RADventure SQL Browse Template
Programmers Documentation

Table of contents

1 Installation

Installation of the SQL browse template is completely automatic. You can start the RadSQL .exe. This installation program will not only install the templates, but also the classes and a demo program. The templates can automatically register themselves in your template registry.

The files installed are the following:

\libsrc	RADSQL b.inc
	RADSQL b.clw
	RADSQL v.inc
	RADSQL v.clw
	RADSQL dr.clw
	RADSQL dr.inc
\template	RADSQL br.tpl
	RADSQL drop.tpl
\examples\SQL Browse	Orademo.app
	Orademo.dct

The RADventure SQL browse templates contains several different templates:

- a browse procedure template complete with insert, update, delete and select button
- a lookup locator control template
- A SQL dropdown template
- Browse update button controls
- Edit in place extension for the browse
- An integrated browse/ Form control template

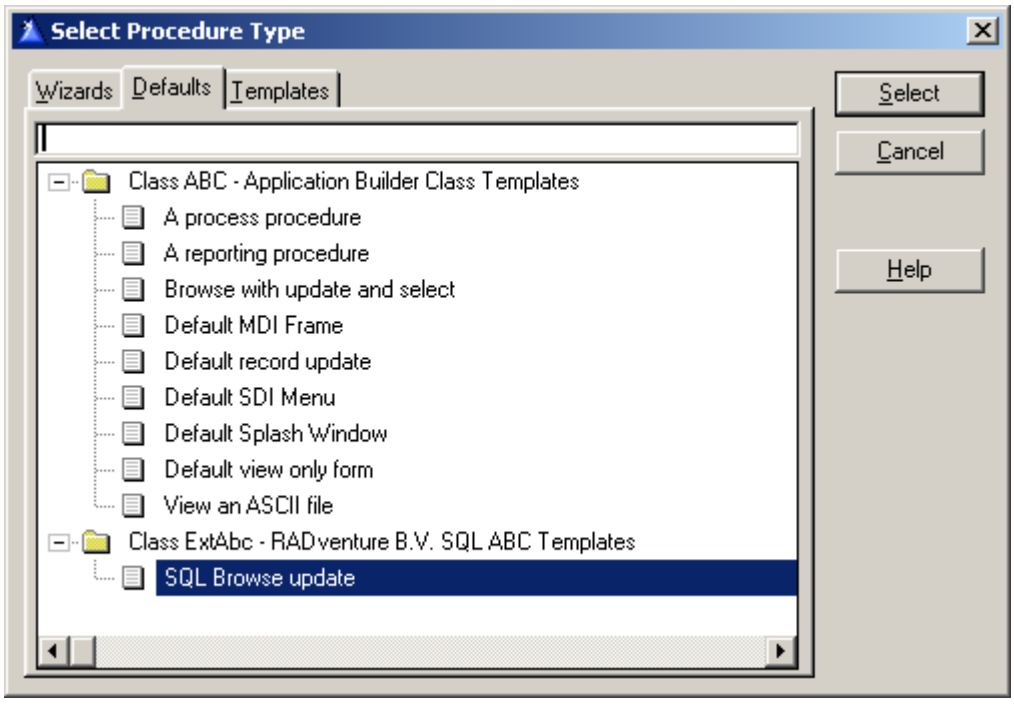
2 SQL Browse Procedure Template

To use the SQL browse template use the following steps:

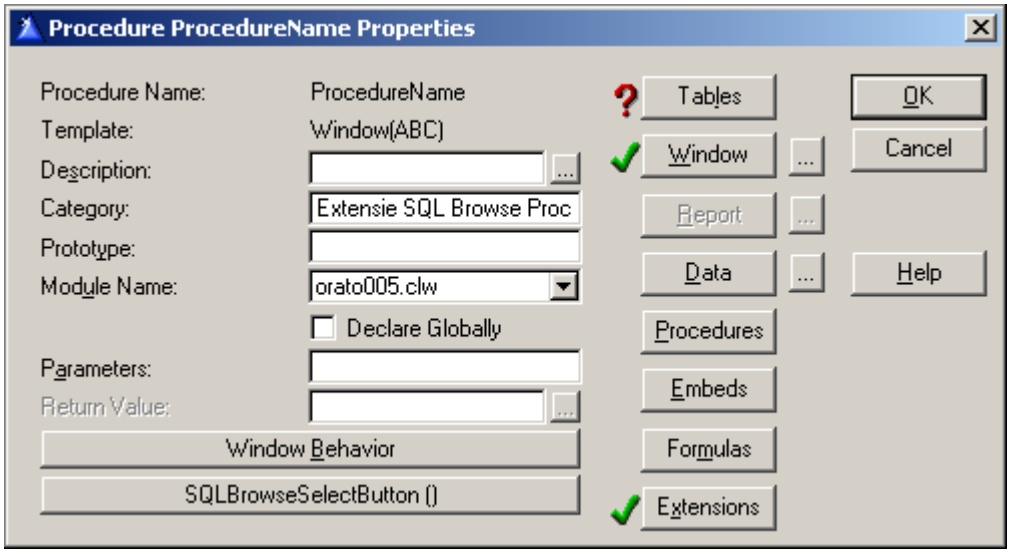
1) Open any valid Clarion application that contains a dictionary
a) Open the Clarion IDE
b) Open your application

2) Click on the menu '**Procedure**' then '**New**'

3) After entering the name of the new procedure select from the available procedure templates:
SQL Browse update
The following screen representation is a sample of the screen that appears at this point (depending on your own loaded template utilities)

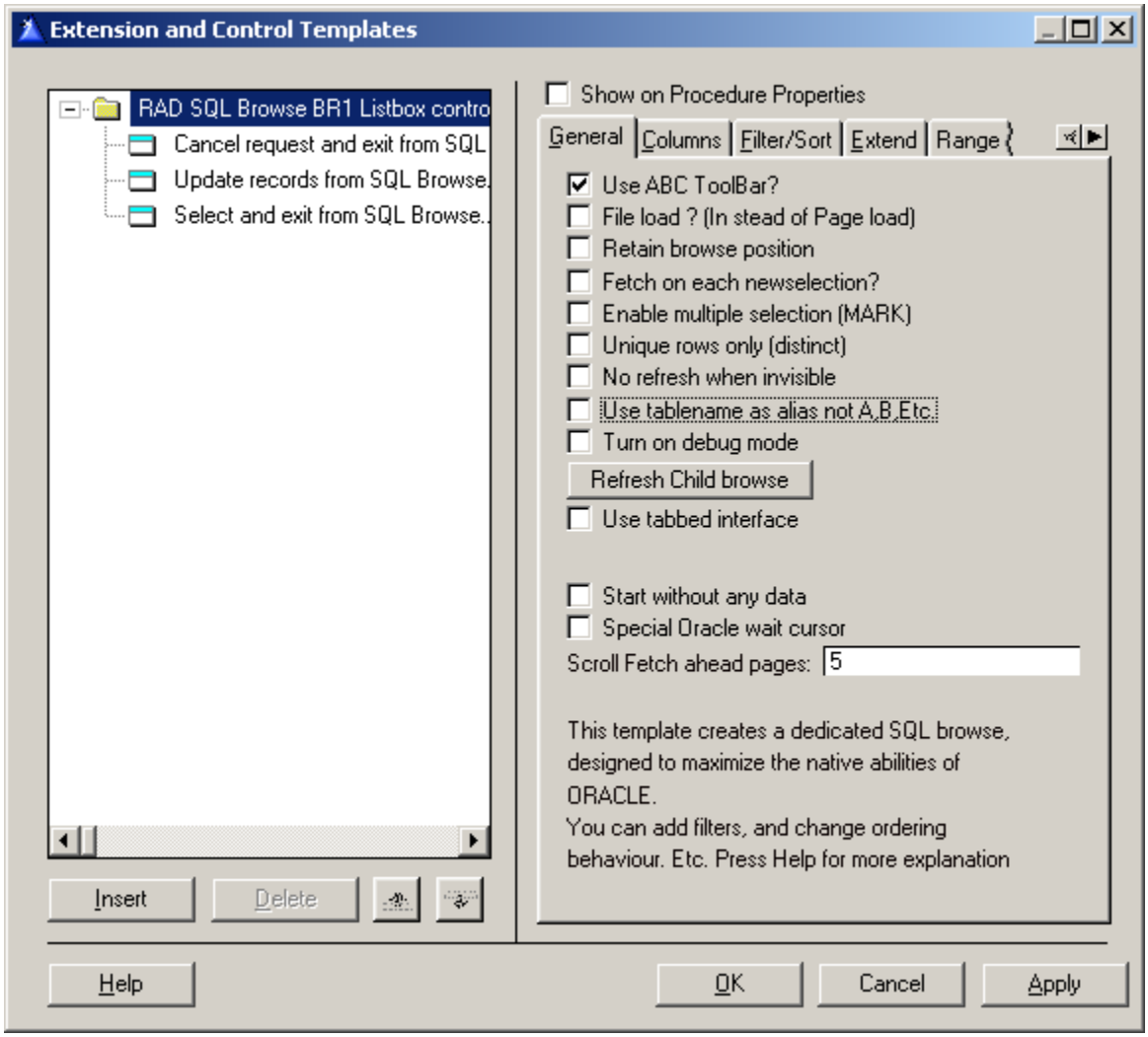


4) Click on select. This window should appear:



This is almost the same window of an ABC Browse procedure (with the '**Tables'** button you must enter the Table(s) the SQL -Browse will use), except this SQL Browse procedure uses an SQL -Browse control, SQL -Update-, Select- and Cancel-Buttons.

