ALLTRIM(String)

Removes both leading and trailing spaces from a string. The string may be derived from any valid xBase expression.

Example:

ALLTRIM("Company Name   ") returns "Company Name".

AT(SearchString, TargetString)

Determine whether a search string is contained within a target.   If found, the function returns the position of the search string within the target string (relative to 1). If not found, the function returns 0 (zero).

Example:

AT("Computer", "Telcon Computer") returns 8.

CHR(Val)

Converts a decimal value to its ASCII equivalent.

Example:

CHR(84) returns 'T'

CTOD(String)

Converts a character string into an xBase date.   The string must be formatted according to the settings of SetDateFormat and SetCentury.

Example:

```
CTOD("12/31/94")
```

DATE()

Returns the system date (today).   Use the DTOC() function to convert the date to a string format .

Example:

If the current date is "3/11/97" then DATE() function would return the date "3/11/97".

DAY(DateField)

Returns the day portion of an xBase date as an integer.

Example:

On "3/11/97" the function DAY() would return the integer 11.

DELETED()

Returns True if the database record is deleted and False if not deleted.

DTOC(DateField)

Converts an xBase date into a character string.

Example:

If the database date field called "DateAdded"   contained March 11, 1997, DTOC(DateAdded) would return '03/11/1997'.

DTOS(DateField)

Converts an xBase date into a string formatted according to standard xBase storage conventions (CCYYMMDD).

Example:

For example, March 11, 1997 would be returned as '19970311'.

EMPTY(Field)

Reports the empty status of any xBase field.   Character and date fields are empty if they consist entirely of spaces.   Numeric fields are empty if they evaluate to zero.   Logical fields are empty if they evaluate to False.

Memo fields that contain no reference to a memo block in the associated memo file are empty.

Example:

```
IIF( Empty( NameField ), '',   'Name:   '+NameField+'<br>' )
```

This example would print a line of HTML to display a name only if the "Name" field has data.

IF(Logical, True Result, False Result)

This is the immediate if function.   If the Logical expression is true, return the True result, otherwise return the False result.   The types of the True Result and the False Result must be the same (i.e., both numeric, or both strings, etc.)   The logical expression must of course evaluate as True or False.

Example:

```
IF( LEN(ALLTRIM(company))>0,   company+"<br>",   "")
```

In this example if the field company has text the company title will appear in the HTML, otherwise no line will be added to the HTML.

IIF(Logical, True Result, False Result)

Identical to the function IF().

This is the immediate if function. If the Logical expression is true, return the True result, otherwise return the False result. The types of the True Result and the False Result must be the same (i.e., both numeric, or both strings, etc.) The logical expression must of course evaluate as True or False.

Example:

```
IIF( LEN(ALLTRIM(company))>0,   company+"<br>",   "")
```

In this example if the field company has text the company title will appear in the HTML, otherwise no line will be added to the HTML.

LEFT(String, Length)

Returns the leftmost characters of the expression for the defined length.

Example:

LEFT("Bluestem", 4) returns 'Blue'.

LEN(Expression)

Returns the length of the expression result as an integer.

Example:

LEN("Bluestem") returns 8.

LOWER(String)

Converts the string expression into lower case.

Example:

LOWER("COMPANY NAME")   returns   "company name".

MONTH(DateField)

Returns the month portion of an xBase date as an integer.

Example:

On "3/11/97" the function MONTH(DATE()) returns the integer 3.

PADC(String, Length, Character)

Centers the passed string between a number of the passed character to make the string the specified length.

Example:

PADC("Bluestem", 14 ,"-")   returns "---Bluestem---".

PADL(String, Length, Character)

Pads the passed string to the specified length with the specified characters.   If the string is longer than the value specified by Length, the string is truncated to this length.

Example:

PADL("$150.00", 10, " ")   returns " $150.00".

PADR(String, Length, Character)

Pads the passed string to the specified length using the specified character.   If the string is longer than the value specified by Length, the string is truncated to this length.

Example:

PADR("Company", 8, " ")+"Name"   returns "Company Name".

RAT(SearchString, TargetString)

Determine whether a search string is contained within a target, starting from the right side of the target string.   If found, the function returns the position of the search string within the target string (relative to 1). If not found, the function returns 0 (zero).

