43MN 402322

WebSphere Development Studio Client for iSeries

VisualAge RPG Introduction

VisualAge RPG: Fundamentals

Claus Weiss, IBM Toronto Laboratory
weiss@ca.ibm.com

www.ibm.com/software/awdtools/iSeries

Contacts

➔**My email:**

weiss@ca.ibm.com

➔**IBM System i5 AD tools homepage:**

http://www.ibm.com/software/awdtools/iSeries

➔ **VARPG Newsgroup:**

news://news.software.ibm.com/ibm.software.varpg

VARPG sessions at COMMON

➔VARPG, Hands on Labs, try it out:

Thursday *all day*, part of the WDSc open lab 51LC – 54LC

- **Intro to VARPG**
- **GUI subfile programming with VARPG (highly recommended)**

- **Advanced topics**
  - **ODBC programming with VARPG**

➔Get handouts for Roy Goldstein presentation:

VisualAge RPG what is it all about 21MO 430086

Agenda

- Overview
- The GUI designer
- The VARPG language
- Features
- Live demo of VARPG

WebSphere Development Studio for iSeries

- Eclipse based

Rational Web Developer (RWD)

For Web, Java, XML, ...

with iSeries extensions

- Remote System Explorer (system objects, RPG, COBOL ....)
- Webtools (web applications, web services)

WebFacing tool

- Browser interface for existing green screens

- Classic tools

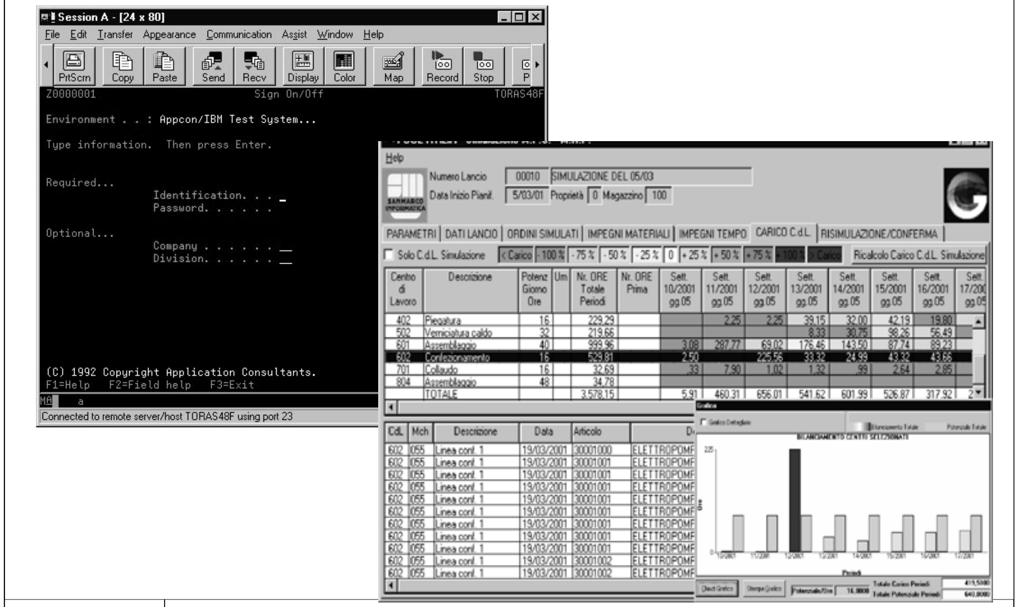
CODE

For RPG, COBOL, CL, C, C++, DDS

VisualAge RPG

For RPG logic with rich Graphical User Interface

Before and After



```
Z0000001                          Sign On/Off                          TORAS48F

Environment . . : Appcon/IBM Test System...

Type information.  Then press Enter.


Required...
                 Identification. . . _
                 Password. . . . . .

Optional...
                 Company . . . . . . __
                 Division. . . . . . __







(C) 1992 Copyright Application Consultants.
F1=Help   F2=Field help   F3=Exit
MA    a
```

Connected to remote server/host TORAS48F using port 23

Numero Lancio 00010 SIMULAZIONE DEL 05/03

Data Inizio Pianif. 5/03/01 Proprietà 0 Magazzino 100

☐ Solo C.d.L. Simulazione

Centro di Lavoro	Descrizione	Potenz Giorno Ore	Um	Nr. ORE Totale Periodi	Nr. ORE Prima	Sett. 10/2001 gg.05	Sett. 11/2001 gg.05	Sett. 12/2001 gg.05	Sett. 13/2001 gg.05	Sett. 14/2001 gg.05	Sett. 15/2001 gg.05	Sett. 16/2001 gg.05	Sett. 17/200 gg.05
402	Piegatura	16		229.29			2.25	2.25	39.15	32.00	42.19	19.80	
502	Verniciatura caldo	32		219.66					8.33	30.75	98.26	56.49	
601	Assemblaggio	40		999.96		3.08	287.77	69.02	176.46	143.50	87.74	89.23	
602	Confezionamento	16		529.81		2.50		225.56	33.32	24.99	43.32	43.66	
701	Collaudo	16		32.69		.33	7.90	1.02	1.32	.99	2.64	2.85	
804	Assemblaggio	48		34.78									
	TOTALE			3.578.15		5.91	460.31	656.01	541.62	601.99	526.87	317.92	2

CdL	Mch	Descrizione	Data	Articolo	D
602	055	Linea conf. 1	19/03/2001	30001000	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001002	ELETTROPOMF
602	055	Linea conf. 1	19/03/2001	30001002	ELETTROPOMF

☐ Grafico Dettagliato

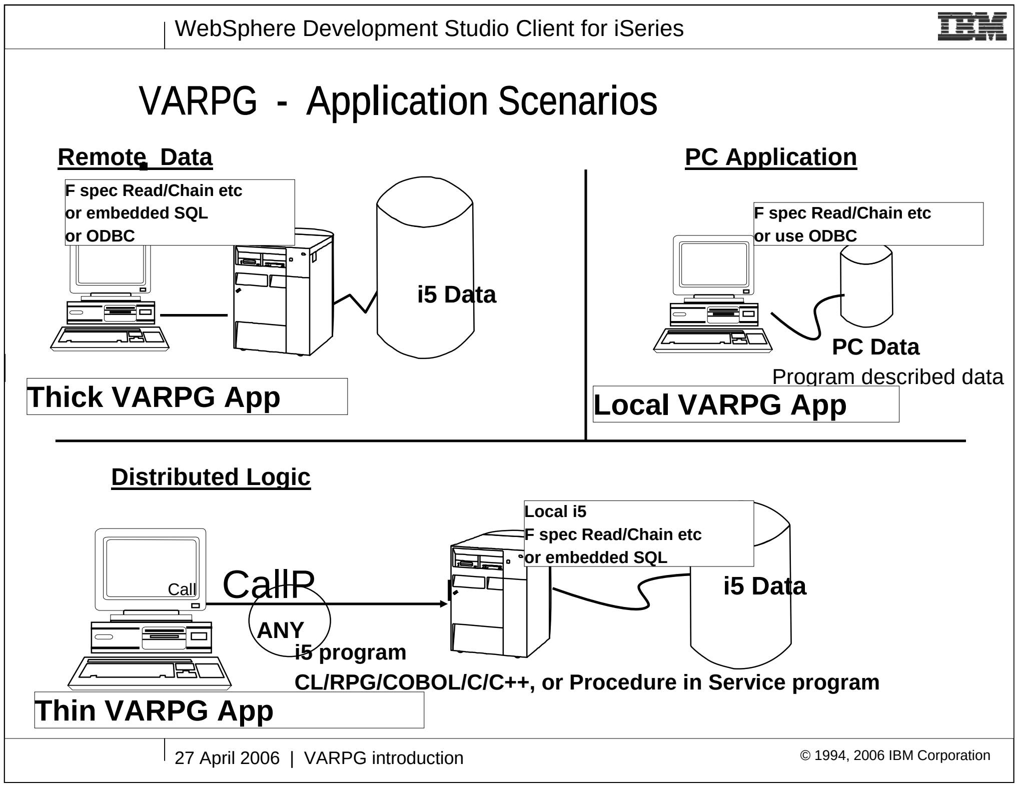
Potenziale/Ore 16.0000

Totale Carico Periodi 419,5000

Totale Potenziale Periodi 640,0000

VisualAge RPG Overview

- **You know RPG ? Yes you can do it**
- **Build client/server applications with RPG!**
  - **GUI builder to design multimedia interfaces**
    - *Easily create GUI screens*
    - *Drag and drop new technologies such as image, audio*
  - **Logic behind GUI is RPG**
    - *Familiar language*
    - *Familiar database*
  - **Integration with CODE editor**
  - ***VARPG project view in Eclipse workbench***
  - **No runtime license required for VARPG applications!**



WebSphere Development Studio Client for iSeries

VARPG - Application Scenarios

Remote Data

F spec Read/Chain etc
or embedded SQL
or ODBC

i5 Data

Thick VARPG App

PC Application

F spec Read/Chain etc
or use ODBC

PC Data

Program described data

Local VARPG App

Distributed Logic

Local i5
F spec Read/Chain etc
or embedded SQL

Call

CallP

ANY

i5 Data

i5 program
CL/RPG/COBOL/C/C++, or Procedure in Service program

Thin VARPG App

27 April 2006 | VARPG introduction

© 1994, 2006 IBM Corporation

This is intended to compare the various application environments supported by VARPG.

The primary target of the product is in the categories of Remote data and Distributed logic.

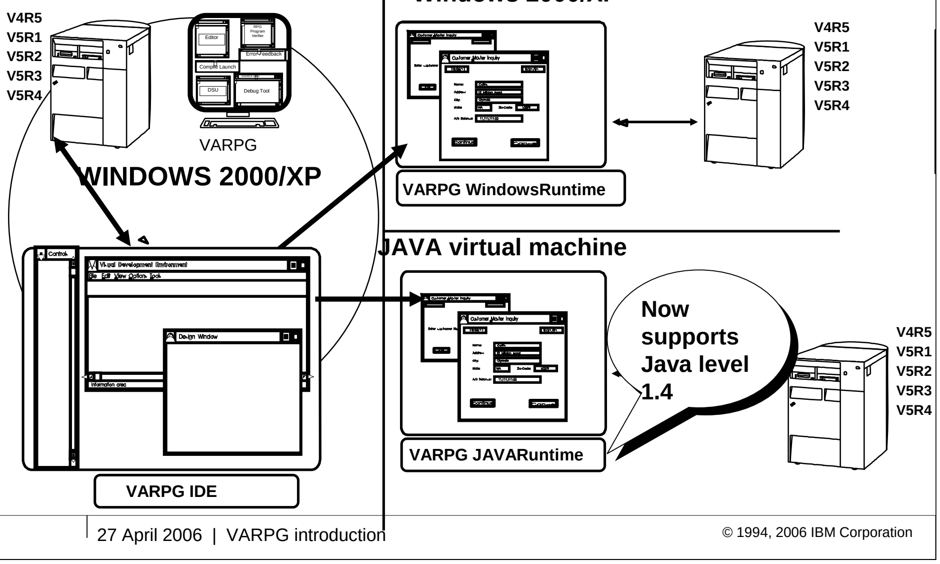
PC applications can be built today, but will be more directly supported in the V3R6 release when local file support (flat files and embedded SQL) is added to the product.

