Database Programming with Ada

Ada Europe 2010, Valencia

Frank Piron, frank.piron@konad.de
KonAd GmbH, In der Reis 5, D-79232 March-Buchheim

Content

- Why Ada for Database Programming?
- The Database Access Library
- The User Interface Library
- Projects
- Conclusion

Why Ada for Database Programming?

The Situation in 2001

- Goal: Development of Oracle Database Applications under different platforms. High demands on stability and performance.

- By the end of the millenium Oracle stopped further development of the Developer 2000 toolset.

- Oracle Forms does not allow 100% integration of system services.

- Oracle Forms modules are single threaded.

Begin of 2002

After 6 month of evaluation → Ada

Why Ada for Database Programming?

Pro

- Similarity between Ada and PL/SQL
- Features of Ada

    Multitasking

    Object Orientation

    System Integration

    Platform Independence

    Standardization

- Ada is a language for the development of big reliable software systems

Why Ada for Database Programming?

Contra

- Ada is not a mainstream language.

  Are there Ada programmers?

  What will the customers say?

  Will Ada be available in 10 years?

- Ada is not easy to learn.
- There are only few libraries for Database Access and GUI-Programming.

Why Ada for Database Programming?

Start in 2002

Building knowledge of Ada95. Decision for GNAT.

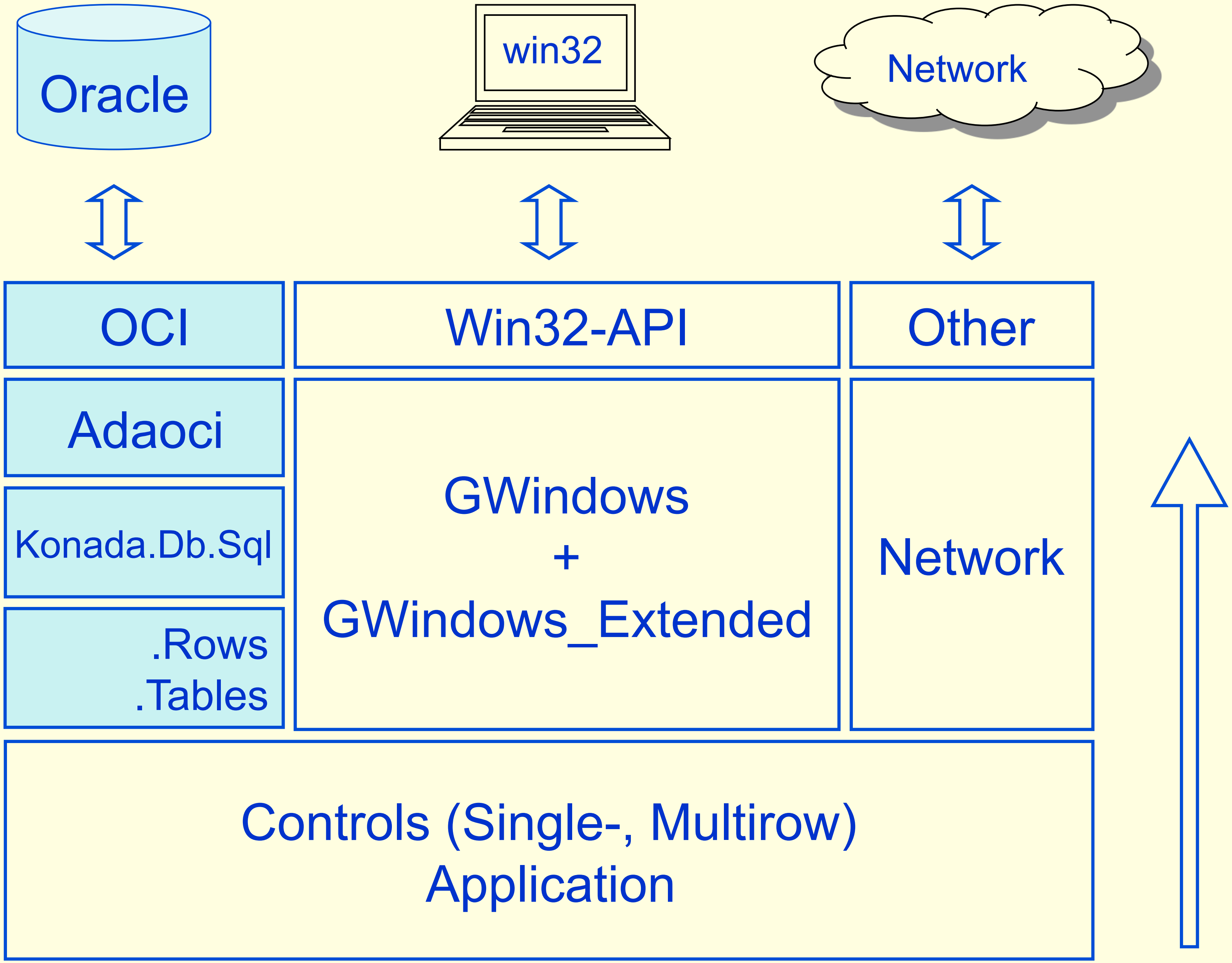
We need libraries for Oracle Database Access and GUI-Programming on Windows.

Web search and evaluation ⇨

✓ Adaoci (Dmitriy Anisimkov) as a starting point for a database access library

✓ GWindows (David Botton) as a basis for the development of GUI-Components

The Database Access Library



Oracle

win32

Network

OCI

Win32-API

Other

Adaoci

Konada.Db.Sql

.Rows
.Tables

GWindows
+
GWindows_Extended

Network

Controls (Single-, Multirow)
Application

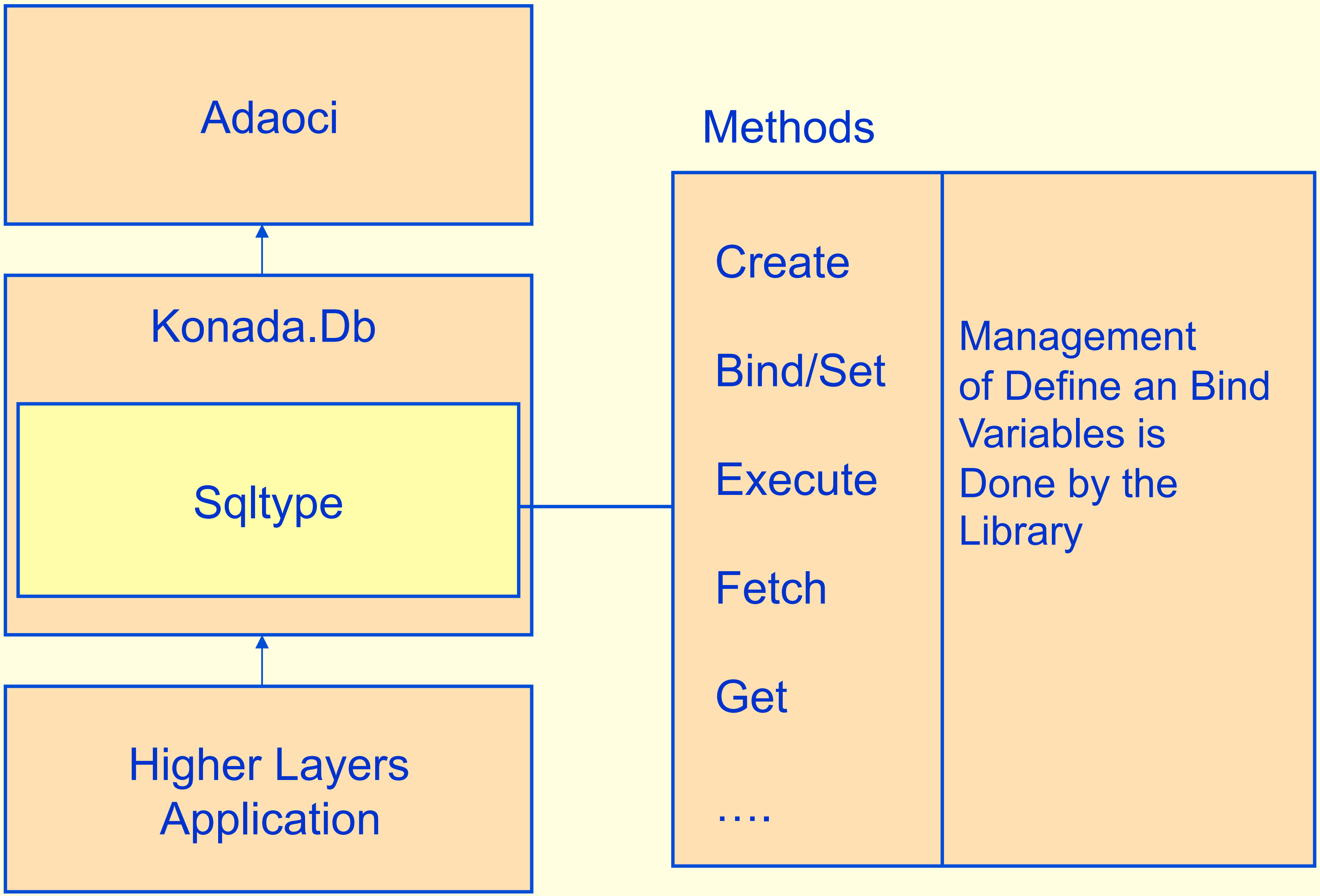
The Database Access Library

Why not e.g. GNADE for database access?

