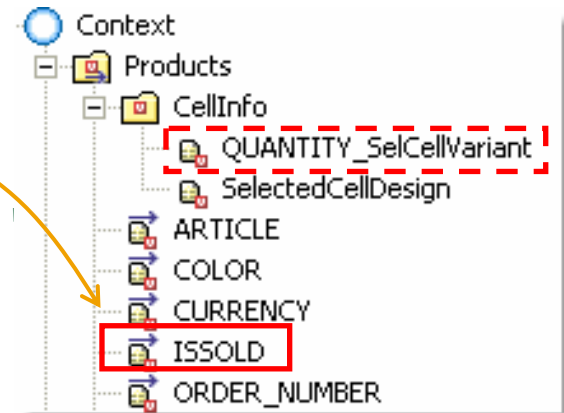
Calculating Variant Keys Per Node Element/Table Row

```
public java.lang.String getCellInfoQUANTITY_SelCellVariant(
    IPrivateTableCellVariantsView.ICellInfoElement element)
{
    //@@begin getCellInfoQUANTITY_SelCellVariant(
    //IPrivateTableCellVariantsView.ICellInfoElement)

    IPrivateTableCellVariantsView.IProductsElement prodEl =
        (IPrivateTableCellVariantsView.IProductsElement) element
            .node().getParentElement();

    return prodEl.getISSOLD()
        ? MyCellVariantKey.SOLD.toString()
        : "" // if VariantKey does not exist
        ;

    //@@end
}
```



With the property **cellDesign** defined on *TableColumn* UI Element and *table cell variant* (*TableStandardCell* UI Element) level you can change the design of the displayed table cells and consequently table rows.

How can table rows have the same cell design when using table cell variants?

Same *table cell design* for all cells in same table row

Using Cell Variants in Web Dynpro Tables

Natural Clothes: Online Shop

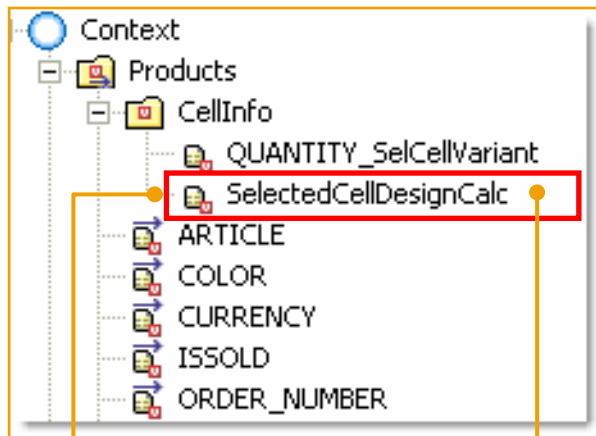
| | Quantity ⇅ | Article ⇅ | Color ⇅ | Price (€) ⇅ | Total per article (€) ⇅ |
|------|------------|------------|---------|-------------|-------------------------|
| Sold | | jacket | blue | 34,60 | 0,00 |
| 0 | | skirt | red | 24,95 | 0,00 |
| 0 | | t-shirt | orange | 29,90 | 0,00 |
| 0 | | trousers | black | 64,90 | 0,00 |
| Sold | | top | black | 44,90 | 0,00 |
| Sold | | dress | colored | 78,90 | 0,00 |
| 0 | | blouse | white | 35,50 | 0,00 |
| 0 | | jeans | blue | 89,90 | 0,00 |
| Sold | | pullover | red | 69,00 | 0,00 |
| 0 | | sweatshirt | green | 61,60 | 0,00 |

Zeile 1 von 25

1

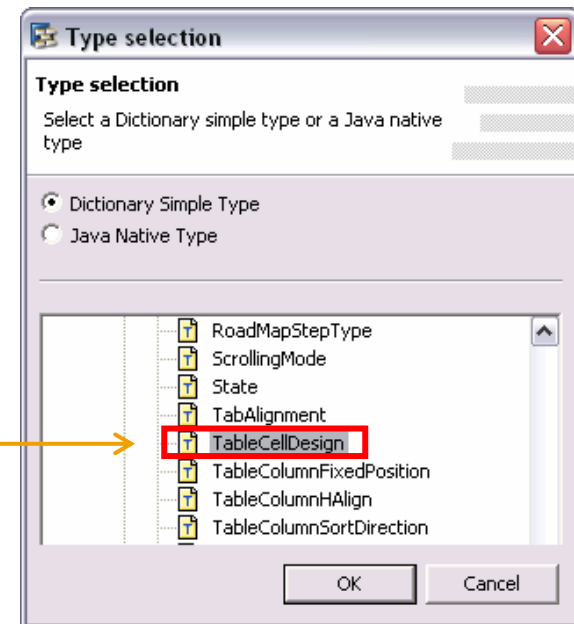
Define Required Calculated Context Attribute

- Every table node element must be enriched with a context attribute storing the **selected cell design** per node element.
- Add non-singleton node **CellInfo** with cardinality 1..1 to (mapped) data node.
- Add **calculated context attribute** of dictionary simple type `com.sap.ide.webdynpro.ui.elementdefinitions.TableCellDesign` to get cell design per node element (table row) at runtime.



