

VARPG Advanced topics

P99

ITSO AS/400 e-business University/Technical Forum

Claus Weiss



Acknowledgement and Disclaimer

Acknowledgement:

Many people contributed to this presentation. In particular thanks goes to:
Larry Schweyer, IBM Corporation, Toronto Lab

Disclaimer:

The information contained in this document has not been submitted to any formal IBM test and is distributed on an as is basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customers' ability to evaluate and integrate them into the customers' operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

Reproduction:

The base presentation is the property of IBM Corporation. Permission must be obtained PRIOR to making copies of this material for any reason.



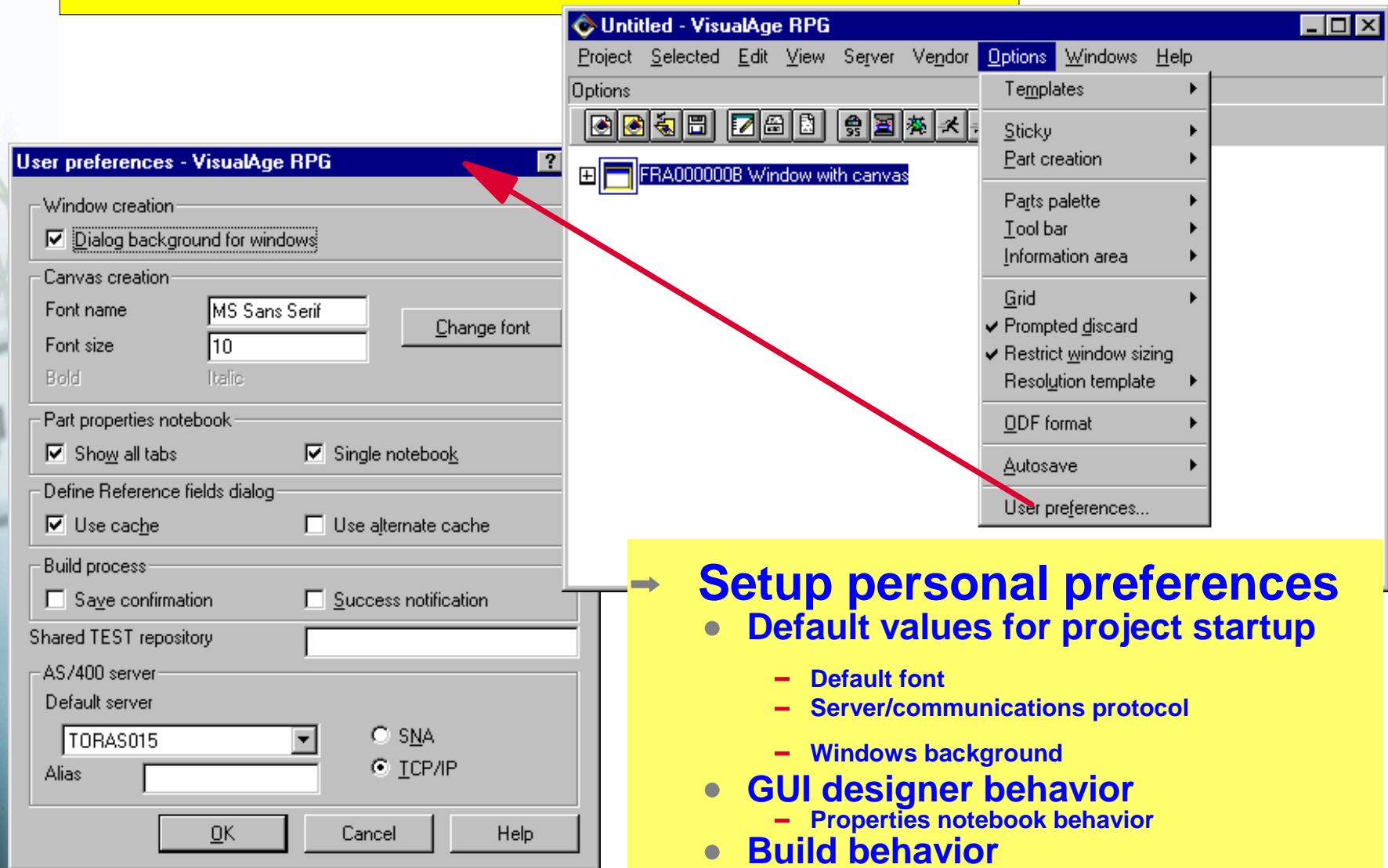
e-business

Agenda

- ➔ User preferences dialog
- ➔ Some subfile features
- ➔ Calling Windows DLL's
- ➔ *Component part (**Mystery part?**)
 - Attributes (how to use them)
- ➔ Calendar part
- ➔ Projects/applications/components
- ➔ Component Reference Part
- ➔ PlugIns
- ➔ Other neat stuff
 - Browse/Merge projects
 - Graph part
 - ODBC part
 - Progress bar, Status bar, Animation control
- ➔ ActiveX part
- ➔ Java source code generation
- ➔ M3 new features

User preferences dialog

GUI designer Menu bar select Options



The image shows two overlapping windows from the VisualAge RPG software. The background window is titled 'Untitled - VisualAge RPG' and has a menu bar with 'Options' selected. The 'Options' menu is open, showing various settings like 'Templates', 'Sticky', 'Part creation', 'Parts palette', 'Tool bar', 'Information area', 'Grid', 'Prompted discard', 'Restrict window sizing', 'Resolution template', 'QDF format', 'Autosave', and 'User preferences...'. A red arrow points from the 'User preferences...' menu item to the foreground window.

The foreground window is titled 'User preferences - VisualAge RPG' and contains several sections of settings:

- Window creation:** Dialog background for windows
- Canvas creation:** Font name: MS Sans Serif, Font size: 10, Change font button, Bold, Italic
- Part properties notebook:** Show all tabs, Single notebook
- Define Reference fields dialog:** Use cache, Use alternate cache
- Build process:** Save confirmation, Success notification
- Shared TEST repository:** [Empty text field]
- AS/400 server:** Default server: TORAS015, SNA, ICP/IP, Alias: [Empty text field]

Buttons at the bottom: OK, Cancel, Help.

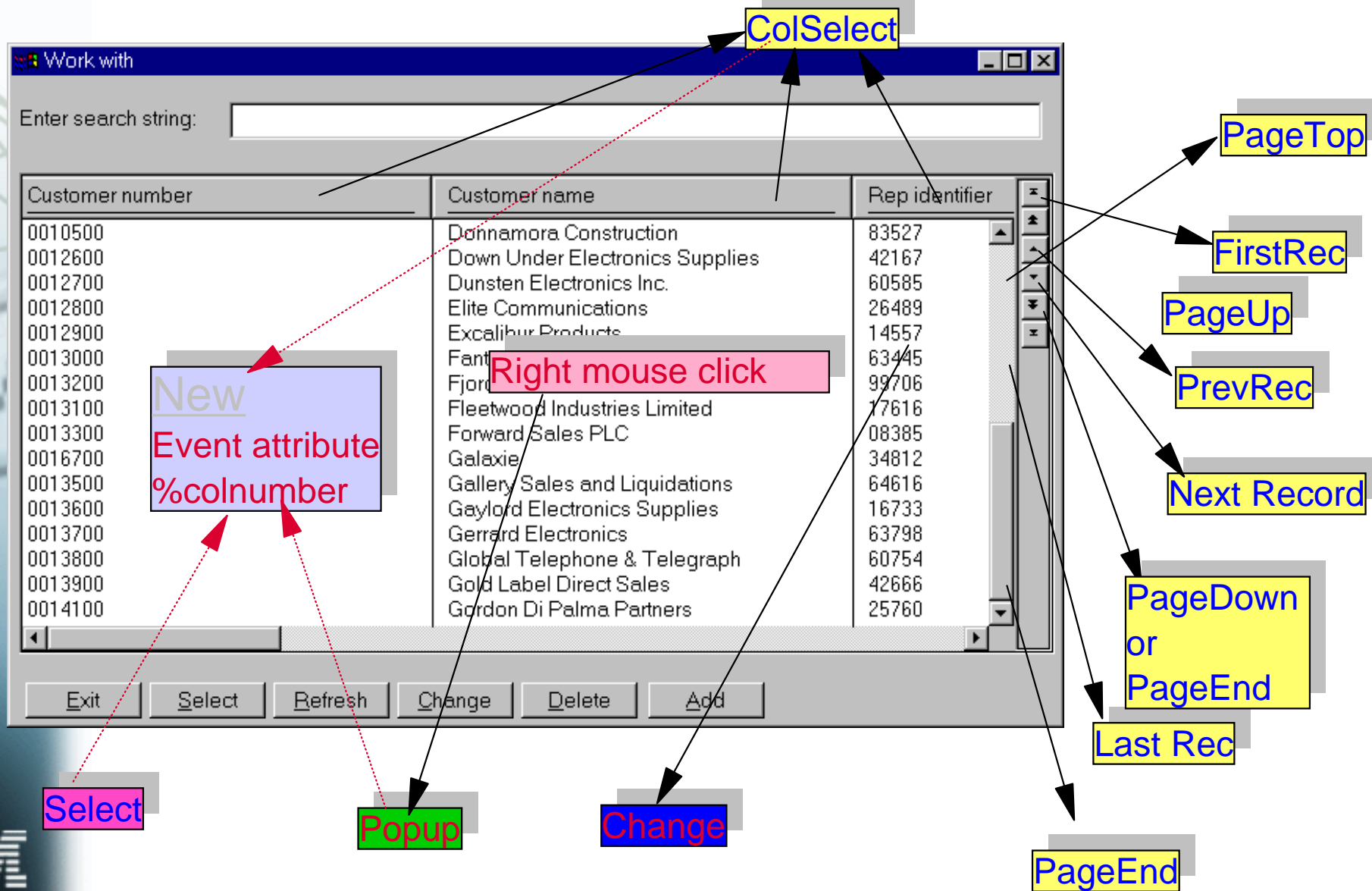
Setup personal preferences

- Default values for project startup
 - Default font
 - Server/communications protocol
 - Windows background
- GUI designer behavior
 - Properties notebook behavior
- Build behavior
 - No message when SAVE is required
 - No message when BUILD is OK



e-business

Enhanced subfile support....Events





e-business

Enhanced subfile support....Colour/Font attributes

Heading colour

HdgBGClr/HdgBGMix

HdgFGClr/HdgFGMix

Subfile colour

BackColor/BackMix

ForeColor.ForeMix

Cell colour

CellFGClr/CellFGMix

CellBGClr/CellBGMix

→ Row colour

→ RowFGClr/RowFGMix

→ RowBGClr/RowBGMix

→ Column Colour

→ ColFGClr/ColFGMix

→ ColBGClr/ColBGMix

Work with

Enter search string:

Customer number	Customer name	Rep identifier
0016000	A/S Mobiltelefon	77107
0017700	Artigos de Escritorio Ltda.	73282
0010800	Baker Electronics	15445
0011000	Bayview Direct Sales	28899
0011100	BelAir Communications	12203
0011200	Bridlewood Electronics Supplies	10716
0011300	Burnham Trading Inc	41480
0011400	Calderone Imports	00477
0010600	Chicago Wholesale Suppliers	61636
0011700	Christies Electronics Limited	32484
0011800	Clement Works	64349
0013400	Communications Fournier	13497
0012200	Communications-R-Us Corporation	22004
0017000	Computer Friends B.V.	48672
0011900	Conroy Communications	25189
0012000	Cumberland Specialties	17983

Exit Select Refresh Change Delete Add

Font -Bold -Italic -Name -Size -Strike -Under





e-business

Calling Windows API's

- Keywords
 - DLL
 - Linkage(*StdCall)
 - For Windows API's
 - No Linkage keyword for optimized

Lots more examples are on our Tips page
just look at

→ <http://www.ibm.com/software/ad/varpg>

Allows calling WINAPI's in DLLs like
Kernel32
user32

→ Description of Windows API's can be found at:

● <http://www.vbapi.com/>

```
* parm1 byVal nBufferlength as Long
* parm2 byVal lpBuffer As String
* define long as 10i 0
d* prototype for Current directory API
dGetCurDir      pr      10I 0 ExtProc('GetCurrentDirectoryA')
d                                     DLL('Kernel32.DLL')
d                                     Linkage(*StdCall)
d                                     10I 0 Value
d                                     255A
d* variable definition
d CurDir          s      255a
d curdirsiz      s      10i 0 Inz(%size(curdir))
c* get current directory name
c                  eval   Rclong=GetCurDir(curdirsiz:curdir)
```

Calling Windows API's

➔ Mapping C datatypes to RPG datatypes

API Data type	Length	VARPG Data type	Declaration
INT, Long	32 bits	Signed Integer length 10	10I 0
UINT, ULONG DWORD, HANDLE	32 bits	UnSigned Integer length 10	10U 0
SHORT	16 bits	Signed Integer length 5	5I 0
USHORT, WORD	16 bits	UnSigned Integer length 5	5U 0
CHAR, BYTE	8 bytes	Character, length 1	1A
BOOL	32bits	Binary, length 1	1B
FLOAT	32 bits	Float length 8	8F
Pointer	32 bits	Pointer	*

➔ **C API definition**

```

• BOOL FlashWindow(HWND hWnd, // handle of window to flash
  BOOL bInvert ); // flash status
  
```

➔ **RPG Prototype**

```

.... DName+++++++ETDsFrom+++To/L+++IDc.Keywords+++++++
0001 D Flash PR 1B 0 ExtProc('FlashWindow')
0002 D DLL('USER32')
0003 D Linkage(*StdCall)
0004 D 10U 0 Value
0005 D 8 1B 0 Value
  
```



e-business

Calling Windows API's

- ➔ C default is passing parameter by value
- ➔ RPG default is passing parameter by reference
 - For RPG prototypes specify VALUE keyword
- ➔ If C specifies passing a pointer
 - Don't specify VALUE keyword in RPG

➔ Declare Function FindNextFile Lib "kernel32.dll" Alias "FindNextFileA" (ByVal hFindFile As Long, lpFindFileData As WIN32_FIND_DATA) As Long

➔ hFindFile

➔ The handle to the file search initiated by FindFirstFile.