VARPG does not support the development of NPT (i.e. 5250) applications.

VARPG Development and Run Time

Development Environment

Run Time Environment



V4R5
V5R1
V5R2
V5R3
V5R4

VARPG

WINDOWS 2000/XP

VARPG IDE

Windows 2000/XP

VARPG WindowsRuntime

V4R5
V5R1
V5R2
V5R3
V5R4

JAVA virtual machine

Now supports Java level 1.4

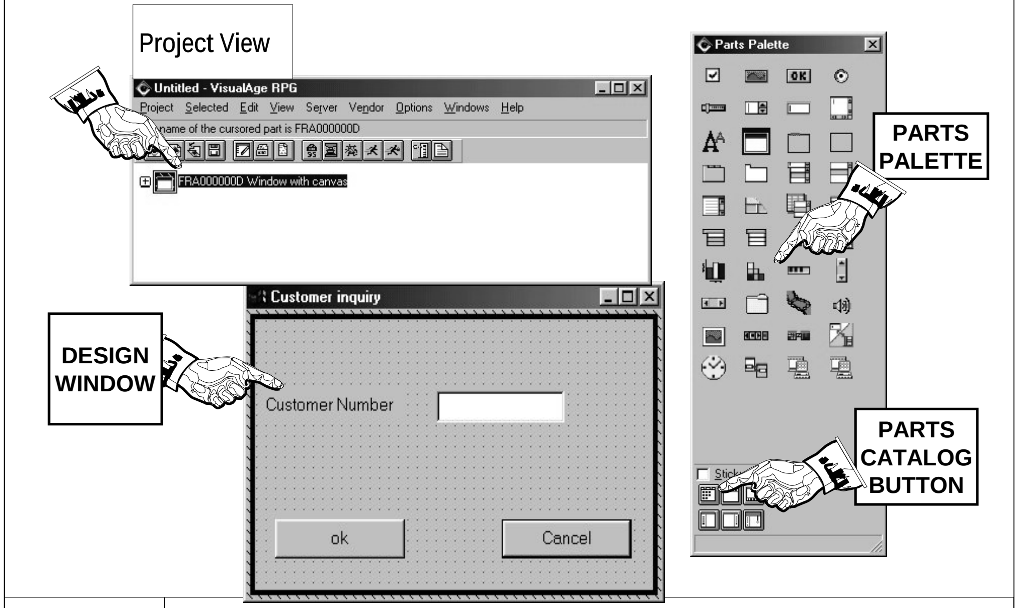
VARPG JAVARuntime

V4R5
V5R1
V5R2
V5R3
V5R4

Agenda

- Overview
- The GUI designer
- The VARPG language
- Features
- Live demo of VARPG

Using The GUI Designer



Project View

DESIGN WINDOW

PARTS PALETTE

PARTS CATALOG BUTTON

Untitled - VisualAge RPG
Project Selected Edit View Server Vendor Options Windows Help
name of the cursored part is FRA000000D
FRA000000D Window with canvas
Customer inquiry
Customer Number
ok
Cancel
Parts Palette
OK
Stick

This chart depicts VARPG GUI builder.

The Parts Palette on the left side contains basic graphical parts that can be dropped onto the development surface to create windows. The palette is customizable and you can create your own customized parts and place them on the palette, for use in other applications.

In this example, the window that is currently open for editing consists of the window part itself, to which has been added an entry field, a text field and two buttons. By double clicking on a part, the developer can define or modify its attributes (see the next page for an example of this). The attributes of a part may also be modified programmatically with the RPG code that defines the application logic. We'll see examples of this later in the presentation.

RPG logic may be associated with events that occur to the parts on the GUI screen, such as pressing a button or passing a mouse over a part.

Parts and Events

Part:

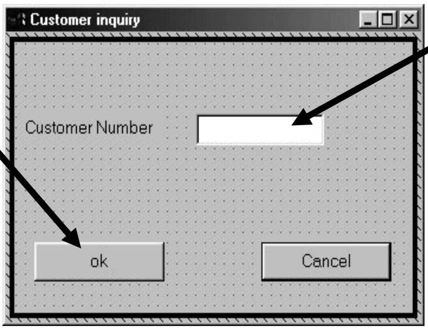
Pushbutton

Attributes:

BackColor
Border
FontSize
....

Events:

Press
Create
Destroy
MouseMove
....



Part:

Entry Field

Attributes:

Backcolor
Text
DataType
....

Events:

Change
Drop
MouseMove
DblClick
....

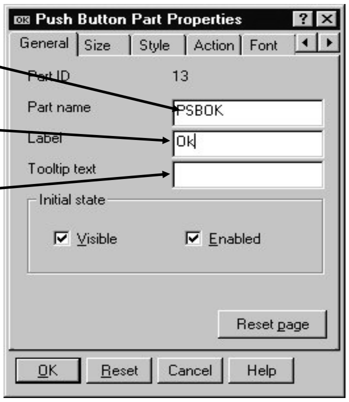
The attributes of a 5250 screen "part" are limited to things such as color, blink/steady, input capability, etc. VARPG parts have a far greater number of attributes.

The initial settings of these attributes are determined by the entries made in the parts notebook, see the next foil for an example of a notebook page. They can also be controlled programmatically. We'll see an example of this later.

Parts are also associated with "Events". For example a button may be PUSHed, an Entry field may be CHANGEd, a Subfile may have a record SELECTed, and so on. These events are associated with action subroutines which contain the logic required to perform the requested task. Again we'll be taking a look at an example of this later in the presentation.

Part Properties (attributes) Notebook

- Part Name
  - ►Used as "Field" name
- Label
  - ►Text appears on part
- Tooltip
  - ►Help info for enduser
- Also specifies :
  - ►Size
  - ►Style
  - ►Font
  - ►Colors
  - ►Description



Part ID 13
Part name PSBOK
Label Ok
Tooltip text
Initial state
☑ Visible ☑ Enabled

Initial state attributes

- •Visible
- •Enabled

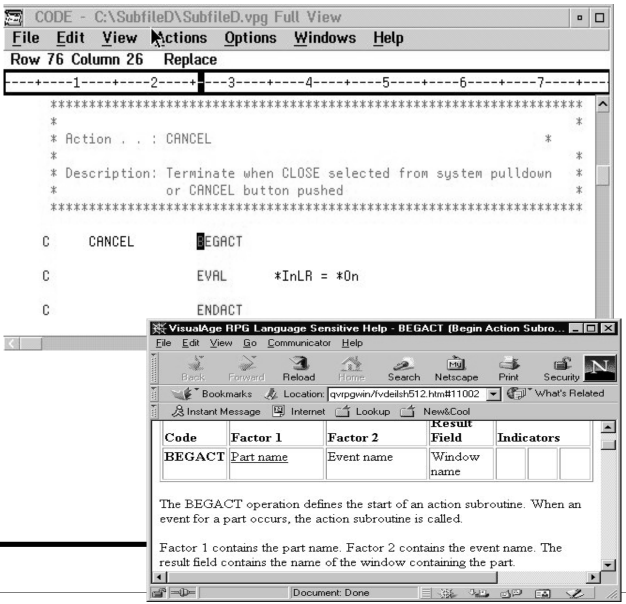
This is an example of a notebook page. This first page names the part and defines any initial text that is to appear on/in the part. In this case, a button, it also determines if the part is to be visible when it's window is first shown and whether the part is enabled (i.e. can be pushed).

There's even room for documentation!

Other pages in the notebook define the part's color (foreground and background), it's font, any help text associated with the part, etc. etc. The notebook also controls optional features of a part, for example whether an image part is to include a magnification panel.

The CODE Editor

- **Programmable**
  - ►**Extensible**
- **Color coding**
- **Context Sensitive**
  - ►**Prompting**
  - ►**Help**
    - **–It's the manual!**
- **Shared with CODE**
- **SEU profile**


CODE - C:\SubfileD\SubfileD.vpg Full View
File Edit View Actions Options Windows Help
Row 76 Column 26 Replace
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+---

```
**********************************************************************
*                                                                    *
* Action . . : CANCEL                                              *
*                                                                    *
* Description: Terminate when CLOSE selected from system pulldown   *
*              or CANCEL button pushed                               *
**********************************************************************

C     CANCEL        BEGACT

C                   EVAL      *InLR = *On

C                   ENDACT
```

VisualAge RPG Language Sensitive Help - BEGACT (Begin Action Subro...
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
Bookmarks Location: qvrpgwin/fvdeilsh512.htm#11002 What's Related
Instant Message Internet Lookup New&Cool

Code	Factor 1	Factor 2	Result Field	Indicators		
BEGACT	Part name	Event name	Window name			

The BEGACT operation defines the start of an action subroutine. When an event for a part occurs, the action subroutine is called.

Factor 1 contains the part name. Factor 2 contains the event name. The result field contains the name of the window containing the part.

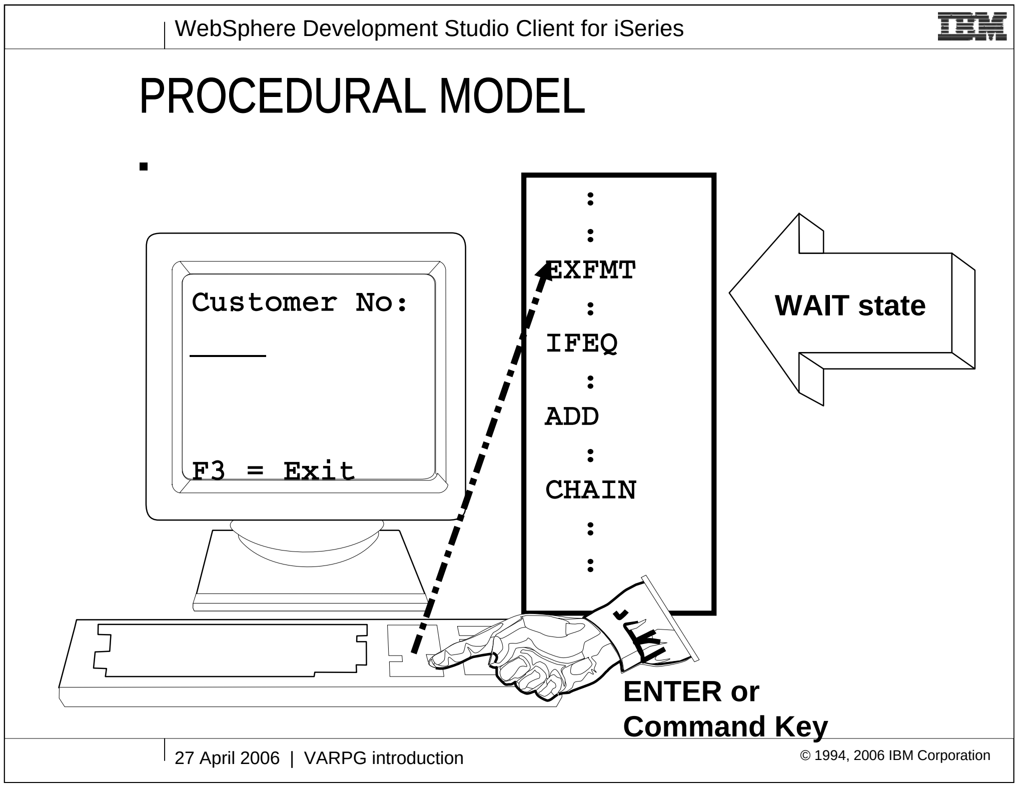
The LPEX editor is fully programmable. If you don't like the defaults, changing them is a simple matter. It is also fully extensible through macros or add-in components.

