

What Year Is It?

Working with Fiscal Years, Business Days and Weekdays

By Noreen Redden

When I get the third question in a month about the same issue, I figure it's time to step back and talk about it. That's what I'll be doing in this article in regards to some questions related to dates in WebFOCUS and FOCUS.

There are a couple of date-related features that were actually implemented in 7.7.01 or earlier but were not documented, pending completion of testing. They will be documented in 7.7.03, but are working fine in 7.7.02, so you might be able to use them now.

The first feature involves fiscal years. Colleges typically start their fiscal years at the beginning of July because they don't want to have to deal with a fiscal year ending and a second semester starting at the same time. Retail businesses, on the other hand, often have fiscal years beginning in February, well after the holiday rush.

For reporting, it might be important to know the week number in relation to the fiscal, rather than calendar, year. Not only do we have to know when the year begins, but also how to label it. If the year begins June 1, then on July 1, 2011 am I in fiscal year 2011 or fiscal year 2012? To figure it out, let's employ fiscal year subroutines that take a date and determine the fiscal year and/or fiscal quarter to which it belongs.

First, the parameters to specify the fiscal year rules:

1. Starting Point of the fiscal year: This may be specified as a just a month (and day 01 assumed) or a month and day. (What happens if it is the first business day in February? I'll answer that at the very end of this article.)
2. Year numbering scheme: Use the year at the start of the fiscal year (FYS) or at the end of the fiscal year (FYE). In other words, if the starting point of the fiscal year is Feb. 1 for this year, the fiscal year would be Feb. 1, 2011 through Jan. 31, 2012. If FYS, then March 1, 2011 is in fiscal year 2011. Same date, with FYE would be fiscal year 2012.
3. Low component: What is the lowest component in the date being supplied? Remember, we need to be able to interpret that integer. If the lowest component is D, then the value represents number of days (displacement) from 12/31/1900. That is obviously important and required if the Starting Point is other than the first of any given month. (If the fiscal year starts on April 15, then April 14 and April 16 are in two different fiscal years). If the lowest component is M, then

the displacement is from 01/1901, and is only expressed in number of months. If the starting point is also a starting point for a calendar quarter (i.e., Jan 1, April 1, July 1, Oct. 1) then the low component may be specified as Q, because the function would be able to translate a calendar quarter to a fiscal quarter.

Once those parameters have been determined, the next determination we need to make is which subroutine to use.

FIYR returns a four-digit integer containing the fiscal year.

FIQTR returns a one-digit integer representing only the Quarter (1-4) with no reference to the year.

FIYYQ returns a "smart date" of the format YYQ.

Then, follow the same argument structure:

```
Output/format = FIYR(inputdate, lowcomponent,  
startmth, startday, yrnumber, out) ;
```

Where:

lowcomponent is D, M, Q.

Startmth and startday define the starting point of the fiscal year.

Yrnumber is the numbering scheme (FYE or FYS).

Out is the name of the output field or the format of the output field in quotes:

FIYR returns 'YY'

FIQTR returns 'Q'

FIYYQ returns 'YYQ'

So, for example:

```
DEFINE FILE MOVIES  
FISCALYY/YY=FIYR(RELDATE, 'D', 10, 1, 'FYE', FISCALYY) ;  
FISCALY/Y=FIYR(RELDATE, 'D', 10, 1, 'FYE', FISCALY) ;  
FISCALI/I4=FIYR(RELDATE, 'D', 10, 1, 'FYE', FISCALI) ;  
NEWTITLE/A20=EDIT (TITLE, '99999999999999999999') ;  
END
```

```
TABLE FILE MOVIES  
PRINT MOVIECODE NEWTITLE RELDATE FISCALYY FISCALY FISCALI  
IF RECORDLIMIT EQ 8  
END
```