➔ lpFindFileData

➔ Receives identifying information about the next matching file that was found.

```

D FindNext      PR          10I 0 ExtProc('FindNextFileA')      Find next file
D                                                       DLL('KERNEL32')
D                                                       Linkage(*StdCall)
D               10U 0 Value                                     Handle
D               * Value                                       -> data structure
*
D fHnd          S           10U 0                               Find handle
D bRC           S           10I 0 Inz(1)                       Long rtn code
*
D pData         S           *   Inz(%Addr(FindData))           -> data structure
D pDir          S           *   Inz(%Addr(Dira))                -> directory name

```

➔ Specify Value for this pointer otherwise it points to the address of the pointer

```
C          Eval          bRC=FindNext(fHnd: pData)
```



e-business

Calling Windows API's

- Type WIN32_FIND_DATA
- dwFileAttributes As Long
- ftCreationTime As FILETIME
- ftLastAccessTime As FILETIME
- ftLastWriteTime As FILETIME
- nFileSizeHigh As Long
- nFileSizeLow As Long
- dwReserved0 As Long
- dwReserved1 As Long
- cFileName As String * 260
- cAlternate As String * 14
- End Typed

Definition of Structure WIN32 Find DATA

Description of fields from API guide

- hdwFileAttributes
 - One or more of the following flags identifying the file's attributes:
 - FILE_ATTRIBUTE_ARCHIVE = &H20
 - An archive file (which most files are).
 - FILE_ATTRIBUTE_COMPRESSED = &H800
 -
- ftCreationTime
 - The time and date of when the file was created.

```

D FindData          DS
D  Attr              10I 0
D  CrtTime           Like(FileTime)
D  AccTime           Like(FileTime)
D  WrtTime           Like(FileTime)
D  SizeHigh          10I 0
D  SizeLow           10I 0
D  Rsv0              10I 0
D  Rsv1              10I 0
D  FileName          260A
D  AltName           14A
  
```



e-business

Calling Windows API's

➔ Definition of Structure FILETIME

- Type FILETIME
- dwLowDateTime As Long
- dwHighDateTime As Long
- End Type

➔ Description of fields from API guide

- Description & Usage
- The FILETIME structure holds a date and time associated with a file. The structure identifies a 64-bit integer specifying the
- number of 100-nanosecond intervals which have passed since January 1, 1601. This 64-bit value is split into the two dwords stored in the structure.

D	FileTime	DS		File time structure
D	CrtLow		10I 0	
D	CrtHigh		10I 0	





Sample RPG procedure to get filenames

e-bus

```

c*  procedures starts here Find file with .RST extension  in directory structure
p getdir          b
d getdir          PI
d dir             255a
d dira           s      255a
d dirb           s      255a
d dirc           s      255a
DFindFirst       PR          10I 0 ExtProc('FindFirstFileA')      Find first file
D                                                         DLL('KERNEL32')
D                                                         Linkage(*StdCall)
D                                                         *   Value          -> directory name
D                                                         *   Value          -> data structure
*
D FindNext        PR          10U 0 ExtProc('FindNextFileA')      Find next file
D                                                         DLL('KERNEL32')
D                                                         Linkage(*StdCall)
D                                                         10U 0 Value          Handle
D                                                         *   Value          -> data structure
*
D fHnd           S          10U 0                                Find handle
D bRC            S          10I 0 Inz(1)                        Boolean rtn code
D pData         S          *   Inz(%Addr(FindData))            -> data structure
D pDir          S          *   Inz(%Addr(Dira))                 -> directory name
*
D FindData       DS                                WIN32_FIND_DATA
D Attr           10I 0
D CrtTime        Like(FileTime)
D AccTime        Like(FileTime)
D WrtTime        Like(FileTime)
D SizeHigh       10I 0
D SizeLow        10I 0
D Rsv0           10I 0
D Rsv1           10I 0
D FileName       260A
D AltName        14A

```



e-business

Sample RPG procedure to get filenames.....

```
D FileTime          DS
D  CrtLow           10I 0
D  CrtHigh         10I 0
C                   Eval    reccount=reccount+1
C
C                   eval    %setatr('winbar':
C                   'winbar':'refresh')=1
C
C                   Eval    Dira= %trim(dir) + '\*.rst' + x'00'
C                   eval    %setatr('winbar':
C                   'stxupdate':'label')=
C                   btext + ' ' + dira
C
C                   Eval    fhnd=FindFirst(pDIR: pData)
C                   if      fhnd=-1
C                   eval    brc=0
C                   endif
C
C                   *
C                   DoW      bRC=1
C                   * If it's a normal file
C                   If      Attr= 32
C                   eval    filename=%Replace(' ':filename:
C                   %scan(x'00':filename:1))
C                   eval    files=%trim(dir) + '\' + %trim(filename)
C                   eval    %setatr('winbar':
C                   'stxupdate':'label')=
C                   btext + ' ' + files
C                   write   subdir
C                   eval    indlist=*on
C
C                   EndIf
C                   eval    files=*blank
C                   eval    filename=*blank
C                   Eval    bRC=FindNext(fhnd: pData)
C                   enddo
```



e-business

Sample RPG procedure to get filenames.....

```
c* look for sub directories
c          if          %getatr('fra0000b':'chksub':'checked')=1
C          Eval        bRC=1
*
C          Eval        Dira= %trim(dir) + '\*.*' + x'00'
C          Eval        fHnd=FindFirst(pDIR: pData)
c          dow         brc=1
c          eval         filename=x'00'
C          Eval        bRC=FindNext(fHnd: pData)
c          eval         filename=%Replace(' ':filename:
c                      %scan(x'00':filename:1))
c* return attribute value of 16 means directory
c* sometimes funny dir names get returned need to filter them out
c          if          attr=16 and %trim(filename) <> '.'
c                      and %trim(filename) <> '..'
c                      and %trim(filename) <> ''
c
c          eval        dirb= %trim(dir) + '\' + %trim(filename)
c          → Recursion of procedure "getdir" to search subdirectory
c
c          callp        getdir(dirb)
c          eval         reccount=reccount-1
c          endif
C          EndDo
c          endif
p getdir              e
```



e-business

*Component part (Mystery Part)

➤ Not on the parts palette or the parts catalog

but

➤ all **VARPG** applications and components got one **Component* part

➤ No events associated with this part

but it features many very useful attributes:

Active	Alarm	AppData	Button
ClipBoard	CurrentDir	Dialog	DirName
DlgOwner	DspHeight	DspWidth	FontName
HelpWindow	MsgData	MsgFile	MsgId
MsgText	Name	OS	Parent
PartCount	PartList	PlugCmd	PlugDLL
PlugId	PlugRC	PlugResult	Printer
SelPrinter	WrkStnName	SwitchTo	

Sample SETATR for *COMPONENT part

➔ Display SAVEAS Dialog window

```
'*Component'      SETATR          2          'DIALOG'
```

➔ New attributes

- DoEvents, FileName, ShData, ShDataLen, ShDataName, ShDataPos, ShowMsgID, **Platform.**





e-business

*Component part attributes

(Mystery Part)

- Active -already running no-> 0/ yes->1
- AppData -allows other components to access this data via component reference part
- Button- returns which button was pressed in File dialog
- Clipboard -allows to exchange data with other applications
- CurrentDir -returns a string with the current directory path
- Dialog -allows to display the standard OPEN or SAVE AS dialog
- DirName -returns Applications/components directory
- DlgOwner - determines whether DIALOG is modal or not
- HelpWindow - displays a specific help panel
- FileName - returns the file name selected from File Dialog
- MsgData - specifies message substitution text
- MsgFile - specifies alternative message file name
- MsgId - specifies message id for MsgText attribute
- MsgText - gets message text for message id specified before
- Name - returns application/component name
- OS - returns whether running in 16 bit to 32 bit environment
- Parent - set to identify parent name for PartList attribute
- PartCount - returns number of children of parent part
- PartList - returns list of children part names
- Printer - returns the printer name from Printer Select Dialog
- PrinterSel - displays the printer Select Dialog

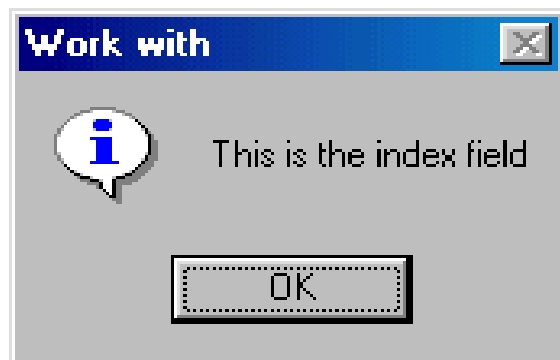


e-business

*Component part attributes

(Mystery Part)

- SwitchTo - activates running copy of application, doesn't allow second invocation
- DOEvents, allows to release control from current Action Subroutine
- File name, used to set file name in dialog
- ShowMsgID, determines whether to show MSGID in message box



- ShData considerations

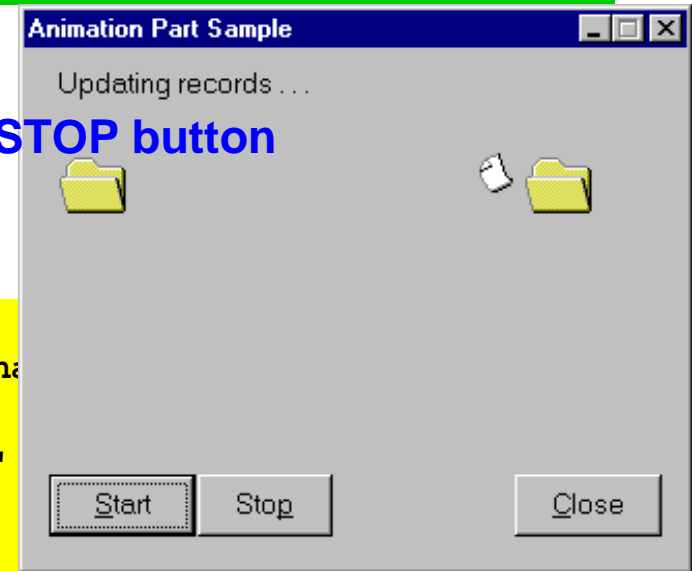
- Use only one instance of ShDATA per application
- Use ShData only to communicate from application to application
- Don't use to communicate between components in same process
- **Not following these rules will cause CALL stack corruption**
 - ShData, SET/Get data into/from this space
 - ShDataLen, size of data to be used, default 1024 bytes
 - ShDataName, name of space, fixed length of 1024
 - ShDataPos, offset to be used for GET/SET data



e-business

*Component part attribute DoEvents

- ➔ Start pushbutton to initiate loop
- ➔ DoEvents allows to get Press event from STOP button



```

* animation file
c      'ANI1'          Setatr    'FILECOPY.AVI' 'filena

C      'ANI1'          Setatr    1          'Mode'

*
* Do 100000 times or until stopped
C          DoW          I < 100000 AND Stop = *OFF
* Do some useful work here
* For now, just loop for a while
C          Do          10000          X
C          Move        'Test'        Field
* Give other events a chance
C       '*Component' Setatr    1          'DoEvents'
C          EndDo
c          add          1          I
C          EndDo
* All done; stop the animation control
C      'ANI1'          Setatr    2          'Mode'
c          .....
C          ENDACT

.....
C      PB_STOP        BEGACT    PRESS        MAIN
C          Move        *on        Stop
C          ENDACT