Third party add-ons have started to appear (a DDS parser for one) and more are expected. The editor is also supported through the user run CODE400-L listserver on the Internet.

The same editor technology is used in VARPG, VisualAge C++, CODE/400, and the new Visual COBOL product.

Agenda

- Overview
- The GUI designer
- The VARPG language
- Features
- Live demo of VARPG



WebSphere Development Studio Client for iSeries

PROCEDURAL MODEL

▪

```
   :
   :
EXFMT
   :
IFEQ
   :
ADD
   :
CHAIN
   :
   :
```

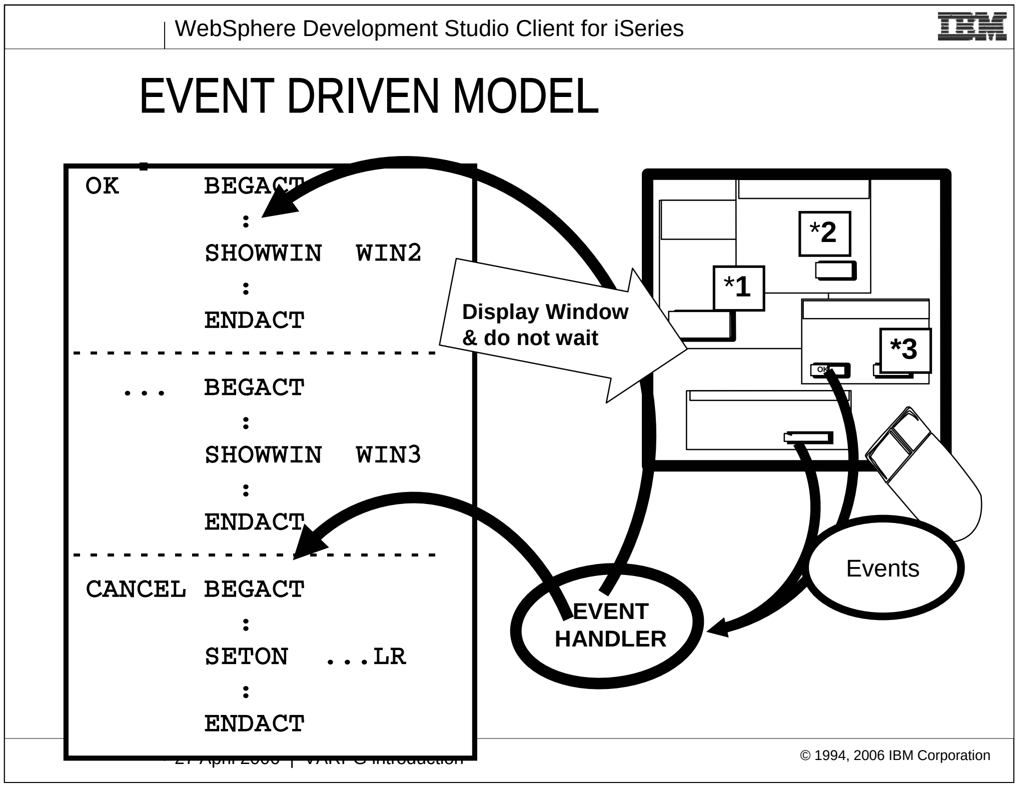
27 April 2006 | VARPG introduction

It's important to understand that the way we program for a GUI application is different than the method we typically use when programming traditional "Green screen" 5250 style applications.

The 5250 style applications use what we'll refer to here as the "Procedural" model. That is to say that we prepare a screen and write it to the display with (in RPG) an EXFMT. Our program then effectively "goes to sleep" and waits for some form of input from the user. This may be the pressing of an F key, Roll Up/Down, Enter, etc. etc.

The significant thing to note is that once the program has been "awakened", it's the programmer's responsibility to determine just what happened. This often takes the form of a series of IF statements which trigger the appropriate logic to match the user's request.

GUI applications tend to use the "Event Driven" model. The VARPG implementation of this is described on the next foil.



WebSphere Development Studio Client for iSeries

EVENT DRIVEN MODEL

```
OK      BEGACT
           :
        SHOWWIN  WIN2
           :
        ENDACT
---------------------
  ...   BEGACT
           :
        SHOWWIN  WIN3
           :
        ENDACT
---------------------
CANCEL  BEGACT
           :
        SETON  ...LR
           :
        ENDACT
```

27 April 2006 | VARPG Introduction

© 1994, 2006 IBM Corporation

In the "Event Driven" model, multiple "screens" (i.e. windows) can be displayed at any given time and the user is free to interact with any of them.

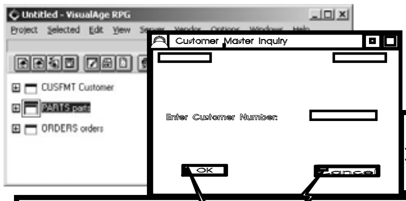
This would very quickly become completely unmanageable if we had to use the procedural approach. Instead we associate logic with the actual fields, buttons, etc. that the user interacts with.

In the example shown, we'll assume that windows *1 and *2 are displayed when the application starts up. The user is free to interact with either of them. If the user clicks on a button in one of the windows, the "event" is monitored by the Event Handler. This identifies the event and routes it to the appropriate action routine.

In our example, the user first clicks on an "OK?" button and the first action subroutine (OK BEGACT) is triggered. Among other things this results in the display of another window (SHOWWIN WIN3).

Had the user clicked on the "Cancel" button, the Event Handler would have activated the third action routine (CANCEL BEGACT).

Adding RPG Code



Global Area

```
FCUSTMSTL  IF E K          Disk   REMOTE
```

```
C      CANCEL          BEGACT    PRESS          CUSFMT
 *Set on LR to end the program
C                      EVAL      *INLR = *ON
C                      ENDACT
```

```
C     OK               BEGACT     Press          CUSFMT
 * Locate customer in file
C     CUST             CHAIN(E)  CUSREC
C                      IF        %found and not %error
                       :
C                      ELSE
 * Customer was found, display update panel.
C                      SHOWWIN   'CUSFLDS'
C                      END
C                      ENDACT
```

This chart illustrates an example of the RPG logic that might be applied to a simple GUI screen.

The logic associated with the Press event of the OK button will evoke the action subroutine (note the BEGACT - BEGin ACTion - operation in the code.) In this example, the data (CUST) entered in the entry field is used to CHAIN to the Customer master file on the AS/400. If the customer is found, a new window is displayed with the customer information. If the customer is not found then we could display a warning box similar to the one shown in a later example.

When the CANCEL button is pressed, the application is ended (by setting on LR).

Note also that there is a global area where File specs and data definition specs are placed.

The VARPG language is based on RPG IV, so if parts of this code do not look familiar, it may be because you are not familiar with the new features of RPG IV, such as free form expressions.

There are also language extensions specific to VARPG, such as the SHOWWIN operation code.

Working with GUI attributes

Free form or EVAL operation code

Use qualified names

AttrValue = Window.Part.Attribute

Window.Part.Attribute = AttrValue

Get text attribute from part custn into variable CUSTNO

```
  C       Eval     custno = window.custn.text
```

Set text attribute of part custn to value in variable CUSTNO

```
  C       Eval     window.custn.text = custno
```

Free form sample:
Custno = Win1.Ent1.Text
Win1.Ent1.BackColor = *Red

E
V
A
L

No indicator usage for attributes

Dynamic Attribute Operations

Built-ins

%GETATR & %SETATR

AttrValue = %GETATR(Window:Part:Attribute)

%SETATR(Window:Part:Attribute) = AttrValue

Get text attribute from part (name contained in variable partname) into variable CUSTNO

```
Eval     custno = %GETATR(‘w1’:partname:’text’)
```

Set text attribute of part (name contained in variable partname) to value in variable CUSTNO

```
Eval     %SETATR(‘w1’:partname:’text’ = custno
```

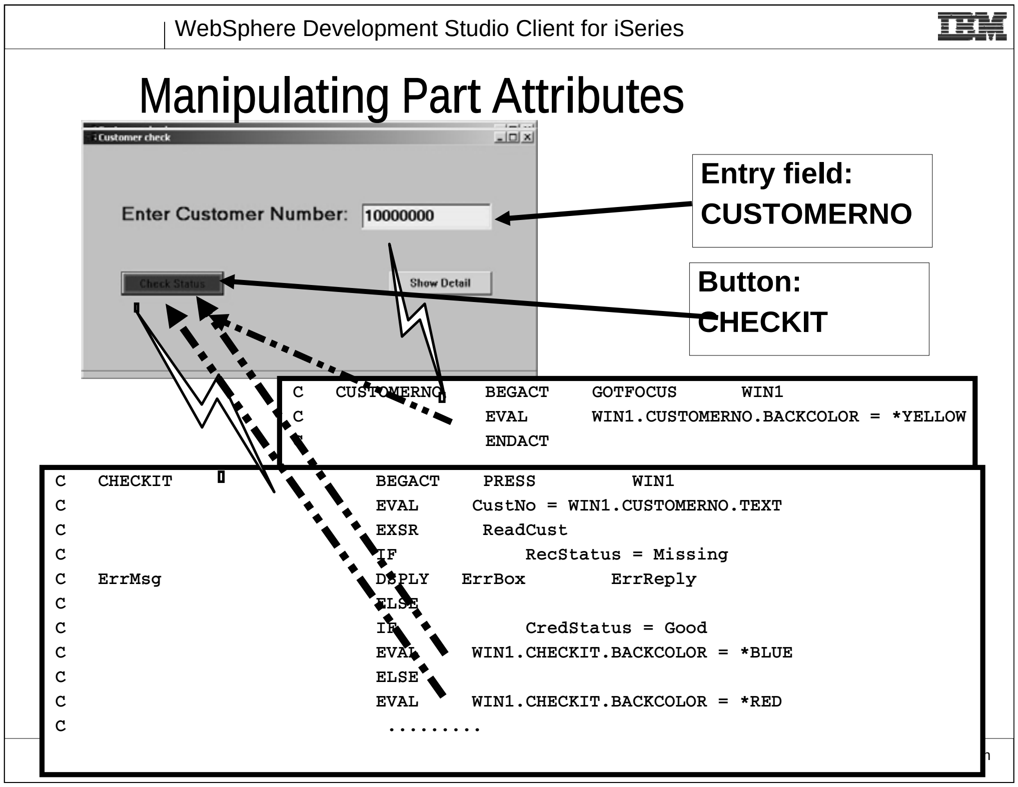
Attribute Operations (old style)