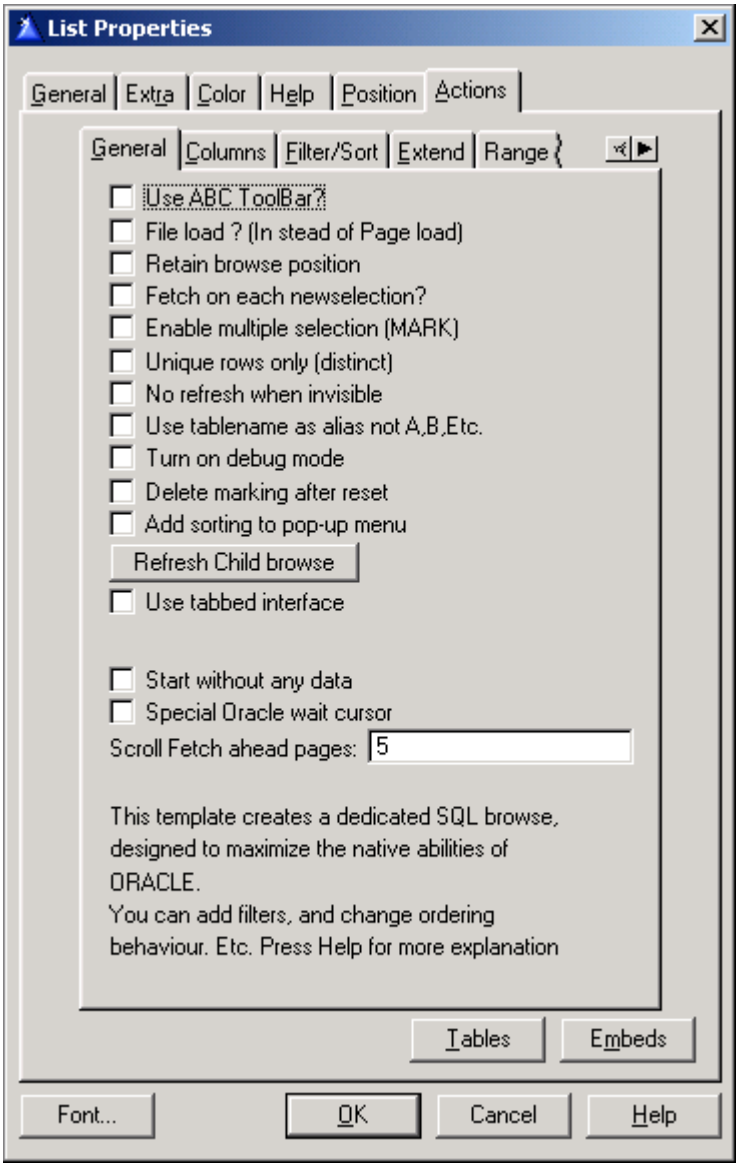
When you press the '**Extensions**' button you will see these SQL controls:



For detailed information about these controls see the documentation of the SQL control templates.

5) When you go to Window-formatter you can populate the SQL Browse with the Table fields in the normal way choosing List Box formatter

3 SQL Browse Box Control Template



3.1 General

You can use the following options on the template.

Use ABC Toolbar
If you want to use the ABC Toolbar buttons for updating, scrolling etc. of the records in the SQL Browse box.

File Load
If you want to fill the Browse box queue immediately with all the records of the table. Not recommended for big tables.

Retain browse position
Retain last selected record in kind of INI-file. When reopening the browse that record will be selected at start.

Fetch on new selection
When checked fetch the new selected record from Table instead of from the view.

Enable multiple selection (MARK)
Enable the user to select more then one record in the browse.

Unique rows only (distinct)
Sets the DISTINCT statement after SELECT in the SQL -statement.

No refresh when invisible
When checked the browse is not refreshed when invisible (for example because the browse is placed on a non selected tab).

Use table name
By default the tables in the SQL -statement are named A, B, C etc.
When you want to overrule this and name them by their real names you must check this option.

Turn on debug mode
When checked a message with the SQL -statement will be shown for executing the statement.

Delete marking after reset
When you want to keep your selected records after a reset, do not check this box. This makes it possible to keep your marked records and select an other subset of data.

Add sorting to pop-up menu
This will make available 9on right mouse click all columns headers to sort on. You will need this when you integrate Clarion-NET, since for the moment in clarion net you cannot double click on a list box header.

Refresh child browse
Enter the child browse(s) that must be refreshed after a new selection in the parent browse

Use tabbed interface

When you place the browse on sheet with more than one tab you can sort the browse on the column field of the selected tab. For example when you select the fifth tab the browse is sorted on the fifth column field.

Use Sheet Control

When you have checked 'Use Tabbed Interface' you have to select the sheet control for this tabbed interface.

Start without any data

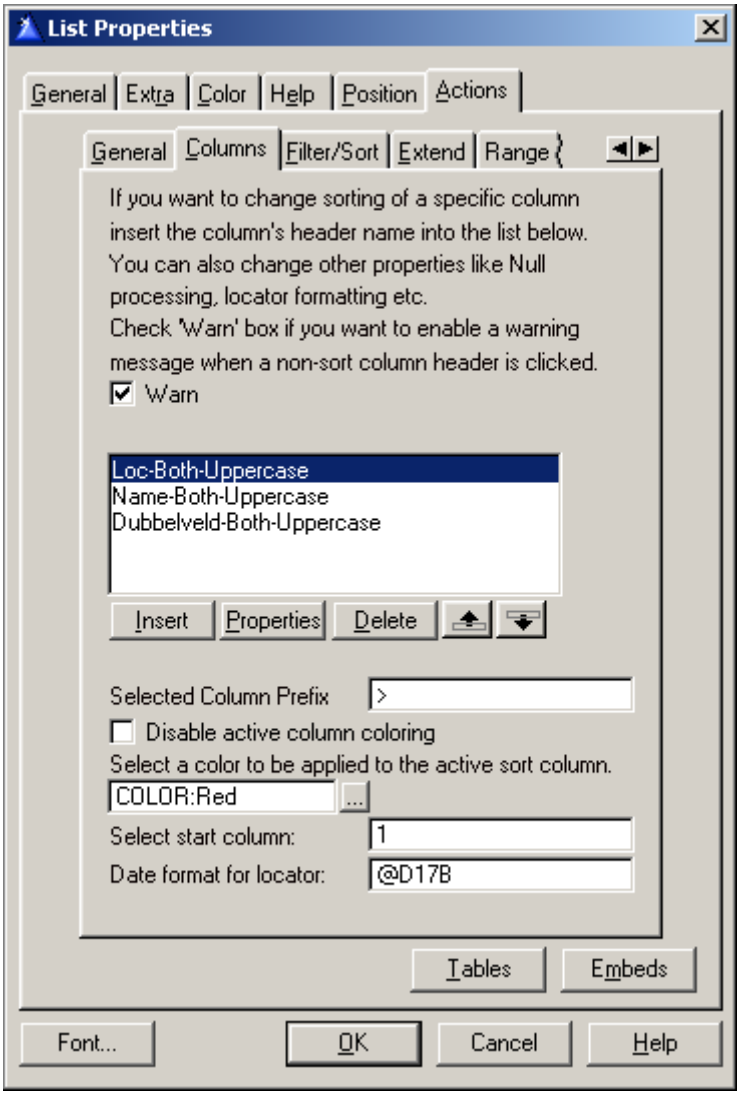
With this option you can start with an empty browse and start filling the browse after some condition or action is completed by setting Self.DoNotRetrieveAnyData = False.

Special Oracle wait cursor

Check this if you want to see the Oracle wait cursor instead of the default wait cursor. This way the user can see that the program is connecting to the Oracle database.

Scroll fetch ahead pages

The number of pages (i.e. number of records in listbox) to read ahead when the user scrolldrags. The default is 5.