```

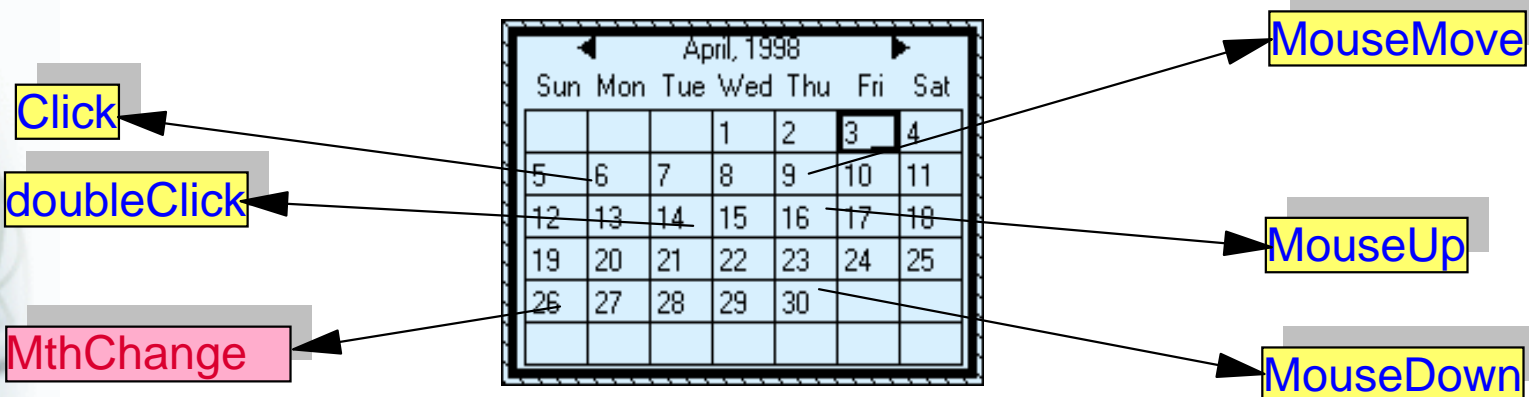


e-business

Calendar part

Calendar part used for selection of date

- Small calendar
- No space for text



DateUnder → Returns blank if not on valid date

Date	DateText		
Day	Month	Year	Weekday
DayIdx	MonthIdx	YearIdx	WeekdayIdx

- Use **date under** to get selected date at click event
- Use these attributes to set/get calendar date and text attributes
- Index attributes for setting calendar information without changing the calendar date



e-business

Calendar part

Calendar part used for scheduling

- Big calendar
- Space for text

Attributes to change the looks

- Border
- Color
- ColorArea
- Colormix
- DayNumPos
- DayNumRect
- FontArea
- FontBold
- FontItalic
- FontName
- FontSize
- FontStrike
- FontUnder
- FrmtString
- MonthArrow
- MonthLen
- OutLineRcl
- Showrects
- Showtext
- YearLen

New in Servicepack 3 (M1)

- DayStart
- VRule
- HRule
- Daylen

April, 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Calendar for Claus Weiss						
Apr, 98						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			1	2	3	4
5	6 Test string	7	8	9	10	11
12	13	14	15	16	17	18 Test string
19	20	21	22	23	24	25
26	27	28	29	30		



e-business

Sample code calendar part

```
C      MYCALENDAR      BEGACT      CLICK      WINCAL
c
c      eval      %setatr('showsel':
c      'showsel':'visible')=1

c      eval      datesels=%getatr('wincal':
c      'mycalendar':'dateunder')
c      if      datesels<>*blank
c      eval      %setatr('wincal':
c      'mycalendar':'date')=datesels
c      eval      daysel=%getatr('wincal':
c      'mycalendar':'day')
c      eval      weekdsel=%getatr('wincal':
c      'mycalendar':'weekday')
c      eval      weekdayi=%getatr('wincal':
c      'mycalendar':'weekdayidx')
c      write      'showsel'
c      endif
c      *
C      ENDACT
```



e-business

Plug ins

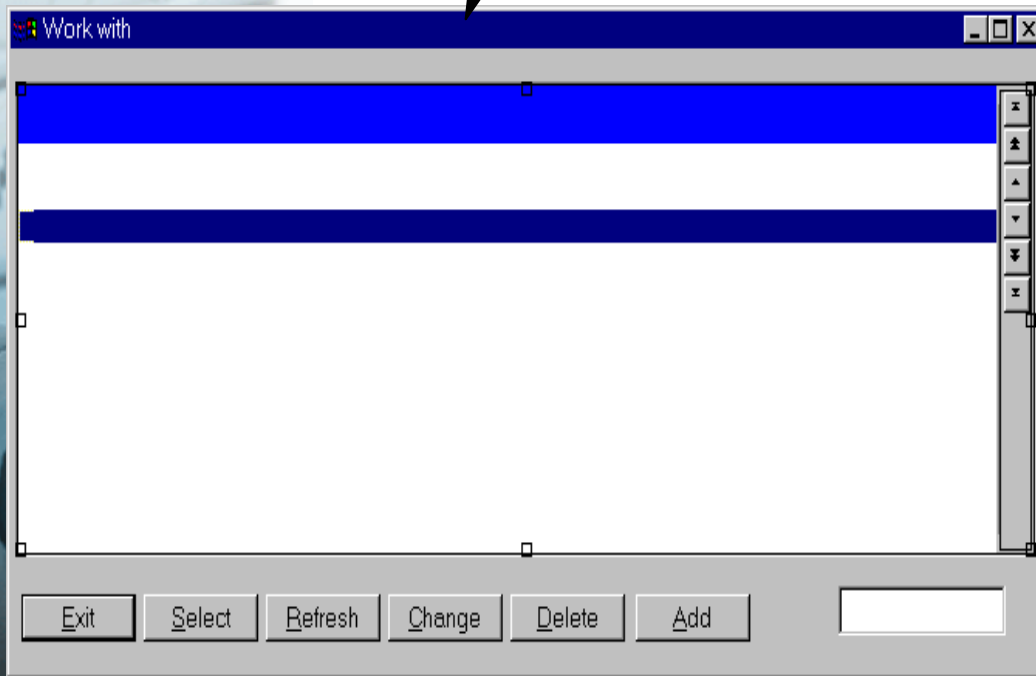
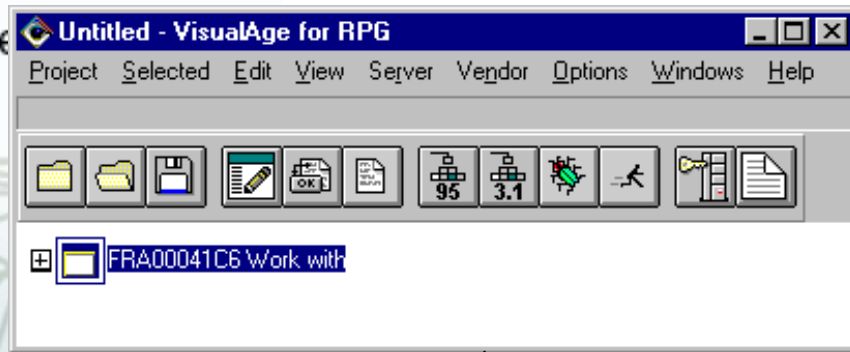
- An easy way to add tools to the GUI designer
- Write your own tools using VARPG, C, or REXX
- 3 samples come with VARPG
- To write your own Plugin you need
 - a Plug-in file that tells the GUI Designer where to find your plugin
 - an executable that gets invoked from the GUI designer menu
- *Component part provides interface for VARPG Plugins
 - attributes **PlugCmd**, **PlugDLL**, **PlugId**, and **PlugRC**
 - PlugDLL and PlugId are used to link the VARPG program and the GUI designer
 - PlugCmd is used to issue commands to the GUI designer
 - PlugRC will provide Return Codes from the GUI designer
-

The IBM logo, consisting of the letters 'IBM' in a bold, sans-serif font, with horizontal lines through the letters.



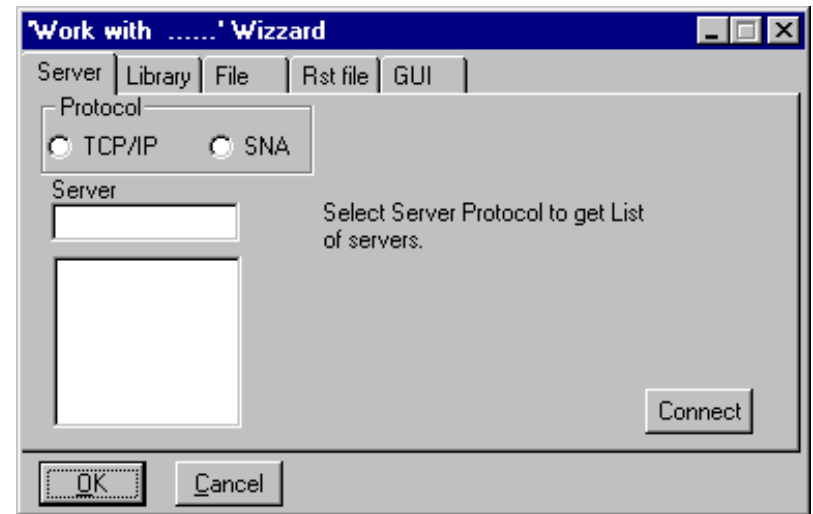
Plug in interface with GUI designer

GUI designer



PLUGIN Targets

- Project
- PartClass
- Part
- Subroutine
- Grid
- LPEX
- PLUGIN
- Registry
- GUI designer





e-business

Sample plug in interface description

Full description in VARPG manual

- *Programming with VisualAge for RPG*

→ Target part

Targets/Commands and the associated Return Values		
Name	partId	Returns the name of the part as shown in the treeview and settings notebook.
OpenDesignWindo	partId [1]	When set to 1, opens and sets focus to the design window to which the indicated part belongs. If set to 0, then the design window is closed instead.
OpenPart	partId	Either opens the part's settings notebook, or if the part is a frame, opens the part's corresponding design

Navigation buttons: Previous, Search..., Print..., Index, Contents, Back, Forward





e-business

Sample code for plugin

```

c* Parameters passed from PLUGIN interface
C   *Entry      Plist
C   Parm        PlugDLL      255
C   Parm        PlugID       255
C   Parm        PartsIDs     255
c   parm        builderpa    50
*****
*
* Description:
* Set focus to Entry field when window gets created
*
*****
C   WINPROMPT   BEGACT   CREATE   WINPROMPT
c               eval    %setatr('winremind':
c               'winremind':'visible')=0
c
c* Get PLUGIN information from parameters
c* Use that information to setup *component part
c*
C   '*COMPONENT' Setatr   PlugDLL   'PlugDLL'
C   '*COMPONENT' Setatr   PlugID    'PlugID'
c* get subfile name
C               Eval     Partid=plugcmd('Project':'CursoredPart':' ')
c               eval     parttype=plugcmd('Part':'ClassName':

```

Interface to GUI designer

→ Use PLUGIN commands to communicate with GUI designer

- Get PartID of current active part from Designer
- Use PartID to get the part type of current active part
- If it is a subfile, do something





e-business

Sample code for plug in continued

```
*
c          eval          subfname=subname
c*      subname          dsply          msg2          rs          9 0
c* Get window name
c          eval          parentid=%subst(partsids: 1 :
c                          %scan('.':Partsids)-1)
c          eval          winname=plugcmd('Part':'Name':
c                          %trim(parentid))
c*      store window name to check whether it is changed later in the process
c*
c          eval          winnamep=winname
c* check whether the subfile has children
c* if not display reminder window
c          eval          lpline=plugcmd('Part':'children':
c                          %trim(partid))
c          if            lpline=*blank
c          eval          %setatr('winremind':
c                          'winremind':'visible')=1
c          return
c          endif
```





Creating a Project for an application



Save as Application - VisualAge RPG

Application name: 1 Subfile example

New application New component New shared component

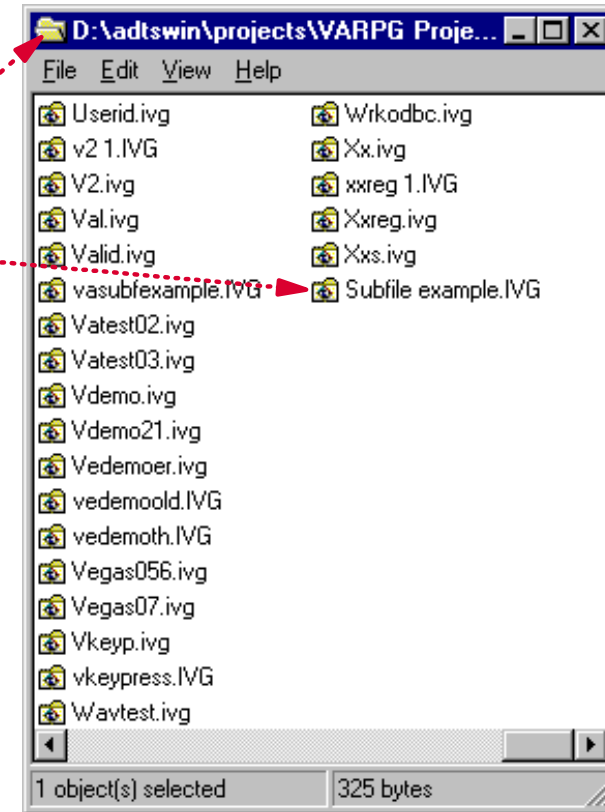
Target folder or project: 2 D:\ADTSWIN\PROJECTS\WARPG Projects Find...