MOVIECODE	NEWTITLE	RELDATE	FISCALYY	FISCALY	FISCALI
-----	-----	-----	-----	-----	-----
001MCA	JAWS	78/05/13	1978	78	1978
005WAR	EAST OF EDEN	55/01/12	1955	55	1955
020TUR	CITIZEN KANE	41/08/11	1941	41	1941
024WAR	DOG DAY AFTERNOON	76/04/04	1976	76	1976
031KKV	SMURFS, THE	88/02/16	1988	88	1988
035CBS	CABARET	73/07/14	1973	73	1973
040ORI	BABETTE'S FEAST	88/03/23	1988	88	1988
043DIS	SHAGGY DOG, THE	59/01/09	1959	59	1959

Remember that to use any "smart date" subroutine in Dialogue Manager -SET, you first have to convert the variable date to a smart date:

```
-SET &INDT=20100105;
-SET &YY = EDIT(&INDT,'9999');
-* checking Dialogue Manager with subroutine "
-SET &MM = 1;
-SET &DD = 15;
-SET &FISCALDT = FIYR((DATECVT(&INDT, 'I8YYMD', 'YYMD')),
- 'D', &MM , &DD , 'FYS ', 'YY');
```

Now let's go back to the quandary I posed: What happens if the starting date is the first business day in February?

Well, the first thing is to figure out what is a business day. The set of BUSDAYS establishes the days of the week that are considered business days. `_MTWTF_` for instance indicates Monday through Friday are the business days, whereas `SMTWT__` would indicate Sunday through Thursday as working days, with Friday and Saturday as the weekends.

The `SET HDAY =xxx` points to the correct holiday file. `SET HDAY=US`, for instance, will look for `HDAYUS.err` in the `/bin` directory. The second new feature in 7.7 allows that file to be saved to a directory other than `/bin` and then a `FILEDEF` issued to point to the dataset. So `SET HDAY = US` and `FILEDEF HDAYUS DISK c:\bi\apps\myapp\holiday.ftm` will find the file. If no `FILEDEF` is issued, then the `/bin` will be searched. Note that when FOCUS for MVS in 7.7 is released, `DYNAM ALLOC` may also be used to point to the file.

Now that we have our holiday file, and we know our business days, we can determine the first business day in any given month, or for the whole year. The first day of the month is easy. It just is `Month/01/year`. The argument `WD+` or `WD-` or `BD+` or `BD-` are very helpful here. That argument says "examine the input date." If it is already a business day (or working day), leave it alone. However, if it is not, then move forward (+) or back(-) until you get to a business day. So:

```
DEFINE FILE FSEQ
```

```

BEGMNT1/WMDYY = DATEMOV (DATE1, 'BOM') ;
BEGMNT2/WMDYY = DATEMOV (BEGMNT1, 'BD+') ;
END
TABLE FILE FSEQ
PRINT BEGMNT1 AS 'FIRST,OF,MONTH' BEGMNT2 AS
'FIRST,BUS,DAY'
BY ORDER AS 'MONTH'
IF ORDER FROM 1 TO 12
END

```

Results in:

MONTH	FIRST OF MONTH	FIRST BUS DAY
-----	-----	-----
1	SAT, 01/01/2011	MON, 01/03/2011
2	TUE, 02/01/2011	TUE, 02/01/2011
3	TUE, 03/01/2011	TUE, 03/01/2011
4	FRI, 04/01/2011	FRI, 04/01/2011
5	SUN, 05/01/2011	MON, 05/02/2011
6	WED, 06/01/2011	WED, 06/01/2011
7	FRI, 07/01/2011	FRI, 07/01/2011
8	MON, 08/01/2011	MON, 08/01/2011
9	THU, 09/01/2011	THU, 09/01/2011
10	SAT, 10/01/2011	MON, 10/03/2011
11	TUE, 11/01/2011	TUE, 11/01/2011
12	THU, 12/01/2011	THU, 12/01/2011

But watch those tricky Business/Working Days, particularly with DATADD.
Let's say we get paid the next business day after the first of the month. So, I'll use the following:

```

DATEADD
DATEADD/WMDYY = DATEADD (BEGMNT1, 'BD', 1) ;

```

I get this:

MONTH	FIRST OF MONTH	FIRST BUS DAY	FIRST BUSINESS DAY AFTER THE 1ST
-----	-----	-----	-----
1	SAT, 01/01/2011	MON, 01/03/2011	TUE, 01/04/2011
2	TUE, 02/01/2011	TUE, 02/01/2011	WED, 02/02/2011
3	TUE, 03/01/2011	TUE, 03/01/2011	WED, 03/02/2011
4	FRI, 04/01/2011	FRI, 04/01/2011	MON, 04/04/2011
5	SUN, 05/01/2011	MON, 05/02/2011	TUE, 05/03/2011

Wrong! At least to my thinking. If the first is Saturday, then adding one business day should bring me to Monday. Nope. The subroutine says it can't add a business day to something that *is not* a business day. So, first move to a business day, and then do your add. So 1/1 moves to 1/3, and then after we add 1 to January 4. It sort of makes sense.

Let's say that a service representative has to call a customer back on the next day after he has the call. So, if the customer calls on Tuesday, the rep has all day Tuesday to do the research, etc., and call on the next business day – Wednesday. However, if the customer opened the call via the Web on Saturday, the rep still needs Monday to do the research, so the next day should be Tuesday. The rule is in the eye of the beholder. But, if your customer really means the very next business day, use BD-. So, move Saturday 1/1 to Friday 12/31 (or Thursday 12/30 if New Year's Eve was a holiday), then add 1 and come up with Monday:

```
NXTBS/WMDYY = DATEADD ( (DATEMOV (BEGMNT1, 'BD-') ), 'BD', 1);
```

First, if the beginning of the month is not a business day, use DATEMOV to move to the last business day of the prior month, then add one business day and ...

MONTH	FIRST OF MONTH	NEXT BUSINESS DAY
-----	-----	-----
1	SAT, 01/01/2011	MON, 01/03/2011
2	TUE, 02/01/2011	WED, 02/02/2011
3	TUE, 03/01/2011	WED, 03/02/2011
4	FRI, 04/01/2011	MON, 04/04/2011
5	SUN, 05/01/2011	MON, 05/02/2011

Almost any time I deal with business days or working days, I at least take a look to make sure I'm getting what I want. Suppose you want the number of business days in a month (maybe to calculate an average). So you use DATEDIFF to subtract the start date from the end date. Generally you would add 1 since in DATEDIFF, the calculation is exclusive of the ending date. However, if the ending date was not itself a business day, then you don't want to add it back in. So, first move the last day to the last business day, and then you can always add 1.

Other new features are affecting dates, such as &DATEFMT, FPRINT, so keep an eye out for new features and your business requirements.

No Space for Dates

By Art Greenhaus

For some time now, the FOCUS language has allowed users to specify the current date using the Dialogue Manager Variable `&DATE` (MM/DD/YY) or the other variants, `&YMD`, `&YYMD`, `&MDY`, `&MDYY`, `&DMY` and `&DMYY`.

After users requested it, other "current date" formats were allowed, using the Dialogue Manager field `&DATEfmt`, where `fmt` can be any valid date format, such as `MTrDYY`, which displayed the current date using the supplied date format. Thus, `&DATEMTrDYY`, on May 12, 2011, would display May 12, 2011.

What is not so obvious is that the length of that output string would be 18 characters. This length results from the length of the year (4) plus the length of the day (2), plus the comma and space between the day and the year (2), plus the maximum length of the month, plus the trailing space (10), allowing for the month of September. While this may not sound like a big deal, when various `&DATE` formats are placed in a `HEADING`, the extra spaces at the end of the variables may not be what was desired. Yes, the `TRUNCATE` function could be used to drop the trailing spaces, but that required additional coding, `BEFORE` use of the variable.