Data Type

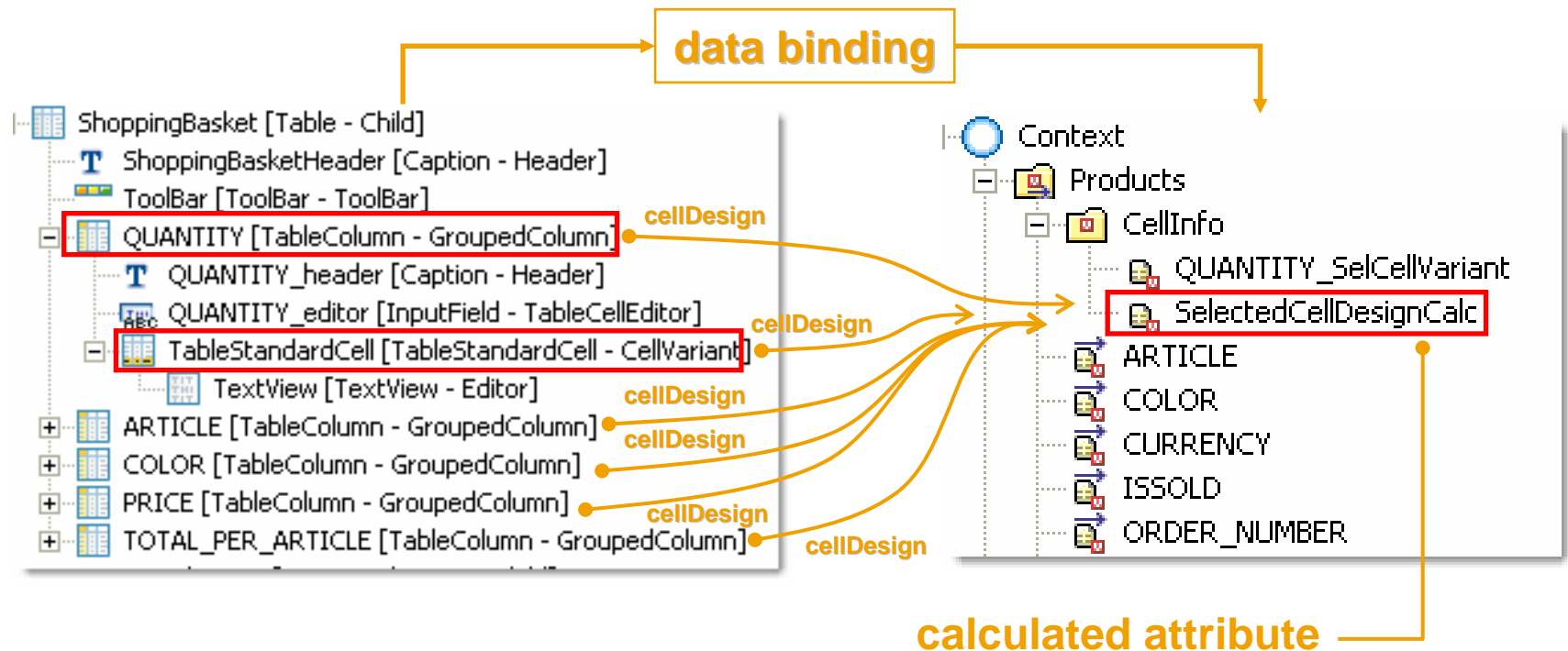
**calculated context attribute
(readOnly = true)**



2

Define Required Data Binding Relations

- Bind property **cellDesign** of all **TableColumn** UI elements to the same calculated context attribute **SelectedCellDesignCalc**.
- Bind property **cellDesign** of associated table cell variants (**TableStandardCell** UI Elements) to the same calculated context attribute **SelectedCellDesignCalc**

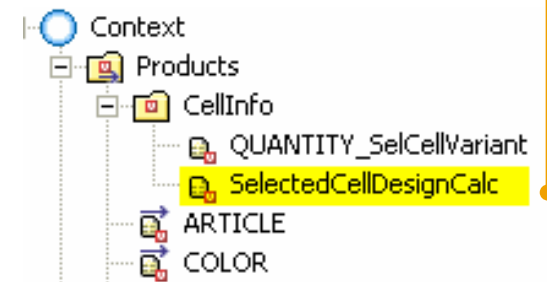


3

Implement Calculated Attribute Getter in View Controller

- Calculate cell design value of type `WTableCellDesign` for every table cell.
- All different `cellDesign` properties (see step 1) are bound to the same calculated attribute → calculated attribute getter method is invoked for every displayed table cell, or for every context attribute of all displayed table node elements.
- Make sure, that the calculated attribute getter returns the same cell design value for the same node element.

```
public com.sap.tc.webdynpro.clientserver.uielib.standard.api
    .WTableCellDesign getCellInfoSelectedCellDesignCalc(
        IPrivateTableCellVariantsView.ICellInfoElement element) {
    //@@begin getCellInfoSelectedCellDesign(
    // IPrivateTableCellVariantsView.ICellInfoElement)
    IPrivateTableCellVariantsView.IProductsElement prodEl =
        (IPrivateTableCellVariantsView.IProductsElement) element
            .node().getParentElement();
    return prodEl.getISSOLD()
        ? WTableCellDesign.BADVALUE_LIGHT
        : WTableCellDesign.STANDARD;
    //@@end
}
```