Example:

RAT("pc", "pc2pc.com") returns 4.

RECCOUNT()

Returns the number of records in the table as a long integer.

RECNO()

Returns the current physical record number as a long integer.

RIGHT(String, Length)

Returns the rightmost characters of the expression for the defined length.

Example:

RIGHT("pc2pc.com", 3) returns 'com'.

SPACE(Length)

Returns a string consisting entirely of spaces for the defined length.

Example:

SPACE(10)+'Text'   returns '   Text'.

STOD(String)

The inverse of   DTOS(). STOD() converts a string formatted according to standard xBase storage conventions (CCYYMMDD) to an xBase Date.

STR(Number, Length, Decimals)

Converts a number into a right justified string with decimals digits following the decimal point.

If the decimals parameter is omitted, the function defaults to zero decimals.   If the length parameter is omitted as well, the length of the result is 10 characters.

Example:

If the field Number contains the value 100 the function STR(Number,5,0) returns the string " 100".

STRZERO(Number, Length, Decimals)

Converts a number into a, zero-padded right justified string with decimals digits following the decimal point. The total length of the string is defined by the length parameter. If the decimals parameter is omitted, the function defaults to zero decimals. If the length parameter is omitted as well, the length of the result is 10 characters.

Example:

STRZERO(1234, 10, 2) returns '0001234.00'

SUBSTR(String, Start, Length)

Returns a portion of the string expression starting at the defined start location for the defined length..

Example:

SUBSTR('Bluestem Technologies', 5, 4) returns 'stem'.

TIME()

Returns the system time as a string in the form HH:MM:SS.

TRANSFORM(Expression, Picture)

Transform converts strings and numeric values into formatted character strings. The function transforms the result of the first expression in accordance with the second picture string.

The picture string is made up of two parts. The first part is the Function string and it is optional for both strings and numeric values (as long as the second Template string is present).

A character string transformation picture may consist of only a Function string or only a Template or both.

A numeric picture must contain a Template string; the Function string is optional.

A logical value must contain only a Template string with Template characters L or Y.

The Function string consists of a leading @ character followed by one or more formatting characters. If the Function string is present, the @ character must be the first character in the picture string with its formatting characters immediately following and it may not contain spaces.

If a Template string exists as well, it follows the Function string. A single space separates the Function string and the Template string.

Function string characters allowed for numeric values are:

B left justify;
C display CR after positive numbers;
X display DR after negative numbers;
Z blank a zero value;
(encloses negative numbers in parentheses.

Function string characters allowed for strings are:

R inserts unassigned template characters;
! converts all alpha characters to upper case.

The @R Function requires a Template; the ! Function does not.

The Template string describes the format on a character by character basis. The Template string is made up of special characters which have specific results and optional unassigned characters which either replace characters or are inserted in the formatted string depending upon the absence or presence of the @R Function string.

Template assigned characters are as follows:

A,N,X,(,# are place holders and are interchangeable;
L displays logical values as T or F;
Y displays logical values as Y or N;
! converts the corresponding character to upper case;
, (comma) or a space (in Europe) in a numeric template separate

the elements of a number;
. (period) or , (comma - in Europe) in a numeric template specify
the decimal position;
* fills leading spaces with asterisks in a numeric template;
$ as the leading character in a numeric template results in a
floating dollar sign being placed in front of the
formatted number.

Example: Where "phone" is a character field holding a phone number with no formatting characters.

TRANSFORM(Phone, "@R (###) ###-####");

returns '(909) 699-6776'. If the formatting characters were actually present in the field, the "@R" function would be omitted

For numeric fields, TRANSFORM(123456.78, "$9,999,999.99") returns ' $123,456.78'.

TRIM(String)

Removes trailing spaces from the string expression.

Example:

TRIM('Bluestem   ') returns 'Bluestem'.

UPPER(String)

Converts the string expression into upper case.   Character fields used in index expressions should always be converted to upper case to insure correct collating sequence.

Example:

Upper('sample') returns 'SAMPLE'.

VAL(String)

Converts a string of numeric characters into its equivalent numeric value.   The conversion stops at the first non-numeric character encountered (or the end of the string).