- We wanted to take full advantage of the Oracle Call Interface like
  - Asynchronous Execution
  - Blob Support
  - Advanced Queueing
- As few third party libraries as possible. Especially no ODBC driver.
- Ability to extend our library with new features of the Oracle Call Interface as soon as possible.

The Database Access Library

Konada.Db.Sql



Adaoci

Konada.Db

Sqltype

Higher Layers Application

Methods

Create

Bind/Set

Execute

Fetch

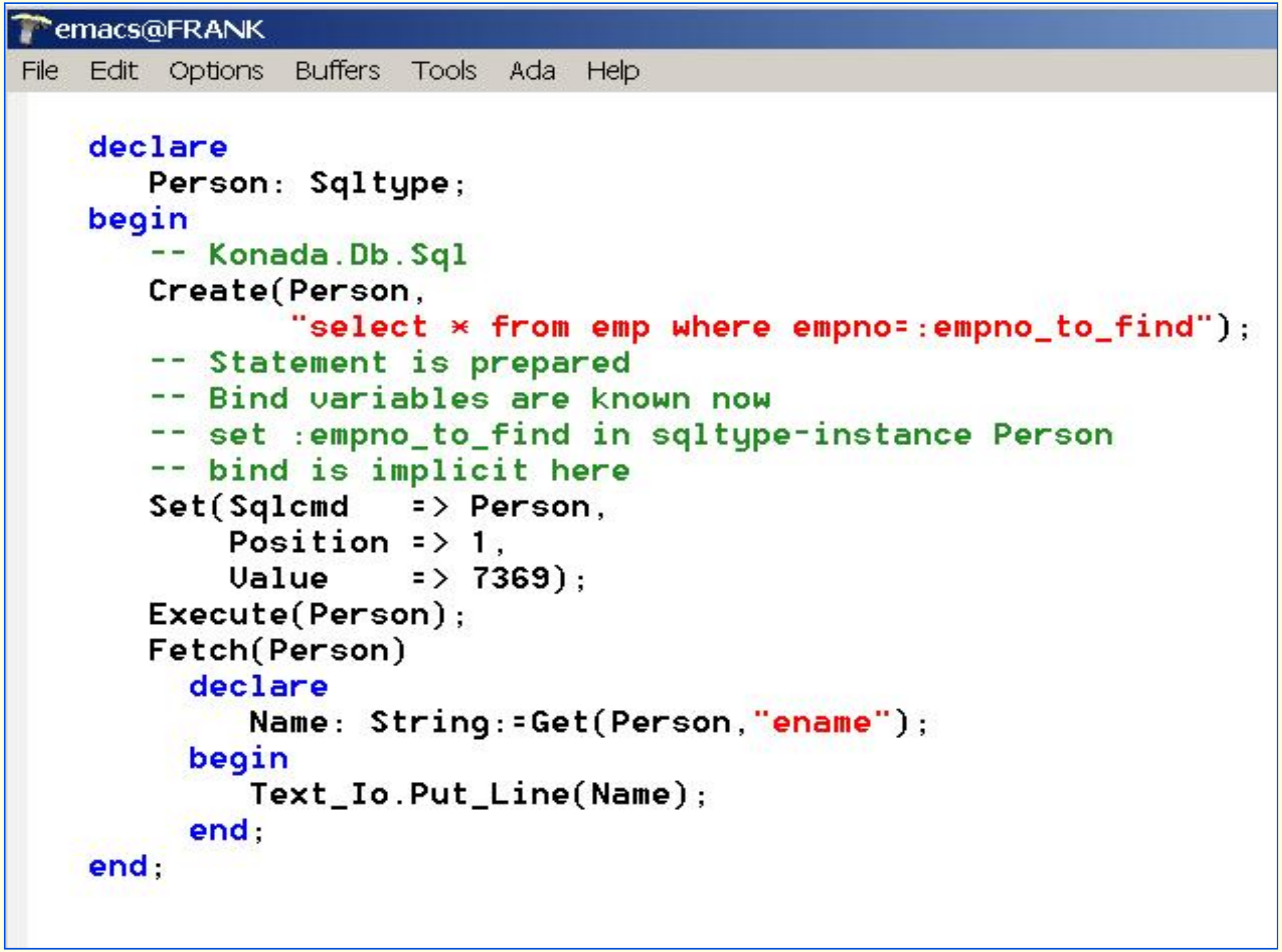
Get

….

Management of Define an Bind Variables is Done by the Library

The Database Access Library

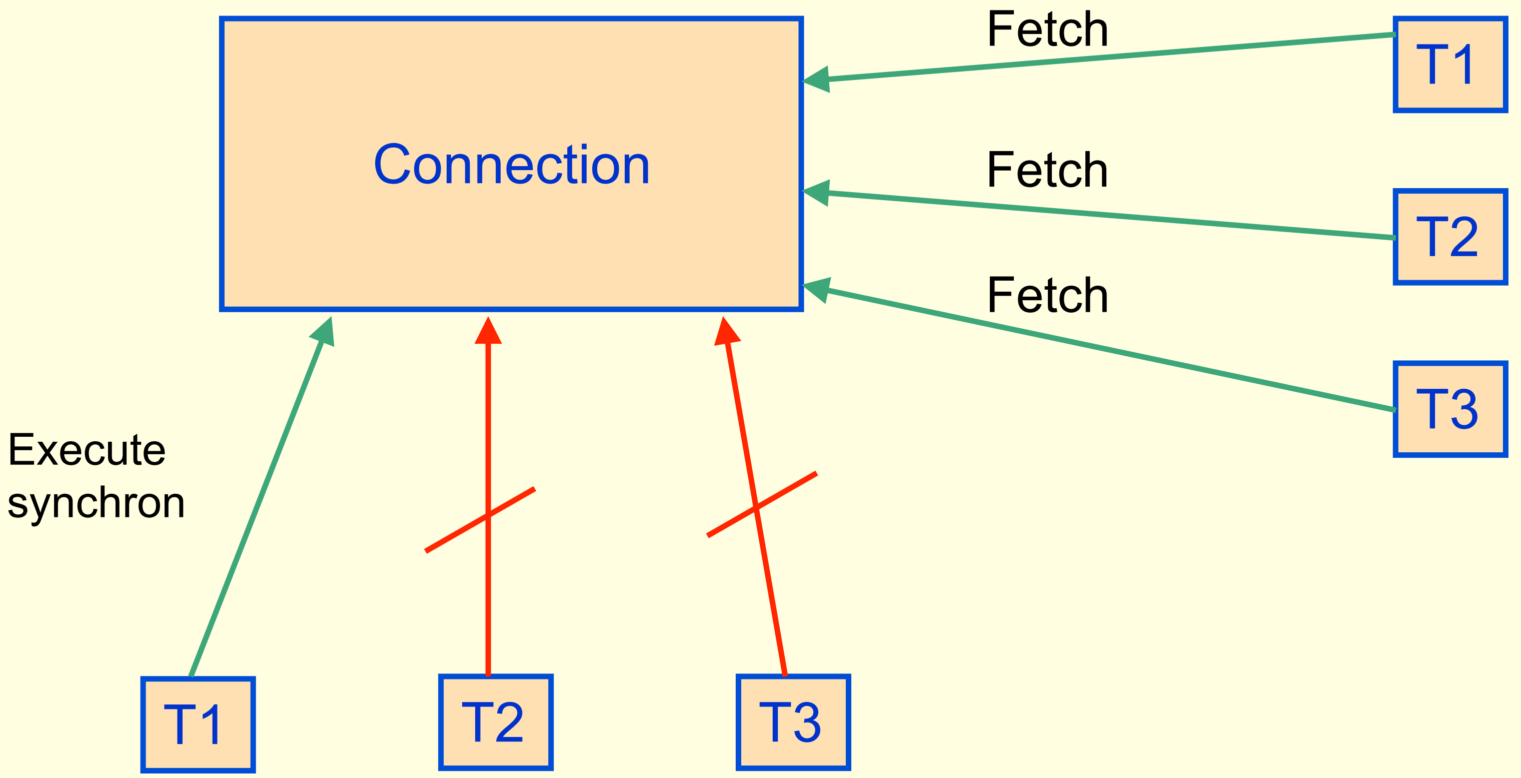
Konada.Db.Sql



```
declare
   Person: Sqltype;
begin
   -- Konada.Db.Sql
   Create(Person,
          "select * from emp where empno=:empno_to_find");
   -- Statement is prepared
   -- Bind variables are known now
   -- set :empno_to_find in sqltype-instance Person
   -- bind is implicit here
   Set(Sqlcmd   => Person,
       Position => 1,
       Value    => 7369);
   Execute(Person);
   Fetch(Person)
     declare
        Name: String:=Get(Person,"ename");
     begin
        Text_Io.Put_Line(Name);
     end;
end;
```