With release 7.7.03, an alternative has been made available, to remedy this issue. Instead of using `&DATEfmt`, users could take advantage of `&DATXfmt` (note the 'X' in place of the 'E' following `&DAT`). This new date format specification automatically dropped trailing spaces. As a way of demonstrating, consider the following:

```
-TYPE DATE &DATE
-TYPE YMD &YMD
-TYPE YYMD &YYMD
-TYPE MDY &MDY
-TYPE MDYY &MDYY
-TYPE DMY &DMY
-TYPE DMYY &DMYY
-TYPE DATEMTrDYY &DATEMTrDYY &DATEMTrDYY.LENGTH
-TYPE DATXMTrDYY &DATXMTrDYY &DATXMTrDYY.LENGTH
```

Here's the output it produces:

```
DATE 05/12/11
YMD 110512
YYMD 20110512
MDY 051211
MDYY 05122011
DMY 120511
DMYY 12052011
```

```
DATEMTrDYY MAY 12, 2011      18
DATXMTrDYY MAY 12, 2011 12
```

Note how the last output line, using the &DATX construct, produces a result of only 12 characters, while the prior output line, using the &DATE construct, was 18 characters long.

In addition to truncating trailing spaces, the 7.7.03 release also allows for a date-time format (displaying a date and/or time), following either &DATE or &DATX. As an example, consider the following:

```
-TYPE &DATXHMTDYYI &DATXHMTDYYI.LENGTH
-TYPE &DATEHMTDYYI &DATEHMTDYYI.LENGTH
-TYPE &DATXHMTDYY &DATXHHIA
-TYPE &DATEHMTDYY &DATEHHIA
```

This produces the following:

```
May 12 2011 11:08 17
May 12 2011 11:08      23
May 12 2011 11:08AM
May 12 2011      11:08AM
```

The first two lines include a time in their display. Note the difference in length. For the last two lines, two variables are displayed, as might be done in a `HEADING`. Note how using the &DATX construct suppresses the trailing blanks.

Creating the Guided Report Form

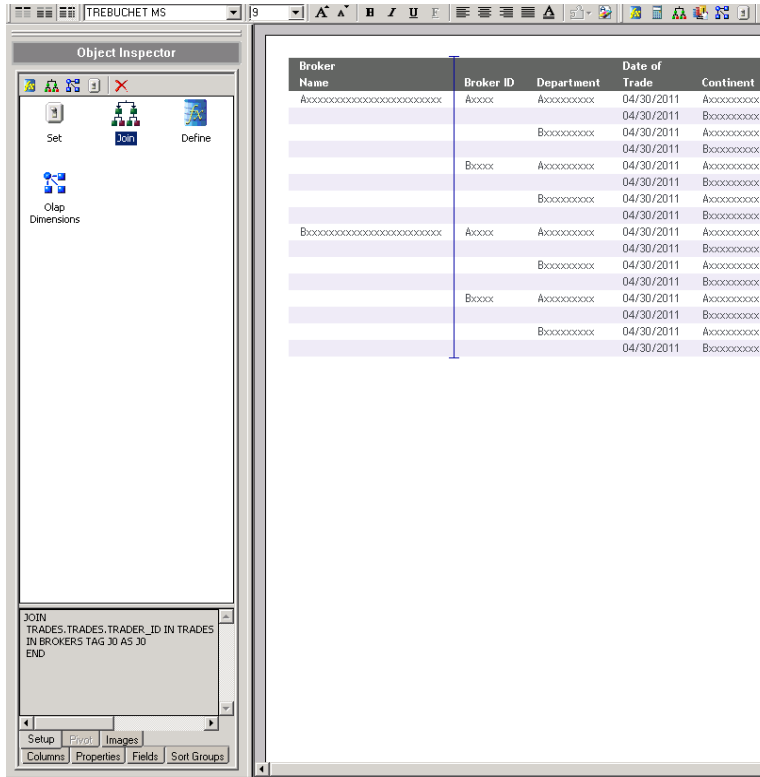
By Lisa Scipio

Following up on my article in the February-March 2011 edition of WebFOCUS Newsletter, "Introducing Guided Report Forms," I now will describe the steps involved in the creation of the Guided Report Form. For detailed explanations of Guided Report Form functions, please refer back to the earlier article.

An advantage of using the Guided Report Form feature is that it provides ease of use to both the developer and the end user. Multiple reports and charts can be created from a single request, and prefixes such as `WHERE`, `DEFINE`, `JOIN`, `SET` and `OLAP Dimensions` are repeated in each frame within a Report Set. Furthermore, since the reports are highly parameterized, the user can control which columns are displayed in each report.