Example:

VAL("2001Years") returns a value of 2001.

YEAR(DateField)

Returns the year portion of an xBase date as an integer.

xBase Functions

dB Page Builder allows users to insert xBase compatible functions into a HTML template. These functions can be used to format database data in the generated HTML. Functions can also be used to conditionally print lines of HTML.

xBase Functions Supported by dB Page Builder

ALLTRIM	Trims both leading and trailing spaces from a string.
AT	Determines if a search string is contained within a string.
CHR	Returns ASCII character represented by a integer.
CTOD	Converts date string to date field.
DATE	Returns todays day in date format.
DAY	Returns day portion of date as integer.
DELETED	Returns true if DBF record is deleted, false if not deleted.
DTOC	Converts date to date string ("YYYYMMDD").
DTOS	Converts date to date string ("MM/DD/YY").
EMPTY	Return True if field is empty.
IF	Evaluates expression and returns value.
IIF	Evaluates expression and returns value.
LEFT	Returns leftmost characters.
LEN	Returns length of string.
LOWER	Converts string to lowercase.
MONTH	Returns month portion of date as integer.
PADC	Pad left and right of string with selected character.
PADL	Pad left of string with selected character.
PADR	Pad right of string with selected character.
RAT	Determines if search string is contained in target.
RECCOUNT	Returns record count as integer.
RECNO	Returns record number as integer.
RIGHT	Returns rightmost characters.
SPACE	Returns spaces.
STOD	Converts date string ("YYYYMMDD") to date.
STR	Converts numeric to string.
STRZERO	Converts numeric to string and pads left with zeros.
SUBSTR	Returns a substring of string.
TIME	Returns time of day as string.
TRANSFORM	Transforms (formats) a string.
TRIM	Removes spaces from right side of string.
UPPER	Converts string to uppercase.
VAL	Converts string of numeric characters to numeric equivalent.
YEAR	Returns year portion of date as integer.

dB Page Builder

Shareware Limitations

The shareware version of dB Page Builder will only generate HTML files for the first 15 database records.   The full version of dB Page Builder will generate HTML files for every database record.   To access the full power of dB Page Builder please contact Bluestem Technologies.

Bluestem Technologies
1104 E. 25th Terrace
Lawrence, KS   66046

Phone: (913) 865-3804
E-mail: bluestem@pchelponline.com
Web: http://www.pchelponline.com/bluestem

Purchase dB Page Builder

About dB Page Builder

Background

Many web site development projects involve placing valuable information stored in databases on web pages. In other cases, developers are manually creating web pages with information that would be easier to maintain in a database. Although there are a wide variety of database server options there are relatively few applications to generate permanent web pages from a database. For many developers, the best available option has been to create a HTML template and then "fill in the blanks" to generate hundreds or thousands of web pages. To make matters worse, some of the most powerful and commonly used web development suites are not well suited for maintaining web sites with thousands of pages.

Adding Database Fields to HTML

DB Page Builder allows the user to open an HTML file much like any other HTML editing tool. We recommend that users design a prototype web page using their favorite HTML editor. Then the user can then open a xBase compatible database (dBase, Clipper, and FoxPro) with data to be placed in web pages. DB Page Builder allows the user to insert xBase compatible expressions into their HTML script. By surrounding expressions with double brackets (e.g. [[expression]]) the user can create an HTML template that dB Page Builder will use to generate hundreds or thousands of HTML files. The software supports a wide variety of xBase functions that allow the user to format field data, insert date and time information, and create IF statements that allow for conditional HTML output. Fields can be evaluated to determine if HTML code should be added to each HTML output file, thereby allowing for database driven HTML customization. DB Page Builder allows users to add and format data from character, numeric, date, logical, and memo fields.

HTML Template Sections

With dB Page Builder, the user can define sections of HTML to be added to the generated HTML file. The primary HTML sections are the header, body, and footer. The HTML placed in the header section will be added to the top of each generated HTML file. Similarly, HTML placed in the footer section will appear at the bottom of each generated HTML file. The body section of the HTML template will be repeated for each record of the selected database. Users will generally place a combination of HTML and database fields in the body section to produce web pages with a list, table, or repeated sections of database information.