The Database Access Library

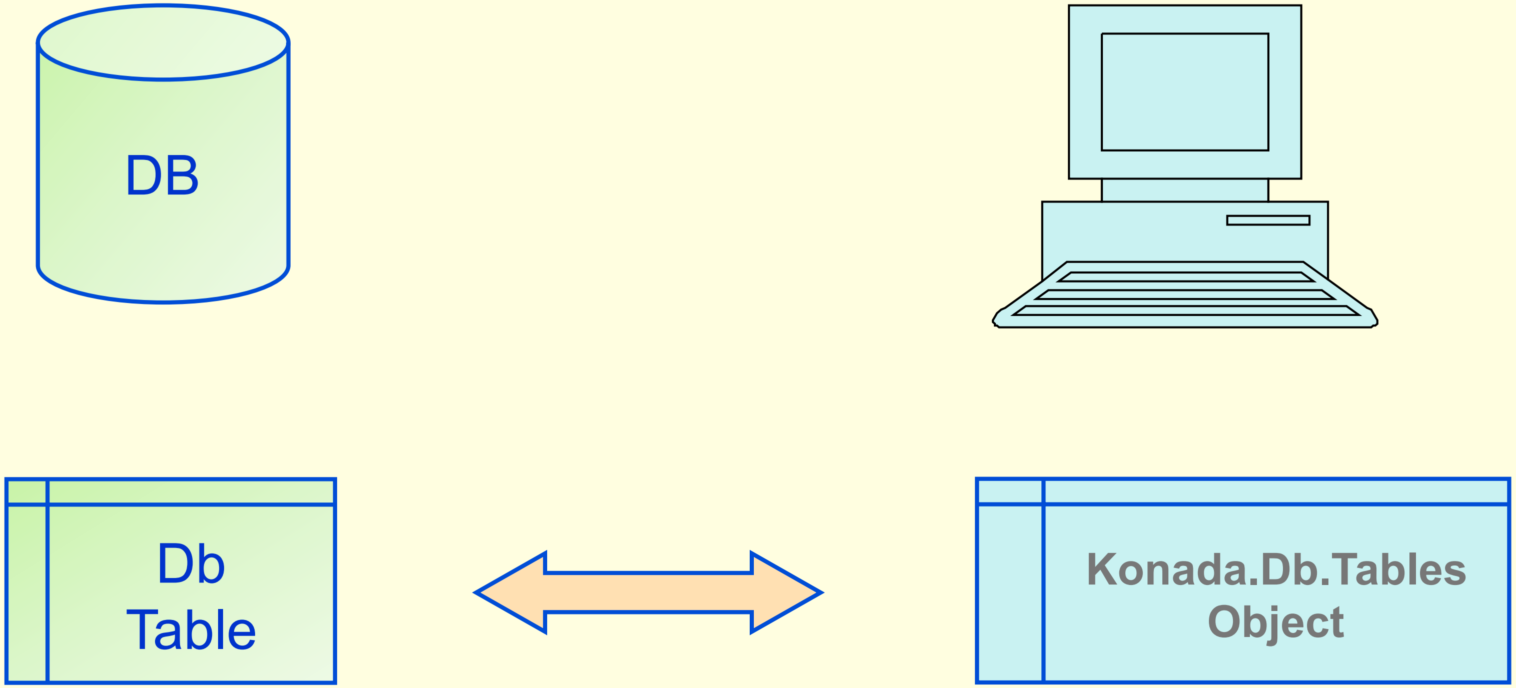
OCI and Ada Tasking



The Database Access Library

Rows and Tables

To get rid of Sql



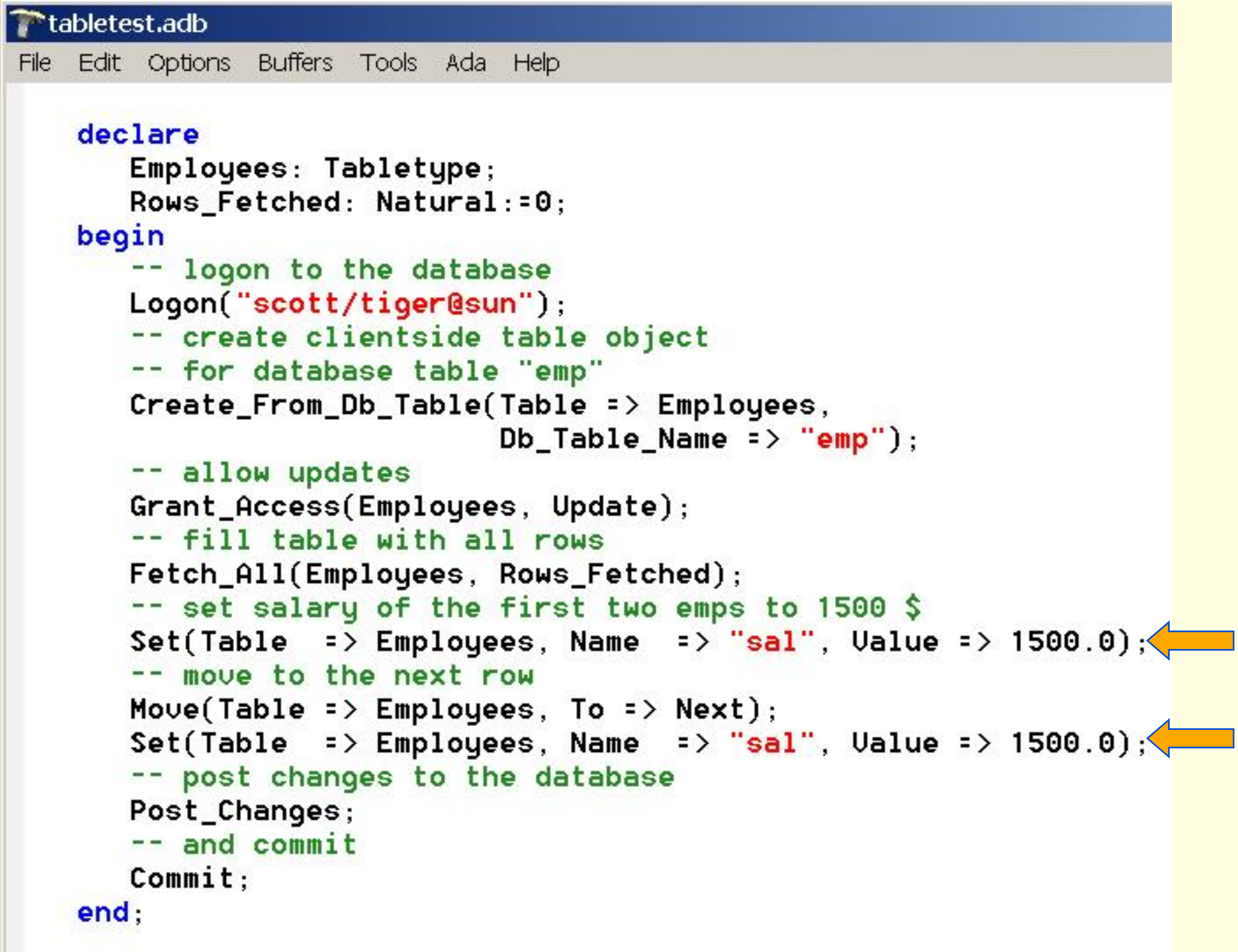
DB

Db Table

Konada.Db.Tables Object

The Database Access Library

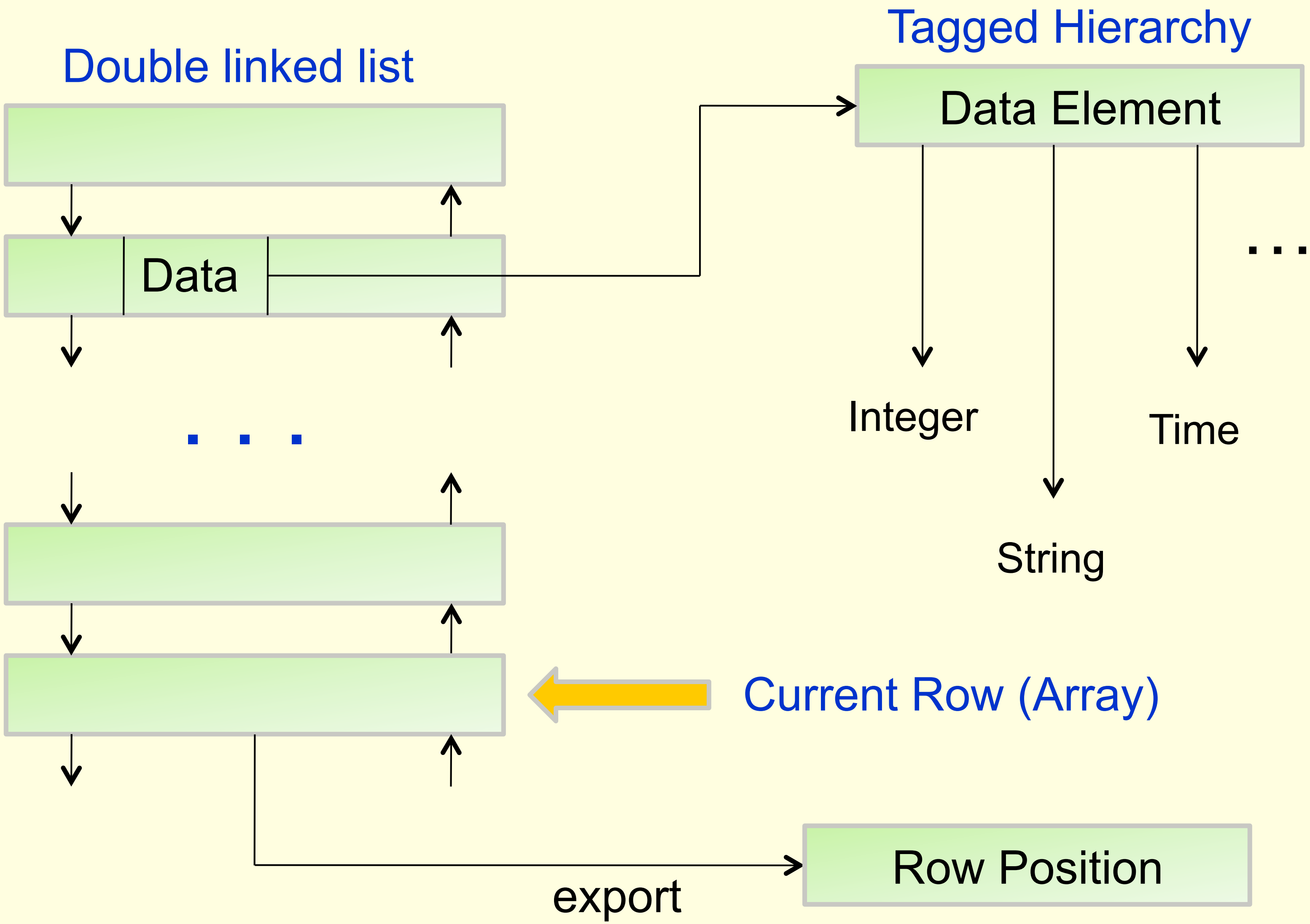
Konada.Db.Tables



```
declare
   Employees: Tabletype;
   Rows_Fetched: Natural:=0;
begin
   -- logon to the database
   Logon("scott/tiger@sun");
   -- create clientside table object
   -- for database table "emp"
   Create_From_Db_Table(Table => Employees,
                        Db_Table_Name => "emp");
   -- allow updates
   Grant_Access(Employees, Update);
   -- fill table with all rows
   Fetch_All(Employees, Rows_Fetched);
   -- set salary of the first two emps to 1500 $
   Set(Table  => Employees, Name  => "sal", Value => 1500.0);
   -- move to the next row
   Move(Table => Employees, To => Next);
   Set(Table  => Employees, Name  => "sal", Value => 1500.0);
   -- post changes to the database
   Post_Changes;
   -- and commit
   Commit;
end;
```