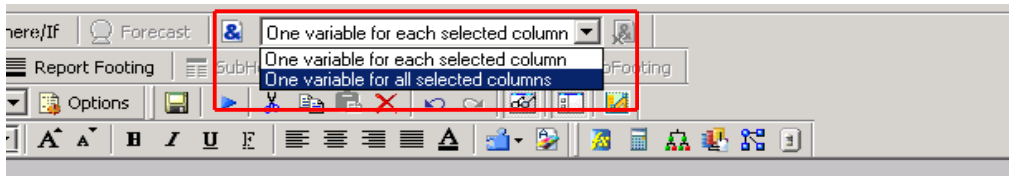
As of WebFOCUS 7.7.02, the option to create a new Guided Report Form is available in the Local Projects, Data Servers and Managed Reporting environments of Developer Studio. The report in the first frame should contain the maximum number of fields. The ability to create procedures using constructs that come before the report request, like `SETs` and

JOINS, without using the Procedure Viewer is turned on in release 7.7.02. **Screen 1** shows a JOIN created in this manner.



Screen 1: JOIN created from the Report Painter invoked from HTML Composer.

To create a variable of field names, we group similar field types by choosing the option to create “One variable for each selected column” or “One variable for all selected columns.” Our selection is saved when we click on the ampersand icon to the left of the combo box. (See **Screen 2.**)



Broker Name	Broker ID	Department	Date of Trade	Continent	Region
Axxxxxxxxxxxxxxxxxxxxxxxx	Axxxx	Axxxxxxxx	04/30/2011	Axxxxxxxx	Axxxxxxxxxxxxxxxx
		Bxxxxxxxx	04/30/2011	Bxxxxxxxx	Bxxxxxxxxxxxxxxxx
		Bxxxxxxxx	04/30/2011	Axxxxxxxx	Axxxxxxxxxxxxxxxx

Screen 2: Options available for creating field variables.

Display options for report output can also be set as variables.

To set Sort Field Options as variables, choose Options from the right-click context menu, select the Sort then the Sorting tab. Sort order, sort limit or total may be set as variables. Variables may be enabled for the following options from the Actions tab: underline, skip line, no split, fold line, page break and restart page numbering options. For all field types, the invisible and skip line options may be set to "Variable" from the General tab of the Field Properties dialog.

Please note that if the option that groups multiple columns in one multi-select control is used, the action is applied to the last value or column selected. For example, if the invisible option is applied to detail fields displayed in a multi-select control, the last value selected is invisible. This is normal FOCUS behavior.

After saving the Guided Report and closing Report Painter, the New Parameters dialog in HTML Composer displays all parameters created; they are positioned to reflect their relationship.

In addition to reports, charts can be created by selecting the desired graph tool from the right-click context menu of a frame. Since reports and charts in a Report Set of a Guided Report Form use the same data files, the Graph Assistant or Advanced Graph Assistant opens immediately to display the data fields for the database used in the first report or chart. No prompt to select a data file appears.

A non-tabbed Report Set can be changed to a tabbed Report Set by selecting DOCUMENT from the Properties Sheet dropdown list, then changing the value for the "Tabbed Layout" attribute to "Yes." Use `app1.css` may also be enabled at the document level; `app1.css` contains additional or alternative styling and applies these modifications to the template when activated.

At runtime, the user customizes his reports and charts by selecting the desired columns, option values and column values (**Screen 3**).