Grouping Data

DB Page Builder also supports creating group order expressions that allow the database records to be separated into categories. Users can define up to three group order expressions. A group order expression is an instruction to dB Page Builder to interrupt printing the body section when the value of one or more database fields changes. Users can define special header and footer sections of HTML to be printed before and after a group order value changes value. The user can also instruct dB Page Builder to create a new web page when the group order value changes. Group order expressions can be used to separate product categories, create separate web pages for each letter of the alphabet, or to separate items by date. Advanced users can create databases with fields specifically designed to separate data effectively.

Creating Outlines

Many web page developers find the need to print the information in a database twice on a single web page. A common example is the use of a content outline that appears at the top of a web page with hyperlinks to content that appears later on the page. Users of dB Page Builder can create a special outline body section of HTML to be printed for each database record within a group order value.

Generating HTML

Once the HTML template has been created the user can then build HTML files. DB Page Builder automatically tests each xBase expression and reports any errors. DB Page builder then begins generating HTML files using database data. DB Page Builder allows the user to select a naming convention for the generated HTML files. Database fields can optionally be used to name output files. DB Page Builder will create HTML with inserted database information at a rate of approximately 2500 records per minute.

What Can I Build With dB Page Builder

dB Page Builder is designed to allow great flexibility in web page generation without sacrificing speed and performance. If you are familiar with HTML and xBase functions, generating HTML templates will be a very easy task. Luckily for those without xBase experience, xBase functions are easy to learn and intuitive to use so learning times will be short. dB Page Builder makes converting existing databases to web sites simple and fast. Inventory, personnel, and other databases can be converted to an HTML in minutes. Even if your data is not currently in a database, dB Page Builder may convince you it should be.

Using the Speedbar

The speedbar provides quick access to dB Page Builder functions.

Speedbar Buttons

Create a new HTML template.
Open HTML template.
Save HTML Template.
Close HTML Template.
Open Project File
Save Project File
Cut highlighted text.
Copy highlighted text.
Paste highlighted text.
Open database.
Close database.
View list of database fields.
View database data.
Set Group Order
Insert Template Section.
Insert xBase expression brackets.
Insert memo field brackets.
Set export location and export HTML file names.
Check HTML script for errors.
Generate HTML files.
Open web browser and display master index.

xBase Expressions

Operators

Evaluation Order

Expression Length
The maximum length of any xBase expression is 255 characters.

Embedded Literal Values
Embedded literal values must be delimited with double quotes.

Examples:

```
[[ '<a href=" '+HTMLFILE+' ">'+COMPANY+'</a><br>' ]]
[[ CTOD("01/02/97") ]]
```

Operators

Operators in dB Page Builder are consistent with other xBase applications.

Alias Operator
dB Page Builder does not require the use of an alias name.   Field names are acceptable by themselves.

String Operators
+   Joins two strings.   Trailing spaces in the strings are placed at the end of each string.
-   Joins two strings.   Trailing spaces are removed from the first string and placed at the end of the second string.

Numeric Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponentiation
**	Exponentiation

Relational Operators

=	Equal to
==	Exactly equal to
<>	Not equal to
#	Not equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than equal to
>=	Greater than equal to
$	Is contained in

Logical Operators

.AND.	Both expressions are true
.OR.	Either expression is true
.NOT.	Either expression is false

(Please note that all logical operators are surrounded by periods.)

Evaluation Order

When an xBase expression contains more than one type of operator the following evaluation order applies:

Field Name (alias name)
String
Numeric
Relational
Logical

Operators contained in parentheses will be evaluated first. Nested parentheses are evaluated starting with the innermost set. Following the evaluation of operators contained in parentheses, operators are evaluated from left to right.

Numeric Operators
Numeric operators are evaluated according to accepted arithmetic principles. Numeric evaluation order is as follows:

operators contained in parentheses
exponentiation
multiplication and division
addition and subtraction

Examples of Using Parentheses
1+2*3+4 = 11
(1+2)*3+4 = 13
(1+2)*(3+4) = 21

Examples of Using Logical Operators
True .AND. True = True
True .AND. False = False
True .AND. .NOT. False = True
True .OR. False = True

Purchasing dB Page Builder

Thank you for considering the purchase of dB Page Builder. Purchasing the full version of dB Page Builder will allow you to generate HTML files from any size database. In addition, you will automatically receive product updates and announcements for upcoming versions.