Source file: 3 Subfile

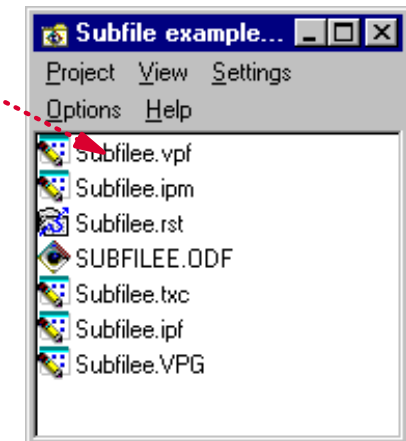
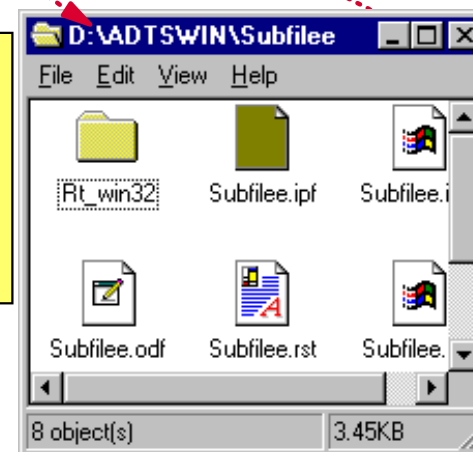
Source directory: 4 D:\ADTSWIN\Subfile

OK Cancel Help

Title of the component or application to be created.

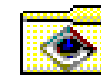


1. Specify project/application name
2. Specify location for 'project' file (can be in a folder or another project)
3. Specify subdirectory to contain source files
4. Specify file 'stub' name





Creating Project for a component



→ Component has to be **STARTED**
no command line invocation

Save as Component - VisualAge RPG

Component name: 1 Listcustomer

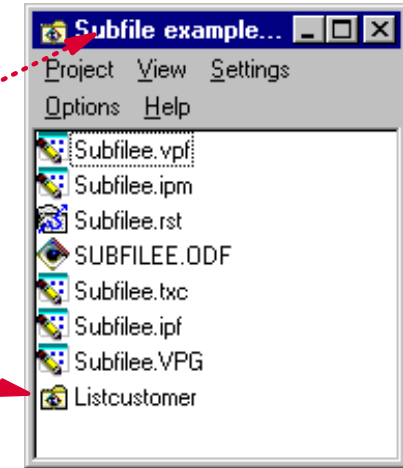
New application New component New shared component

Target folder or project: 2 D:\ADTSWIN\PROJECTS\VARPG Projects\Subfil Find...

Source file: 3 Listcust

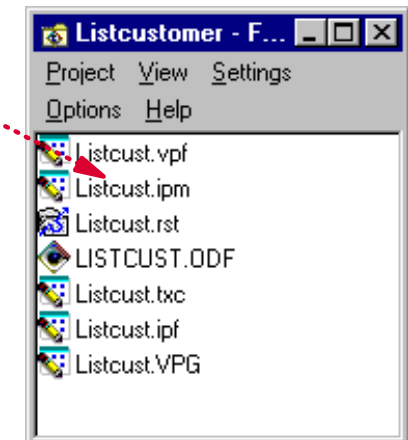
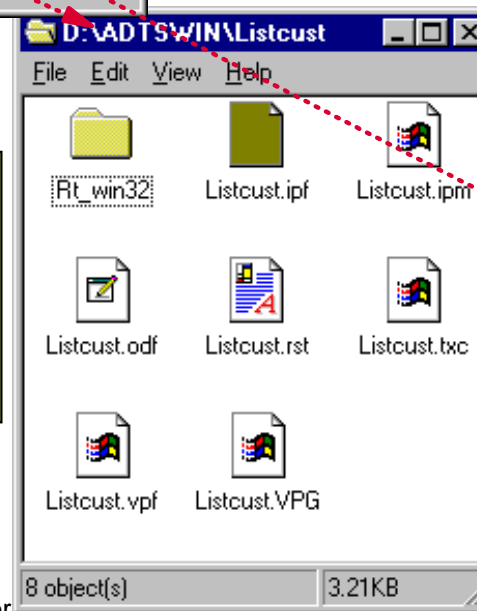
Source directory: 4 D:\ADTSWIN>Listcust

OK Cancel Help



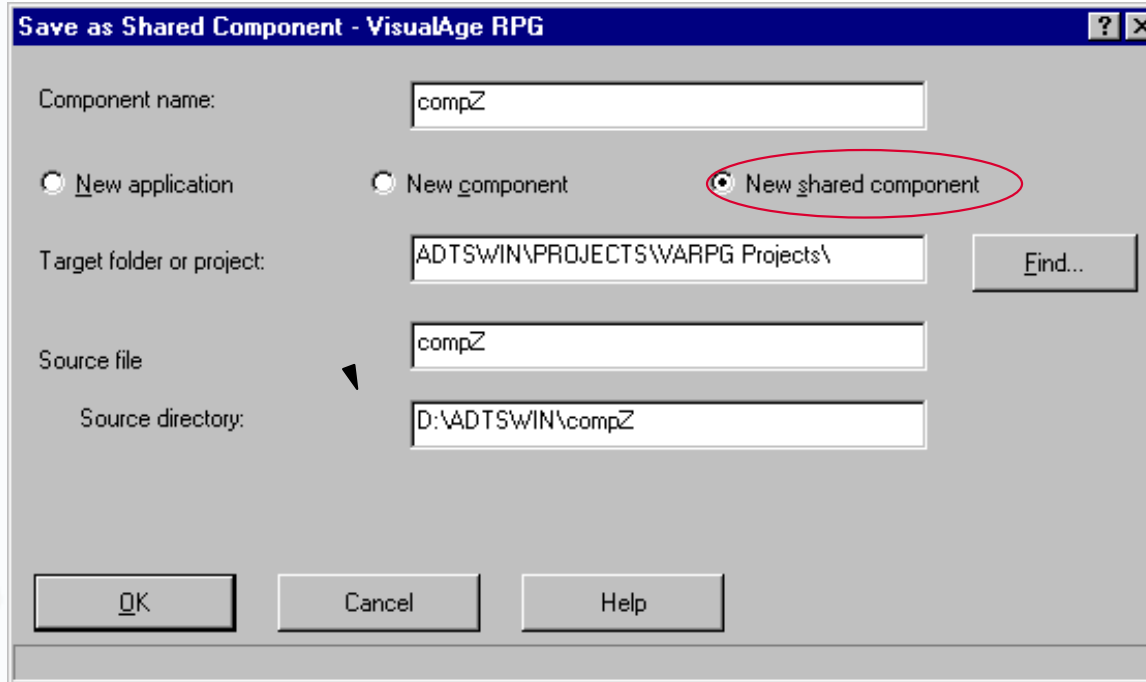
1. Specify project/application name
2. Specify location for 'project' file (can be in a folder or another project) suggested location application project
3. Specify subdirectory to contain source files
4. Specify file 'stub' name

→ VARPG packager will include component into application package





→ Shared by several applications



Save as Shared Component - VisualAge RPG

Component name: compZ

New application New_component New_shared component

Target folder or project: ADT\$WIN\PROJECTS\WARPG Projects\ Find...

Source file: compZ

Source directory: D:\ADT\$WIN\compZ

OK Cancel Help

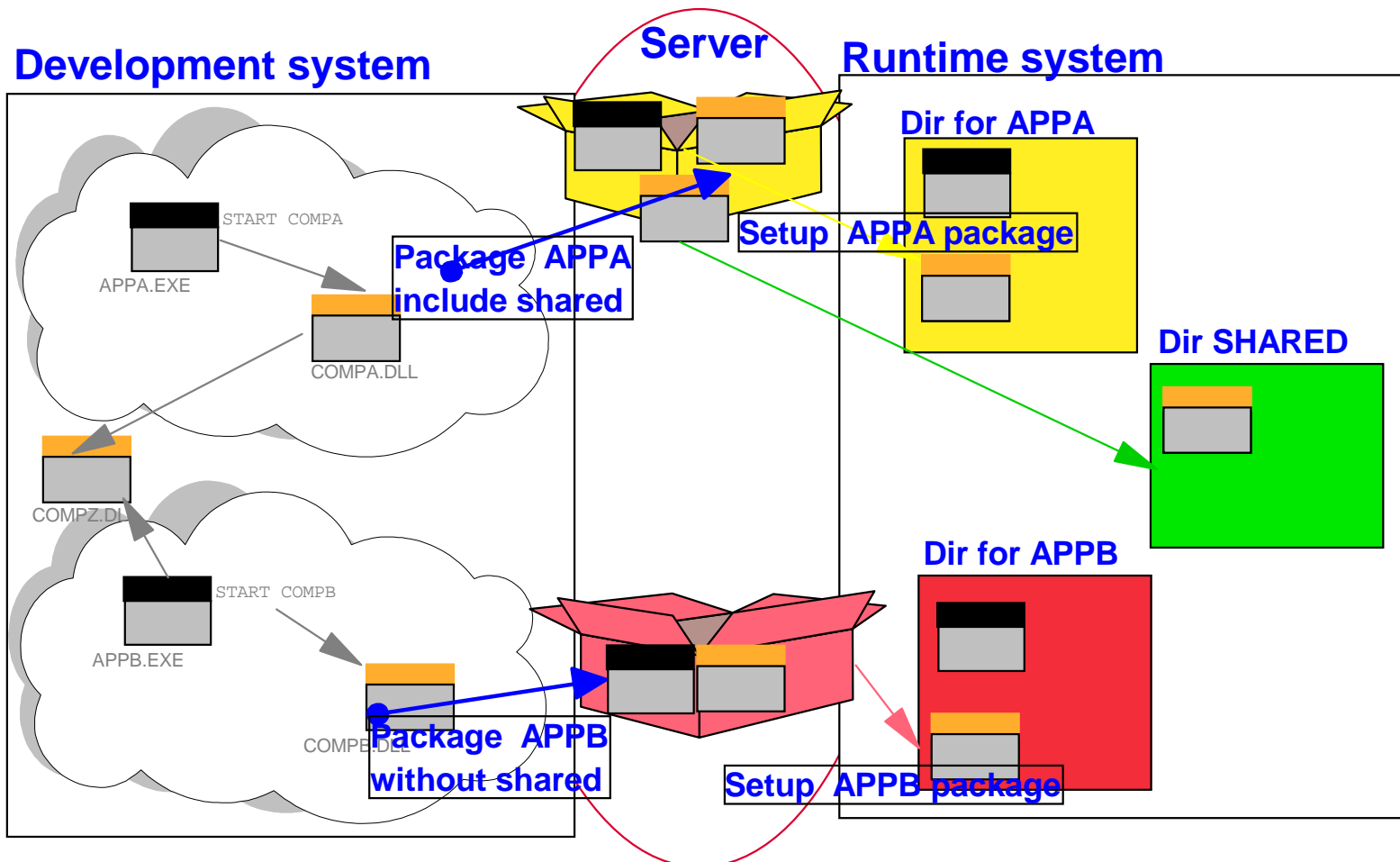
- When packaging application, choice to include shared component in application package
- If another application already uses shared component no need to repackage shared component
- At application install time shared components get installed in "SHARED" directory, so the DLL can be found when STARTING them



e-business

Application packaging

- Run VARPG packager for each application
- Specify server (IFS) and directory for package location
- Run Setup command to install application on user workstation
- Application shortcut in Start/Program gets created

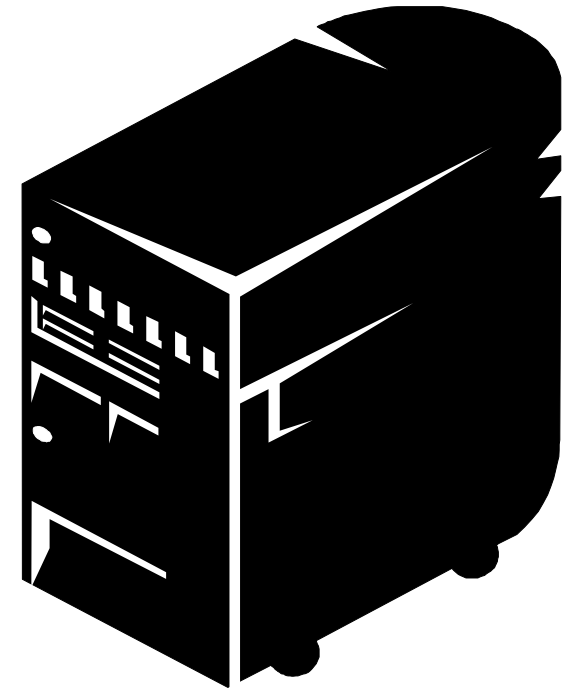
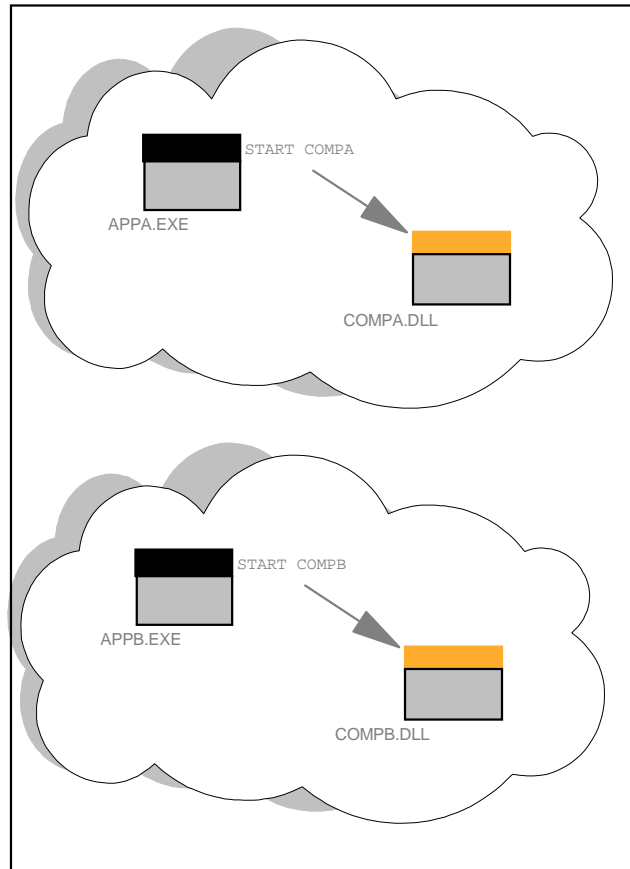




e-business

Application

- Stand alone program (.EXE)
- Can be invoked from command line or desktop icon
- Can be CALLED from other programs
- Consists of itself and zero or more components
- Specified during Save (As)
- Run in separate process