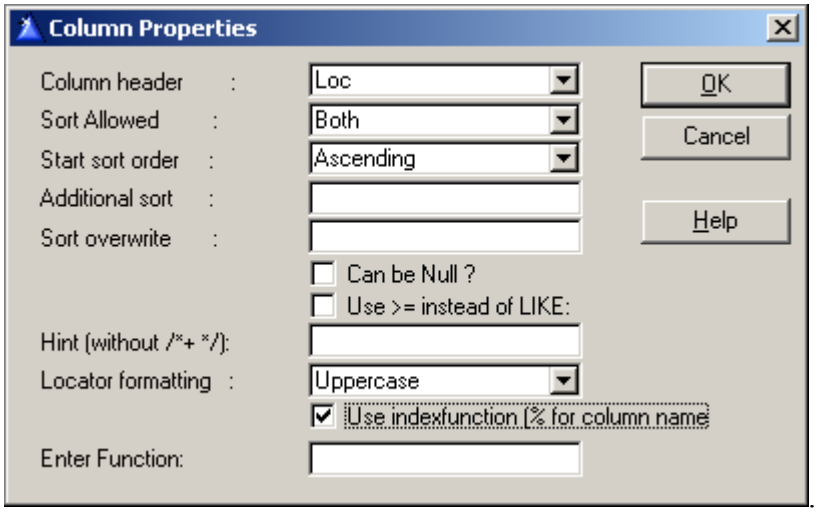
3.2 Columns

Warn

Check this when you want to give the user a message when the user double-clicks on a non-sorted column header.

Columns Listbox

Here you can add the columns on which you want to sort by double-clicking on the column header.



Column header

Fieldname of the column.

Sort Allowed

The sort order allowed for this column:

-Both (Ascending and Descending)
-Ascending (From low to high)
-Descending (From high to low)
-No Sort (No sort allowed: message after clicking the column header)
-Omit Sort (No sort used)

Start sort order

If Sort Allowed is Both, you can choose the sort (Ascending or Descending) to start.

Additional sort

Additional field to sort on.

Sort overwrite

Overwrite the sort of the column field by another field. Useful when you have a local variable in your browse that is based on one or more table fields.

Can be NULL

Also records where the column field value is NULL will be shown

Use >= instead of LIKE

When you use the SQL locator for a alphanumeric field the Where clause of the SQL -statement default uses LIKE. As a result you see only those records that start with the locator value. If you also want to see the subsequent records you can use this option.

Hint

Here you can specify an Oracle Hint.

Locator formatting

Format for the SQL locator for the columnfield.

Use index function

Use a index function for this column.

Enter function

When you have checked the "Use index function" you have to enter the function. For example if you have a UPPER function for the column field you can enter UPPER(%).

Selected Column Prefix

Prefix on column header when column is the active sort order. Default ">" and "<" for ascending and descending.

Disable Active Column Coloring

No specific coloring for the active SORT column.

Color for active sort column

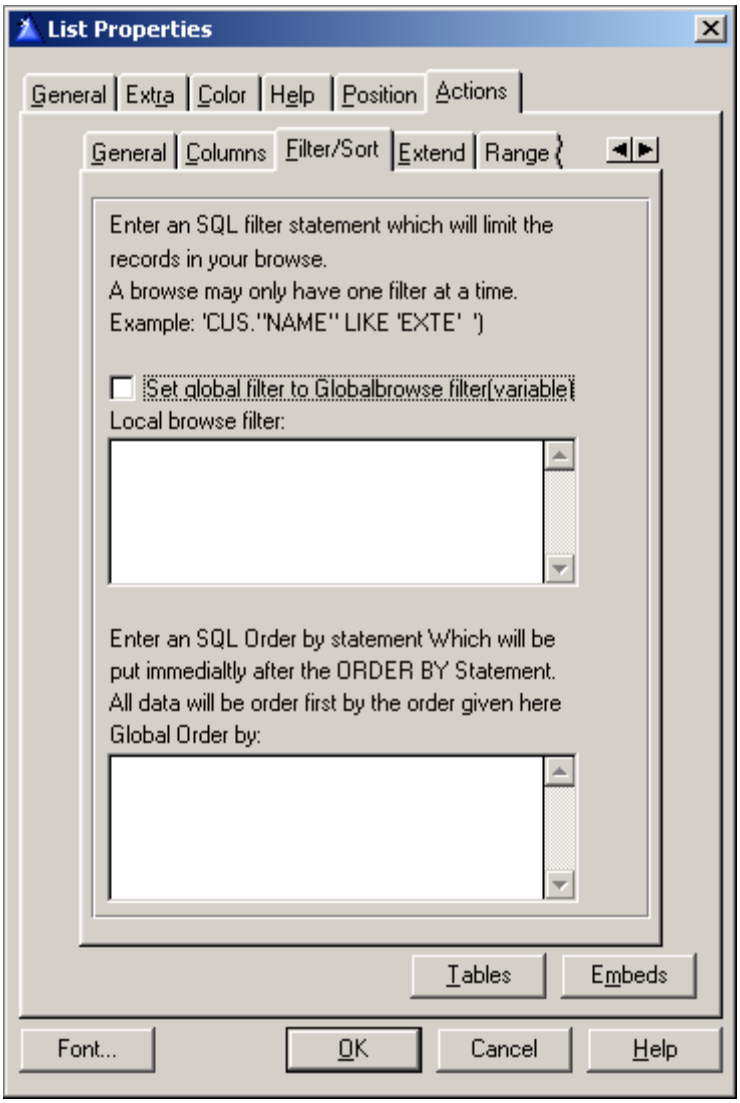
Select color for the active SORT column.

Select start column

Select column that will be active on opening the SQL browse.

Date format for locator

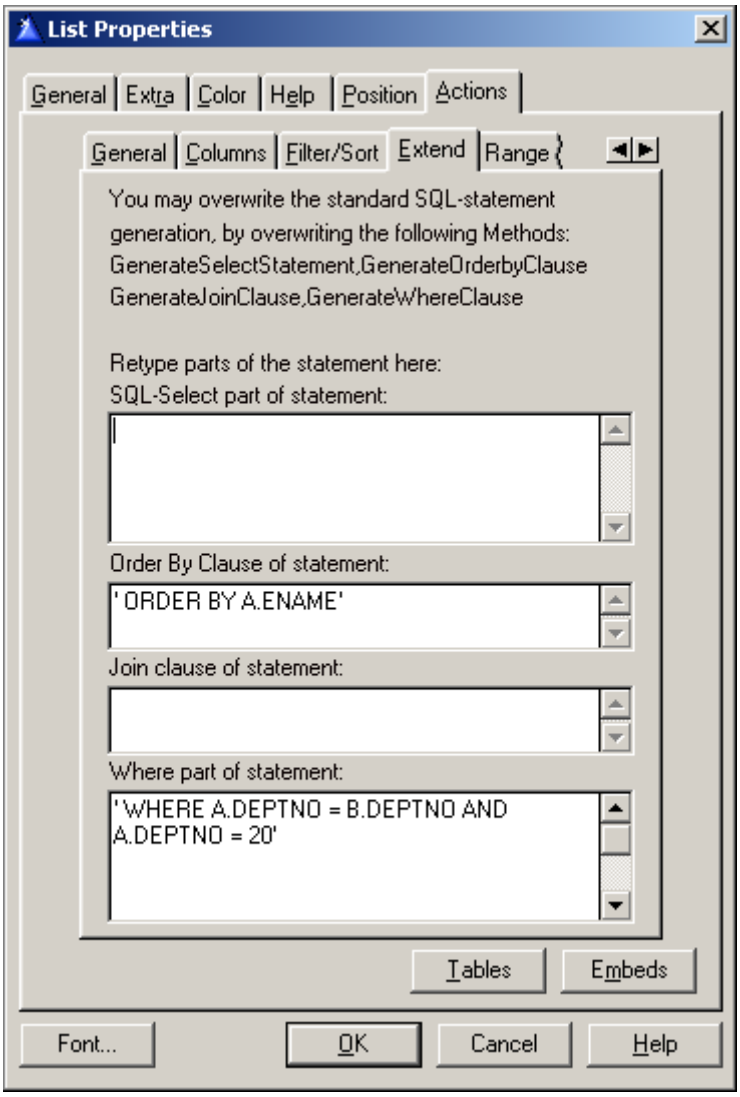
Date format for SQL locator on date fields in the browse.



3.3 Filter/Sort

Set global filter to Globalbrowse filter
A global filter stored in a global variable "Glo:BrowseFilter" that can be applied through the whole application. When you want to use this global filter in this procedure you have to check this option.
Local browse filter
Here you can enter your own filter statement which will be placed after the WHERE statement. For example you can enter 'A.DNAME LIKE ''AC%''', where A is the Alias for SCOTT.DEPT.
Global order by

Here you can enter the field(s) to always sort on first.