The Database Access Library

Konada.Db.Tables - Implementation

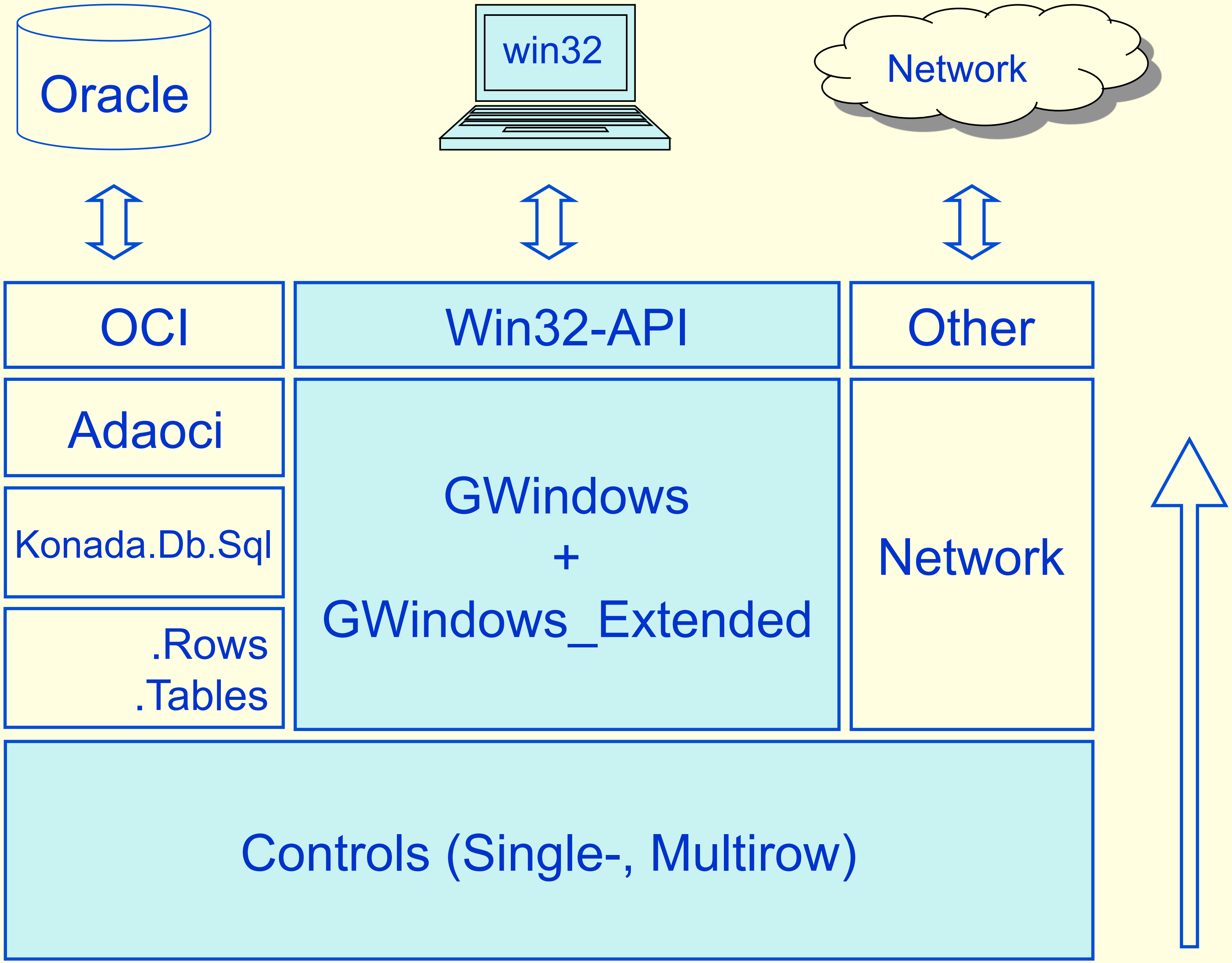


Double linked list

Tagged Hierarchy

Current Row (Array)

The User Interface Library



The User Interface Library

Goals

GUI –Components for several purposes:

- Display and manipulation of single and multiple Data Records
- Automatic and programmatic layout (no GUI-Builder)
- Flexible Event-Model
- Tree-Navigation with the tree reflecting the data model
- Storage of GUI-Layout into the database. The Modules adjust without recompilation

The User Interface Library

Why start with Gwindows?

GUI-centered thick binding to the win32-API by David Botton

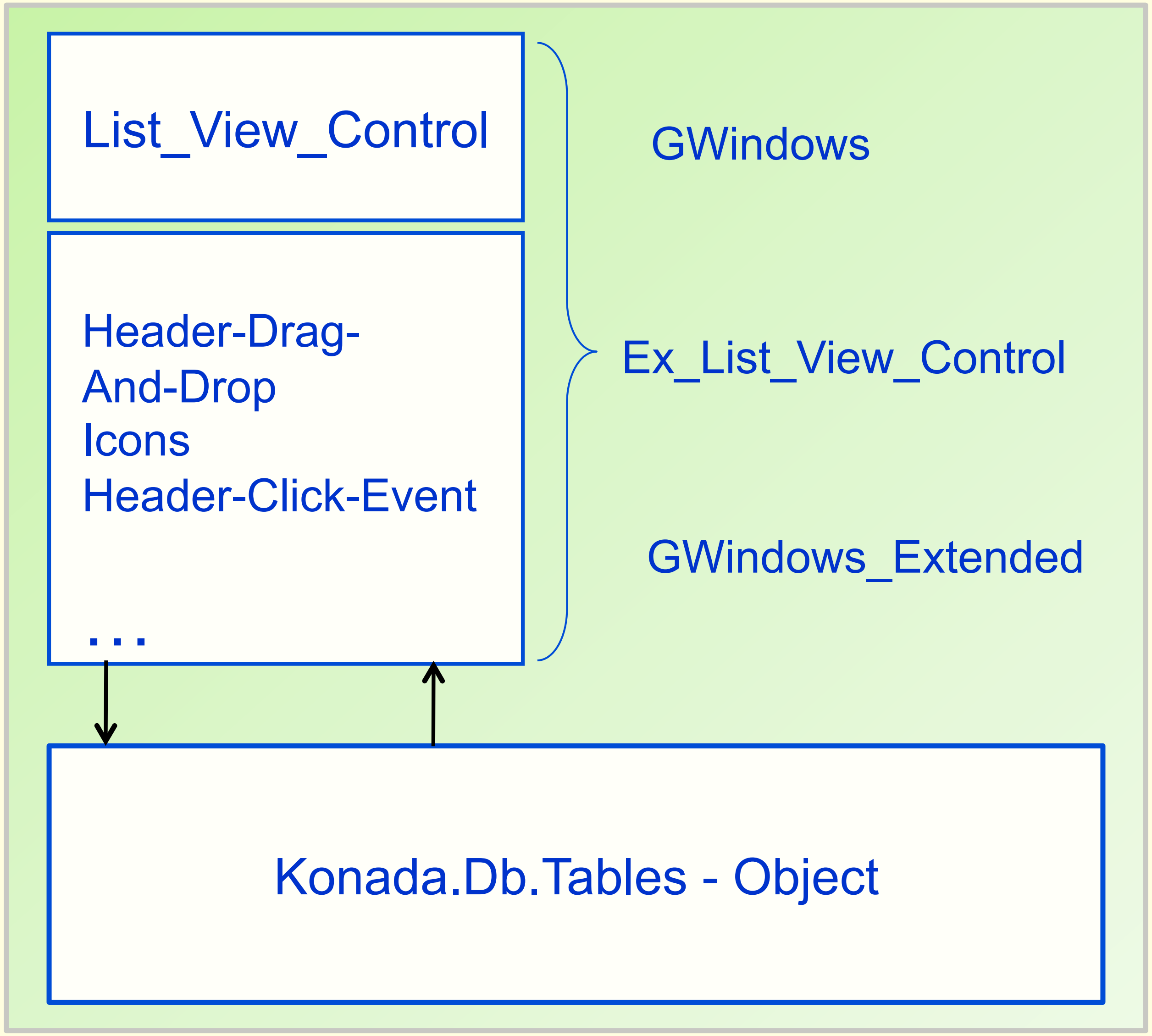
✓ Easy to use

✓ Easy to extend

✓ Available under the GMGPL

The User Interface Library

Example: Multi Record Control



List_View_Control

Header-Drag-
And-Drop
Icons
Header-Click-Event

...