The introductory price of dB Page Builder is $99. You may mail your check or money order written in US currency only, directly to the address below, payable to Bluestem Technologies. To order dB Page Builder online access our web page at http://www.pchelponline.com/bluestem.

Bluestem Technologies
1104 E. 25th Terrace
Lawrence, KS   66046

Phone: (913) 865-3804
e-mail: bluestem@pchelponline.com
Web: http://www.pchelponline.com/bluestem

Single User License Pricing:

dB Page Builder: $99
dB Web   Builder: $99
dB Page Builder & dB Web Builder Suite: $149

Please contact Bluestem Technologies to discuss other licensing options.

Creating Tables

In this project dB Page Builder will be used to create HTML tables from a database. The following steps will take you through the entire process of building HTML files.

Step #1 Open the VideoTbl.htm template file.
To load the Video Table template click on the open file button. Select the "VideoTbl.htm" file located in the Tables subdirectory of the Tutorial directory. The VideoTbl.htm file contains a template to create a table with video catalog information.

View VideoTbl.htm Source

Step #2 Open the Video.dbf database.

dB Page Builder supports Clipper, DBase, FoxPro, and Successware style DBF databases. To open the Video.dbf database click on the open database button. Select the "Video.dbf" database located in the Tables subdirectory of the Tutorial directory.

Step #3 Viewing Database Data & Database Field List
To view the contents of the Video database click on the View Database button. A spreadsheet style or browse listing of the database will then appear. This screen will not allow users to edit the database.
Click on the View Field List button to display a list of database fields. This screen indicates the name and type of every field in the open database. To insert a field name into the HTML template double click on the field name located on this screen.

Step #4 Set Export Location and File Names
Click on the Export Location button to display the Export Location & HTML File Names dialog. Using the dialog set the export location to the Tables subdirectory of the Tutorial directory. Leave the HTML File Names set to Sequentially Numbered Files. Close the dialog when finished.

Step #5 Set Group Order
Click on the Set Group Order button to produce the Group Order dialog. Click on the Use Group checkbox for Group Order #1 to active the group. Click on the New Page checkbox to instruct dB Page Builder to generate a new page when the value of the group expression changes The Group Expression box allows the user to enter a expression to group the data by. The Group Expression can be any valid xBase expression. Usually the group order expression will be the name of a database field. When the value of the field changes the group header and group footer will be printed. For the Video Catalog project we want to create a new web page for each video category. To accomplish this, type the following field name into the Group Expression box: Category.

Step #6 Check the HTML Template
Click on the Check HTML Template button to perform an analysis of all xBase expressions. If dB Page Builder reports an error, correct the indicated line or index.

Step #7 Generate HTML Files
Click on the Build HTML button to generate web pages and indexes.

View Generated REC00001.htm

View Generated REC00007.htm

Step #8 View the Results

Click on the Browser button to open the web browser associated with HTML files. A web browser must be associated for this feature to work correctly. When dB Page Builder generates HTML it automatically creates a master index of its completed work. The browser feature will open the master.htm file located in the currently selected export directory. The web page includes links to each HTML file created. Take a look at the 2 web pages generated by the tutorial. Now its time to start thinking about what dB Page Builder can do with your databases.

Return to Tutorial

Creating a Video Catalog

This project is designed to introduce you to the process of creating HTML templates. The video database uses a variety of field types that will be used to demonstrate the power of using xBase functions to format output data. The following steps will take you through the entire process of building HTML files.

Step #1 Open the VideoCat.htm template file.
To load the Video Catalog template click on the open file button. Select the "VideoCat.htm" file located in the VideoCat subdirectory of the Tutorial directory. The VideoCat.htm file contains a template to create a simple video catalog.

View VideoCat.htm Source

xBase Expressions used in the VideoCat.htm Template

Expression: <H1>[[CATEGORY]]</H1>
Result: The character field "Category" is inserted into the HTML. Since this expression appears in the group header it appears only once per category.