3.4 Extend

SQL Select part of statement

Overwrite SELECT part of generated SQL -statement with your own SELECT.

Order By Clause of statement

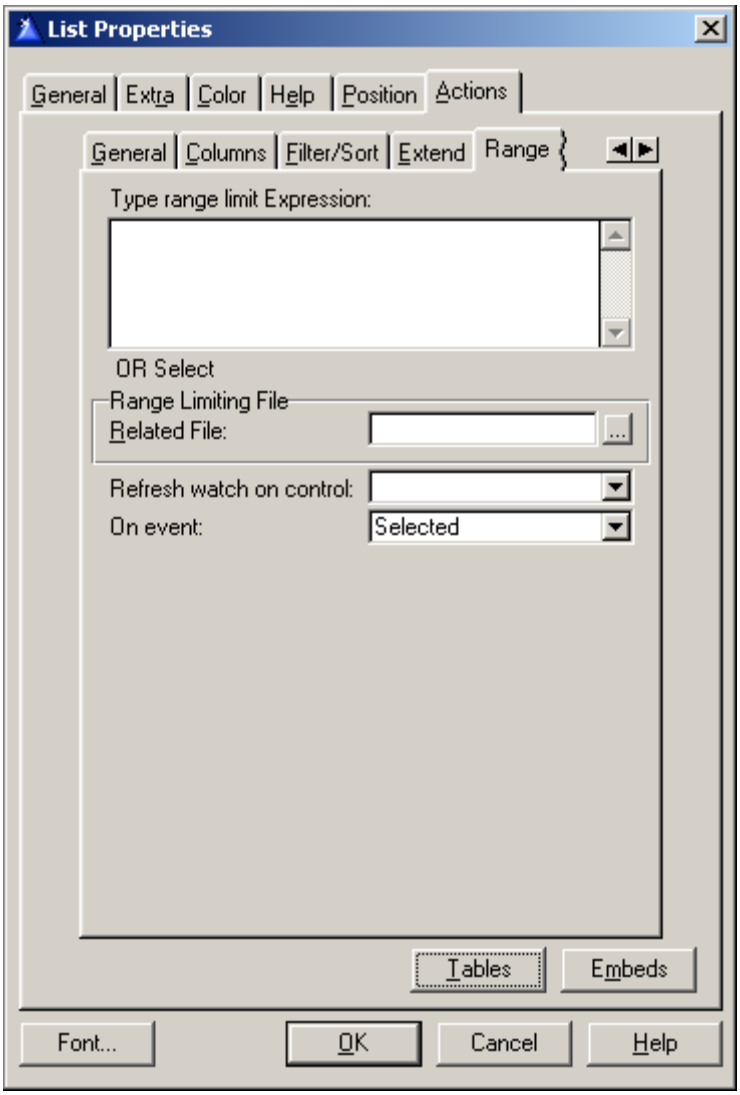
Overwrite ORDER BY part of generated SQL -statement with your own ORDER BY.

Join Clause of statement

Overwrite JOIN part of generated SQL -statement with your own JOIN.

Where part of statement

Overwrite WHERE part of generated SQL -statement with your own WHERE.



3.5 Range

Range Limit expression

You can enter a Range Limit expression that will be placed after the WHERE part of the SQL statement.

Range Limiting File

Or you can select the related Table for the Range Limit.

Refresh watch on control

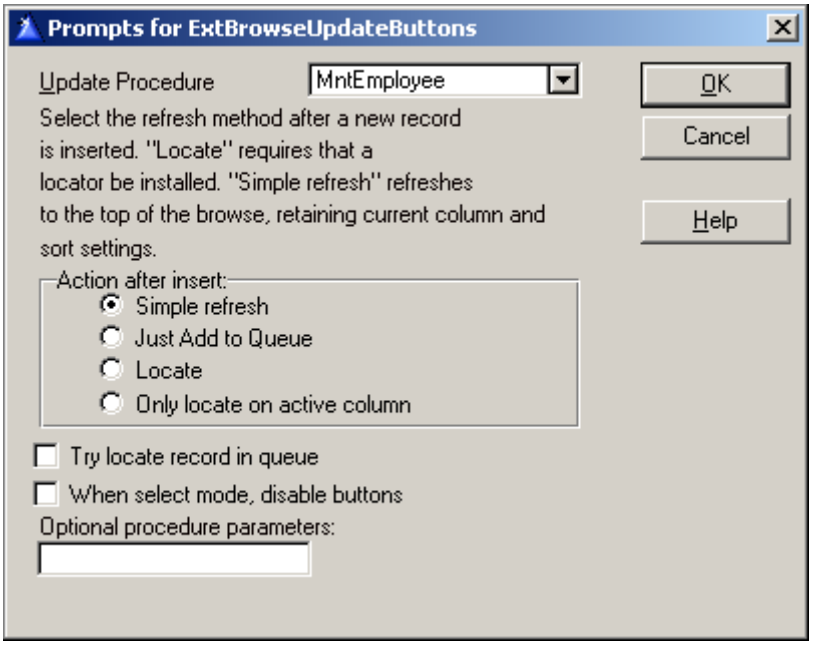
Force refresh browse after an event for this control

On event

Select the event to force the browse refresh.

The last four tabs of the Browsebox actions '**Hotfields/Buffering**', '**Colors**', '**Icons**' and '**Classes**' are similar to the normal ABC browsebox, except that the default class is ABC:ExtSQL B

4 SQL Browse Box Update Buttons Control Template



4.1 Prompts for BrowseUpdateButtons

Update Procedure

The name of the update procedure (Form). If you enter a new procedure name the Application Generator adds the new procedure to the Application Tree.

Action after insert

Chose here between:

Try locate record in queue

Try locate record in queue after update.

When selected mode, disable buttons

If checked the update buttons are disabled when the Browse procedure is called with globalrequest is SelectRecord.

Optional procedure parameters

5 SQL Browse Box Select Button Control Template

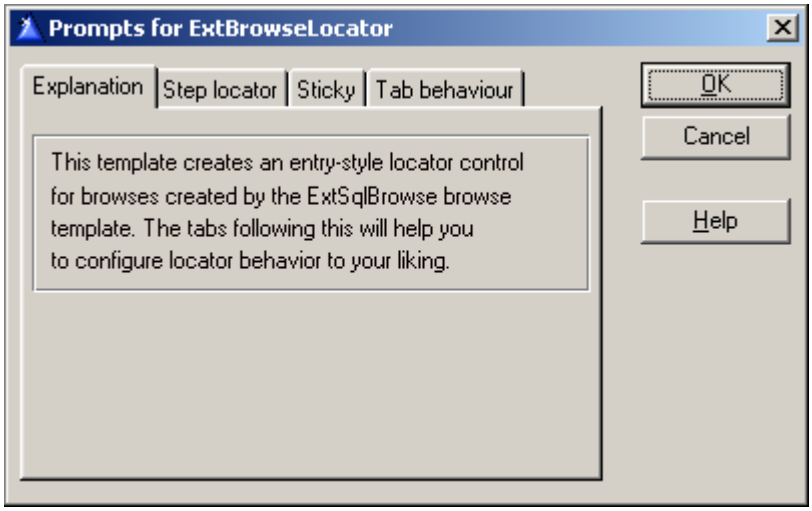
This control template has no programmers input.
The select button updates the records content, closes the browse and returns RequestCompleted.

The select button is hidden when the browse procedure is not called with the SelectRecord request.

SQL Browse Box Cancel Button Control Template

This control template has no programmers input.
The cancel button closes the browse and returns RequestCancelled.

6 SQL Browse Box Locator Control Template



Step Locator

Default the locator is "entry-style" : the locator executes after leaving the locator with TAB entry. If you want to locate on every entry in the locator, you have the check the "Step locator" and turn the IMM attribute of the locator on.

Sticky

If you check the "sticky locator", the selection made in the locator on the active column is saved when you select another column where you can locate further in the saved selection.