GWindows

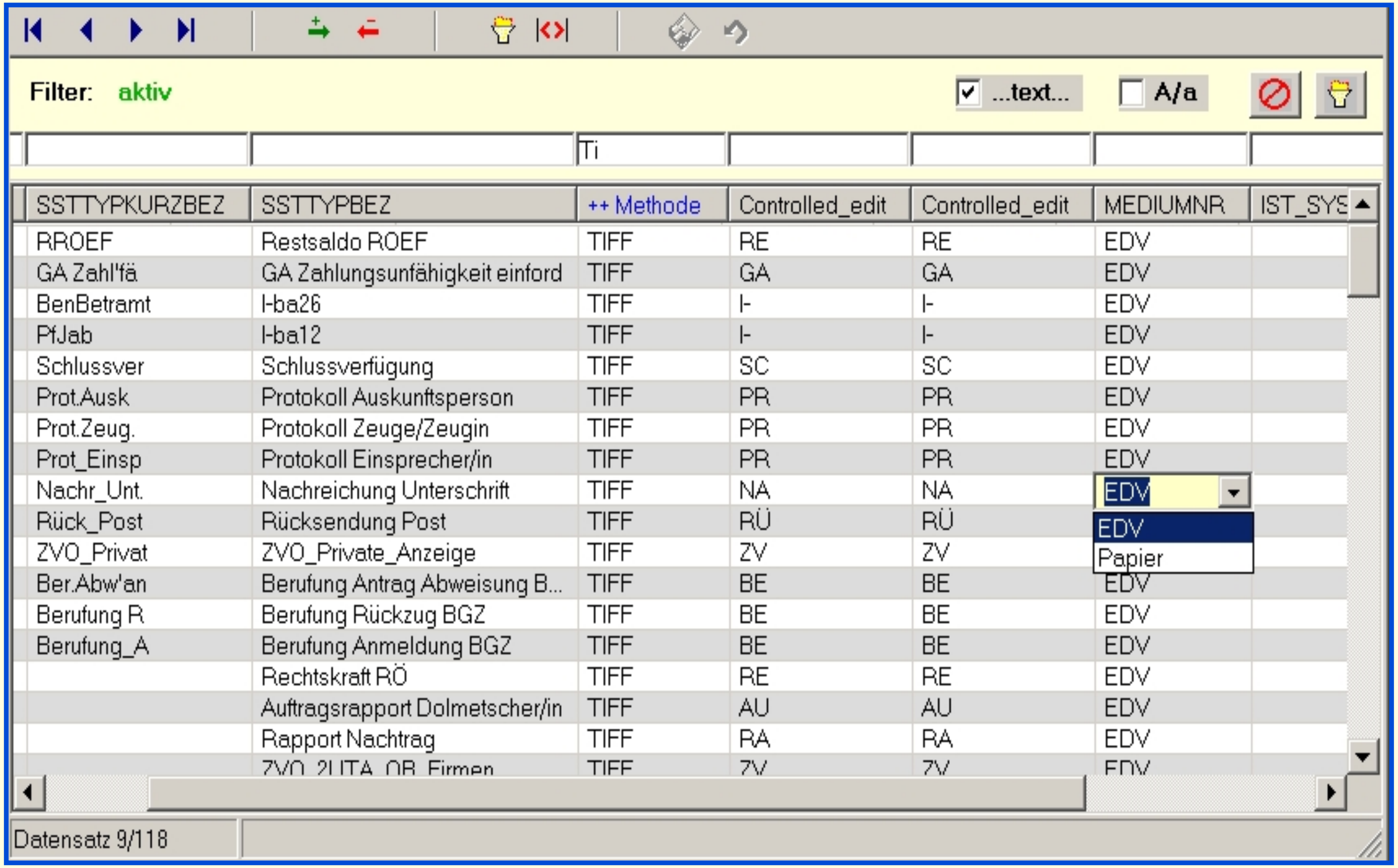
Ex_List_View_Control

GWindows_Extended

Konada.Db.Tables - Object

The User Interface Library

Multi Record Control



Filter: aktiv

☑ ...text... □ A/a

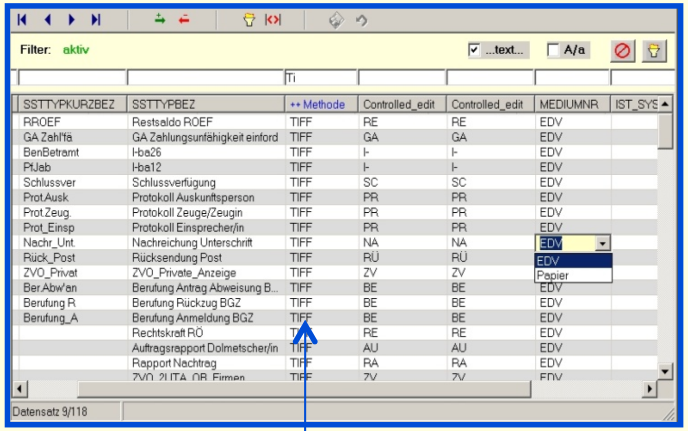
Ti

SSTTYPKURZBEZ	SSTTYPBEZ	++ Methode	Controlled_edit	Controlled_edit	MEDIUMNR	IST_SYS
RROEF	Restsaldo ROEF	TIFF	RE	RE	EDV	
GA Zahl'fä	GA Zahlungsunfähigkeit einford	TIFF	GA	GA	EDV	
BenBetramt	I-ba26	TIFF	I-	I-	EDV	
PfJab	I-ba12	TIFF	I-	I-	EDV	
Schlussver	Schlussverfügung	TIFF	SC	SC	EDV	
Prot.Ausk	Protokoll Auskunftsperson	TIFF	PR	PR	EDV	
Prot.Zeug.	Protokoll Zeuge/Zeugin	TIFF	PR	PR	EDV	
Prot_Einsp	Protokoll Einsprecher/in	TIFF	PR	PR	EDV	
Nachr_Unt.	Nachreichung Unterschrift	TIFF	NA	NA	EDV	
Rück_Post	Rücksendung Post	TIFF	RÜ	RÜ		
ZVO_Privat	ZVO_Private_Anzeige	TIFF	ZV	ZV		
Ber.Abw'an	Berufung Antrag Abweisung B...	TIFF	BE	BE	EDV	
Berufung R	Berufung Rückzug BGZ	TIFF	BE	BE	EDV	
Berufung_A	Berufung Anmeldung BGZ	TIFF	BE	BE	EDV	
	Rechtskraft RÖ	TIFF	RE	RE	EDV	
	Auftragsrapport Dolmetscher/in	TIFF	AU	AU	EDV	
	Rapport Nachtrag	TIFF	RA	RA	EDV	
	ZVO_2LITA_OR_Firmen	TIFF	ZV	ZV	EDV	