Expression: <H3>[[TITLE]]</H3>
Result: The character field "Title" is inserted into the HTML.

Expression: {{DESC}}
Result: The memo field "Desc" is inserted into the HTML.

Expression: Starring: {{STARRING}}<br>
Result: The memo field "Starring" is inserted into the HTML.

Expression: Director: [[DIRECTOR]]<br>
Result: The character field "Director" is inserted into the HTML.

Expression: Released: [[Str(Year(RELEASED),4)]]<br>
Result: The year portion of the date field "Released" is converted into a character string and then inserted into the HTML.

Expression: Price: [[Transform(PRICE, "$9,999,999.99")]]<br>
Result: The numeric field "Price" is formatted to reflect currency and then inserted into the HTML.

Expression: [[IF(Empty(AWARDS), '', 'Awards: '+Alltrim(Awards)+'<br>')]]
Result: If the character field "Awards" is empty nothing is added to the HTML. If the character field "Awards" is not empty the field and associated HTML is inserted into the output file.

Expression: [[IF(RATING='R', '<IMG SRC="R.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="R">', '')]]
Result: If the character field "Rating" is equal to "R" then the HTML to display the R.GIF file is added.

Expression: [[IF(RATING='PG', '<IMG SRC="PG.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="PG">', '')]] Result: If the character field "Rating" is equal to "PG" then the HTML to display the PG.GIF file is added.

Expression: Last Updated: [[DTOC(Date())]]<br>
Result: The current data in a character string format is inserted into the HTML.

Step #2 Open the Video.dbf database.

dB Page Builder supports Clipper, DBase, FoxPro, and Successware style DBF databases. To open the Video.dbf database click on the open database button. Select the "Video.dbf" database located in the VideoCat subdirectory of the Tutorial directory.

Step #3 Viewing Database Data & Database Field List

To view the contents of the Video database click on the View Database button. A spreadsheet style or browse listing of the database will then appear. This screen will not allow users to edit the database.

Click on the View Field List button to display a list of database fields. This screen indicates the name and type of every field in the open database. To insert a field name into the HTML template double click on the field name located on this screen.

Step #4 Set Export Location and File Names

Click on the Export Location button to display the Export Location & HTML File Names dialog. Using the dialog set the export location to the VideoCat subdirectory of the Tutorial directory. Leave the HTML File Names set to Sequentially Numbered Files. Close the dialog when finished.

Step #5 Set Group Order

Click on the Set Group Order button to produce the Group Order dialog. Click on the Use Group checkbox for Group Order #1 to active the group. Click on the New Page checkbox to instruct dB Page Builder to generate a new page when the value of the group expression changes The Group Expression box allows the user to enter a expression to group the data by. The Group Expression can be any valid xBase expression. Usually the group order expression will be the name of a database field. When the value of the field changes the group header and group footer will be printed. For the Video Catalog project we want to create a new web page for each video category. To accomplish this, type the following field name into the Group Expression box:

Category

Step #6 Check the HTML Template

Click on the Check HTML Template button to perform an analysis of all xBase expressions. If dB Page Builder reports an error, correct the indicated line or index.

Step #7 Generate HTML Files

Click on the Build HTML button to generate web pages and indexes. Don't blink because dB Page Builder will generate the web pages in under a second. Registered users can reasonably expect to process over 2,000 database records a minute on a Pentium computer.

View Generated REC00001.htm
View Generated REC00007.htm

Step #8 View the Results

Click on the Browser button to open the web browser associated with HTML files. A web browser must be associated for this feature to work correctly. When dB Page Builder generates HTML it automatically creates a master index of its completed work. The browser feature will open the master.htm file located in the currently selected export directory. The web page includes links to each HTML file created. Take a look at the 2 web pages generated by the tutorial. Now its time to start thinking about what dB Page Builder can do with your databases.