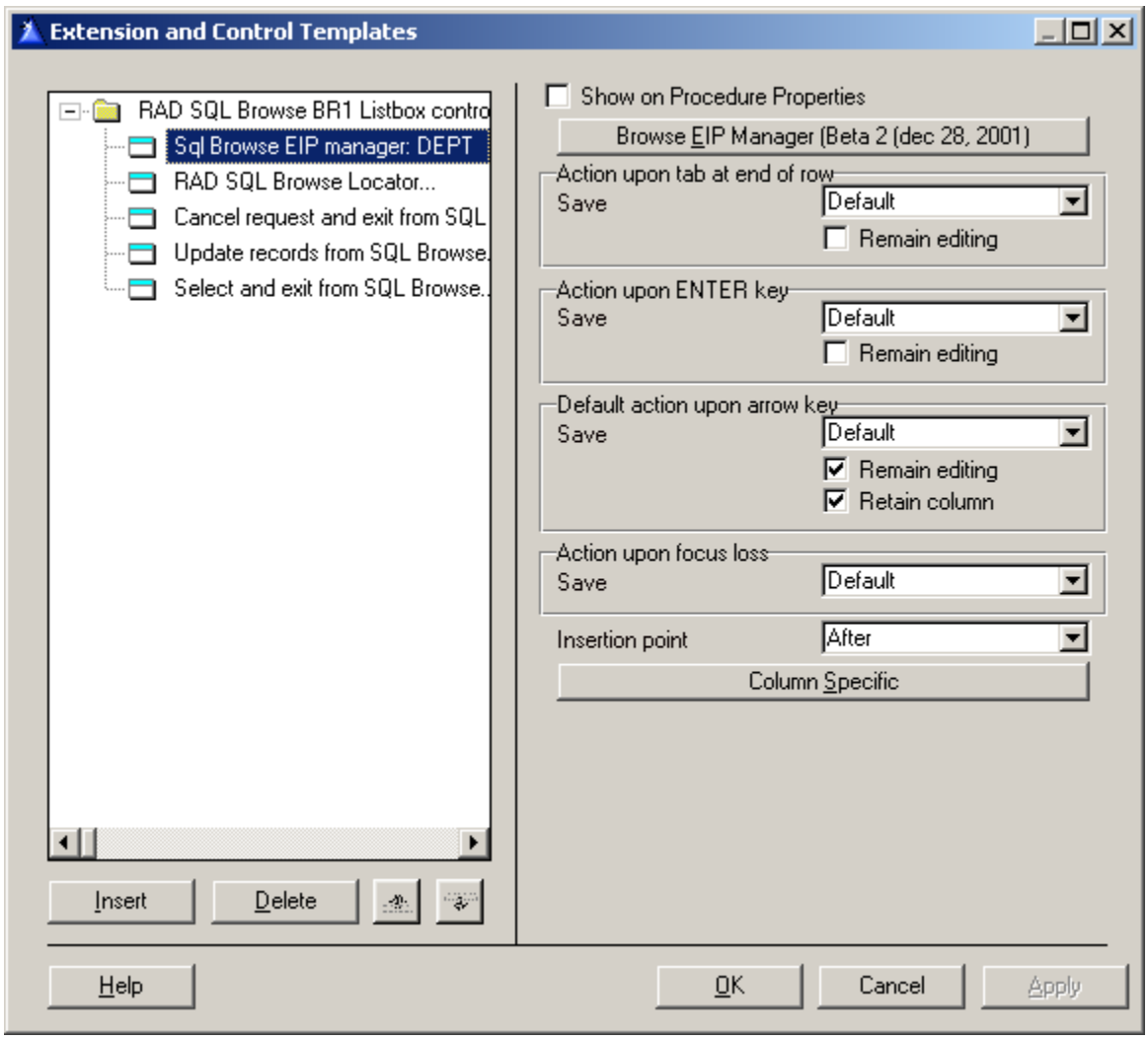
Depends on byte field

If you use the sticky locator, you have to enter a byte check field on the window to set (when byte = 1) or reset (when 0) the sticky locator.

Tab behavior

When you check "Jump form browse box to locator after TAB" the locator is selected again after Tabbing out of locator.

SQL Browse Box EditInPlace Extension Template



Save

The Configure Edit in place dialog offers the Save option for four different keyboard actions. These options determine whether
changes to an edited record are saved or abandoned upon the following keyboard actions: TAB key at end of row, ENTER key, up or down arrow key, focus loss (changing focus to another control or window, typically with a mouse-click). Choose from:
Default
Always on TAB key end of row.
Prompted on the other key actions.

Always
Always save the record.
Never
Never save the record, abandon the changes.

Prompted
Ask the end user whether to save or cancel the changes.

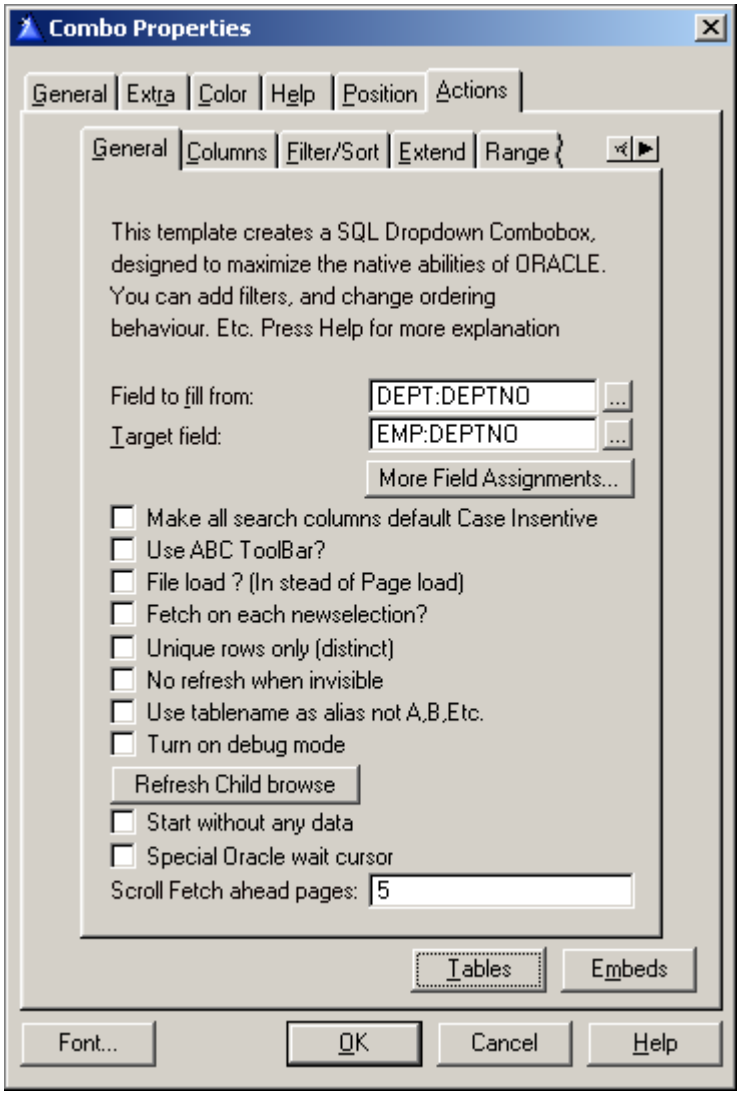
Remain editing
If checked the user continues to edit after the specified key action

Remain column
If checked the continues to edit in the same listbox column

Insertion point
The Configure Edit in place dialog offers the Insertion Point option for initial new record placement in the list. The droplist choices before, after, and append— indicate where the edit-in-place row will appear in the list when inserting a record. Before and after indicate placement in relation to the highlighted record, and append places the edit-in-place row at bottom of the list.

Column specific
Press this button, then press the Insert button to disable a specific column for Edit in Place or to specify the CLASS of object to use when editing a specific list box column.

7 SQL DropDownCombo Control Template



The SQL DropCombo control is basically a SQL Browselist with an additional locator/entryfield and an optional updateprocedure. Therefore, the most options for this control are the same as the options for the SQL Browse control. Here the extra options are mentioned.

Field to fill from

The field in the lookup file whose value is assigned to the Target Field. Press the ellipsis (...) button to select from the Select Field dialog.