20 Minutes

EXERCISE 1

Using Table Cell
Variants



Designing Tables – Sizing, Layouting, Grouping

Sizing and Layouting Table UIs

Grouping Table Columns

| | Use Case Scenario |
|--|--|
| 1 How to size tables with <i>absolute widths</i> | <ul style="list-style-type: none">■ My table width is driven by the widths of the embedded table columns■ I want to avoid varying table widths when scrolling |
| 2 How to size tables with <i>relative widths</i> | <ul style="list-style-type: none">■ My table must span 100% width of its embedding container UI element■ I want to set minimum widths for all columns |
| 3 How to size tables with <i>fixed table layout</i> | <ul style="list-style-type: none">■ My table must be sized with absolute not relative or minimum widths■ I want to avoid varying column width when scrolling |
| 4 How to <i>wrap texts</i> in <i>Text-View</i> table cell editors | <ul style="list-style-type: none">■ I want to reduce the column width by wrapping the texts of the displayed TextView cell editors. |
| 5 How to apply <i>horizontal table scrolling</i> | <ul style="list-style-type: none">■ My table must display several columns and gets a large width even when applying <i>text wrapping</i>■ I want to display a subset of fixed columns and scroll the other column horizontally. |

Relevant Properties for Sizing Web Dynpro Tables Widths

To change the widths of Web Dynpro Tables and Table Columns three specific UI element properties must be defined:

- *Table* UI Element Property `fixedTableLayout`
- *Table* UI Element Property `width`
- *TableColumn* UI Element Property `width`

Depending on the defined combination of property values different sizing results can be achieved.

1

Table
width

500px

TableColumn

2

width

150px

50%

50%

3

Table
`fixedTableLayout`

true

| Table Header | | |
|----------------|-------------------------------|----------------|
| Column1 Header | Column2 HeaderColumn2 Head... | Column3 Header |
| Column | Column | Column |
| Column | Column | Column |
| Column | Column | Column |
| Column | Column | Column |
| Column | Column | Column |

Row 1 of 10

When to Apply *Absolute* Table Widths



Set absolute table column widths when the table width *is driven by or can increase with* the displayed table columns !

- Keeps the table width independent of the embedding container width
- Table width adapts to the width of its table column content (cell editor texts, column header texts)

Container embedding table

table must not span the container width

| Name | Ticker Symbol | Stock Market Index |
|-------------------|---------------|--------------------|
| ADIDAS | ADS.DE | DAX |
| ALTANA | ALT.DE | |
| ALLIANZ N | ALV.DE | |
| BASF | BAS.DE | |
| BAYER | BAY.DE | |
| BMW | BMW.DE | |
| COMMERZBANK | CBK.DE | |
| DEUTSCHE BOERSE N | DB1.DE | |
| DEUTSCHE BANK N | DBK.DE | |
| DAIMLERCHRYSLER N | DCX.DE | |

NOTE: Slide content only valid for *Table UI* property `fixedTableLayout = false`

Absolute Table Widths are *Minimum* Widths

With *Table* UI element property `fixedTableLayout = false` all absolute table widths values are *minimum* widths.

Without defining any table width property, your table gets **implicitly sized by its content**

- Widths of column header texts
- Widths of cell editor content in visible rows

By defining suitable minimum widths you can reduce the probability of an implicit increase of table widths based on larger header or cell editor content widths!

The top screenshot shows a table with the following structure:

| Name | Ticker Symbol | Stock Market Index |
|------|---------------|--------------------|
| | | |
| | | |
| | | |
| | | |

The bottom screenshot shows the same table with data rows:

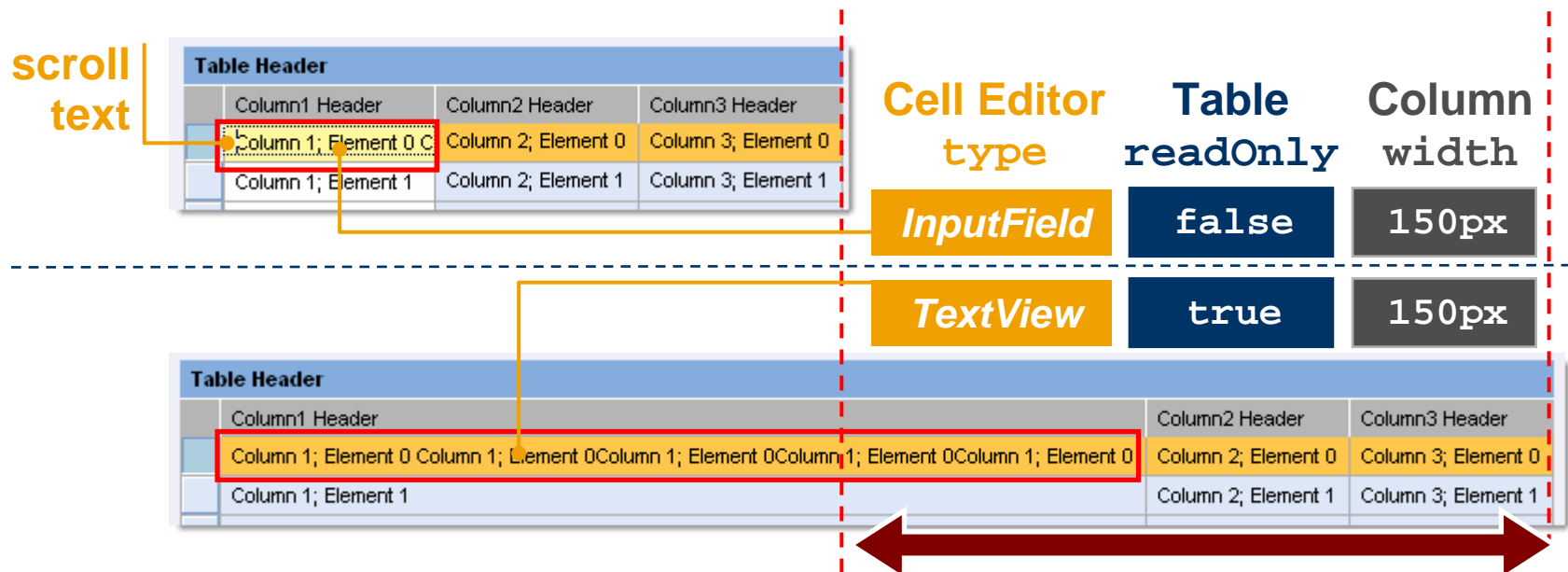
| Name | Ticker Symbol | Stock Market Index |
|---------------------|---------------|--------------------|
| QCOM QUALCOMM | QCOM | NASDAQ |
| QLGC QLOGIC | QLGC | |
| RIMM RESEARCH MOTIO | RIMM | |
| ROST ROSS INC | ROST | |

NOTE: Slide content only valid for *Table* UI property `fixedTableLayout = false`

The Cell Editor Type Can Affect the Columns Width

The type of the rendered table cell editor can affect the column width (with *Table UI* element property `fixedTableLayout=false`):

- In a *readOnly* table an *InputField* cell editor is rendered as a *TextView*.
- In an *InputField* You can scroll the “*hidden*” text content → the absolute table column size does not get enlarged and remains 150px
- In a *TextView* You cannot scroll → the absolute (*minimum*) table column size gets enlarged → the table width gets enlarged
- **SOLUTION:** Set *Table UI* property `fixedTableLayout` to `true`



NOTE: Slide content only valid for *Table UI* property `fixedTableLayout = false`



Implicit Sizing of
Table Widths

*Absolute Widths
are Minimum Widths*

DEMO

When to Apply *Relative* Table Widths



Set a relative table width when the table width is driven by the width of its parent container but not of the table content!

- Keeps the table width relative to the width of the embedding container
- Table width only adapts to the width of its table columns when “large” column header texts or cell content gets displayed.

Ticker Symbols

Index: DAX Search

| Name | Ticker Symbol | Stock Market Index |
|-------------------|---------------|--------------------|
| ADIDAS | ADS.DE | DAX |
| ALTANA | ALT.DE | |
| ALLIANZ N | ALV.DE | |
| BASF | BAS.DE | |
| BAYER | BAY.DE | |
| BMW | BMW.DE | |
| COMMERZBANK | CBK.DE | |
| DEUTSCHE BOERSE N | DB1.DE | |
| DEUTSCHE BANK N | DBK.DE | |
| DAIMLERCHRYSLER N | DCX.DE | |

Container
embedding
table

width = 100%

table must span the container width

Sizing Tables with *Relative* Widths

A relative table width is controlled by the *defined* width of its embedding (scroll) container

When defining *relative* widths a table gets still implicitly sized by its content

With *Table* UI element property `fixedTableLayout = false` all *relative* table width values are *minimum* widths related to the *absolute* width of the embedding container

Set the *Table* UI element width to 100% to span the whole width of the embedding container by default!

| Sizing Web Dynpro Table Widths | | |
|--------------------------------|---------------------|---------------------|
| ON/OFF | | |
| Table Header | | |
| Column1 Header | Column2 Header | Column3 Header |
| Column 1; Element 5 | Column 2; Element 5 | Column 3; Element 5 |
| Column 1; Element 6 | Column 2; Element 6 | Column 3; Element 6 |
| Column 1; Element 7 | Column 2; Element 7 | Column 3; Element 7 |
| Column 1; Element 8 | Column 2; Element 8 | Column 3; Element 8 |
| Column 1; Element 9 | Column 2; Element 9 | Column 3; Element 9 |