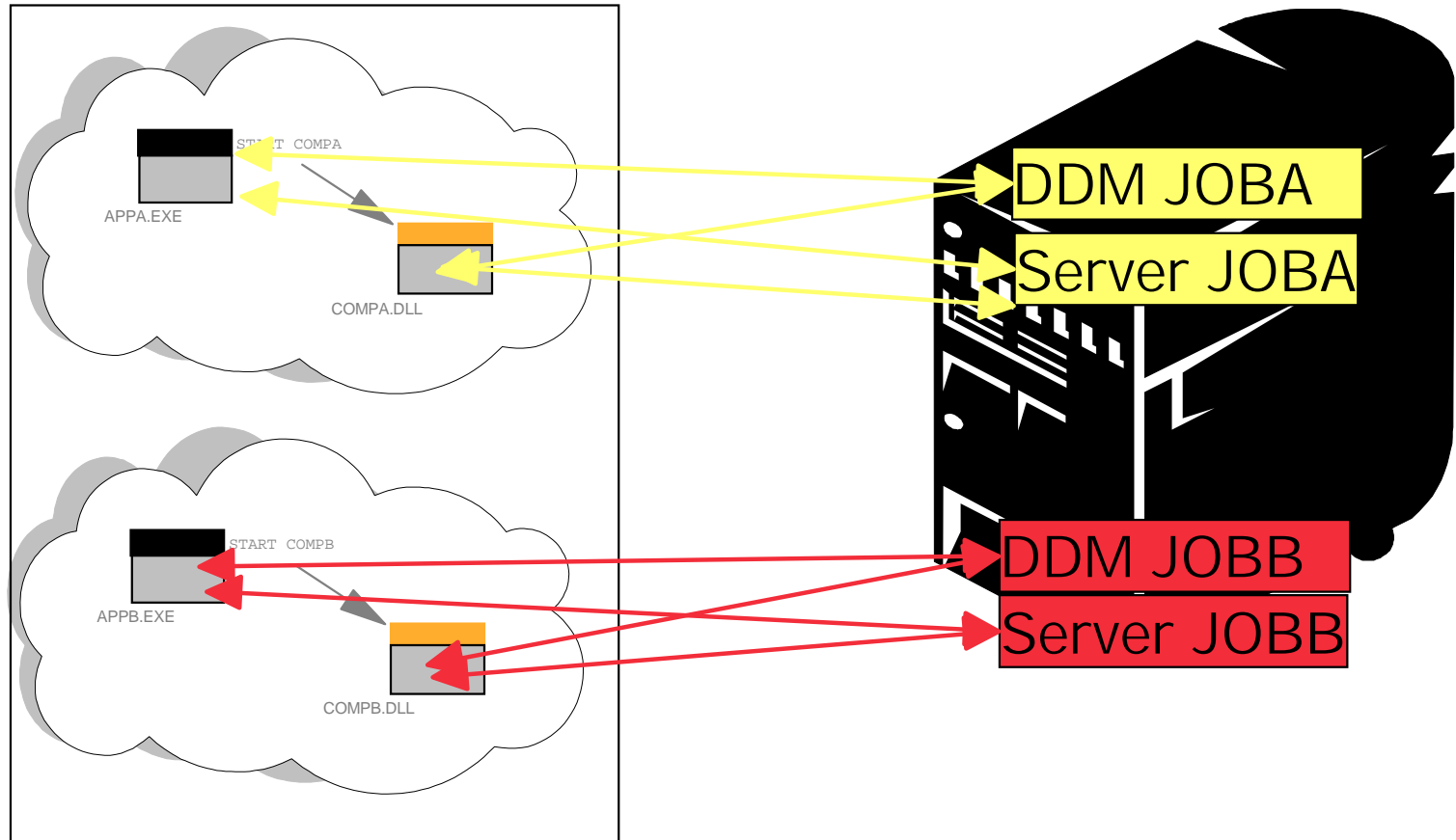
IBM



e-business

Application

- Stand alone program (.EXE)
- Can be invoked from command line or desktop icon
- Can be CALLED from other programs
- Consists of itself and zero or more components
- Specified during Save (As)
- Run in separate process

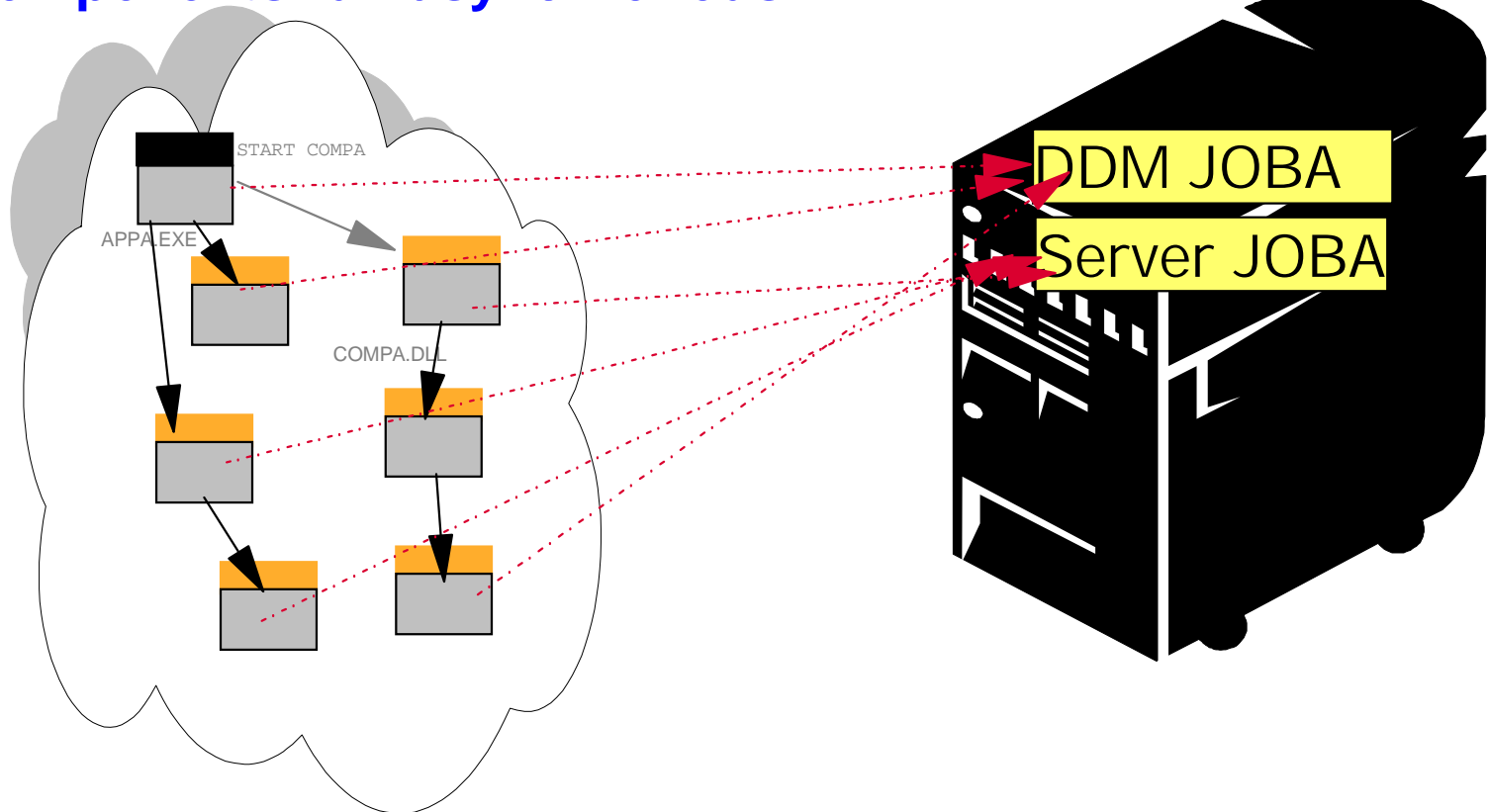




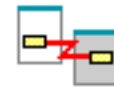
e-business

Application structure

- One main application (.EXE)
- **STARTS** several components (without Linkage keyword)
- Components can **START** other components
- Can be packaged with one packaging step
- All components use one set of jobs
- Component reference and parameters to communicate
- Components run asynchronous

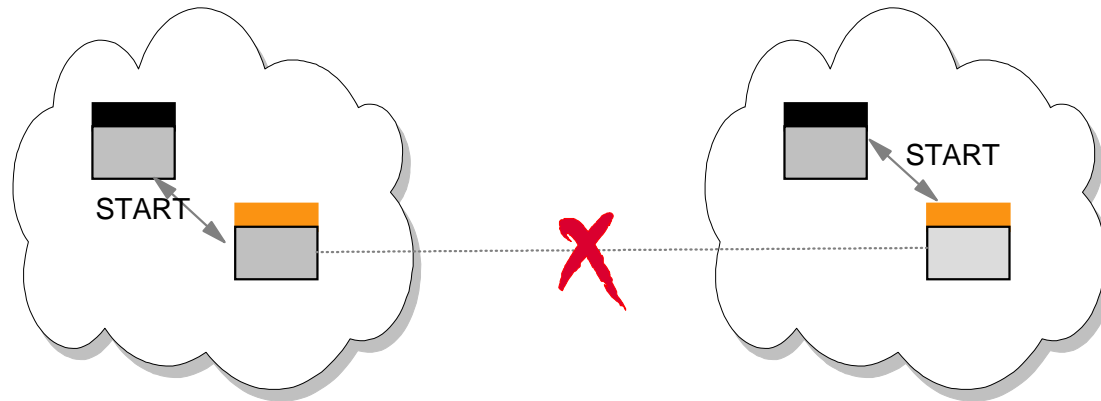


IBM



- Used to communicate between application and components or between components
- Provides two functions:
 - Set and Get Attributes of parts in other components
 - Monitor for Events in other components
- Referenced component must be STARTed
- Referenced component must be in same application (process)

APPA $\xrightarrow{\text{CALL}}$ APPB

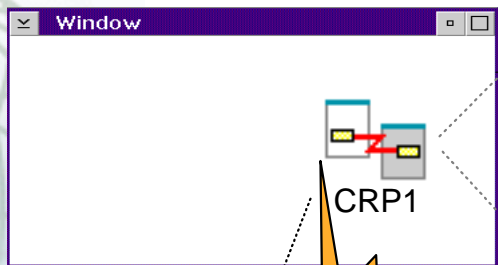




e-business

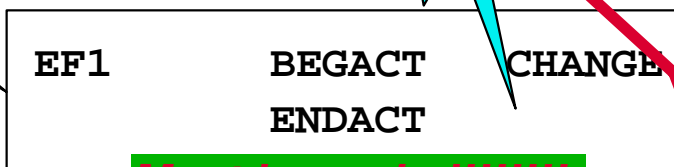
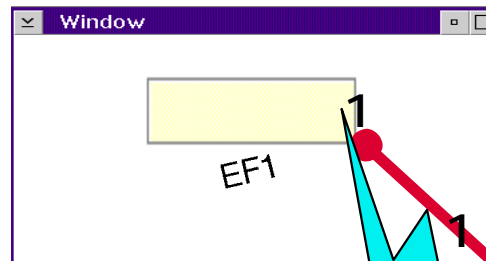
Component Reference Part - Example

COMPA



Component = COMPB
 Window = MAIN
 Part = EF1
 Event = CHANGE

COMPB



Must be coded!!!!!!

```
CRP1    BEGACT    NOTIFY
'WIN1'  SETATR   1
'Visible'
        ENDACT
```

```
'EF1'  SETATR  '1'  TEXT
```

- ➔ Component reference part gets set to watch CHANGE event
 - in COMPB/MAIN/EF1
- ➔ When EF1 changes, CHANGE event gets fired
- ➔ Change Event notifies CRP1
- ➔ NOTIFY event gets triggered



e-business

Referencing attributes and monitoring events

Specify referenced component in Settings

Notebook:

Reference component name

COMPB

Specify attribute to be referenced in Component Reference Part Settings Notebook:

Reference window name

MAIN

Reference part name

EF1

Reference part attribute name

TEXT

Specify event to monitor in Settings

Notebook:

Notification source window name

MAIN

Notification source part name

PB1

Notification source event name

PRESS

→ Attributes can also be set during runtime in VARPG logic

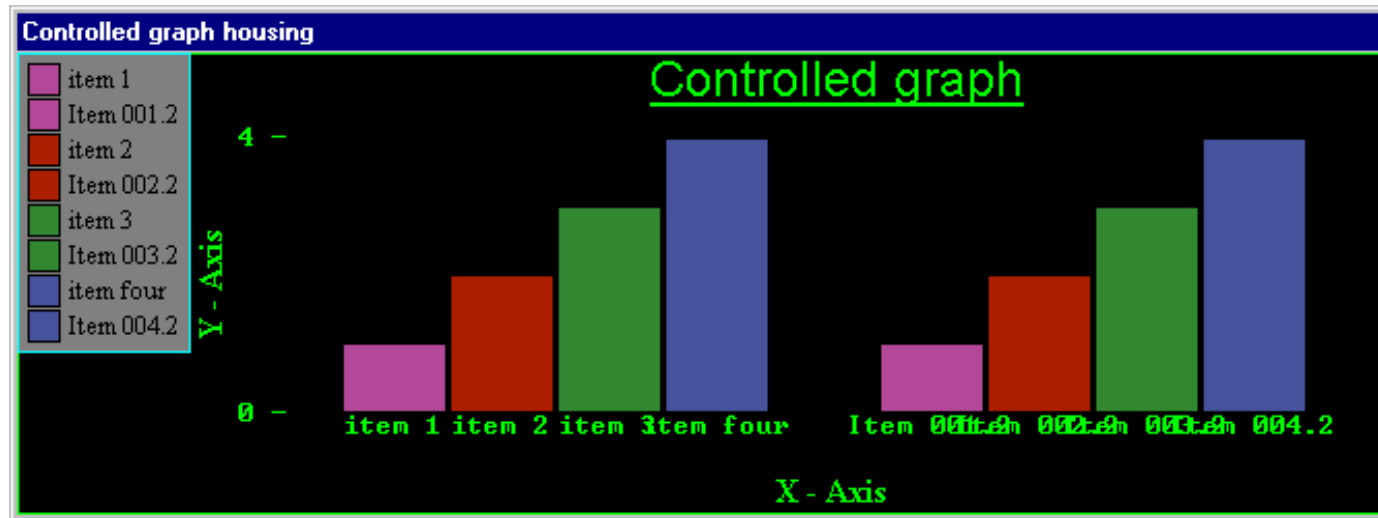
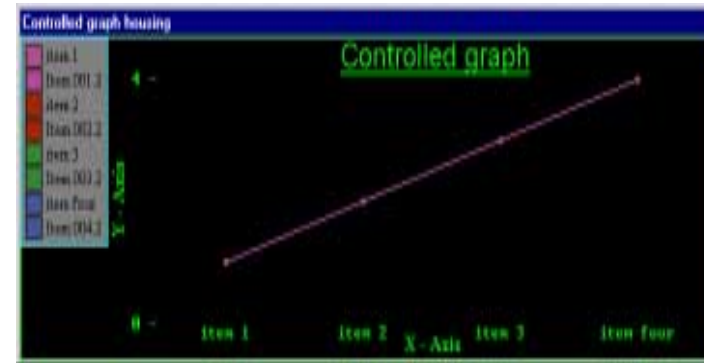
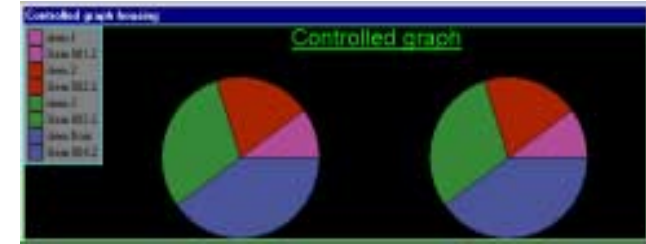


e-business

Graph part

→ Four different formats to choose from:

- Pie
- Bar
- Line
- Line and bar



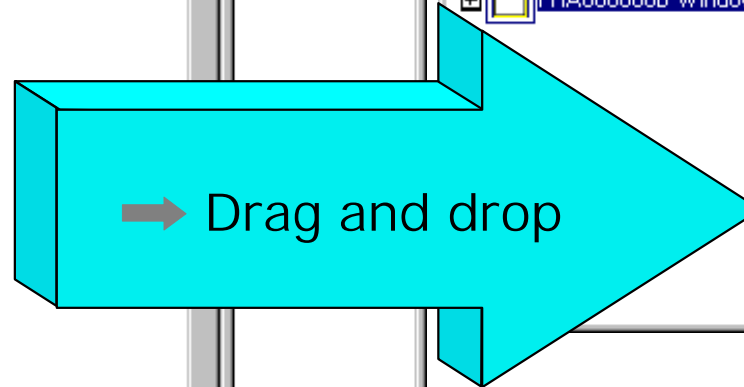
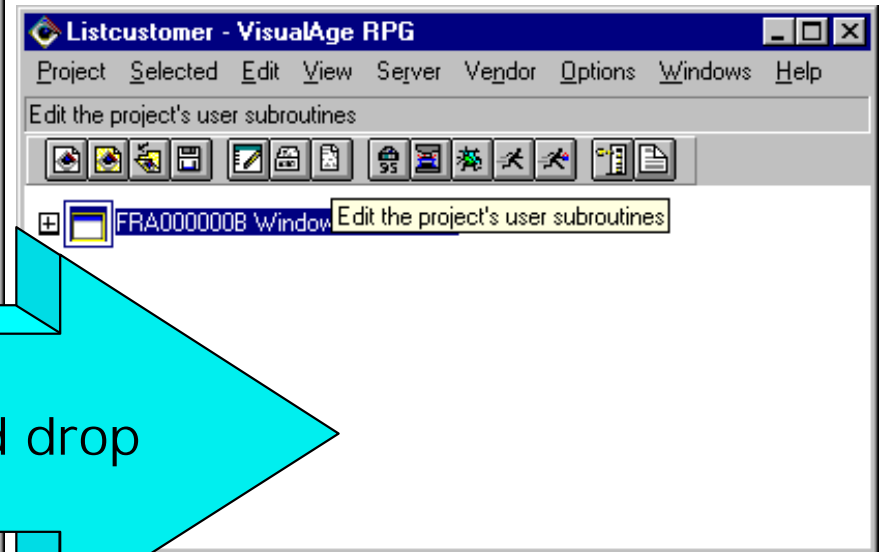
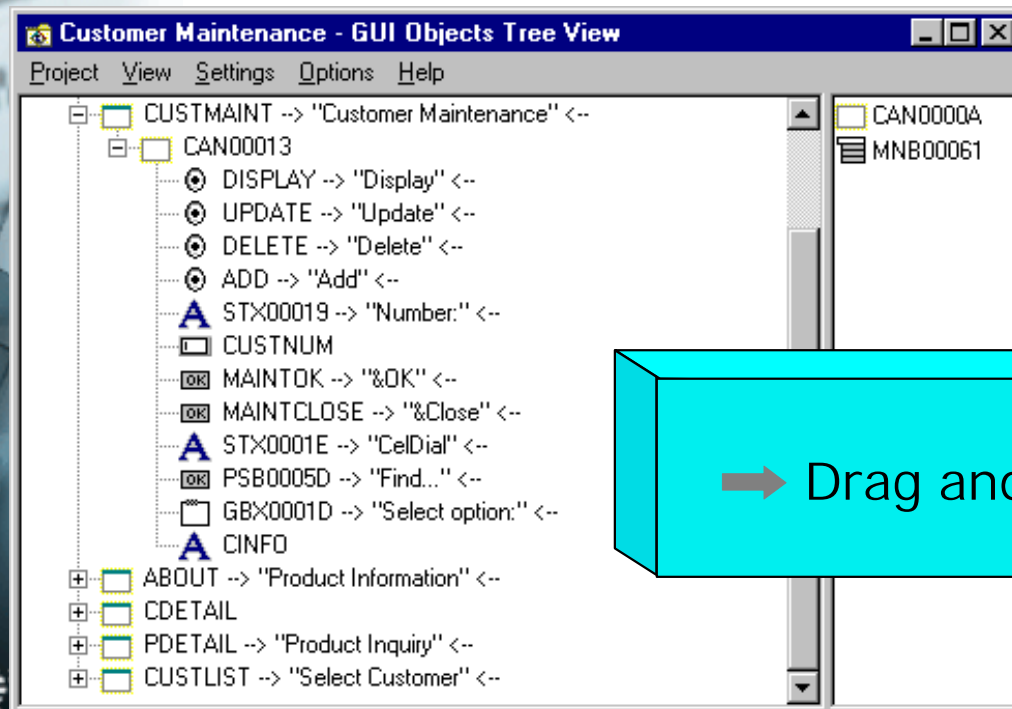


e-business

Browse merge

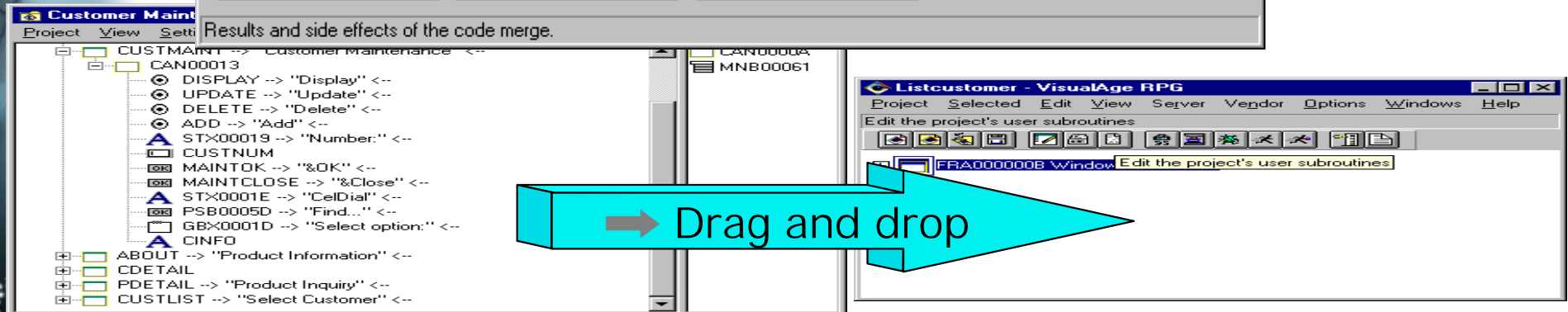
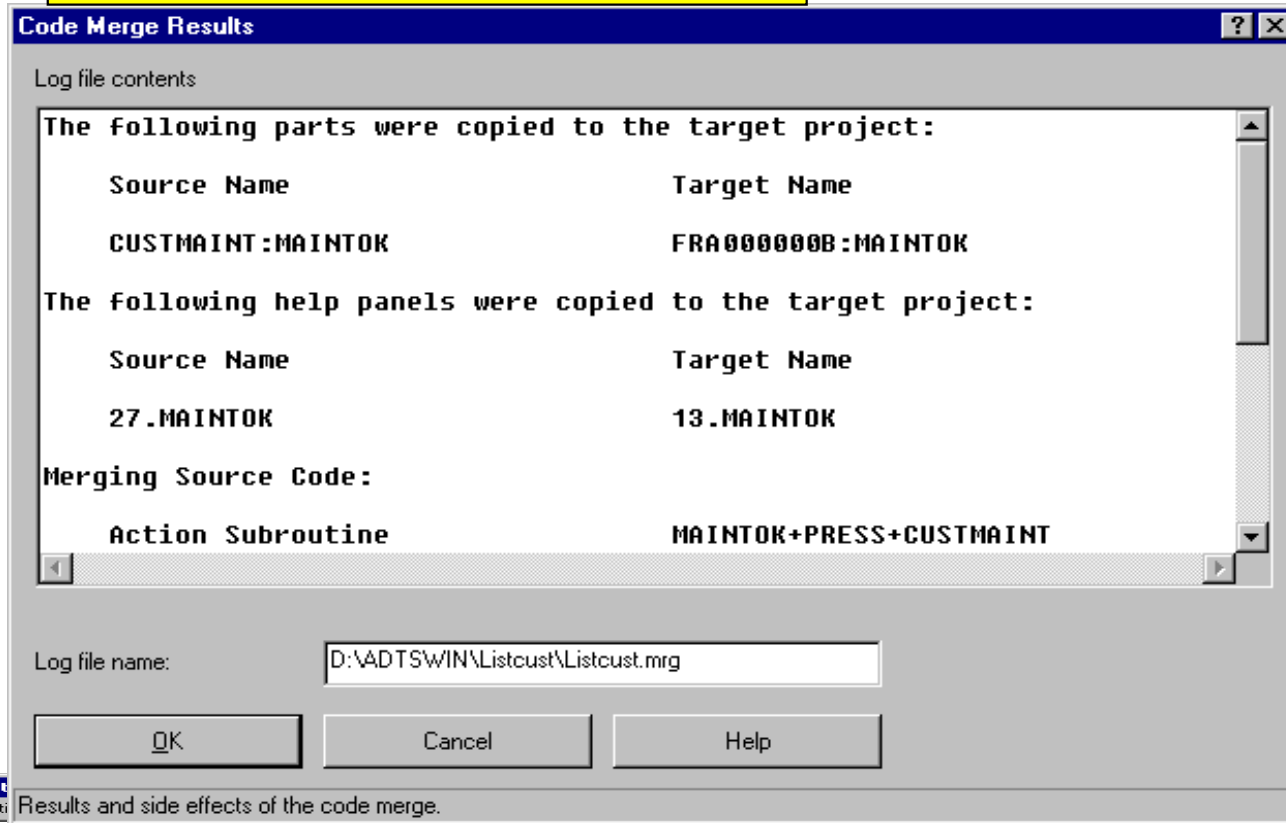
➔ Browse/merge