Datensatz 9/118

The User Interface Library

Multi Record Control and Data Container



Filter: aktiv

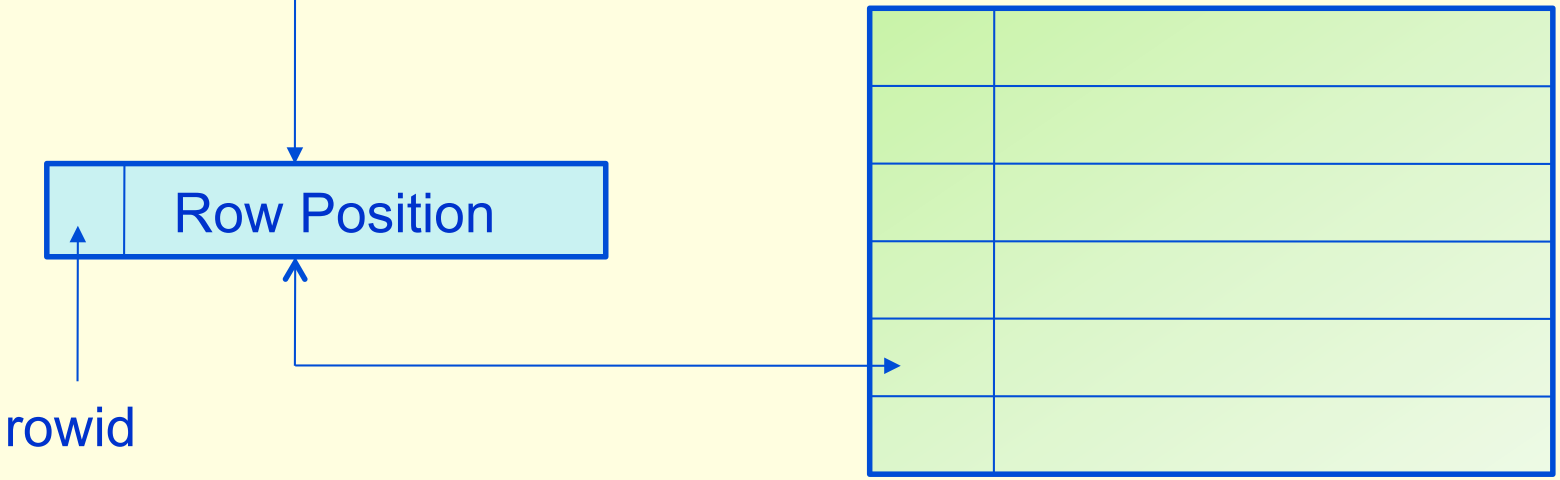
☑ ...text... ☐ A/a

	SSTTYPKURZBEZ	SSTTYPBEZ	++ Methode	Controlled_edit	Controlled_edit	MEDIUMNR	IST_SYS
			Ti				
	RROEF	Restsaldo ROEF	TIFF	RE	RE	EDV	
	GA Zahl'fä	GA Zahlungsunfähigkeit einford	TIFF	GA	GA	EDV	
	BenBetramt	I-ba26	TIFF	I-	I-	EDV	
	PfJab	I-ba12	TIFF	I-	I-	EDV	
	Schlussver	Schlussverfügung	TIFF	SC	SC	EDV	
	Prot.Ausk	Protokoll Auskunftsperson	TIFF	PR	PR	EDV	
	Prot.Zeug.	Protokoll Zeuge/Zeugin	TIFF	PR	PR	EDV	
	Prot_Einsp	Protokoll Einsprecher/in	TIFF	PR	PR	EDV	
	Nachr_Unt.	Nachreichung Unterschrift	TIFF	NA	NA	EDV	
	Rück_Post	Rücksendung Post	TIFF	RÜ	RÜ	EDV	
	ZVO_Privat	ZVO_Private_Anzeige	TIFF	ZV	ZV	Papier	
	Ber.Abw'an	Berufung Antrag Abweisung B...	TIFF	BE	BE	EDV	
	Berufung R	Berufung Rückzug BGZ	TIFF	BE	BE	EDV	
	Berufung_A	Berufung Anmeldung BGZ	TIFF	BE	BE	EDV	
		Rechtskraft RÖ	TIFF	RE	RE	EDV	
		Auftragsrapport Dolmetscher/in	TIFF	AU	AU	EDV	
		Rapport Nachtrag	TIFF	RA	RA	EDV	
		ZVO_2UTA_OB_Firmen	TIFF	ZV	ZV	EDV	

Datensatz 9/118

List View Control

Konada.Db.Tables

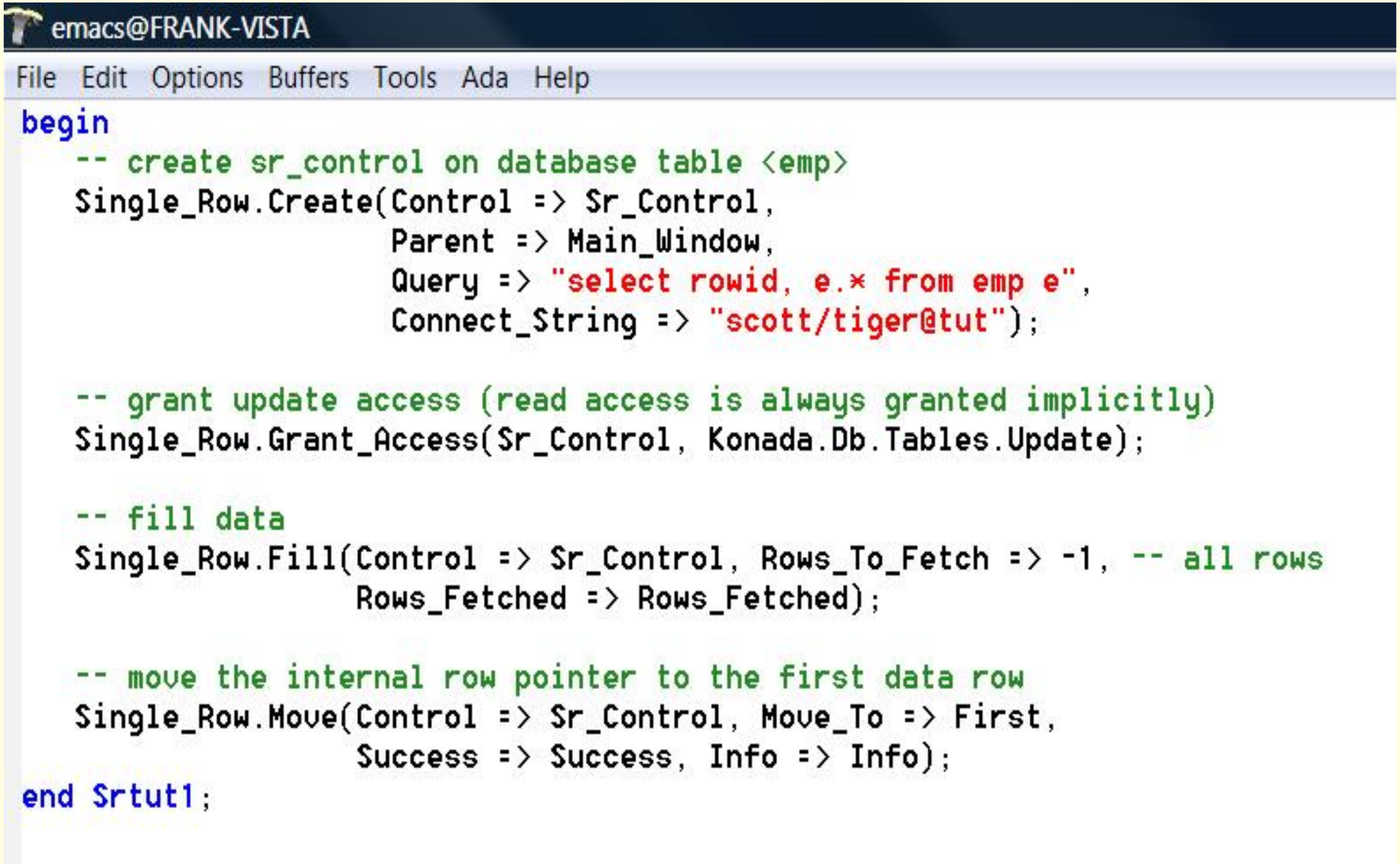


Row Position

rowid

The User Interface Library

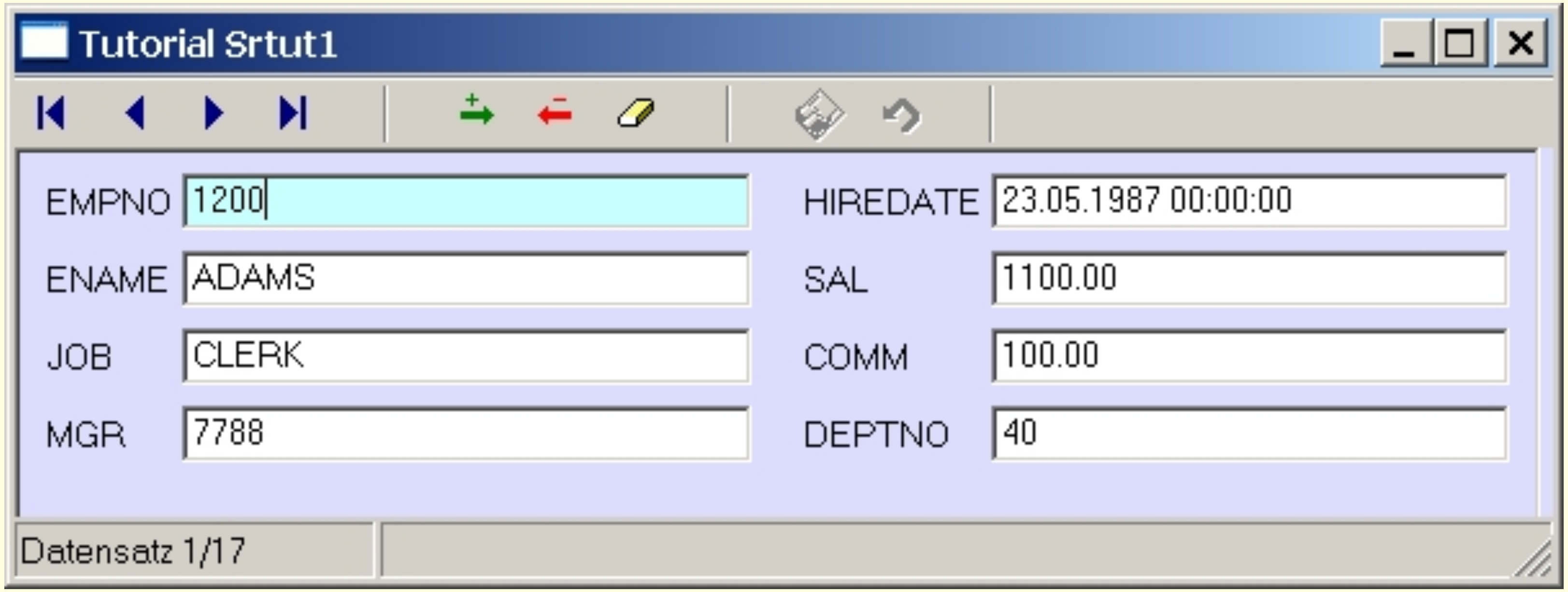
Code sample – Simple Single Record Control



```
begin
   -- create sr_control on database table <emp>
   Single_Row.Create(Control => Sr_Control,
                     Parent => Main_Window,
                     Query => "select rowid, e.* from emp e",
                     Connect_String => "scott/tiger@tut");

   -- grant update access (read access is always granted implicitly)
   Single_Row.Grant_Access(Sr_Control, Konada.Db.Tables.Update);

   -- fill data
   Single_Row.Fill(Control => Sr_Control, Rows_To_Fetch => -1, -- all rows
                   Rows_Fetched => Rows_Fetched);

   -- move the internal row pointer to the first data row
   Single_Row.Move(Control => Sr_Control, Move_To => First,
                   Success => Success, Info => Info);
end Srtut1;
```