BRANDING LOGO HERE

Report set title

Please select sort field(s) Underline: Skip Line: Page Break:
 [Broker Name] [On] [On] [On]

No Split: [On]

Please select sort field(s) Invisible: [On]

Please select sort field(s) Please select detail field(s) Invisible: Skip Line:
 [Broker Name] [On] [On]

Continent Type of Instrument Buy / Sell Amount Region Country Holder

Broker Name	Broker Name	Region	Country
ALBERT E. STAPLEY	ALBERT E. STAPLEY	NORTH AMERICA	UNITED STATES
		FAR EAST	JAPAN
		NORTH AMERICA	UNITED STATES
		CENTRAL AMERICA	GUATEMALA
		EAST EUROPE	CZECH, REP
		CENTRAL AMERICA	GUATEMALA
		EAST EUROPE	CZECH, REP
		WEST EUROPE	FRANCE
		WEST EUROPE	ENGLAND

1B
800M

Contains commands for working with the selected items.

Screen 3: An example of an executed Guided Report Form.

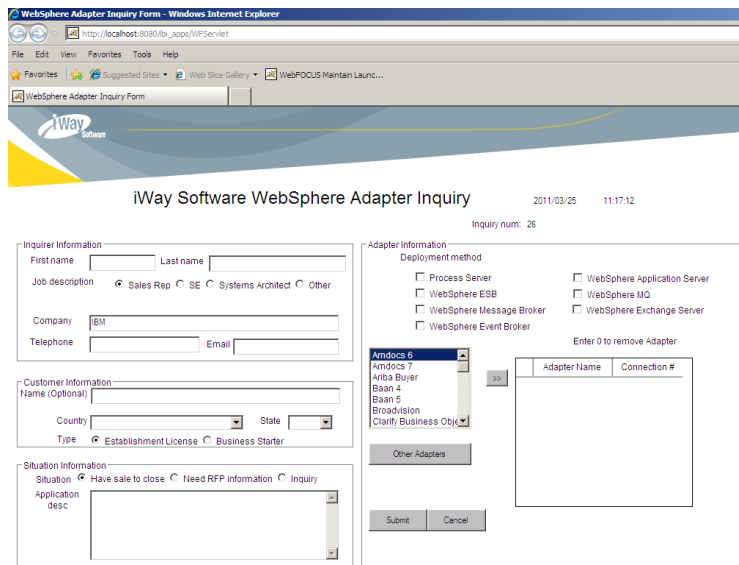
For further information on the Guided Report Form, please refer to "Getting Started with Guided Reports" in the "Designing a User Interface for a Web Application With the HTML Composer" documentation manual.

Maintain and Group Boxes

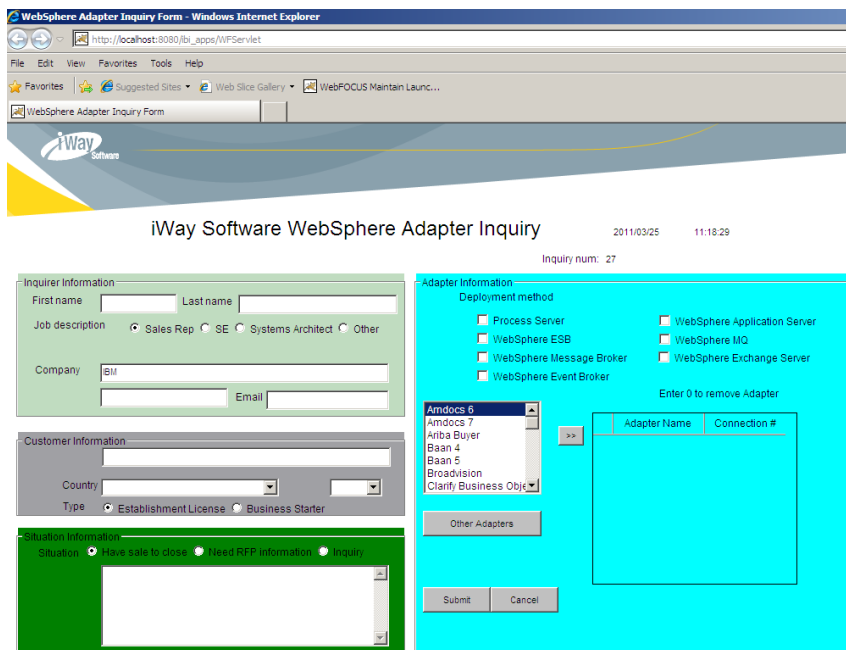
By Mark Derwin

When building a form in Maintain, you may want to divide the screen into different sections.