width = 100%

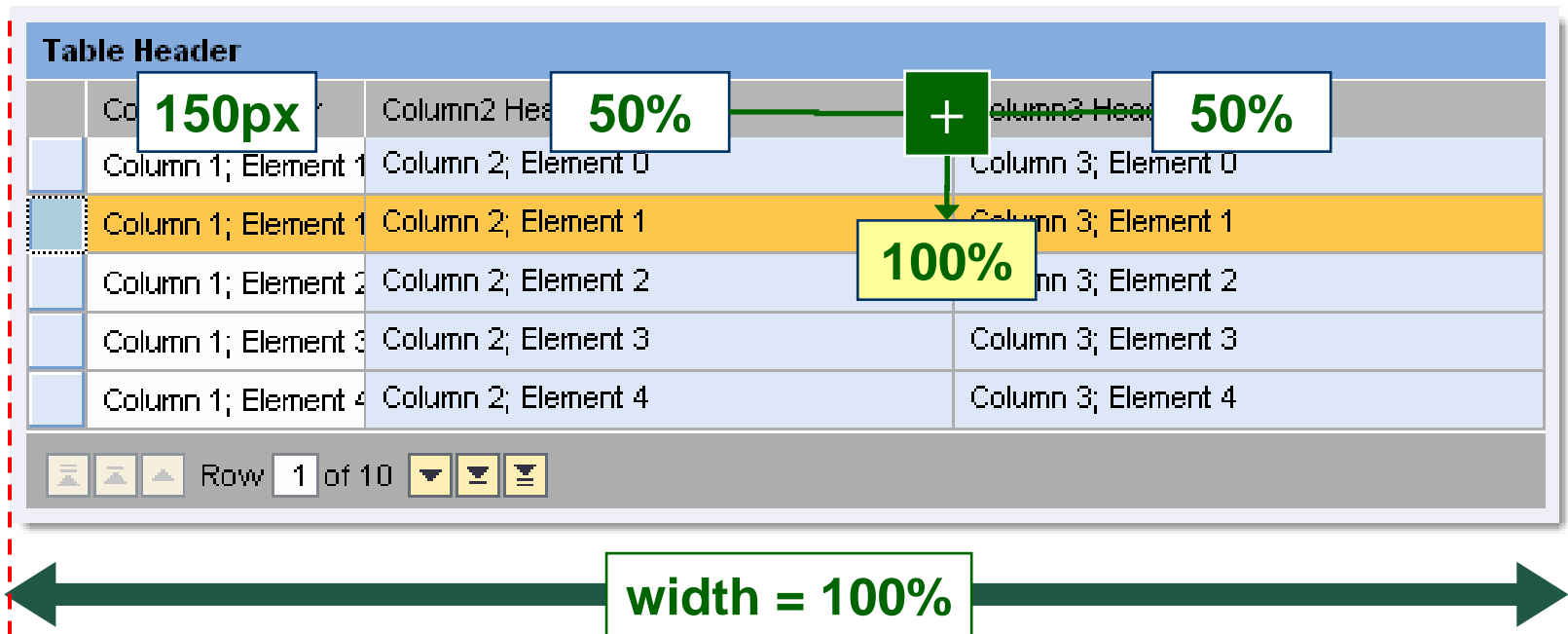
| Table Header |
|---|
| Column1 Header |
| Cell Text Cell Text Cell Text Cell Text Cell Text Cell Text Cell Text Cell Text Cell Text |
| Column 1; Element 6 |
| Column 1; Element 7 |
| Column 1; Element 8 |
| Column 1; Element 9 |

NOTE: Slide content only valid for *Table* UI property `fixedTableLayout = false`

Combining *Absolute* with *Relative* Table Widths (1)

You can combine *relative* and *absolute* widths in the UI elements *Table* and *TableColumn*.

- Set **relative** table width
- Set **absolute** table width
- Set some columns with *absolute* widths
- Set other columns with *relative* widths summed up to 100%



NOTE: Slide content only valid for *Table* UI property `fixedTableLayout = false`



Relative Table Widths
Mixing *Relative* with
Absolute Widths

DEMO

Text Wrapping of Cell Editor Content in Table Columns

Apply text wrapping of *TextView* cell editor content to ...

- keep the column width close to the defined minimum widths, as long as cell text does not enlarge it,
- *avoid* or *delay* horizontal scrolling based on long texts.

NOTE: Text wrapping only works for text containing whitespace

| Column1 Header | Column2 | Column3 Header |
|---------------------|---------------------|---------------------|
| Column 1; Element 0 | Column 2; Element 0 | Column 3; Element 0 |
| Column 1; Element 1 | Column 2; Element 1 | Column 3; Element 1 |

TextView textWrapping **true** false

| Column1 Header | Column2 | Column3 Header |
|---------------------|---------------------|---------------------|
| Column 1; Element 0 | Column 2; Element 0 | Column 3; Element 0 |
| Column 1; Element 1 | Column 2; Element 1 | Column 3; Element 1 |
| Column 1; Element 2 | Column 2; Element 2 | Column 3; Element 2 |
| Column 1; Element 3 | Column 2; Element 3 | Column 3; Element 3 |
| Column 1; Element 4 | Column 2; Element 4 | Column 3; Element 4 |

TextView textWrapping **false** false

NOTE: Slide content only valid for *Table UI* property `fixedTableLayout = false`

Using a Fixed Table Layout

By setting the *Table UI* element property `fixedTableLayout` to `true` you can achieve *fixed widths* instead of *minimum table (column) widths*.