Fixed form

GETATR & SETATR

Only apply to the default (active) window

Factor 1	Op-code	Factor2	Result
Part	GETATR	Attribute	AttrValue
Part	SETATR	AttrValue	Attribute


WebSphere Development Studio Client for iSeries

Manipulating Part Attributes

Enter Customer Number: 10000000

Entry field: CUSTOMERNO

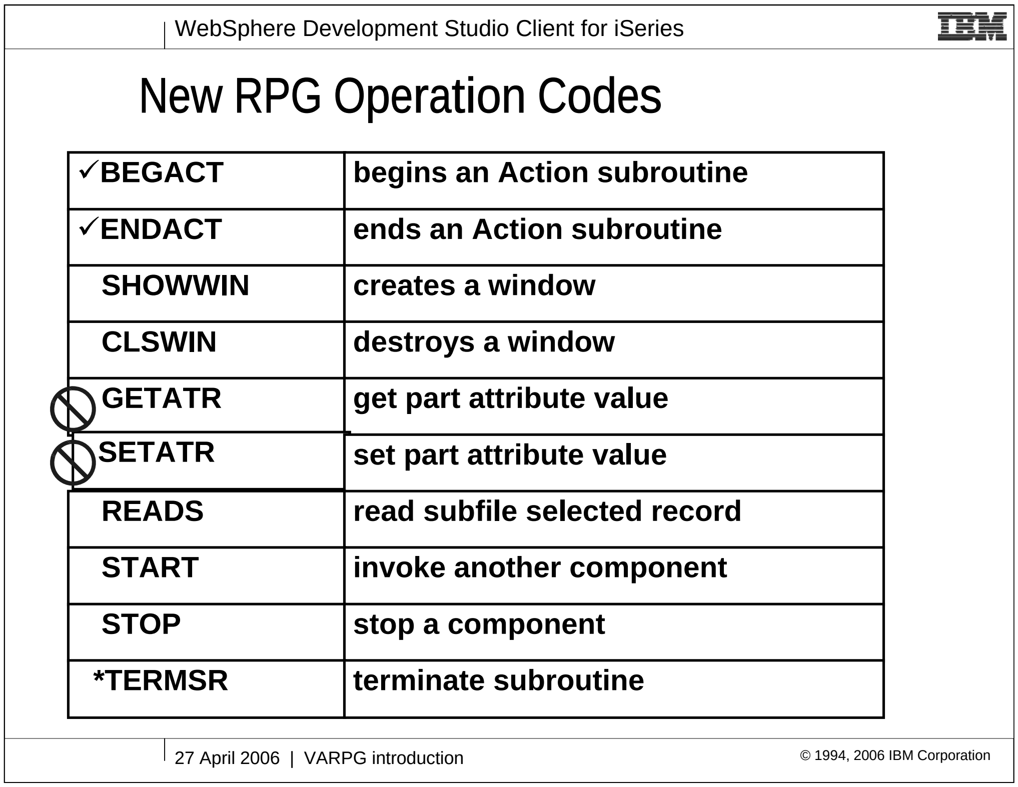
Button: CHECKIT

```
C    CUSTOMERNO     BEGACT     GOTFOCUS       WIN1
C                   EVAL       WIN1.CUSTOMERNO.BACKCOLOR = *YELLOW
                    ENDACT
```

```
C    CHECKIT                  BEGACT     PRESS          WIN1
C                             EVAL      CustNo = WIN1.CUSTOMERNO.TEXT
C                             EXSR       ReadCust
C                             IF             RecStatus = Missing
C    ErrMsg                   DSPLY    ErrBox         ErrReply
C                             ELSE
C                             IF             CredStatus = Good
C                             EVAL      WIN1.CHECKIT.BACKCOLOR = *BLUE
C                             ELSE
C                             EVAL      WIN1.CHECKIT.BACKCOLOR = *RED
C                              .........
```

When the Customer Number entry field (the CUSTOMERNO part) gets focus, the background color of the button CHECKIT is changed to pale gray. The GOTFOCUS event can be triggered in a number of ways, for example, by clicking with the mouse on the part or by using the Tab key.

When the CHECKIT button is PRESSed the logic checks the credit status for the entered Customer Number. If the customer is determined to be over the limit, the color of the CHECKIT button is changed to Red. If the customer's credit rating is good, the color of the button is changed to Blue.
In the example program, the CHECKIT button is also set as the default action for the key. This allows the user to perform the credit check by simply entering the Customer Number and pressing .



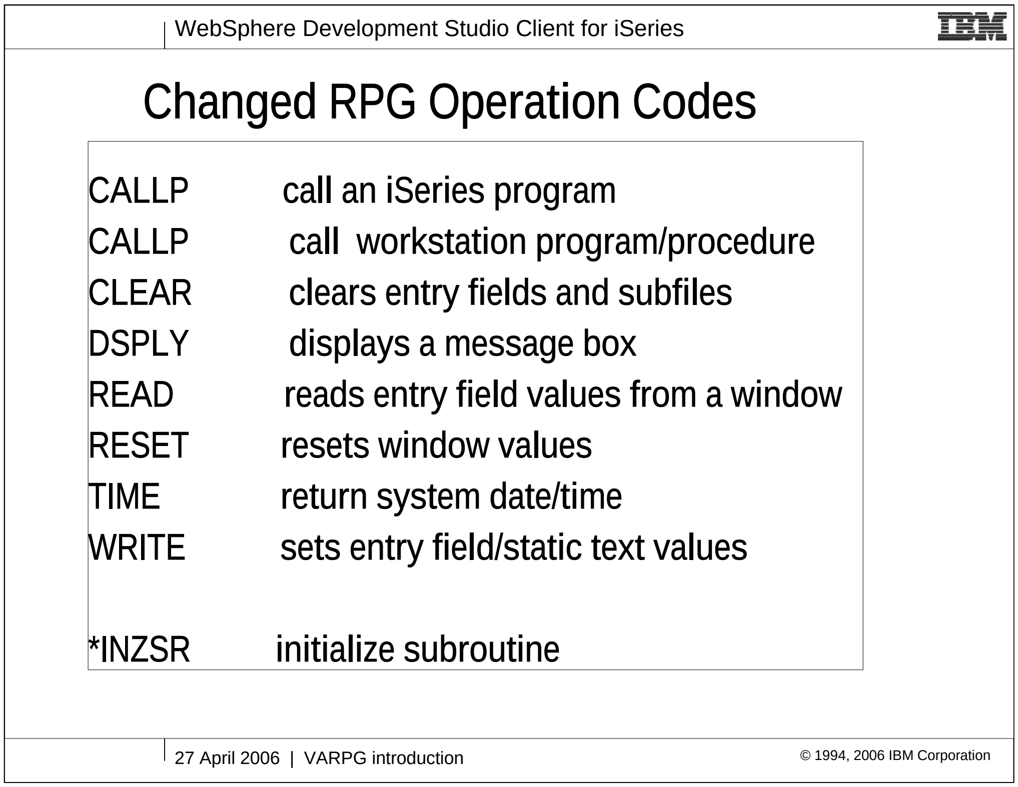
New RPG Operation Codes

✓BEGACT	begins an Action subroutine
✓ENDACT	ends an Action subroutine
SHOWWIN	creates a window
CLSWIN	destroys a window
⊘ GETATR	get part attribute value
⊘ SETATR	set part attribute value
READS	read subfile selected record
START	invoke another component
STOP	stop a component
*TERMSR	terminate subroutine

We will now introduce the VARPG operation codes that don't exist in RPGIV.

Since we already discussed BEGACT, ENDACT, GETATR, SETATR we will not get back to these new operation codes.

TERMSR is not an operation code, but we will handle it in this section as well.



Changed RPG Operation Codes

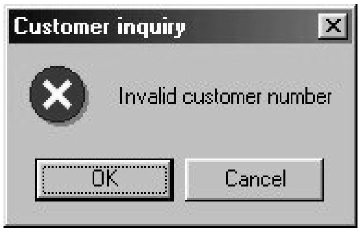
CALLP	call an iSeries program
CALLP	call workstation program/procedure
CLEAR	clears entry fields and subfiles
DSPLY	displays a message box
READ	reads entry field values from a window
RESET	resets window values
TIME	return system date/time
WRITE	sets entry field/static text values
*INZSR	initialize subroutine

Several of the normal RPGIV operation code had to be modified to adapt to the new runtime environment.