The User Interface Library

The Effect of the previous code snippet



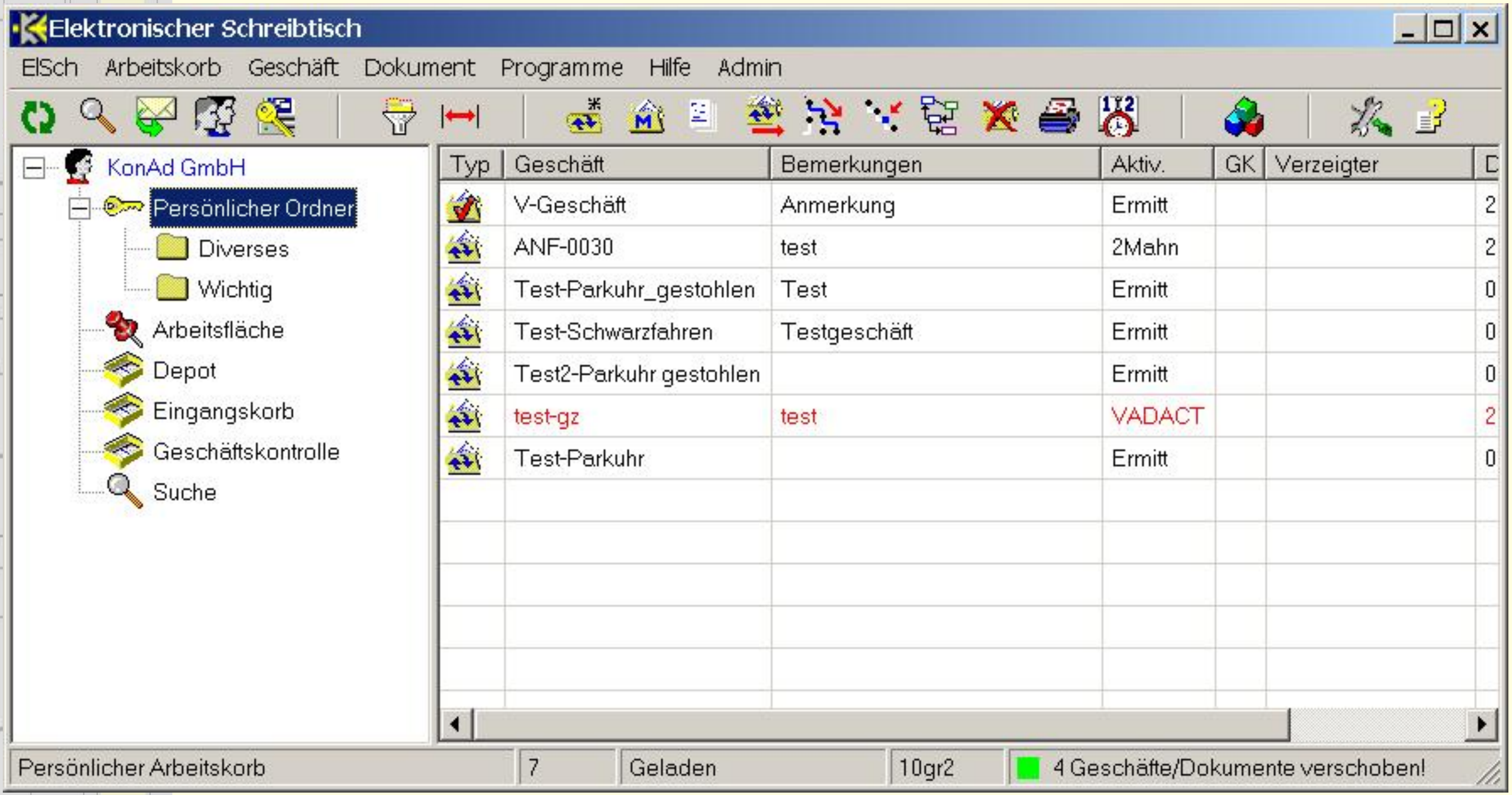
EMPNO	1200	HIREDATE	23.05.1987 00:00:00
ENAME	ADAMS	SAL	1100.00
JOB	CLERK	COMM	100.00
MGR	7788	DEPTNO	40

Datensatz 1/17

Projects

- ElSch – Workflow Client
- Complete ERP-Solution for hydraulic and pneumatic element manufacturers (Demo)
- Interface between Workflow System and Archive System on the Solaris platform
- Further development of our dynamic GUI control

Projects - ElSch

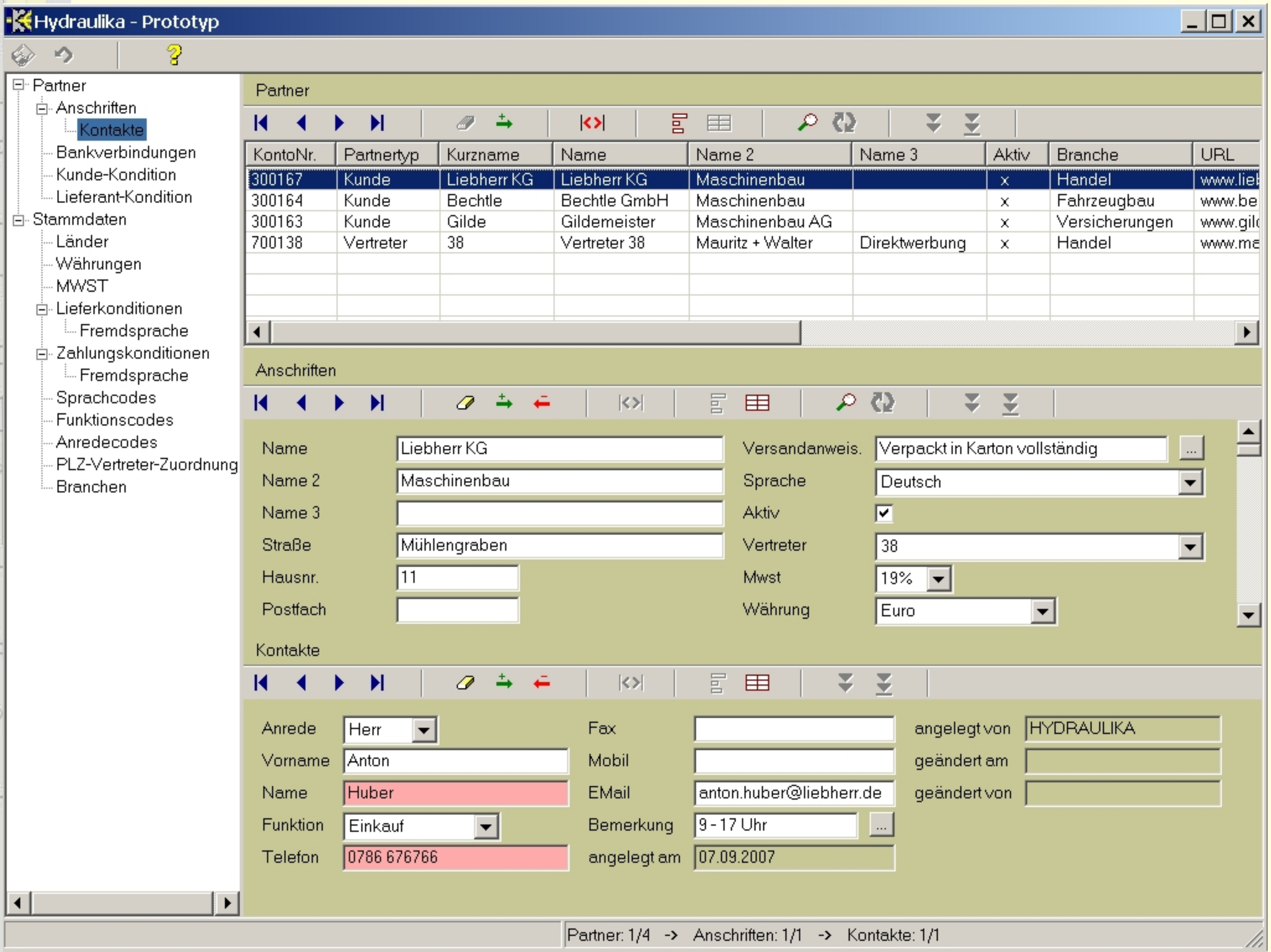


Typ	Geschäft	Bemerkungen	Aktiv.	GK	Verzeigter	D
	V-Geschäft	Anmerkung	Ermitt			2
	ANF-0030	test	2Mahn			2
	Test-Parkuhr_gestohlen	Test	Ermitt			0
	Test-Schwarzfahren	Testgeschäft	Ermitt			0
	Test2-Parkuhr gestohlen		Ermitt			0
	test-gz	test	VADACT			2
	Test-Parkuhr		Ermitt			0

Persönlicher Arbeitskorb | 7 | Geladen | 10gr2 | 🟩 4 Geschäfte/Dokumente verschoben!