Target field

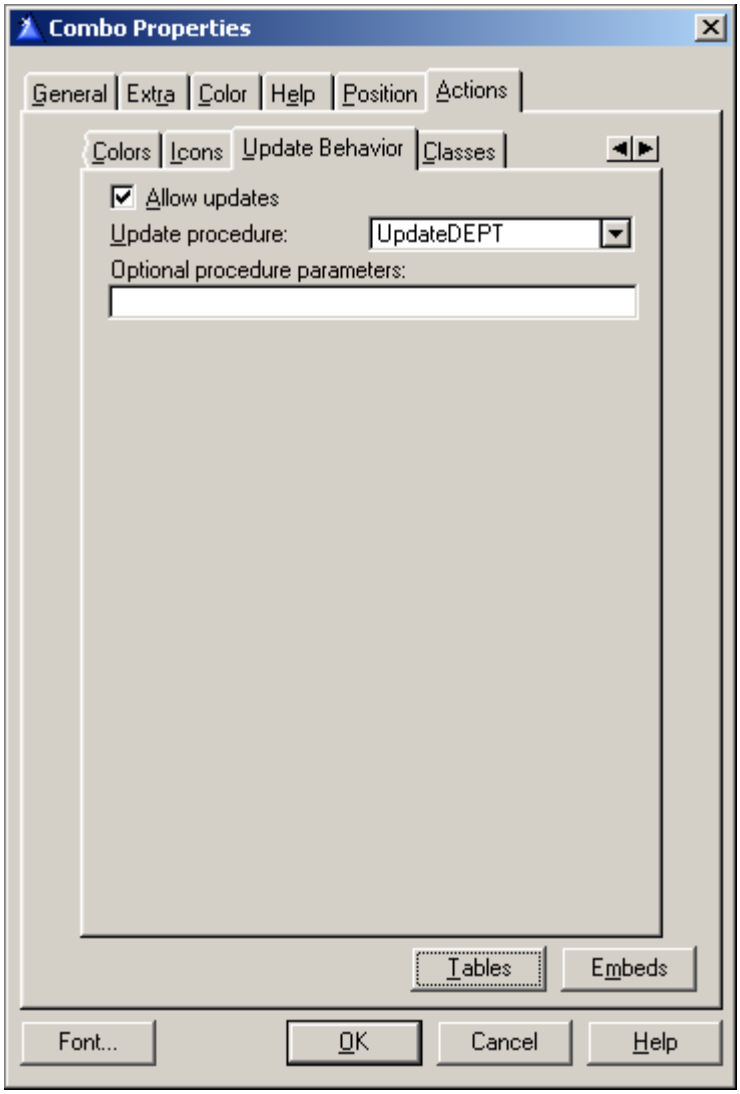
The field that receives the value from the Field to fill from. Press the ellipsis (...) button to select from the Select Field dialog.

More Field Assignments

Add additional fill from/ target fields combinations.

Search columns default Case Insensitive

Make the locator/entry field case insensitive for locating records in the browse.



Allow updates

Allow the user to insert records after entering a value in the locator/entry field that does not exists in the table.

Update procedure

Name the procedure to call to add the new record, or leave this field blank if no update procedure is needed.

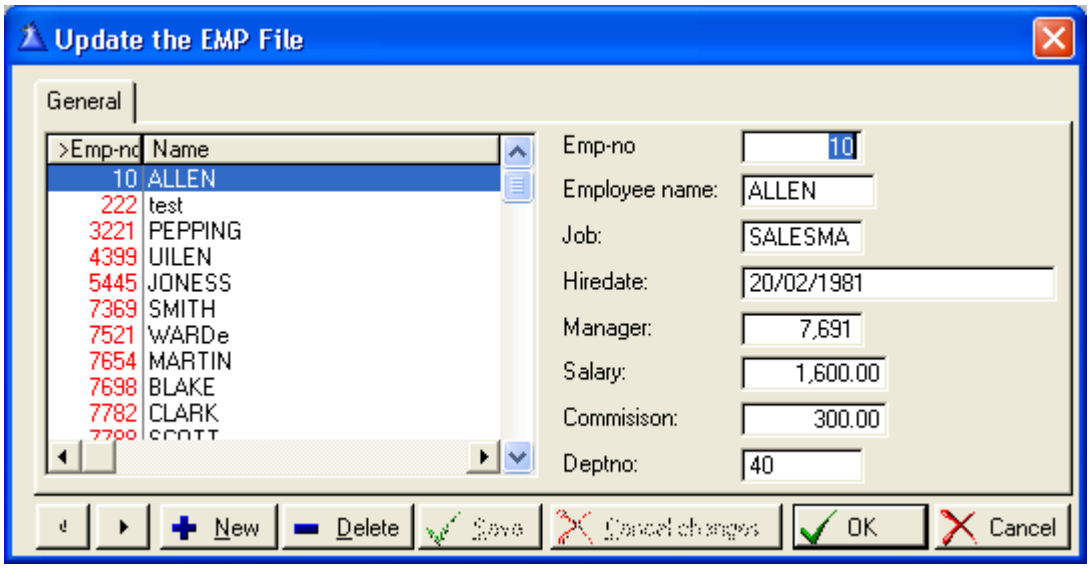
Optional parameters

Here you can enter the parameters for the update procedure if needed.

8 SQL Browse update combination

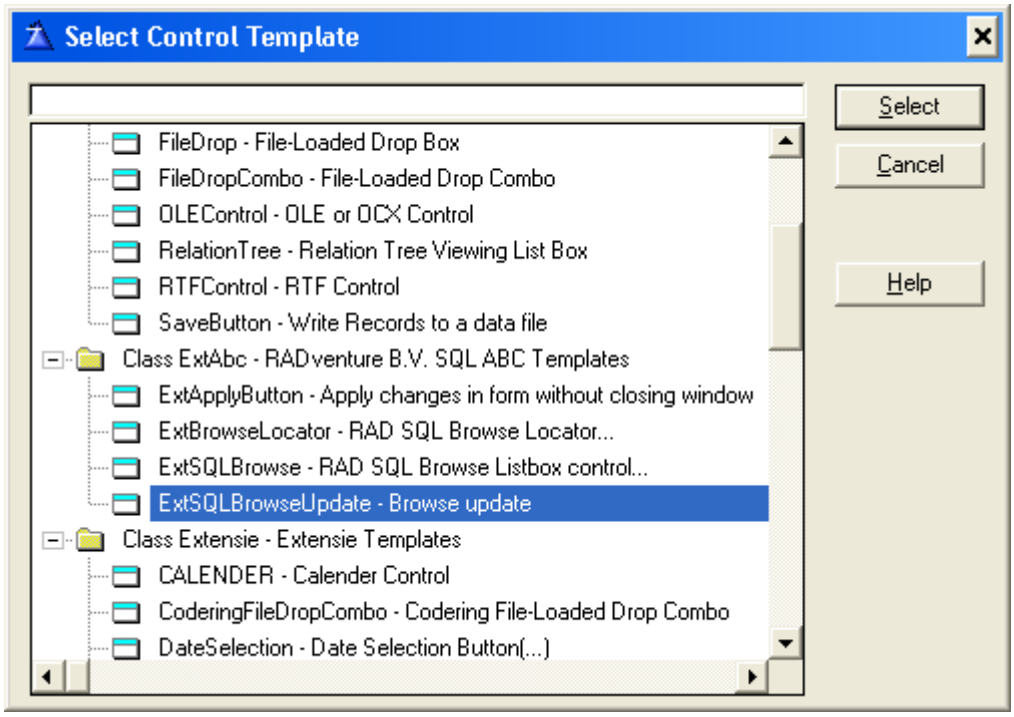
The SQL browse/update combination combines the browse and form into one window. This almost makes separation between browse and form into two procedures obsolete.

The result of this template is a screen as shown below:



To build this procedure, first make a default form procedure. Add to this procedure the RadSQL browse update (list box). Then add the SQL Browse update template. This will automatically add 6 buttons to your form. There is no coding required. A lot of extra embed points are added.

The control ‘Browse update’ is selected via the following control select window:


Select Control Template
Select
Cancel
Help
FileDrop - File-Loaded Drop Box
FileDropCombo - File-Loaded Drop Combo
OLEControl - OLE or OCX Control
RelationTree - Relation Tree Viewing List Box
RTFControl - RTF Control
SaveButton - Write Records to a data file
Class ExtAbc - RADventure B.V. SQL ABC Templates
ExtApplyButton - Apply changes in form without closing window
ExtBrowseLocator - RAD SQL Browse Locator...
ExtSQLBrowse - RAD SQL Browse Listbox control...
ExtSQLBrowseUpdate - Browse update
Class Extensie - Extensie Templates
CALENDER - Calender Control
CoderingFileDropCombo - Codering File-Loaded Drop Combo
DateSelection - Date Selection Button(...)

9 SQL View Manager

9.1 SQL View Managers Properties

ViewTablelist &Ext_TableList

The ViewTableList is a queue with the tables used in the view

ViewFieldpairs &Ext_FieldpairQueue

This queue links the Queue field to the View field. Pointers do this.

SQL Joinfields &Ext_Joinfields
A queue of the fields used in the join

SelectStatement CString(2000)
This string is the generated complete SQL select statement

GlobalSortOrder CString(500),private
Global order by clause will always be directly after order by

OrderByclause CString(1500)
The order by clause for the SQL statement

Fromclause CString(500)
The from clause for the SQL statement

Joinclause CString(1000)
The join clause for the SQL statement

Whereclause CString(2000)
The Where clause for the SQL statement

ViewSQL Statement CString(2000)
The complete SQL Statement for the view

OldSQL statement CStrin
The old SQL statement

Locatorfilter CString(500)
Filter forced by locator

SQL Filter CString(1000)
Filter on View

RangeFilter CString(1000),protected
Range applied

QBEFilter CString(1000)
Filter from QBE

Hint CString(100),protected
Hint

ManagedView &View
The managed view

RecordsInPage ULong
Number of records in listbox

Pagesize Short
Number of records in listbox

NoReFill Byte,protected
No refill of view

StickyLocator Byte,protected
Sticky locator active

QBEActive Byte
Is the QBE active ?