While you can use the line tool to separate fields and objects, I prefer to use the Group Box. The Group Box lets you partition your form into logical sections, as illustrated on **Screen 1**. But what if you want your sections to really stand out, as in **Screen 2**? The option to change the background color of the Group Box is not one of its properties.



Screen 1



Screen 2

That's not a problem. We can easily change the background color of the Group Box in the Maintain code.

In order to change a property of an object, you need to know the form name and the object name. You also must issue a `WINFORM SHOW_INACTIVE`. Any time you want to operate on a form before it is displayed, you must issue that command.

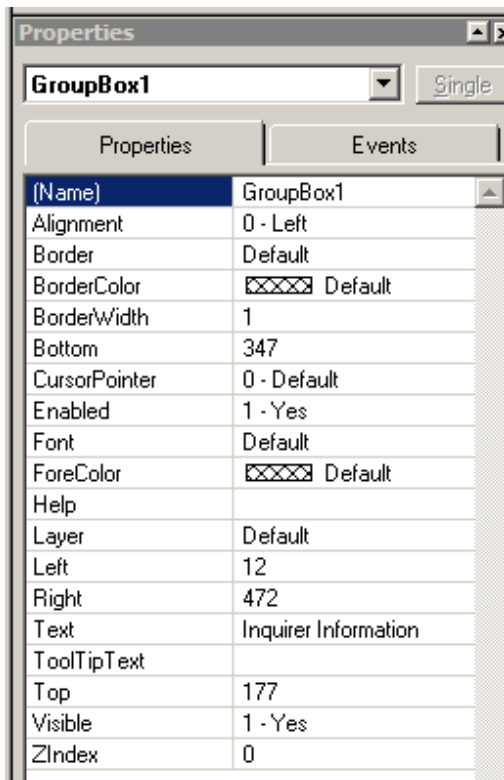
In the following example, my form is named `Form1` and my group boxes are 1 – 4 sequentially. So, in this case I use:

```
Winform show_inactive Form1
Form1.GroupBox1.SetBackColor(192, 220, 192)
Form1.GroupBox2.SetBackColor(0, 225, 255)
Form1.GroupBox3.SetBackColor(160, 160, 164)
Form1.GroupBox4.SetBackColor(0, 128, 0)
Winform show Form1
```

Before the form is displayed, I set the colors for all of the boxes. In this example there are spaces between the boxes. You could create them so the entire screen is two (or more) colors and really make the sections stand out.

There is one important thing to keep in mind when using Group Boxes. Sometimes, when running an application, you will not be able to enter anything into your edit boxes, or select anything from your lists.

If this happens, and the objects are surrounded by a Group Box, there is an easy fix. Open your application in the Maintain Development Environment and bring up the form. Click on the Group Box and look at the Zindex Property (**Screen 3**).



Screen 3

Zindex tells the form in what order to display the objects. Objects with a higher Zindex are displayed on top of objects with a lower Zindex. If the Zindex for the Group Box is higher than the objects it is displaying, you will not be able to enter into or select from those objects.

The best thing to do is change the Zindex of the Group Box to zero. This way the objects in the box will always be "in front" of the box and you won't have to worry about not being able to enter data.

Group Boxes are great to break up your form into logical sections. Use this technique to make your sections stand out even more.

Bye-Bye Internet Explorer 6

By Susan Trommer

Earlier this year Microsoft created “The Internet Explorer 6 Countdown” website <http://www.theie6countdown.com/default.aspx> to and urge its global user community to move off of the 10-year-old Internet Explorer 6 (IE6) browser.

The site shows a map of the world with IE6 percentage usage per country and region, which as of June 2011 was globally at 10.9 percent. It also contains resource sections, including “Join the Cause,” “Educate Others” and “Tell Your Friends,” at the bottom of the page to assist organizations, developers and individual users in supporting this mission.

The “Join the Cause” section provides developers with code they can copy and paste to display a banner on their website only to IE6 users informing them that they are using an outdated version of the browser. Microsoft also informs developers on this page that they can email their corporate logo to display in this section of Microsoft’s site to recognize participating websites.