- PRO: Avoid / Delay horizontal scrolling based on long texts in table
- CON: Hidden cell editor or column header texts are displayed in tooltips
- CON: Cannot be combined with cell editor text wrapping

The diagram shows a table with three columns and four rows. The first column is labeled 'Column 1 Header', the second 'Column 2 Header', and the third 'Column 3 Header'. The first row of data contains 'Column 1; Element 0', 'Column 2; Element 0', and 'Column 3; Element 0'. The second row contains 'Column 1; Element 1', 'Column 2; Element 1', and 'Column 3; Element 1'. The third row contains 'Column 1; Element 2', 'Column 2; Element 2', and 'Column 3; Element 2'. The fourth row contains 'Column 1; Element 4', 'Column 2; Element 4', and 'Column 3; Element 4'. A tooltip is shown over the first cell of the second row, displaying the text 'Column 1; Element 0 jalfjaslflajsdf lasdf'. Another tooltip is shown over the first cell of the third row, displaying the text 'Column 2 HeaderColumn2 HeaderColumn2 HeaderColumn2 HeaderColumn2 HeaderColumn2 HeaderColumn2 HeaderColumn2 Header'. Annotations include: 'TableColumn widths' pointing to the table columns, 'Table width' pointing to a box containing '500px', and three boxes containing '150px', '50%', and '50%' pointing to the first, second, and third columns respectively. A box at the bottom right contains the text 'fixedTableLayout = true'.

TableColumn widths

Table width

150px

50%

50%

500px

fixedTableLayout = true

NOTE: Slide content only valid for *Table UI* property `fixedTableLayout` = `true`

Apply scrollable and fixed table columns to reduce the number of columns being displayed

- **PROS:** Display large tables with large number of columns, without ...
 - scrolling the embedding container,
 - reading cell content in tooltips based on heavily shrunk column widths
- **CONS:**
 - User must explicitly scroll columns to see “hidden” column content
 - Potential variation of displayed table columns based on varying column content widths (see note)

Table – scrollableColCount

2

| Col 1 | Col 2 | Col 3 | Col 4 | Col 5 | Col 6 | Col 7 | Col 8 | Col 9 | Col 10 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| cell | cell | cell | cell | cell | cell | cell | cell | cell | cell |
| cell | cell | cell | cell | cell | cell | cell | cell | cell | cell |

TableColumn
fixedPosition

left

notFixed

right




Table Text Wrapping
Using Fixed Table Layout
Scrolling Table Columns

DEMO



15 Minutes

EXERCISE 2

Sizing and Layouting
Table UIs

Tables columns headers can be grouped hierarchically by “inserting” **TableColumnGroup** UI elements between **Table** and **TableColumn** UI elements:

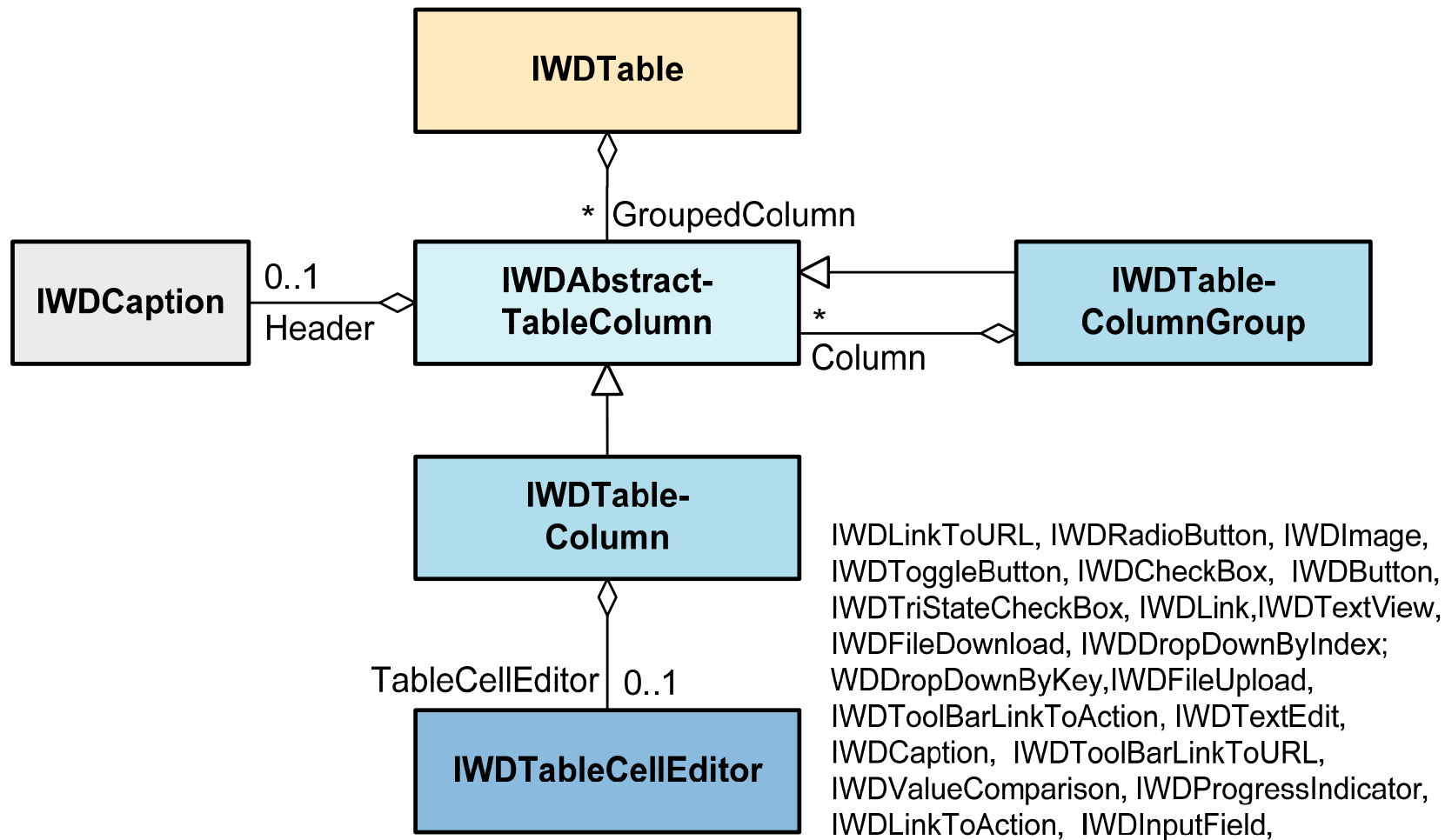
Nested Grouping of Table Columns

Natural Clothes: Online Shop

| Products | | | | | |
|-----------------|----------|--------|-----------------------|-----------|--|
| Article Details | | | Price Information | | |
| Quantity | Article | Color | Total per article (€) | Price (€) | |
| | jacket | blue | 0,00 | 34,60 | |
| | skirt | red | 0,00 | 24,95 | |
| | t-shirt | orange | 0,00 | 29,90 | |
| | trousers | black | 0,00 | 64,90 | |

- 1 TableColumnGroupPRODUCTS [TableColumnGroup - GroupedColumn]
- TableColumnGroupsPRODUCTS_Header [Caption - Header]
- 2 TableColumnGroupARTICLE [TableColumnGroup - Column]
- TableColumnGroupARTICLE_Header [Caption - Header]
- QUANTITY [TableColumn - Column]
- ARTICLE [TableColumn - Column]
- COLOR [TableColumn - Column]
- 2 TableColumnGroupPRICE [TableColumnGroup - Column]
- TableColumnGroupPRICE_Header [Caption - Header]
- TOTAL_PER_ARTICLE1 [TableColumn - Column]
- PRICE [TableColumn - Column]

Grouped Table Columns – Metamodel

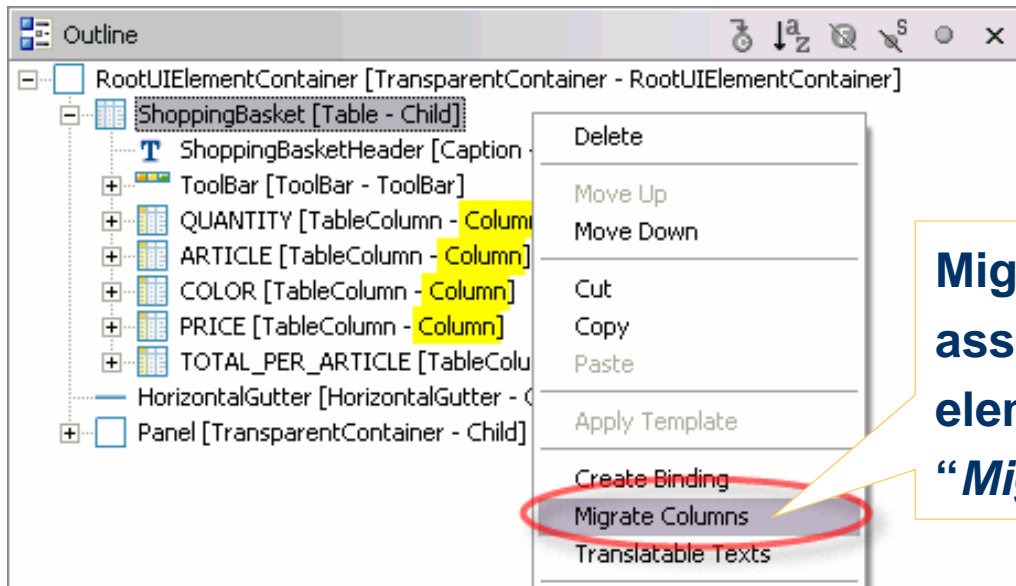


Migrating Deprecated Table UI Element Association

The UI element association *Column* from a *Table* UI element to a *TableColumn* UI element is deprecated:

- Do not use the *Table* context menu item **Insert Column** any more!
- Use the new association **GroupedColumn** instead

| Deprecated Association | New Association | Association Target UI Element |
|------------------------|----------------------|--|
| Column | GroupedColumn | <ul style="list-style-type: none">■ IWDTableColumn■ IWDTableColumnGroup |



Migrate deprecated UI element associations of the Table UI element with context function “Migrate Columns”