- Capability of browsing another project while in the GUI designer
- In addition copy parts from another application
 - ➔ Including all linked action subroutines, also includes user subroutines used in the action subroutines
 - ➔ In case of name conflicts, MERGE will rename the parts and partnames used in the action subroutines



Browse merge

- ➔ Merging a pushbutton
- ➔ Result screen





e-business

Some cool features

➔ Listbox and Combobox

- Fast capability for searching
 - ➔ New attributes **Case, Searchtype, Search**
- Key is now supported default separator is ";"
 - ➔ **Example:** %setatr('WIN':'LBXA':'additemend')='CA;California'

➔ Entry field

- New attributes:
 - ➔ **Autoselect** (selects all data in it when tabbed to)
 - ➔ **FieldExit**
 - ➔ **Cut, Copy, Paste, and Delete** (Clipboard support)
- Enhanced capability for validity checking
 - ➔ if error, field gets reversed image
 - ➔ error message can be "User text, or message id"
- M3 additions
 - ➔ Auto format/Auto validate ----- kicks in when user exits field





e-business

ODBC part

- ➔ Need ODBC SDK free from MS
- ➔ First specify database to connect to
- ➔ Create record set with SQL QUERY + ExecuteSQL
- ➔ Bind columns in table to RPG variable

- column
- address of variable
- datatype and length

```

➔ D First          S              20
➔ D fptr          S              *   INZ(%Addr(first))
➔ *
➔ C   'ODBC1'     Setatr         1   'Column'
➔ C   'ODBC1'     Setatr         20  'BufferLen'
➔ C   'ODBC1'     Setatr         fptr 'BufferPtr'
➔ C   'ODBC1'     Setatr         1   'BufferType'

```

- ➔ Now you can use SQL statements to access your data

```

➔ D SelAll        C              `SELECT * FROM "CUSTOMERS"'
➔ *
➔ C   'ODBC1'     Setatr         SelAll `SQLQuery'
➔ C   'ODBC1'     Setatr         1     `ExecutesQL'

```

➔ Attributes

- ConnectStr
- Connect
- Connected
- SQLQuery
- ExecuteSQL
- Column
- BufferPtr
- BufferLen
- BufferType
- Rows
- FetchNext
- FetchPrior
- CurrentRow
- UpdateRow
- DeleteRow
- Buffertype
- AllowChg

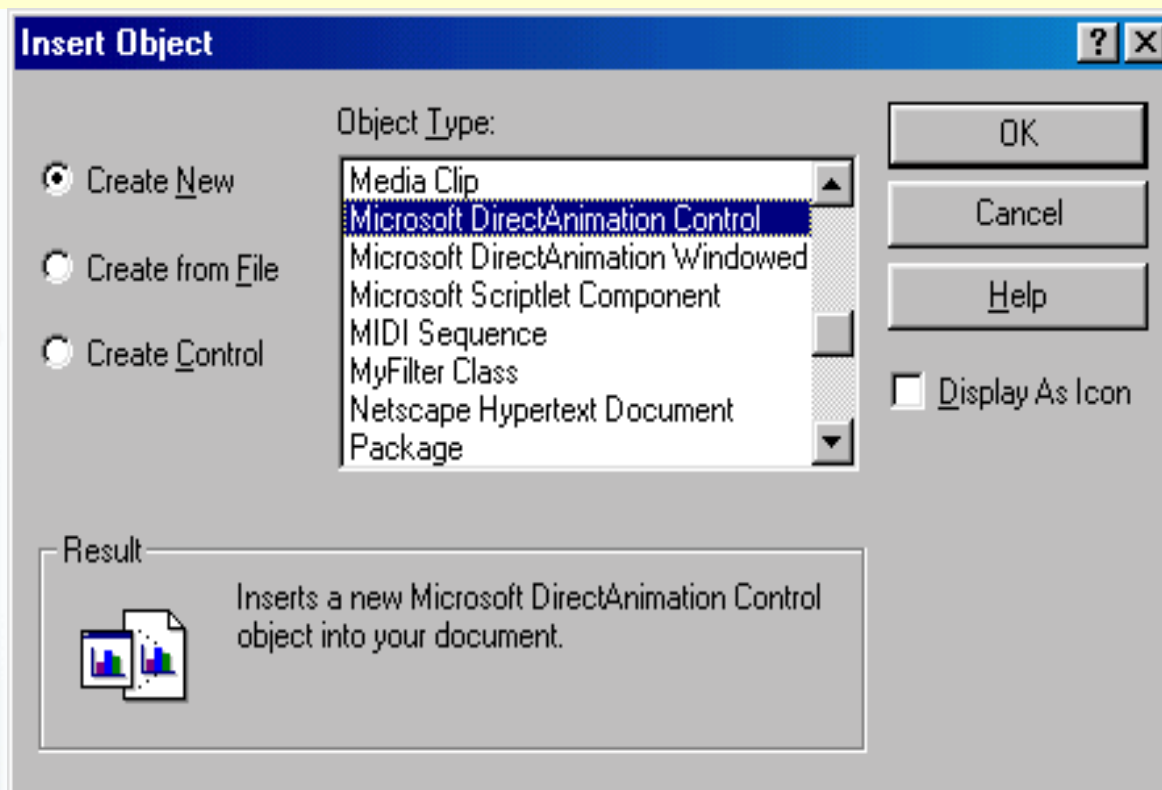




e-business

ActiveX part

→ Only support for ActiveX controls that have C++ interfaces



- Attributes
- Activate
 - AddEvent
 - DeActivate
 - HasPrpPage
 - OCXProp
 - OCXValue
 - RMVEvent
 - ShowProp
- Events
- Create
 - Destroy
 - OCXEvent





e-business

Status/Animation/Progress parts

➔ Status bar

- Attached to Window by attribute *StatusBar*
- Attributes: *SBLabel SBPosition SBStyle*

➔ Animation control

- Show video files in AVI format
- Attributes: *Filename Mode*

➔ Progress bar

- Attached to Window or separate part
 - ➔ Attached to window by attribute: *ProgresBar*
- Attributes: *PBRange PBSetPos PBStep PBStepSize*

The IBM logo, consisting of the letters 'IBM' in a bold, sans-serif font with horizontal stripes.

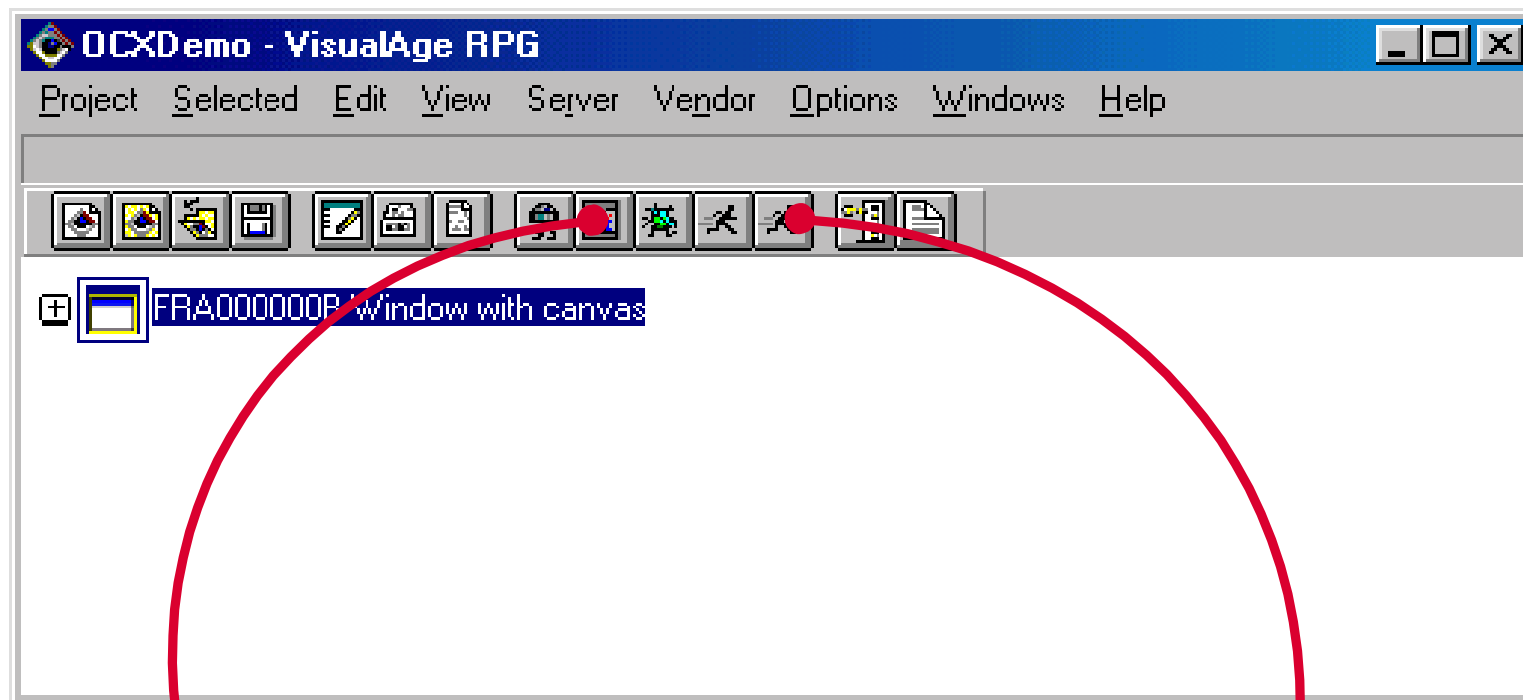


e-business

Sample code progress bar and animation part

```
C      PSBPRGANI      BEGACT      PRESS      WPROGRESS
c      read          'wprogress'
C      SHOWWIN      'WINBAR'
c      eval          counter=1
c* create the progress bar in window winbar
c      eval          %setatr('winbar':
c                      'winbar':'Progresbar')=1
c* set range
c      eval          %setatr('winbar':
c                      'winbar':'PBrange')=entend
c* tell animation part which movie to play and tell it to play it
c      eval          %setatr('winbar':
c                      'animation1':'filename')=amovie
c      eval          %setatr('winbar':
c                      'animation1':'mode')=1
c      dow          counter<entend
c      eval          counter=counter+1
c      eval          count=count+1
c      if          count>=10
c* advance progress bar by one step
c      eval          %setatr('winbar':
c                      'winbar':'PBstep')=1
c      eval          count=0
c      endif
c      write        'wprogress'
c      enddo
c      clswin      'winbar'
c      endact
```