We will go through all the changes in detail in the following foils

DSPLY Operation Code

- ●Displays Dialogue box
  - ●Three styles
    - ➔*INFO *WARN & *HALT
  - ●Use literal, field, or Message
  - ●Input responses
    - ➔*OK *Cancel *Retry *Abort *Ignore
    - ➔*Enter *YesButton *NoButton



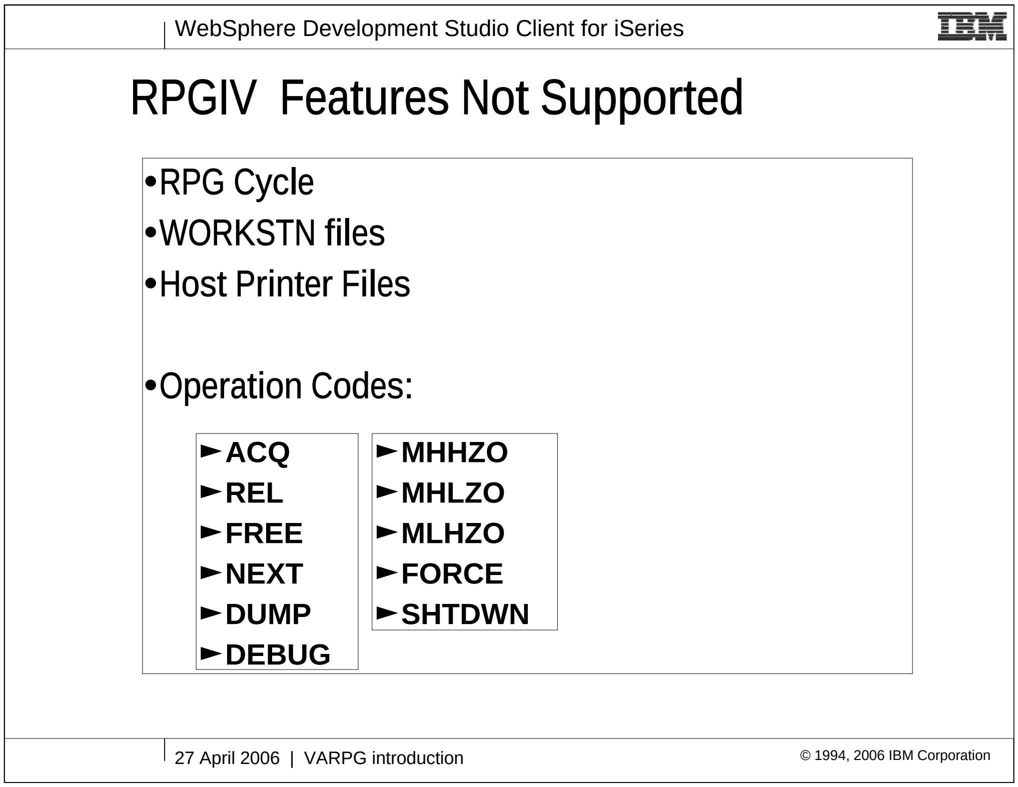
Customer inquiry

Invalid customer number

```
D ErrMsg           C  .....          'Invalid Customer Number'

D ErrBox           M  .....          STYLE(*HALT)
D                                    Button(*ok:*cancel)
C ErrMsg           DSPLY     ErrBox        Reply
C                  If        Reply=*ok
```

The DSPLY Op-code has been enhanced to allow VARPG programs to put up message boxes.

A value is returned in the Reply field which indicates the specific button that was pressed to clear the message. This value can be tested against the figurative constants listed (e.g. *OK, *Cancel, etc.)



RPGIV Features Not Supported

- RPG Cycle
- WORKSTN files
- Host Printer Files

- Operation Codes:
  - ACQ
  - REL
  - FREE
  - NEXT
  - DUMP
  - DEBUG
  - MHHZO
  - MHLZO
  - MLHZO
  - FORCE
  - SHTDWN

Some functions available in RPGIV have been totally dropped from VARPG

Everything related to the RPG cycle has been dropped with the exception of the LastRecord (LR) indicator.

WORKSTN files are not available in VARPG, so this device has been dropped from the F specs.

On the workstation we don't have printer files. If local printing is needed it has to be done with internal (program) described O specs, or by invoking a workstation report writer via DLL function calls or the ActiveX part.

If you want to use a PrinterFile we suggest to CALL a HostProgram that can then work with external described PrinterFiles

The operation codes dropped are not needed in the GUI event driven environment and haven't been very much used in the Green screen RPGIV environment

Program Source Structure

- RPG IV V5R3 Syntax
  - Monitor support
  - Free form
  - Qualified data structures
  - etc

```
                   H
                    *
                   F      Define files
                    *
                   D
Definition Specs   D
                    *
                   I      Local input files
                   I      Special files
                   I
                    *
                   C
                   C
                    *
                   O      Printer files
                   O      Local output files
                    *
                   P
                   D
                   C
                   C
                   P
                   **-Array data
```

All logic written in subroutines or procedures

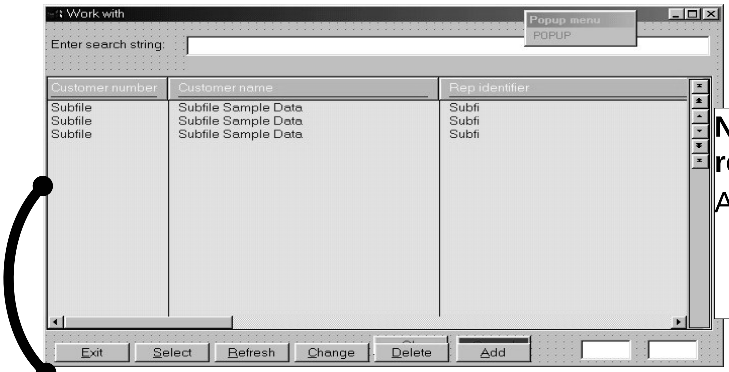
- **Action Subroutines**
- **User Subroutines**
- **Sub Procedures**
- ***INZSR**
- ***TERMSR**
- ***PSSR**
- ***ENTRY**
- **KLISTs**

✓ **One source file per application**

Agenda

- Overview
- The GUI designer
- The VARPG language
- Features
- Live demo of VARPG

What about Subfiles?



Customer number	Customer name	Rep identifier
Subfile	Subfile Sample Data	Subfi
Subfile	Subfile Sample Data	Subfi
Subfile	Subfile Sample Data	Subfi

No subfile control record

Additional op codes

- Clear
- Reset

SUBFILE is a GUI object

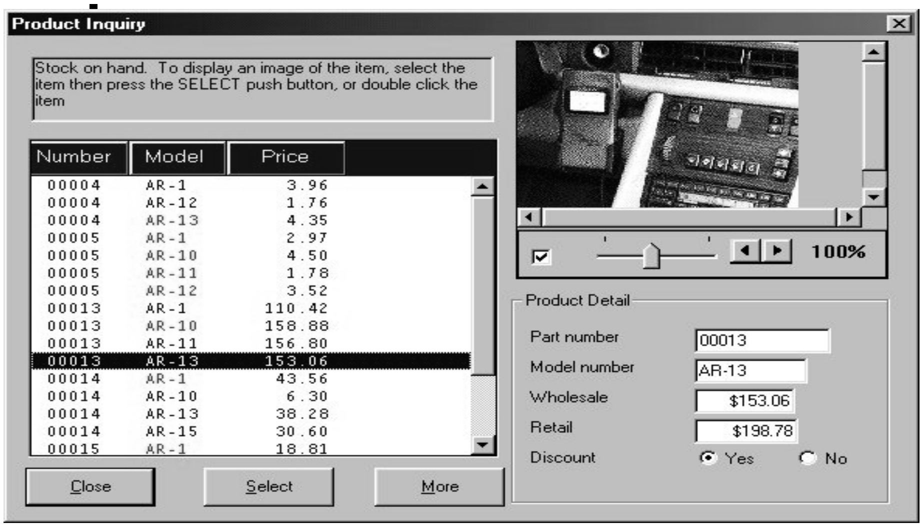
- **Just drag and drop database reference fields or define your own field list**
- **All existing Subfile op-codes are supported**
  - ***READC, DELETE, CHAIN, UPDATE***
- **Plus new READS (Read Selected)**

Subfiles are a powerful and frequently used facility in AS/400 programming. VARPG provides a GUI equivalent that supports all of the existing subfile Op-codes, and adds a new READS (Read Selected) operation to further simplify subfile use.

If you take advantage of VRPG's import capability to convert AS/400 screen definitions to their GUI equivalent, subfiles will be automatically converted to their VARPG equivalent.

The normal performance considerations for subfile programming (i.e. fill a page at a time and get the next page when the user rolls up) are probably even more important in C/S programming than for "green screen" so the techniques you've developed will not go to waste.

Subfile and Image Example



Product Inquiry

Stock on hand. To display an image of the item, select the item then press the SELECT push button, or double click the item

Number	Model	Price
00004	AR-1	3.96
00004	AR-12	1.76
00004	AR-13	4.35
00005	AR-1	2.97
00005	AR-10	4.50
00005	AR-11	1.78
00005	AR-12	3.52
00013	AR-1	110.42
00013	AR-10	158.88
00013	AR-11	156.80
00013	AR-13	153.06
00014	AR-1	43.56
00014	AR-10	6.30
00014	AR-13	38.28
00014	AR-15	30.60
00015	AR-1	18.81

Product Detail

Part number: 00013

Model number: AR-13

Wholesale: $153.06

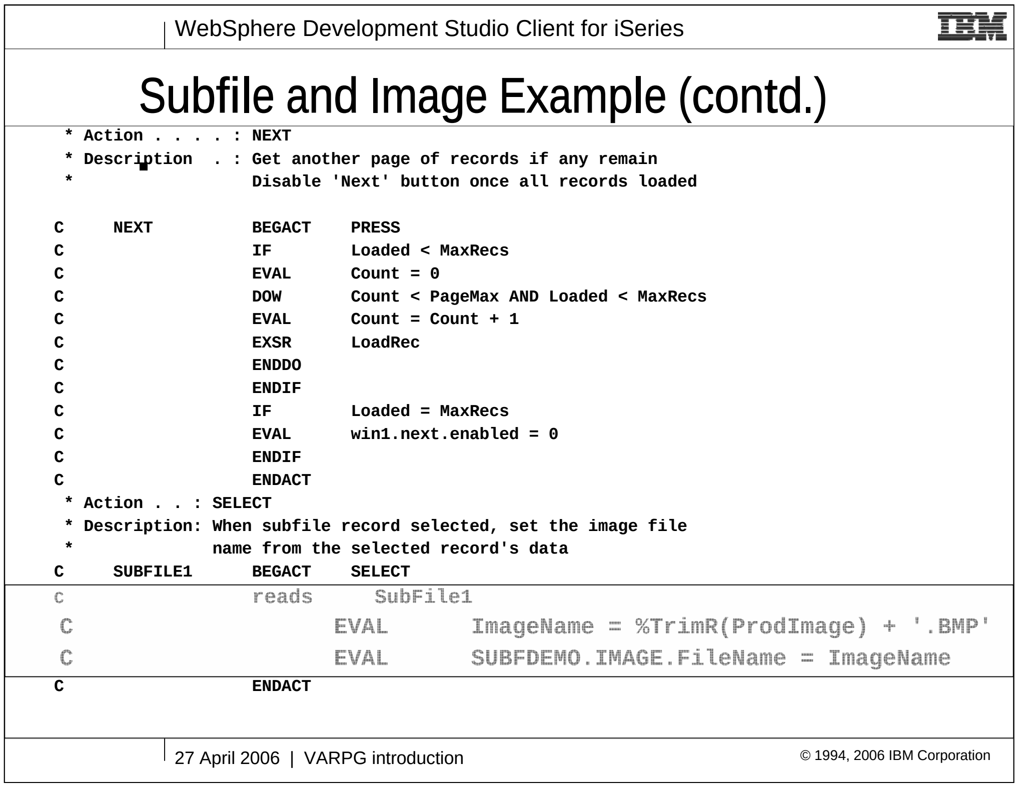
Retail: $198.78

Discount: ● Yes ○ No

This and the following page are taken directly from the VARPG code used in the demo program. The subfile part on the right hand side is the one shown in the previous foil.