ElSch – Workflow Client. Running since 2005
Stadtrichteramt Zürich

Projects – ERP Solution



Partner

KontoNr.	Partnertyp	Kurzname	Name	Name 2	Name 3	Aktiv	Branche	URL
300167	Kunde	Liebherr KG	Liebherr KG	Maschinenbau		x	Handel	www.lie
300164	Kunde	Bechtle	Bechtle GmbH	Maschinenbau		x	Fahrzeugbau	www.be
300163	Kunde	Gilde	Gildemeister	Maschinenbau AG		x	Versicherungen	www.gil
700138	Vertreter	38	Vertreter 38	Mauritz + Walter	Direktwerbung	x	Handel	www.me

Anschriften

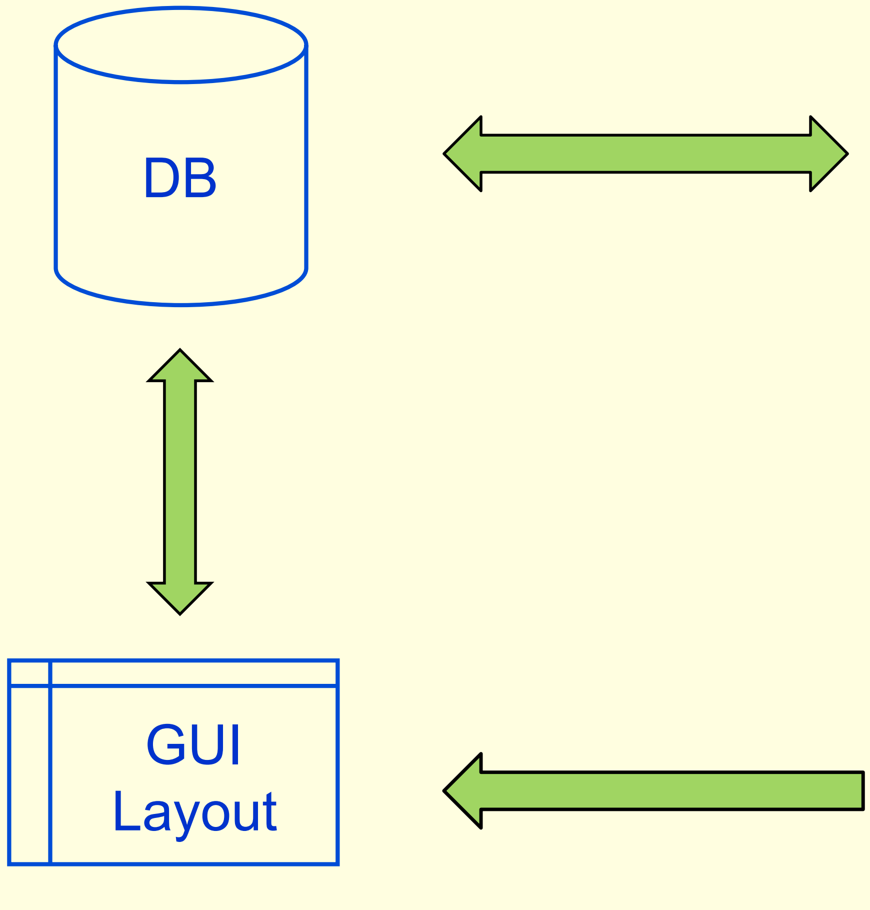
Name	Liebherr KG	Versandanweis.	Verpackt in Karton vollständig
Name 2	Maschinenbau	Sprache	Deutsch
Name 3		Aktiv	☑
Straße	Mühlengraben	Vertreter	38
Hausnr.	11	Mwst	19%
Postfach		Währung	Euro

Kontakte

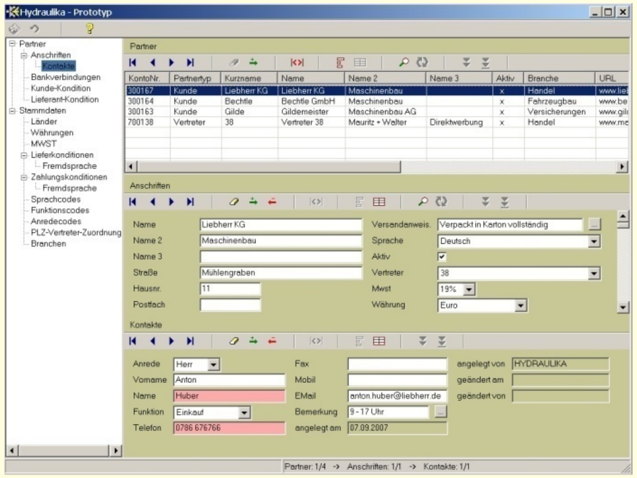
Anrede	Herr	Fax		angelegt von	HYDRAULIKA
Vorname	Anton	Mobil		geändert am	
Name	Huber	EMail	anton.huber@liebherr.de	geändert von	
Funktion	Einkauf	Bemerkung	9 - 17 Uhr		
Telefon	0786 676766	angelegt am	07.09.2007		

Partner: 1/4 -> Anschriften: 1/1 -> Kontakte: 1/1

Projects
Dynamic GUI Control



Application



Partner

KontoNr.	Partnertyp	Kurzname	Name	Name 2	Name 3	Aktiv	Branche	URL
300167	Kunde	Liebherr KG	Liebherr KG	Maschinenbau		x	Handel	www.lie
300164	Kunde	Bechtle	Bechtle GmbH	Maschinenbau		x	Fahrzeugbau	www.be
300163	Kunde	Gilde	Gildemeister	Maschinenbau AG		x	Versicherungen	www.gil
700138	Vertreter	38	Vertreter 38	Mauritz + Walter	Direktwerbung	x	Handel	www.me

Anschriften

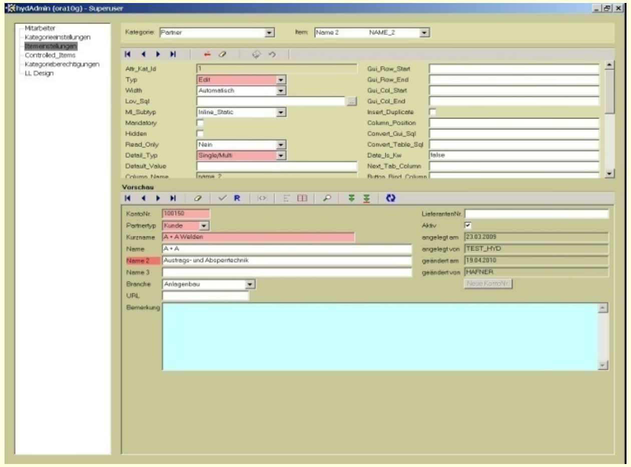
Name	Liebherr KG	Versandanweis.	Verpackt in Karton vollständig
Name 2	Maschinenbau	Sprache	Deutsch
Name 3		Aktiv	☑
Straße	Mühlengraben	Vertreter	38
Hausnr.	11	Mwst	19%
Postfach		Währung	Euro

Kontakte

Anrede	Herr	Fax		angelegt von	HYDRAULIKA
Vorname	Anton	Mobil		geändert am	
Name	Huber	EMail	anton.huber@liebherr.de	geändert von	
Funktion	Einkauf	Bemerkung	9 - 17 Uhr		
Telefon	0786 676766	angelegt am	07.09.2007		

Partner: 1/4 -> Anschriften: 1/1 -> Kontakte: 1/1

GUI - Maintenance



Kategorie: Partner   Item: Name 2 NAME_2

Attr_Kat_Id	1	Gui_Row_Start	
Typ	Edit	Gui_Row_End	
Width	Automatisch	Gui_Col_Start	
Lov_Sql		Gui_Col_End	
Ml_Subtyp	Inline_Static	Insert_Duplicate	□
Mandatory	□	Column_Position	
Hidden	□	Convert_Gui_Sql	
Read_Only	Nein	Convert_Table_Sql	
Detail_Typ	Single/Multi	Date_Is_Kw	false
Default_Value		Next_Tab_Column	
Column_Name	name_2	Button_Bind_Column	

Vorschau

KontoNr.	100150	LieferantenNr.	
Partnertyp	Kunde	Aktiv	☑
Kurzname	A + A Welden	angelegt am	23.03.2009
Name	A + A	angelegt von	TEST_HYD
Name 2	Austrags- und Absperrtechnik	geändert am	19.04.2010
Name 3		geändert von	HAFNER
Branche	Anlagenbau		
URL			
Bemerkung			

Conclusion

Database Programming with Ada is real business, but

- We had to learn and turn
- The customers have to trust our choice
- We often have to give answers to the question:

„Why not Java?“

Here is one:

„We like programming in Ada.“