The “Educate Others” section is very direct in stating the benefits of upgrading to a newer IE browser version. Those benefits include improved speed, tabbed browsing and better privacy settings. If that isn’t enough to persuade you, select the link to view the browser comparison pages. The comparison page provides information for Internet Explorer browser versions (IE9, IE8, and IE7) in one tab and Internet Explorer and other browsers (Firefox 4, Chrome 10, IE9) in a second tab.

Now that I’ve done my share to support the cause, I’ll turn the focus to WebFOCUS. Information on WebFOCUS browser support is available on the Information Builders Technical Support site page “Web Browser Support for WebFOCUS” http://techsupport.informationbuilders.com/tech/wbf/wbf_tmo_027.html.

This page is reviewed and updated for each WebFOCUS release as well as on an as-needed basis. The WebFOCUS Installation Manuals also provide browser support information in the End User Machine Requirements topic with a reference to the *Web Browser Support for WebFOCUS* page that has the most up-to-date information. Information on known issues for a specific browser vendor and/or version is available in the *WebFOCUS Release Notes* document available for each release.

Within the “Web Browser Support for WebFOCUS” page, the Internet Explorer Browser Version Support section provides WebFOCUS release certification and support information by IE browser version. Further down the page, you’ll find a table with rows for each WebFOCUS release that provides certification and support information for IE, Firefox, Safari, and Chrome browsers, as well as Adobe Acrobat Reader and Java Virtual Machine (JVM) support.

The browser certification plan for the WebFOCUS Release 7.7.03 is as follows:

- IE browser versions 9 (32bit), 8 and 7.
**Information Builders adheres to the support schedule and recommendations of third-party support vendors so IE6 SP3 will not be listed as a supported browser.*
- Firefox 3.6.16.
**Firefox 4 is actively being evaluated but testing will not be completed for Release 7.7.03.*
- Areas currently listed as supported with Safari and Chrome with the latest maintenance version.

Certification for future WebFOCUS releases will continue to evaluate new browser and third-party software versions as well as expand testing efforts based on industry trends and the business requirements of our customers. Please contact Customer Support to provide information on issues found during your evaluation of the latest browser versions and communicate your requests for browser certification.

Share What You Know through Focal Point

Focal Point is Information Builders' online community for developers where more than 5,000 members come together to share ideas, ask questions and resolve technical issues. Focal Point is an interactive network with members from almost every profession and industry, collaborating on solutions and sharing tips.

As part of that interaction, Focal Point members occasionally post articles with plenty of technical information that other members are bound to find useful. The following links will take you to some of these member-submitted articles. If you have any tips and techniques to share, please contact Kathleen Butler (Kathleen_Butler@ibi.com). These tips and techniques also typically are shared at Information Builders' local User Groups.

Large or multiple Compound Reports with NOBREAK, poor performance
<http://www.informationbuilders.com/support/developers/ReportsNOBREAK>
By Warren Hinchliffe, Link Market Services

JavaScript to Validate Date Entry
<<http://www.informationbuilders.com/support/developers/javascript-validate-date-entry>>
By Tom Flynn, SunGard Higher Education, and Jeralee Seaburn, Hercules Tire

Converting and Shortening an Integer to Alphanumeric
<<http://www.informationbuilders.com/support/developers/IntToAlphaNum>>
by Tom Flynn, SunGard Higher Education; Gerard van der Paal, BI-Solutions; and John Lewis, Partner Intelligence

WebFOCUS Excel Templates
<http://www.informationbuilders.com/support/developers/ExcelTemplate>>

By Anthony Alsford, now working for Information Builders

McGyver Dynamic Subtotals/SubFoots

<<http://www.informationbuilders.com/support/developers/McGyverDynamic>>

by David Smith, Brinker International

Open Drilldown in a New Tab

<<http://www.informationbuilders.com/support/developers/drilldownnewtab>>

By Marilyn Patchen Beyer, Minnesota Department of Education