Create Java source

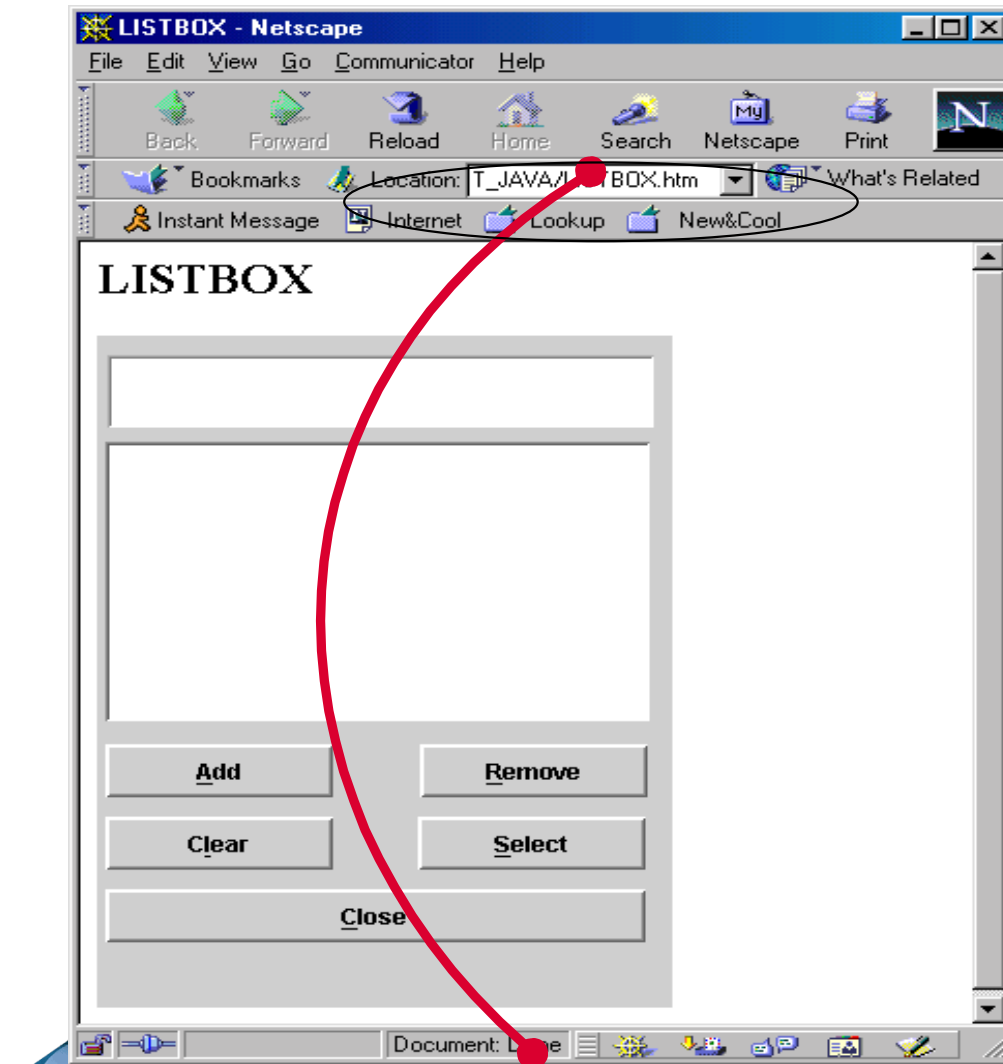


→ Create Java source

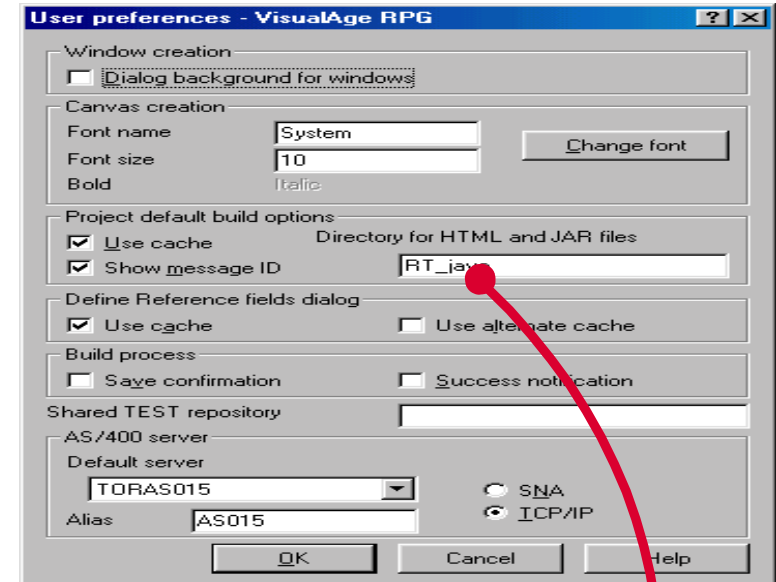
→ Run Java application



Run Java applet



→ Target HTML file in browser

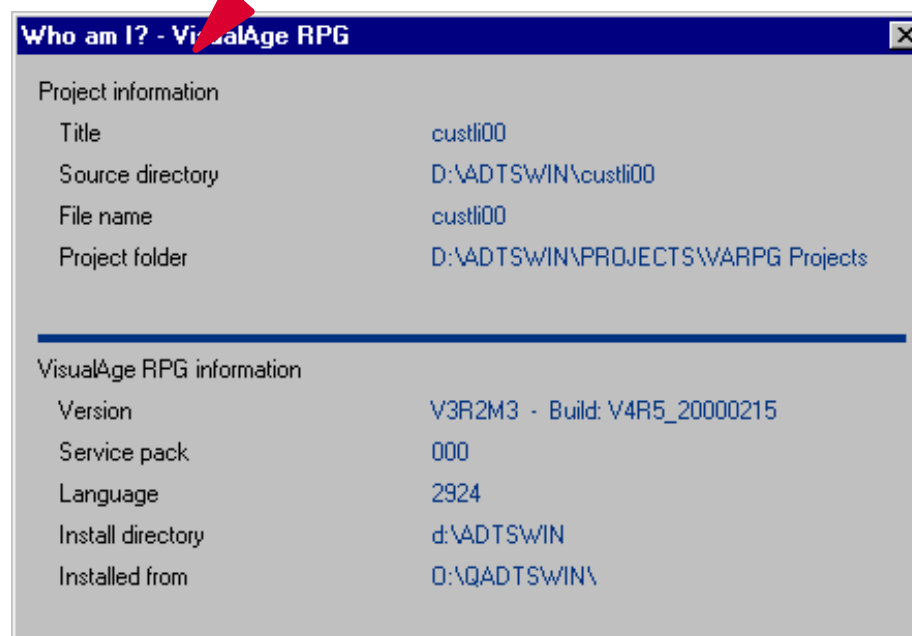
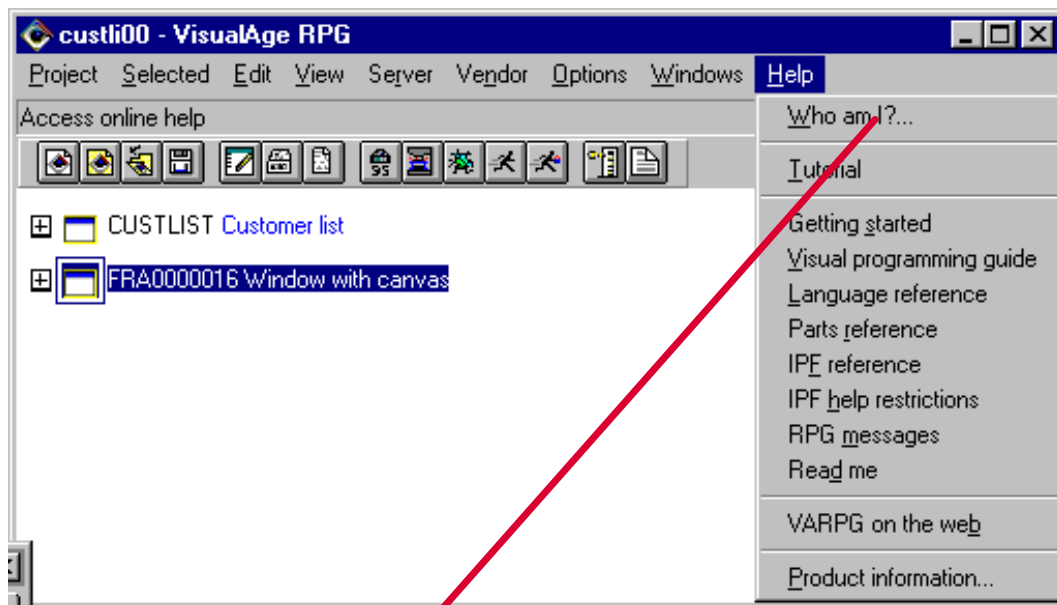


→ Specify HTML target



e-business

Who am I ? New in M3





e-business

Properties notebook page, Size

New in M3

→ All parts

- Property notebooks for got a new page

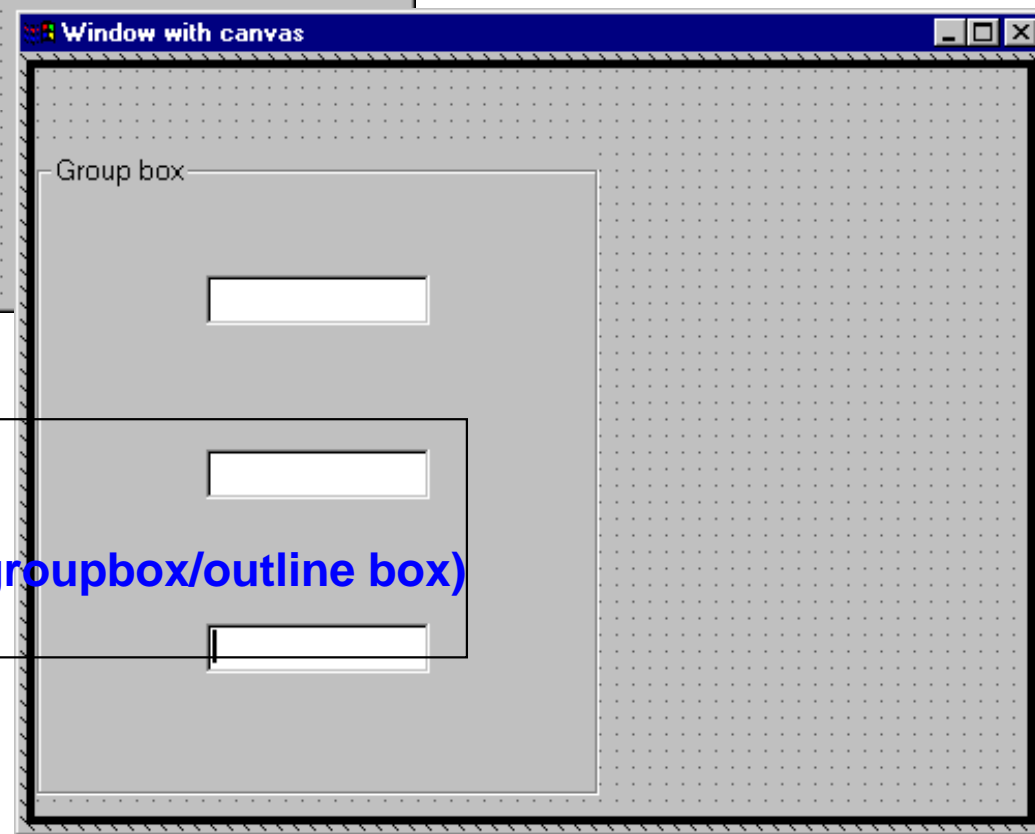
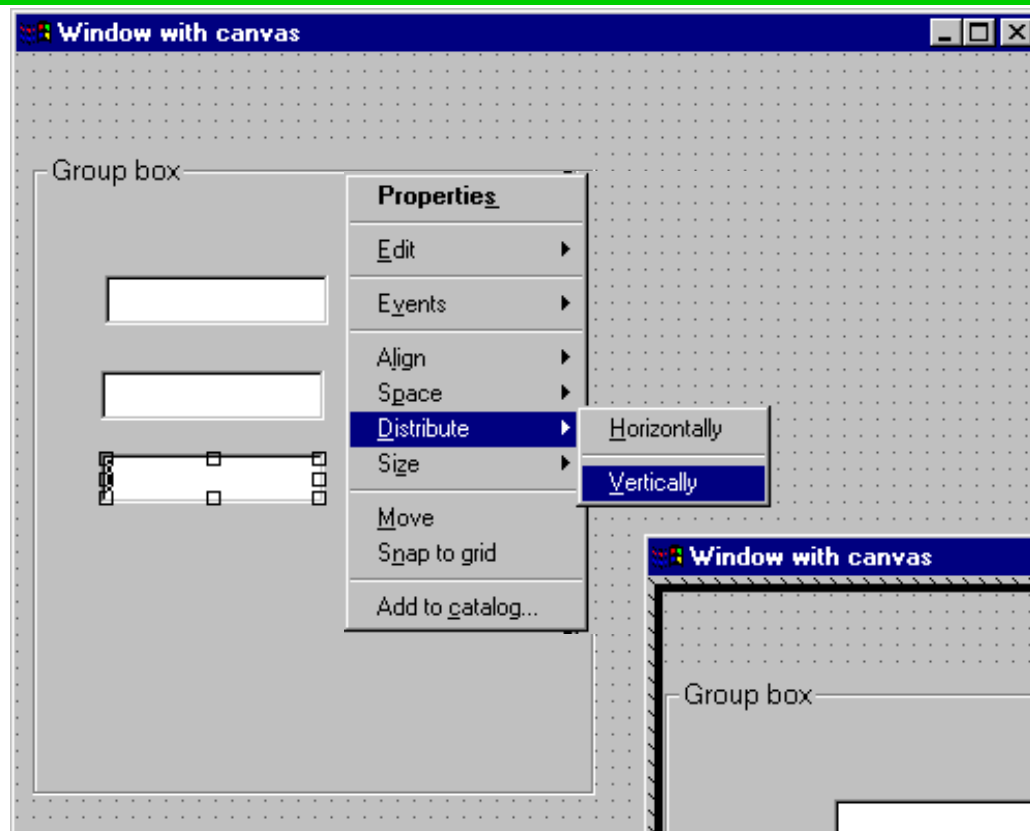
The screenshot shows a dialog box titled "Entry Field Part Properties" with a "Size" tab selected. The dialog has several tabs: Link, Font, Color, Description, Style, Data, Editing, Validation, General, Size, and Reference. The "Size" tab contains four input fields: "Left" with the value "88", "Top" with "104", "Width" with "110", and "Height" with "26". A "Reset page" button is located at the bottom right of the dialog. At the bottom of the dialog are three buttons: "Apply", "Reset", and "Help".



e-business

Spacing relative to an area

New in M3



- ➔ **Select parts**
- ➔ **If area is not window**
 - **Select bounding area (groupbox/outline box)**
- ➔ **Distribute**





e-business

Summary

- ✓ VARPG provides a complete C/S solution when combined with CODE/400 (PDM, SEU, SDA) on Windows
- ✓ Create your own sets of user defined parts
- ✓ Create your own tools with the *Plug in* interface
- ✓ VARPG provides access to Windows API's
- ✓ Use all new RPGIV features on the Client and on the Server
- ✓ Access data
 - ✓ in DB/400
 - ✓ in local ASCII files
 - ✓ or with embedded SQL in all DB/2 supported servers
 - ✓ in *ODBC* databases
- ✓ ActiveX support
- ✓ Java source generation
- ✓ Try to exploit the rich capabilities of VARPG
- ✓ Enjoy working with VisualAge RPG