Below it we have the "Next", "OK" and "Cancel" buttons. The logic "behind" the "Next" button is shown on the next page. The "OK" is linked to the same code as the SELECT event on the subfile (i.e. it causes the selected subfile image to be displayed). This code is also shown on the next page.

The image part on the right side uses all of the VARPG image capabilities (i.e. it has scroll bars and a magnification panel). As you'll see, very little programming is needed to display the image)



Subfile and Image Example (contd.)

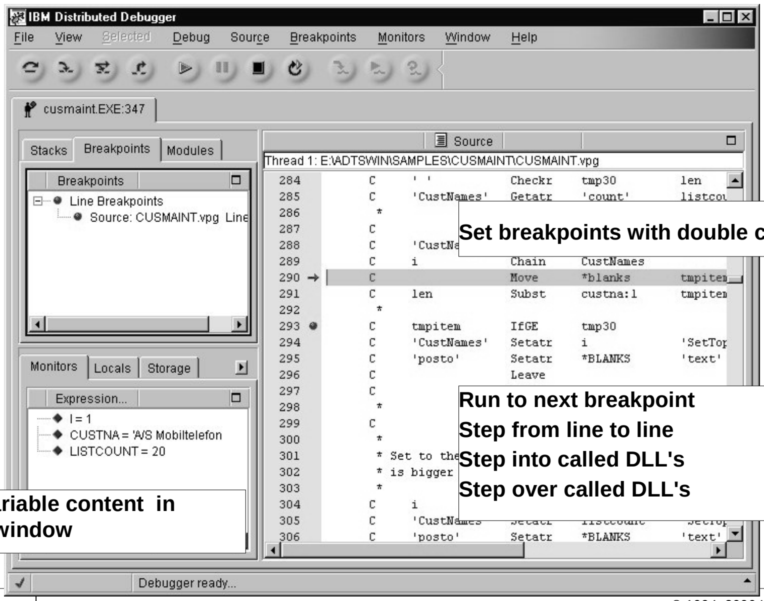
```
 * Action . . . . : NEXT
 * Description  . : Get another page of records if any remain
 *                  Disable 'Next' button once all records loaded

C     NEXT          BEGACT    PRESS
C                   IF        Loaded < MaxRecs
C                   EVAL      Count = 0
C                   DOW       Count < PageMax AND Loaded < MaxRecs
C                   EVAL      Count = Count + 1
C                   EXSR      LoadRec
C                   ENDDO
C                   ENDIF
C                   IF        Loaded = MaxRecs
C                   EVAL      win1.next.enabled = 0
C                   ENDIF
C                   ENDACT
 * Action . . : SELECT
 * Description: When subfile record selected, set the image file
 *              name from the selected record's data
C     SUBFILE1      BEGACT    SELECT
c                   reads       SubFile1
C                   EVAL      ImageName = %TrimR(ProdImage) + '.BMP'
C                   EVAL      SUBFDEMO.IMAGE.FileName = ImageName
C                   ENDACT
```

The first routine (NEXT - PRESS) is simply checking to see if there are further records to display and adding them to the subfile if any exist. The only VARPG specific code here is the SETATR statement at the end of the routine. This causes the "Next" button to be disable once all records have been loaded.

The second routine (SUBFILE1 - SELECT) first reads the selected subfile record (READS) then does an EVAL using the hidden field in the subfile record to form the full file name for the image that we want to display. Once this is done, it only remains to use SETATR (in this case the free-form %SetAtr is used) to set the file name into the "FileName" attribute of the image part.

Now what could be simpler!

Built in source level debugger



Thread 1: E:\ADTSWIN\SAMPLES\CUSMAINT\CUSMAINT.vpg

```
C     ' '          Checkr   tmp30        len
C     'CustNames'  Getatr   'count'      listcou
 *
C
C     'CustNa
C     i            Chain    CustNames
C                  Move     *blanks      tmpitem
C     len          Subst    custna:l     tmpitem
 *
C     tmpitem      IfGE     tmp30
C     'CustNames'  Setatr   i            'SetTop
C     'posto'      Setatr   *BLANKS      'text'
C                  Leave
C
 *
C
 *
 * Set to the
 * is bigger
 *
C     i
C     'CustNa
C     'posto'      Setatr   *BLANKS      'text'
```

I = 1
CUSTNA = 'A/S Mobiltelefon
LISTCOUNT = 20

Debugger ready...

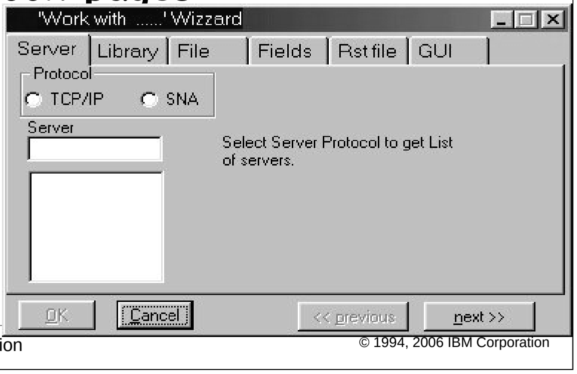
Set breakpoints with double click

Run to next breakpoint
Step from line to line
Step into called DLL's
Step over called DLL's

Watch variable content in monitor window

Wide variety of built in parts

- **Try different parts, like:**
  - ►**Radio buttons**
  - ►**Check boxes**
  - ►**Container**
  - ►**Notebook/Notebook pages**
  - ►**Calendar**
  - ►**Pop-Up Menus**
  - ►**Multi line edit**

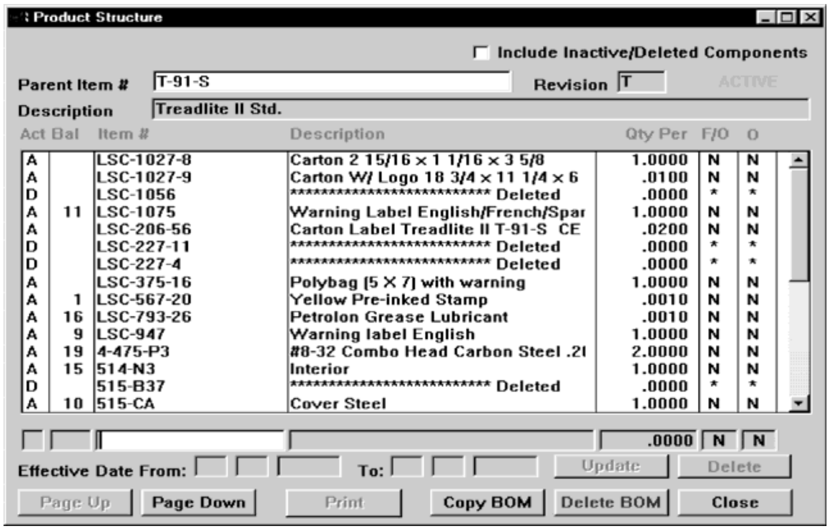


Select Server Protocol to get List of servers.

Many more features with new releases

- **ODBC support**
- **Many GUI subfile features**
  - **Sorting of columns**
  - **Changing location of columns at runtime**
- **Additional Parts that come in handy**
  - **Graph part**
  - **Progress bar,**
  - **Animation part**
  - **ActiveX part**
- **Set Server and communications API's**
- **Synchronization with RPGIV language definition**
- **JAVA source generation now at 1.4 level**
- **Each new release contains many features user requested**

Linemaster (USA) MAPICS frontend



☐ Include Inactive/Deleted Components

Parent Item # T-91-S Revision T ACTIVE

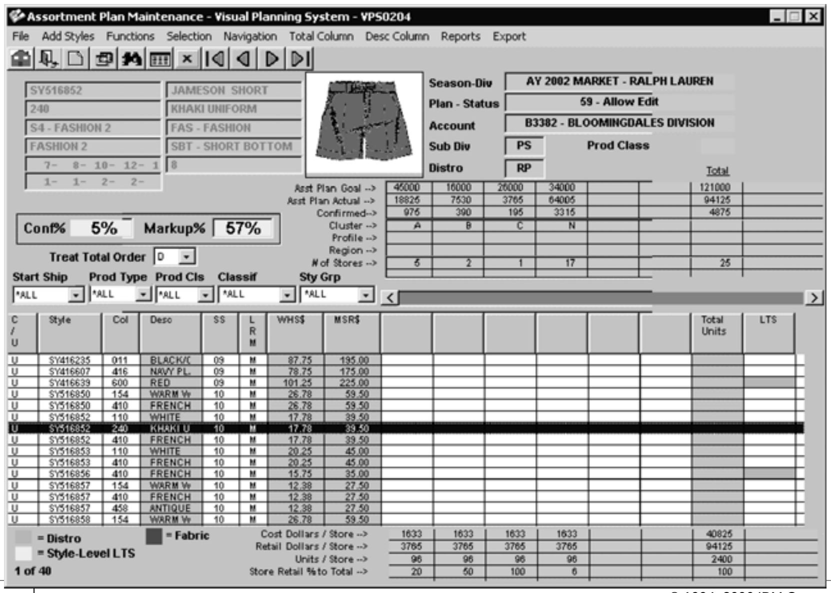
Description Treadlite II Std.