DebugMode Byte
If 1 turns debug mode on

UseTableasAlias Byte
If 1 return table name as alias not A,B, etc.

Distinct Byte
If 1 adds distinct keyword

Viewopened Byte,private
The view was opened if 1

Eof Byte,protected
End of file reached

9.1.1 SQL View Managers Methods

AddViewfieldpair (link the Queue field to the View field)

AddViewfieldpair (Ref_Queuefield, Ref_ViewField, In_Tablename, In_SQL field, in_fieldno, in_colno)

Ref_QueueField	*?	Reference to field in queue
Ref_ViewField	*?	Reference to field in view
In_TableName	String	Tablename
In_SQL Field	String	Fieldname
In_FieldNo	Byte	Fieldnumber in table
In_ColNo	Byte	Fieldnumber in browse

AddViewfield (Add field to the view, not in queue p.e. joinfield)

AddViewfield (In_Tablename, In_SQL field, In_fieldno)

In_TableName	String	Tablename
In_SQL Field	String	Fieldname
In_FieldNo	Byte	Fieldnumber in table

AddTable (Add Table to TableList)

AddTable (In_table, In_fileno, In_Qbeadded),Virtual

In_Table	String	Tablename
In_FileNo	Byte	Tablenumber
In_QBEadded	Byte	QBE added

AddJoinFields (Add joinfieldpairs)

AddJoinFields (Table1, Field1, Table2, Field2, Outerjoin, QbeAdded)

Table1	String	First tablename
Field1	String	Field of first table
Table2	String	Second tablename
Field2	String	Field of second table
OuterJoin	Byte	Use outerjoin
QBEadded	Byte	QBE added

AddMarkField (Add markfield to Queue)

AddMarkField (Ref_Queuefield)

Ref_QueueField	*?	Reference to queue

OpenView (Open the view)

OpenView ()

GenerateSelectStatement (Generate SELECT part of SQL statement)

GenerateSelectStatement (),Virtual

GenerateFromClause (Generate FROM part of SQL statement)

GenerateFromClause (),Virtual

GenerateOrderByClause (Generate ORDER BY part of SQL statement)

GenerateOrderByClause (),Virtual

GenerateJoinClause (Generate JOIN part of SQL statement)

GenerateJoinClause (),Virtual

GenerateWhereClause (Generate WHERE part of SQL statement)

GenerateWhereClause (),Virtual

AddGlobalSortOrder (Global sortorder added to the ORDER BY part of SQL statement)

AddGlobalSortOrder (),Virtual

AddGlobalFilter (Possibility to override the global SQL Filter. New global filter will be added to WHERE part of SQL statement)

AddGlobalFilter (),Virtual

ExecuteSQL (SQL statement executed by View{Prop:SQL } = SQL statement)

ExecuteSQL (),Virtual

SetLocatorFilter (Set filter from locatorfield)

SetLocatorFilter (In_LocFilter),Virtual

In_LocFilter	String	Locator filter statement

SetFilter (Set SQL filter for the WHERE part of SQL statement)

SetFilter (In_Filter),Virtual

In_Filter	String	Filter statement

SetRange (Possibilty to enter or override a RangeFilter for the WHERE part of SQL statement)

SetRange (),Virtual

SetGlobalSortOrder (Set the global ORDER BY)

9.1.1.1.1 SetGlobalSortOrder (In_Order)

In_Order	String	Order statement

SetStickyLocator (Enable/disable stickylocator that saves previous locator filters)

9.1.1.1.2 SetStickyLocator (In_Sticky)

In_Sticky	Byte	True or false to enable or disable stickylocator

Init (Initialize view)

9.1.1.1.3 Init ()

Next (Next record of the view, returns 1 if EOF else 0)

9.1.1.1.4 Next (),Byte,Virtual

In your SQL Browse you can use the derived Next method to validate the record by setting Self.ManualFiltered to True in the embed after the Parent.Next if you want to exclude the validated record from the listbox queue.

Previous (Previous record of the view, returns 1 if ErrorCode else 0)

9.1.1.1.5 Previous (),Byte,Virtual

GetAlias (Returns the alias of the tablename)

9.1.1.1.6 GetAlias (In_Tablename),String

In_TableName	String	Tablename

SetSort (Set sort for ORDER BY part of SQL statement)

9.1.1.1.7 SetSort (In_tablename, In_SetSortfield, Ordertype)

In_TableName	String	Tablename
In_SetSortfield	String	Field to sort on
Ordertype	String	Ascending/Descending

AppendSort (Add additional sortfield for ORDER BY part of SQL statement)

9.1.1.1.8 AppendSort (In_tablename, In_SetSortfield, Ordertype)

In_TableName	String	Tablename
In_SetSortfield	String	Field to sort on
Ordertype	String	Ascending/Descending

ResetView (Reset of the view, calls ExecuteSQL with new SQL statemet)

9.1.1.1.9 ResetView (),Virtual

Destruct (Destruct view object)

9.1.1.1.10 Destruct ()

ClearQueue (Clear records of queue and dispose of queue)

9.1.1.1.11 ClearQueue (QueueInstance)

QueueInstance	*Queue	Reference to a queue

10 SQL Browse Manager

11 SQL Browse Managers Properties

AddFieldpairs (Link Queuefield to Viewfield)

11.1.1.1.1 AddFieldpairs (Ref_Queuefield, Ref_ViewField, In_Tablename, In_SQL field, IN_Type, In_CanbeNull, In_SortASCDES, In_DefSort, IN_Case, IN_USEGE, In_fieldno, in_colno), Virtual

Ref_Queuefield	*?	Reference to the queue
Ref_Viewfield	*?	Reference to the view
In_Tablename	String	Tablename
In_SQL field	String	SQL field
In_Type	String	Type
In_CanBeNull	Byte	Can be NULL value
In_SortASCDES	String	Sort Ascending/Descending
In_DefSort	String	Default sort
In_Case	String	Case
In_UseGE	String	Use Greater or Equal (instead of LIKE)
In_FieldNo	Byte	Field number
In_ColNo	Byte	Column number

AddIndexFunction (Add indexfunction for columnfield)

11.1.1.1.2 AddIndexFunction (IndexFunction)

IndexFunction	String	The indexfunction for the field

AddSort (Add extra sortorder for columnfield)

11.1.1.1.3 AddSort (in_columnnr, in_sortorder)

In_Columnnr	Byte	Column number
In_SortOrder	String	Sort order for field

OverwriteSort (Override sortorder for columnfield with other field)

11.1.1.1.4 OverwriteSort (in_columnnr, in_sortorder)

In_Columnnr	Byte	Column number
In_SortOrder	String	New sort order for field

SetListcontrol (Set listcontrol for SQL browse class)

11.1.1.1.5 SetListcontrol (ListControl)

Listcontrol	Signed	The listcontrol

GetColumn (Return active column number)

11.1.1.1.6 GetColumn (),byte, Virtual

FillPage (Fill listbox queue from view)

11.1.1.1.7 FillPage (), Virtual

UpdateWindow (Enable/disable buttons depending on listbox)

11.1.1.1.8 UpdateWindow (), Virtual

SetFileLoad (Set filling listbox to fileload instead of pageload)

11.1.1.1.9 SetFileLoad ()

Init (Initialize browse class object)

11.1.1.1.10 Init (<Filemanager>), Virtual

FileManager	Class	The filemanager class

TakeEvent (Handle events for listbox ,locator and buttons)

11.1.1.1.11 TakeEvent (), Virtual

TakeKey (Handle alertkeys for the listbox)

11.1.1.1.12 TakeKey (), Virtual

InsertRecord (Insert record)

11.1.1.1.13 InsertRecord (SuppressClear), Virtual

Suppressclear	Byte	Suppress clear of filebuffer, default false

UpdateRecord (Update highlighted record)

11.1.1.1.14 UpdateRecord (), Virtual

DeleteRecord (Delete highlighted record)

11.1.1.1.15 DeleteRecord (), Virtual

SelectRecord (Select highlighted record)

11.1.1.1.16 SelectRecord (), Virtual

TakeScroll (Handle scroll events)

11.1.1.1.17 TakeScroll (E)

E Signed Scroll event

ScrollOne (Scroll one record up or down)

11.1.1.1.18 ScrollOne (Ev)

Ev Signed Event scrollup or scrolldown

ScrollPage (Scroll one page)

11.1.1.1.19 ScrollPage (Direction)

Direction Signed Direction up or down

Previous (Get previous record)

11.1.1.1.20 Previous (),Byte, Virtual

SetQueueRecord (Fill queuerecords fields with values from view)

