

# How... to insert a column into a list box

## Part 4 – Complex data types

### Requirements

- ClassiX® system installed
- Codewright or another text editor
- knowledge of:

<a href="#">basic ClassiX® design conventions</a> <a href="#">classes</a> <a href="#">inheritance</a> <a href="#">module</a>	<a href="#">widget</a> <a href="#">action list</a> <a href="#">event</a> <a href="#">message</a> <a href="#">variables</a>	<a href="#">monitor window</a> <a href="#">ObjectListView</a> <a href="#">SetFormat</a> <a href="#">object inspector</a>
---	--	---

### Task:

In the last guide, we have learned how to add simple data fields to a list box. Now, we want to find a complex data field element, understand the access mechanics and add it to the list box.

### Content

- Complex data types
- Back to practice

## Complex data types

We have seen more data types in the object inspector than just the simple INTEGER and STRING:

Datenfeld	S	Typ
allocations		REL_MN
attributeSet		POINTER
clearingObjects		REL_MN
generalTerms		REL_1M
mlDescription	⇨	MULTIPLE_STRING
monitors		REL_MN
number	⇨	STRING
pattern		REL_1M
qualityAssurance	⇨	INTEGER
read.comment	⇨	STRING
sparePart	⇨	INTEGER
structures		REL_MN
supplyEnum	⇨	ENUMSHORT
uniqueID		STRING
validity		POINTER

### Type POINTER

The pointer directly displays the address of another data object (“reference to another object”). This is a simple relation:

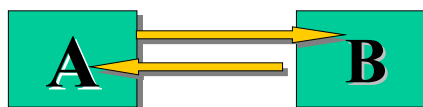
#### Pointer: Einfache Relation



### Type REL\_11

The pointer directly points to the address of another database object. This database object in turn, points back to the 1<sup>st</sup> pointer:

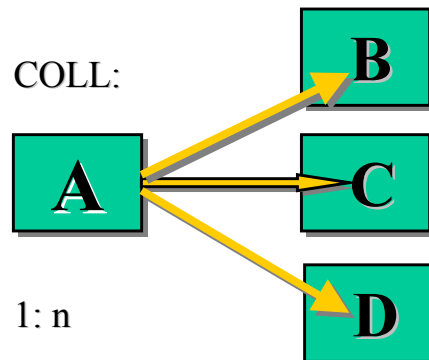
#### REL\_11:



## COLLECTIONS:

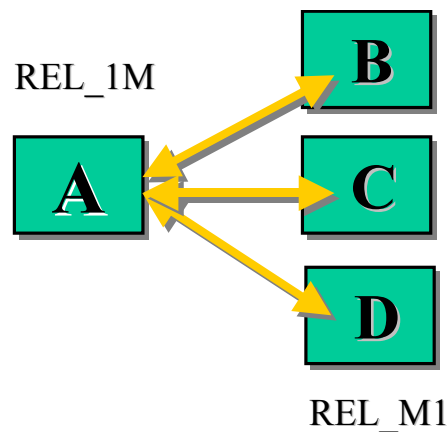
### Type COLL

This is list of pointers:



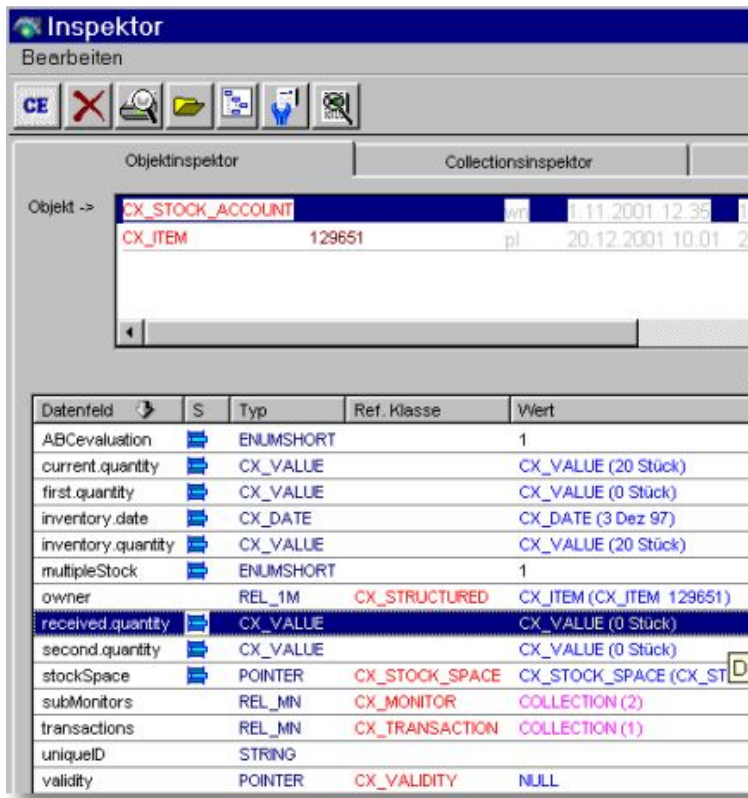
Example: Document header data element points to document lines.

### Type REL\_1M, REL\_M1, REL\_MN (multiple references)



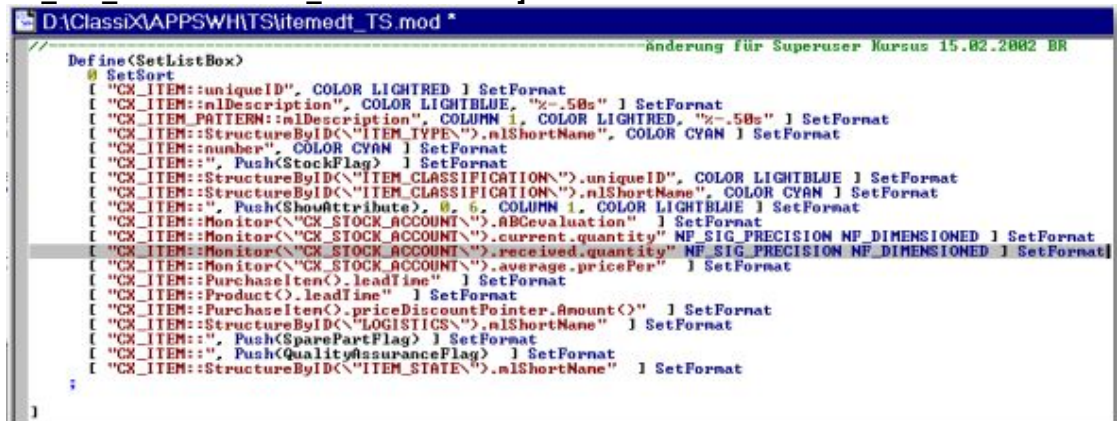
## Back to practice:

In the last “cooking recipe”, we have added a simple data field to the list box. Now, we want to display the **Eingangsmenge** in the list box. The object inspector displays no data field though, which could be identified as the **Eingangsmenge**. This data field can be found in a collection. Via double click on the data fields (data members), which have been marked as a collection, we find the field as a “received quantity” under the collection “monitors”. Most (main) data fields, which are connected to transaction data, are kept in monitors – and therefore can be found in monitors/...



If we want to add this data field to our list box, the following statement needs to be inserted into the **Define(SetListBox)** in the module itemedt\_TS:

**[ "CX\_ITEM::Monitor(\CX\_STOCK\_ACCOUNT").received.quantity"**  
**NF\_SIG\_PRECISION NF\_DIMENSIONED ] SetFormat**



The list box modification will be completed after saving the module and closing and re-opening ClassiX®.

## **Abstract:**

We have learned what pointers and collections are and also, how to search a data field, which cannot be addressed directly, but needs to be referenced with a collection instead and how to display it with SetFormat anzeigt.

## **Further information**

**Part 5 – List box **cosmetics****

**Part 6 – How to find /modify a column head line**