Act	Bal	Item #	Description	Qty Per	F/O	O
A		LSC-1027-8	Carton 2 15/16 x 1 1/16 x 3 5/8	1.0000	N	N
A		LSC-1027-9	Carton W/ Logo 18 3/4 x 11 1/4 x 6	.0100	N	N
D		LSC-1056	************************** Deleted	.0000	*	*
A	11	LSC-1075	Warning Label English/French/Spar	1.0000	N	N
A		LSC-206-56	Carton Label Treadlite II T-91-S CE	.0200	N	N
D		LSC-227-11	************************** Deleted	.0000	*	*
D		LSC-227-4	************************** Deleted	.0000	*	*
A		LSC-375-16	Polybag (5 X 7) with warning	1.0000	N	N
A	1	LSC-567-20	Yellow Pre-inked Stamp	.0010	N	N
A	16	LSC-793-26	Petrolon Grease Lubricant	.0010	N	N
A	9	LSC-947	Warning label English	1.0000	N	N
A	19	4-475-P3	#8-32 Combo Head Carbon Steel .2(	2.0000	N	N
A	15	514-N3	Interior	1.0000	N	N
D		515-B37	************************** Deleted	.0000	*	*
A	10	515-CA	Cover Steel	1.0000	N	N

.0000 N N

Effective Date From: To:

Capricorn (USA) Apparel industry package



Assortment Plan Maintenance - Visual Planning System - VPS0204

SY516852 | JAMESON SHORT
240 | KHAKI UNIFORM
S4 - FASHION 2 | FAS - FASHION
FASHION 2 | SBT - SHORT BOTTOM
7- 8- 10- 12- 1 | 8
1- 1- 2- 2-

Season-Div: AY 2002 MARKET - RALPH LAUREN
Plan - Status: 59 - Allow Edit
Account: B3382 - BLOOMINGDALES DIVISION
Sub Div: PS
Prod Class:
Distro: RP

Conf% 5%
Markup% 57%

Treat Total Order: D

							Total
Asst Plan Goal -->	45000	16000	26000	34000			121000
Asst Plan Actual -->	18826	7530	3765	64005			94125
Confirmed-->	976	390	195	3315			4875
Cluster -->	A	B	C	N			
Profile -->							
Region -->							
# of Stores -->	5	2	1	17			25

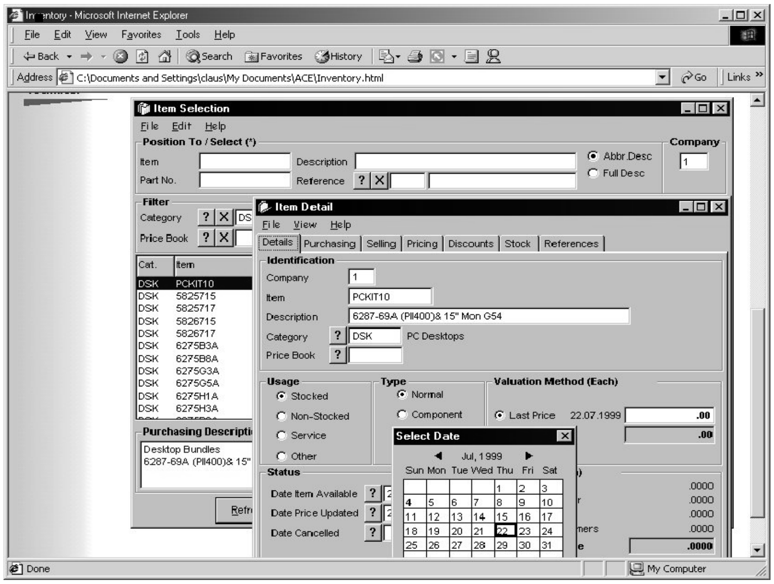
Start Ship: *ALL
Prod Type: *ALL
Prod Cls: *ALL
Classif: *ALL
Sty Grp: *ALL

C / U	Style	Col	Desc	SS	L R M	WHS$	MSR$							Total Units	LTS
U	SY416235	011	BLACK/C	09	M	87.75	195.00								
U	SY416607	416	NAVY PL.	09	M	78.75	175.00								
U	SY416639	600	RED	09	M	101.25	225.00								
U	SY516850	154	WARM W	10	M	26.78	59.50								
U	SY516850	410	FRENCH	10	M	26.78	59.50								
U	SY516852	110	WHITE	10	M	17.78	39.50								
U	SY516852	240	KHAKI U	10	M	17.78	39.50								
U	SY516852	410	FRENCH	10	M	17.78	39.50								
U	SY516853	110	WHITE	10	M	20.25	45.00								
U	SY516853	410	FRENCH	10	M	20.25	45.00								
U	SY516856	410	FRENCH	10	M	15.75	35.00								
U	SY516857	154	WARM W	10	M	12.38	27.50								
U	SY516857	410	FRENCH	10	M	12.38	27.50								
U	SY516857	458	ANTIQUE	10	M	12.38	27.50								
U	SY516858	154	WARM W	10	M	26.78	59.50								

▒ = Distro
□ = Style-Level LTS
■ = Fabric
1 of 40

Cost Dollars / Store -->	1633	1633	1633	1633			40825
Retail Dollars / Store -->	3765	3765	3765	3765			94125
Units / Store -->	96	96	96	96			2400
Store Retail % to Total -->	20	50	100	6			100

Versatile (Australia) inventory sample


Inventory - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Go Links
Item Selection
File Edit Help
Refr
Item Detail
File View Help
Details Purchasing Selling Pricing Discounts Stock References
Done
My Computer

Address C:\Documents and Settings\claus\My Documents\ACE\Inventory.html

Position To / Select (*)

Item

Part No.

Description

Reference

● Abbr.Desc
○ Full Desc

Company
1

Filter

Category DS

Price Book

Cat.	Item
DSK	PCKIT10
DSK	5825715
DSK	5825717
DSK	5826715
DSK	5826717
DSK	6275B3A
DSK	6275B8A
DSK	6275G3A
DSK	6275G5A
DSK	6275H1A
DSK	6275H3A

Purchasing Descripti

Desktop Bundles
6287-69A (PII400)& 15"

Identification

Company 1

Item PCKIT10

Description 6287-69A (PII400)& 15" Mon G54

Category DSK PC Desktops

Price Book

Usage
● Stocked
○ Non-Stocked
○ Service
○ Other

Type
● Normal
○ Component

Valuation Method (Each)
● Last Price 22.07.1999 .00
.00

Status

Date Item Available

Date Price Updated

Date Cancelled

Select Date

Jul, 1999

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	31

)

.0000

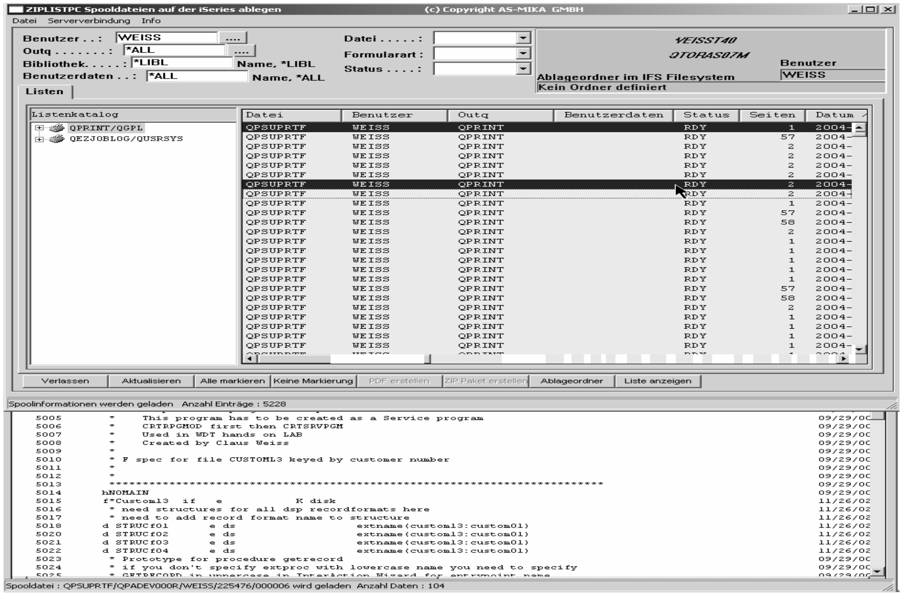
r .0000

.0000

mers .0000

e .0000

AS-MIKA (Germany) Spoolfile sample



Benutzer . . : WEISS
Outq . . . . : *ALL
Bibliothek. . . . . : *LIBL Name, *LIBL
Benutzerdaten . . : *ALL Name, *ALL
Datei . . . . . :
Formularart :
Status . . . . :

VEISST40
QTORAS07M
Benutzer: WEISS
Ablageordner im IFS Filesystem
Kein Ordner definiert

Listenkatalog
- QPRINT/QGPL
- QEZJOBLOG/QUSRSYS

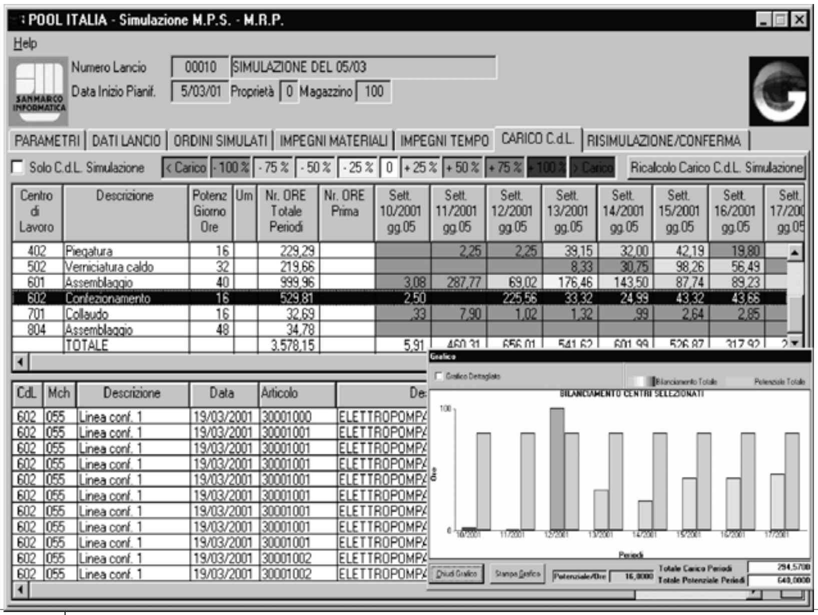
Datei	Benutzer	Outq	Benutzerdaten	Status	Seiten	Datum
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	57	2004-
QPSUPRTF	WEISS	QPRINT		RDY	2	2004-
QPSUPRTF	WEISS	QPRINT		RDY	2	2004-
QPSUPRTF	WEISS	QPRINT		RDY	2	2004-
QPSUPRTF	WEISS	QPRINT		RDY	2	2004-
QPSUPRTF	WEISS	QPRINT		RDY	2	2004-
QPSUPRTF	WEISS	QPRINT		RDY	2	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	57	2004-
QPSUPRTF	WEISS	QPRINT		RDY	58	2004-
QPSUPRTF	WEISS	QPRINT		RDY	2	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	57	2004-
QPSUPRTF	WEISS	QPRINT		RDY	58	2004-
QPSUPRTF	WEISS	QPRINT		RDY	2	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-
QPSUPRTF	WEISS	QPRINT		RDY	1	2004-

Spoolinformationen werden geladen   Anzahl Einträge : 5228

```
  5005        *     This program has to be created as a Service program                  09/29/0C
  5006        *     CRTRPGMOD first then CRTSRVPGM                                       09/29/0C
  5007        *     Used in WDT hands on LAB                                             09/29/0C
  5008        *     Created by Claus Weiss                                               09/29/0C
  5009        *                                                                          09/29/0C
  5010        * F spec for file CUSTOML3 keyed by customer number                        09/29/0C
  5011        *                                                                          09/29/0C
  5012        *                                                                          09/29/0C
  5013        ***************************************************************************09/29/0C
  5014       hNOMAIN                                                                     09/29/0C
  5015       f*Customl3  if   e             K disk                                       11/26/02
  5016        * need structures for all dsp recordformats here                           11/26/02
  5017        * need to add record format name to structure                              11/26/02
  5018       d STRUCf01      e ds                  extname(customl3:custom01)            11/26/02
  5020       d STRUCf02      e ds                  extname(customl3:custom01)            11/26/02
  5021       d STRUCf03      e ds                  extname(customl3:custom01)            11/26/02
  5022       d STRUCf04      e ds                  extname(customl3:custom01)            11/26/02
  5023        * Prototype for procedure getrecord                                        09/29/0C
  5024        * if you don't specify extproc with lowercase name you need to specify     09/29/0C
```