11.1.1.1.21 SetQueueRecord (), Virtual

TakeNewSelection (Handle new selection in listbox)

11.1.1.1.22 TakeNewSelection (), Virtual

FetchRecord (Retrieve selected record)

11.1.1.1.23 FetchRecord (In_forcefetch), Virtual

In_forcefetch Byte If true first a reget of the view

AddLocator (Add locator for listbox)

11.1.1.1.24 AddLocator (Iby_locatorcontrol,Iby_promptControl)

Iby_Locatorcontrol	Signed	Locatorcontrol of listbox
Iby_Promptcontrol	Signed	Prompt of locator

AddItem (Add EditInPlace)

11.1.1.1.25 AddItem (In_EIPManager)

In_EIPManager	SQL BEipManager EditInPlace manager class

ResetLocator

11.1.1.1.26 ResetLocator (), Virtual

LocateRecord (Locate record from locators value)

11.1.1.1.27 Locaterecord (), Virtual

ResetBrowse (Reset of browse)

11.1.1.1.28 ResetBrowse (Forcerefill), Virtual

ForceRefill	Byte	If true a reset of the view is performed

ResetSort (Reset sort order based on selected column)

11.1.1.1.29 ResetSort (In_col), Virtual

In_Col	Byte	Column number

ResetChildren (Reset of child browses)

11.1.1.1.30 ResetChildren (), Virtual

RegetFromFile (Reget record values from file after completed update)

11.1.1.1.31 RegetFromFile (), Virtual

LocateAfterInsert (Locate record in listbox after insert)

11.1.1.1.32 LocateAfterInsert (), Virtual

SetToDate (Return Oracle dateformat)

11.1.1.1.33 SetToDate (In_date, In_format), String

In_Date	Long	Date/Timefield
In_Format	Long	Date or time

SetSortColColor (Set color for active sort column)

11.1.1.1.34 SetSortColColor (In_color)

In_color	ULong	Date/Timefield

SetForceFetch (Set forcefetch to true or false for FetchRecord)

11.1.1.1.35 SetForceFetch (InForceFetch)

InForceFetch	Byte	True or false

SetColLocatorFilter (Set locatorfilter for active column)

11.1.1.1.36 SetColLocatorFilter (Locatorvalue), Virtual

LocatorValue	String	Value from locator

SetHint (Set hint for active column)

11.1.1.1.37 SetHint (In_col, in_hint)

In_Col	Byte	Column number
In_hint	String	Oracle hint

SetAlerts (Add popup for update and marking)

11.1.1.1.38 SetAlerts (), Virtual

NoRecords (Dummy, programmer can handle no records in listbox)

11.1.1.1.39 NoRecords (), Virtual

RunForm (Run updateform for insert, update or delete record)

11.1.1.1.40 Runform (in_Request),Virtual

In_Request	Byte	Insert-, Change- or DeleteRecord

AddToolbarTarget (Hook ABC toolbar into SQL Browse)

11.1.1.1.41 AddToolbarTarget (T)

T	ToolbarClass	Toolbarclass

UpdateToolbarButtons (Connect browsebuttons with toolbarbuttons)

11.1.1.1.42 UpdateToolbarButtons ()

SetQBEActive (Set QueryByExample is active to true or false)

11.1.1.1.43 SetQBEActive (In_QBEActive)

In_QBEActive	Byte	True or false

Destruct (Dispose of browseclass object)

11.1.1.1.44 Destruct ()

Debug (If in special debugmode show debugmessage)

11.1.1.1.45 Debug (DebugMessage)

Debugmessage	String	Message

GetPrimaryKey (Return primary key of table)

*11.1.1.1.46 GetPrimaryKey (File F), *Key, Virtual*

F	File	Filelabel

RetainBrowsePosition (Retain browse position at opening browse or after update)

11.1.1.1.47 RetainBrowsePosition (StartCol, UseQueue), Virtual

StartCol	Byte	Column number to sort on

UseQueue	Byte	If true retain from last selected record. If false from INI-file

SaveBrowsePosition (Save browseposition in INI-file)

11.1.1.1.48 SaveBrowsePosition (), Virtual

SetMarking (Turn marking on or off)

11.1.1.1.49 SetMarking (SetMarkOnOrOff), Virtual

SetMarkOnOrOff	Byte	Turn marking on or off

SetMarkRange (Turn mark on or off for range of records in listbox)

11.1.1.1.50 SetMarkRange (BeginPosition, EndPosition, SetMarkOnOrOff, CurrentChoice), Virtual

BeginPosition	Long	Beginposition in listbox
EndPosition	Long	Endposition in listbox
SetMarkOnOrOff	Byte	Turn mark on or off for range
CurrentChoice	Long	If not 0 this record is not turned on or off

12 SQL EditInPlace Manager

12.1 SQL EditInPlace Managers Methods

InitControls (Initialize edit in place fields of listbox)

12.1.1.1.1 InitControls ()

ClearColumn (Handle EIP record when new selection)

12.1.1.1.2 ClearColumn ()

Construct (Make new Edit in Place object)

12.1.1.1.3 Construct ()

Init (Initialize EditInPlace object)

12.1.1.1.4 Init (),Byte

Kill (Dispose of EditInPlace object)

12.1.1.1.5 Kill ( ),Byte

CopyDbFields (Copy Queuefields to viewfields before insert or change record is completed)

12.1.1.1.6 CopyDbFields ( ),Virtual

TakeCompleted (Handle EditInPlace of row is completed)

12.1.1.1.7 TakeCompleted (Force)

Force	Byte	With or without asking user

TakeEvent (Handle events)

12.1.1.1.8 TakeEvent ( ),Byte

TakeNewSelection (Handle selection of new row)

12.1.1.1.9 TakeNewSelection ( ),Byte

SetRequest (Set request for EditInPlace action)

12.1.1.1.10 SerRequest (In_Request)

In_Request	Byte	Insert or Update

NoEIPCol (Set column to no EditInPlace allowed)

12.1.1.1.11 NoEIPCol (In_Col)

In_Col	Byte	Column number where no EIP is allowe

13 SQL DropCombo Manager

13.1 SQL DropCombo Managers Methods

Construct (Initialize queues used by drop combo class)

13.1.1.1.1 Construct ()

Destruct (Free queues used by drop combo class)

13.1.1.1.2 Destruct ()

SetListControl (Initialize listcontrol used by drop combo class)

13.1.1.1.3 SetListControl (ListControl)

Listcontrol Signed The Field Equate Label of the droplist

TakeEvent (Event handling for list- and locatorcontrol)

13.1.1.1.4 TakeEvent (), Virtual

FetchRecord (Retrieve selected record)

13.1.1.1.5 FetchRecord (In_ForceFetch), Virtual

In_ForceFetch	Byte	If true first a reget of the view

AddDropVariables (Initialize droplist variables)

13.1.1.1.6 AddDropVariables (Iby_locatorcontrol, Iby_promptControl, UseField, FillField, FillFieldReference, TargetReference)

Iby_locatorcontrol	Signed	Field Equate Label of locatorcontrol
Iby_promptcontrol	Signed	Field Equate Label of locators prompt
UseField	*?	The use variable of the droplist
FillField	String	Field to fill from
FillFieldReference	*?	Reference to the field to fill from
TargetReference	*?	Reference to the field to fill, the targetfield

LocateRecord (Locate record for value entered in the locatorcontrol)

13.1.1.1.7 LocateRecord (), Virtual

ResetSort (Reset sort order based on active column)

13.1.1.1.8 ResetSort (In_Col), Virtual

In_Col	Byte	Column number

AddField (Add field combinations: fill from fields/targetfields)

13.1.1.1.9 AddField (Source, Destination), Virtual

Source	*?	Reference to field to fill from
Destination	*?	Reference to field to fill , the target field

RunForm (Set global request for updateprocedure)

13.1.1.1.10 RunForm (In_Request), Virtual

In_Request	Byte	GlobalRequest for updateprocedure, in case of this droplist only insertrecord

InsertRecord (Prime record before going to update procedure and reset of the browse if response is requestcompleted)

13.1.1.1.11 InsertRecord (SurpressClear), Virtual

SurpressClear	Byte	If true suppress clearing of filebuffer before calling RunForm with request is insertrecord

StartUpdateProcedure (Get of view or clearing of filebuffer depending on surpressclear before calling insertrecord)

13.1.1.1.12 StartUpdateProcedure (SurpressClear), Virtual

SurpressClear	Byte	If true suppress clearing of filebuffer before calling Insertrecord

14 RADventure Support

RADventure Support	
Email:	tools@radventure.nl
Telephone:	+31 (0)346 29 09 80
Fax:	+31 (0)346 29 09 05
Post:	PO Box 1069, 3600 BB Maarssen, The Netherlands

15 Copyright