15 Minutes

EXERCISE 3

Adding Grouped
Table Columns



Web Dynpro Table UI Elements

Designing Tables – Sizing, Layouting, Grouping

Further Notes and Topics

Outlook – Table in SAP NetWeaver CE 7.1

Summary

Additional technical information on the Web Dynpro Table subject matter can be found in SAP Online Help and on SDN:

| Topic | Link |
|-------------------------------------|---|
| Table in SAP Online Help | http://help.sap.com/saphelp_nw2004s/helpdata/en/b8/f87842fdb70f53e10000000a155106/frameset.htm |
| Table Tutorial | SDN Tutorial “Creating a Tree Structure in a Table” https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/com.sap.km.cm.docs/library/uuid/49f2ea90-0201-0010-ce8e-de18b94aee2d |
| Trees in Table | SDN Tutorial “Creating a Tree Structure in a Table” Link URL: <i>See Table Tutorial</i> |
| Master Detail Table Tutorial | SDN Tutorial “Context Programming and Data Binding” Link URL: <i>See Table Tutorial</i> |
| Table Selection Behavior | SDN Article “Enhancing Table Performance” https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/2343 |

Additional technical information on the Web Dynpro Table subject matter can be found in SAP Online Help and on SDN:

| Topic | Link |
|--|--|
| Downloading Files in Tables on-demand | SDN Article “Uploading and Downloading Files in Web Dynpro Tables - SAP NetWeaver 7.0” https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/6603 |
| Excel Export in Tables | SDN Wiki Code Tutorial “Exporting Table Data Using On-Demand Streams - SAP NetWeaver 7.0” https://wiki.sdn.sap.com/wiki/x/0mQ |
| Sorting in Tables | SDN WebLog „Enhanced Web Dynpro Java TableSorter for SAP NetWeaver 04s“ https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/3287 |
| Vetoable Lead Selection Change | Web Dynpro Runtime API JavaDoc: <u>WDTableSelectionMode</u> |



Web Dynpro Table UI Elements

Designing Tables – Sizing, Layouting, Grouping

Further Notes and Topics

Outlook – Table in SAP NetWeaver CE 7.1

Summary

The UI element **Table** has been enhanced in SAP NetWeaver CE 7.1

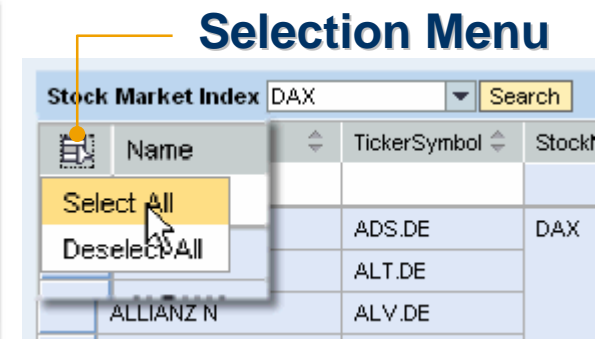
- Multi sorting ability and visualize filtered column
- Horizontal and vertical table scrolling
- Provide scroll tip for scroll status (index scroll tips)
- Wrapping of table column headers
- Generic selection menu (Select/deselect all)
- Selectable columns



A screenshot of a table with columns: Name, Street, ZipCode, City. The table is partially visible, showing rows for 'Giese Ing, Gesellschaft', 'Glöckler GMBH', 'Gora GMBH', and 'GPA-Jakob GMBH'. A scroll bar is visible on the right side of the table. A tooltip is displayed over the scroll bar, showing a large letter 'G' and the text 'Row 64'. An orange line connects this tooltip to the 'Scroll tip' label below.

| Name | Street | ZipCode | City |
|-------------------------|-------------------|---------|---------|
| Giese Ing, Gesellschaft | Schwarzwaldring20 | 76275 | Ettl... |
| Giese Ing, Gesellschaft | Schwarzwaldring20 | 76275 | Ettl... |
| Glöckler GMBH | Pforzheimer. 105 | 7505 | Br... |
| Gora GMBH | Gewerbestr.7 | 76467 | Bie... |
| GPA-Jakob GMBH | Greschbachstr.15 | 76229 | Ka... |

Scroll tip



A screenshot of a table titled 'Stock Market Index' with a search bar and a dropdown menu. The table has columns: Name, TickerSymbol, Stockf. The table is partially visible, showing rows for 'ADS.DE', 'ALT.DE', and 'ALLIANZ N', 'ALV.DE'. A selection menu is open over the table, showing 'Select All' and 'Deselect All' options. An orange line connects this menu to the 'Selection Menu' label above.

| Name | TickerSymbol | Stockf |
|-----------|--------------|--------|
| | ADS.DE | DAX |
| | ALT.DE | |
| ALLIANZ N | ALV.DE | |

Selection Menu



DEMO



Web Dynpro Table UI Elements

Designing Tables – Sizing, Layouting, Grouping

Interacting with Table UIs

Further Notes and Topics

Outlook – Table in SAP NetWeaver CE 7.1

Summary

- **Have an overview of the new Web Dynpro table functions in SAP NetWeaver 7.0**
- **Know about table layouting functions**
- **Understand how to use cell variants**
- **Know how to size tables using absolute and relative widths**
- **Have a grasp of table grouping**
- **Know about some of the functions in SAP NetWeaver CE 7.1**



SAP Public Web:

SAP Developer Network (SDN): www.sdn.sap.com and
www.sdn.sap.com/irj/sdn/developerareas/webdynpro

Business Process Expert (BPX) Community: www.bpx.sap.com



Related SAP Education and Certification Opportunities

<http://www.sap.com/education/>



Related SAP Content

[CD254 Developing State-of-the-Art Table UIs in WD Java - Exercises](#)