Spooldatei : QPSUPRTF/QPADEV000R/WEISS/225476/000006 wird geladen Anzahl Daten : 104

San Marco (Italy) ERP sample



Numero Lancio 00010 SIMULAZIONE DEL 05/03

Data Inizio Pianif. 5/03/01 Proprietà 0 Magazzino 100

□ Solo C.d.L. Simulazione

Centro di Lavoro	Descrizione	Potenz Giorno Ore	Um	Nr. ORE Totale Periodi	Nr. ORE Prima	Sett. 10/2001 gg.05	Sett. 11/2001 gg.05	Sett. 12/2001 gg.05	Sett. 13/2001 gg.05	Sett. 14/2001 gg.05	Sett. 15/2001 gg.05	Sett. 16/2001 gg.05	Sett. 17/200 gg.05
402	Piegatura	16		229,29			2,25	2,25	39,15	32,00	42,19	19,80	
502	Verniciatura caldo	32		219,66					8,33	30,75	98,26	56,49	
601	Assemblaggio	40		999,96		3,08	287,77	69,02	176,46	143,50	87,74	89,23	
602	Confezionamento	16		529,81		2,50		225,56	33,32	24,99	43,32	43,66	
701	Collaudo	16		32,69		,33	7,90	1,02	1,32	,99	2,64	2,85	
804	Assemblaggio	48		34,78									
	TOTALE			3.578,15		5,91	460,31	656,01	541,62	601,99	526,87	317,92	2

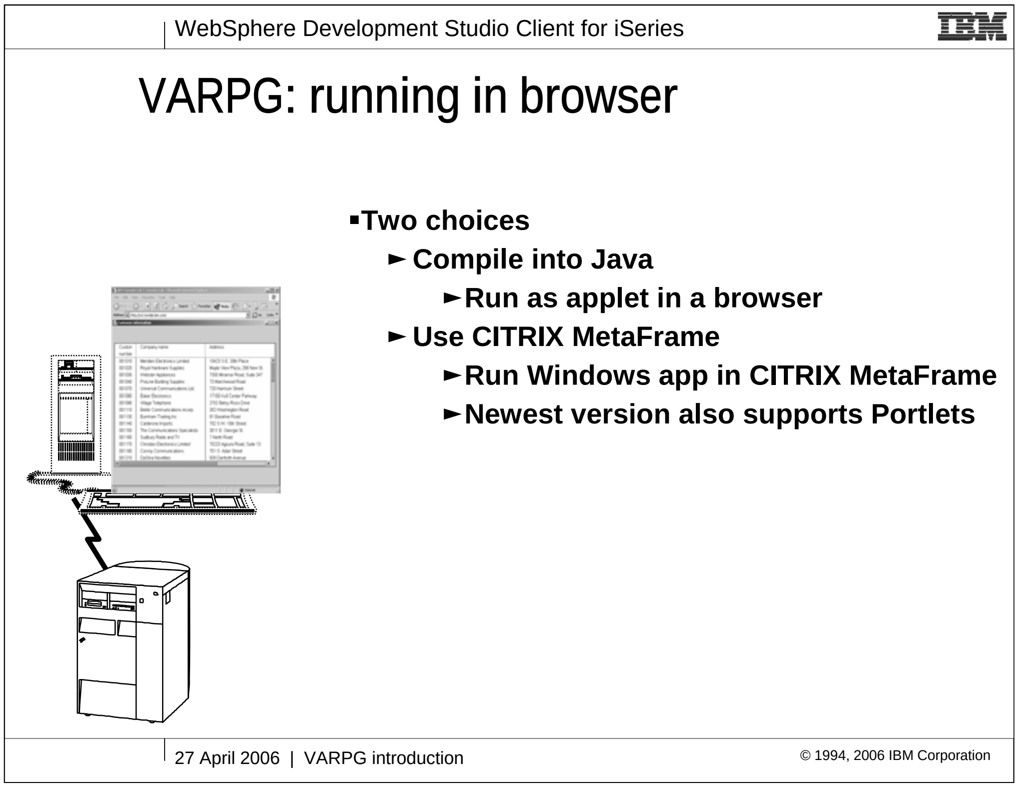
CdL	Mch	Descrizione	Data	Articolo	De
602	055	Linea conf. 1	19/03/2001	30001000	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001001	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001002	ELETTROPOMPA
602	055	Linea conf. 1	19/03/2001	30001002	ELETTROPOMPA

□ Grafico Dettagliato

Potenziale/Ore 16,0000

Totale Carico Periodi 294,5700

Totale Potenziale Periodi 640,0000



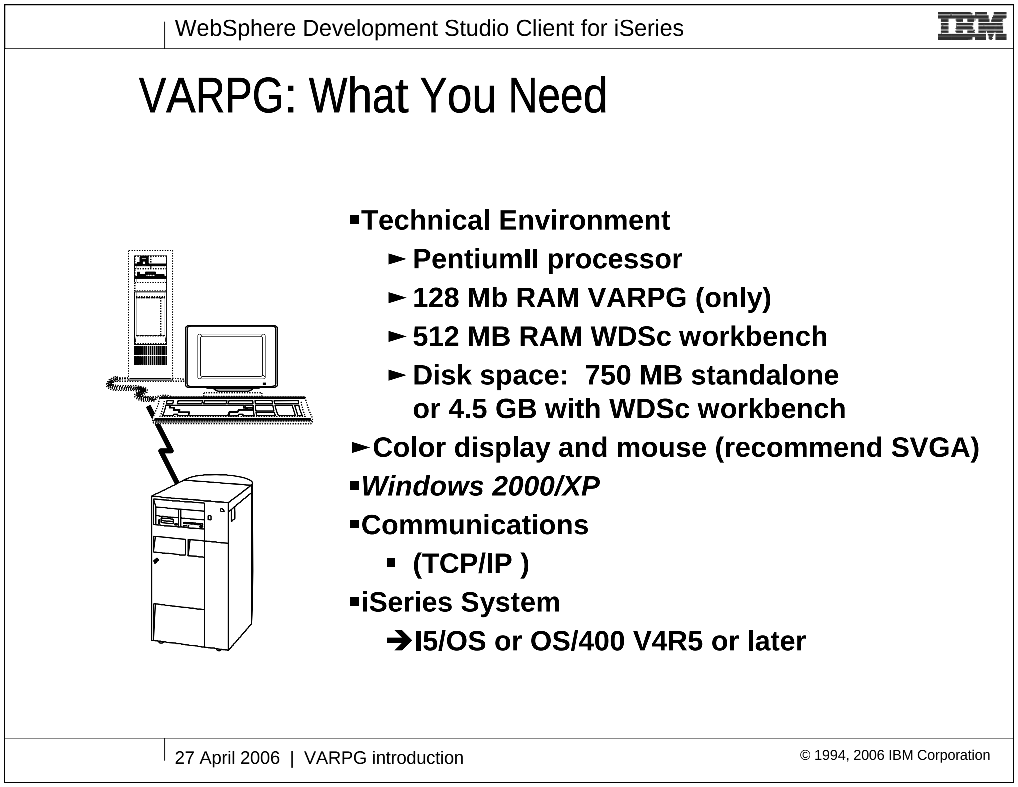
VARPG: running in browser

- Two choices
  - ► Compile into Java
    - ►Run as applet in a browser
  - ► Use CITRIX MetaFrame
    - ►Run Windows app in CITRIX MetaFrame
    - ►Newest version also supports Portlets

This chart depicts the hardware and software recommendations for using the VARPG development environment.

Note that the application runtime requirements are simply a reflection of the runtime requirements of the client operating system.

There is NO RUN TIME LICENSE required on the user's workstation to run applications created using VARPG Client/2.



VARPG: What You Need

- **Technical Environment**
  - ► **PentiumII processor**
  - ► **128 Mb RAM VARPG (only)**
  - ► **512 MB RAM WDSc workbench**
  - ► **Disk space: 750 MB standalone or 4.5 GB with WDSc workbench**
- ►**Color display and mouse (recommend SVGA)**
- ***Windows 2000/XP***
- **Communications**
  - **(TCP/IP)**
- **iSeries System**
  - ➔**I5/OS or OS/400 V4R5 or later**

This chart depicts the hardware and software recommendations for using the VARPG development environment.

Note that the application runtime requirements are simply a reflection of the runtime requirements of the client operating system.

There is NO RUN TIME LICENSE required on the user's workstation to run applications created using VARPG Client/2.

Summary

- ✓ VARPG provides a complete C/S solution when combined with ***<u>WDSc workbench</u>***
- ✓ GUI designer provides edit/compile/debug integration
- ✓ Use built in communications layer to access iSeries
- ✓ Use RPGIV features on the Client and on the Server
- ✓ Access data
  - ✓in DB2 UDB for iSeries
  - ✓in local ASCII files
  - ✓or with embedded SQL in all DB2 supported servers
  - ✓in *ODBC* databases

➔>>><u>Use your RPG skills and build GUI applications</u> <<<

Agenda

- Overview
- The GUI designer
- The VARPG language
- Features
- Live demo of VARPG

Enjoy working with VisualAge RPG!

-- Demo --

Acknowledgment and Disclaimer

Acknowledgment:
Many people contributed to this presentation. In particular, thanks goes to Claus Weiss - IBM Corporation for creating the original presentation.

Disclaimer:
This presentation may contain examples of code and names of companies or persons. The code is given for presentation purposes and has not been tested by IBM. Therefore IBM does not guarantee the reliability, serviceability, or function of the code and the code is provided "AS IS". IBM EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE AND MERCHANTABILITY. Any names appearing in this presentation are designed to be fictitious and IBM makes no representations as to the accuracy of the names or data presented in accordance therewith.

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

Trademarks and Disclaimers

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Instruction: Refer to the following URL: http://www.ibm.com/legal/copytrade.shtml. Edit the list below, IBM subsidiary statement, and special attribution companies which follow so they coincide with your presentation.

AS/400
AS/400e
eServer
@server

e-business on demand
IBM
IBM (logo)
iSeries

i5/OS
OS/400
System i5

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, other countries, or both.
Intel, Intel Logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Other company, product or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.