Return to Tutorial

Tutorial

This tutorial is designed to introduce you to some of the capabilities of dB Page Builder. Three sample projects have been created to demonstrate some of dB Page Builders uses. The tutorial is designed to demonstrate how xBase expressions can be used to insert database data into HTML files and provide a variety of data formatting options.

dB Page Builder Basics

Template Sections
Creating HTML Templates
Set Group Order

Example Projects

Project #1: Creating a Video Catalog
Project #2: Creating Tables
Project #3: Using Outlines

Creating HTML Templates

Overview

dB Page Builder uses a slightly modified version of HTML scripting to generate web pages from a database. dB Page Builder uses two varieties of double brackets to insert xBase compatible expressions and memo fields into a standard HTML script. When dB Page Builder builds HTML files the xBase expressions are evaluated and the resulting character strings are inserted into the generated HTML files. When a memo field is evaluated the contents of the memo field are added to the HTML file. For programmers familiar with xBase applications including dBase, FoxPro, and Clipper the supported xBase functions will be very familiar. Users unfamiliar with xBase conventions should review the xBase functions list in this help file. Learning to create xBase expressions should be a relatively easy and painless process.

Designing Web Pages

dB Page Builder is not intended to compete with the long list of web page design tools. dB Page Builder is designed to take a well designed web page to generate 1000 more like it using database data. Create a prototype web page based on the information stored in one record of the source database. Once the page is complete open the HTML file in dB Page Builder. Use xBase expressions and memo fields to replace the sample text in your prototype web page. dB Page Builder can then create a web page for each record in your database. If you decide to redesign your web page, dB Page Builder will allow you to regenerate all of the web pages from the source database far faster than manually editing each page.

Rules for xBase Expressions

1. Each xBase expression must be surrounded by double brackets (i.e. [[Date()]]).
2. Only one xBase expression ([[ ]]) or memo field ({{ }}) is allowed per line.
3. xBase expressions have a 255 character length limit.
4. The entire xBase expression must appear on a single line.

Rules for Inserting Memo Fields

1. Each memo field name must be surrounded by double braces (i.e. {{ MemoFieldName }}).
2. Only the memo field name should appear within the double braces. HTML commands should be placed outside of the double braces.
3. Only one xBase expression ([[ ]]) or memo field ({{ }}) is allowed per line.
4. The entire memo declaration must appear on a single line.

Setting Export Location & File Names

Accessing the Export Location & HTML File Names Dialog

To view the Export Location & HTML File Names dialog select Export Location/File Names from the Export Menu or click the button with the same title.

Setting Export Location

The Export Location window allows the user to select the drive and directory to export HTML files to. When the Browser button is clicked, the export directory is searched for a master index. The Master Index is generated each time the user builds HTML files and contains links to every file created. The Master Index allows users to proof new web pages.

Setting HTML File Names

The HTML File Names windows allows users to choose between three file naming conventions: Sequentially Number Files, File Name and Extension from DBF, and File Name Without Extension from DBF.

Sequentially Number Files - Uses the record number of each database record to name the created HTML file. The user can enter a file name prefix and file extension to use when naming HTML files. If the default prefix "REC" is used with the default file extension "HTML" the generated files with have the names REC00001.HTML - REC99999.HTML.

File Name and Extension from DBF - This option allows the user to select a database field that contains a valid file name, including extension, to be used as the file name for created HTML files. The user can also optionally assign a file name prefix to be added to the start of each file name.

File Name Without Extension from DBF - This option allows the user to select a database field that contains a valid file name. The file extension is selected by the user.. The user can also optionally assign a file name prefix to be added to the start of each file name.

VideoTbl.htm Source

The following is a listing of the VideoTbl.htm template file.   This template is desgined to create a table based catalog of video titles.   The HTML used to generate web pages is divided into four sections in this template.   The start and end of each section is identified by a section declaration that appears in blue.

```
<<dbHeader>><!-- Start of web page header. -->
<html>
<head>
<title>SAMPLE</title>
</head>
<body bgcolor="#FFFFFF">
<</dbHeader>><!-- End of web page header. -->

<<dbGroup1Header>><!-- Start of group #1 header. -->
<p><font size="4"><strong>Category: [[Alltrim(Category)]]</strong></font></p>
<div align="left">
<table border="1" cellpadding="10" cellspacing="0">
    <tr>
        <td align="center" bgcolor="#C0C0C0"><strong>Title</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Rating</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Director</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Released</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Cost</strong></td>
    </tr>
<</dbGroup1Header>><!-- End of group #1 header. -->

<<dbBody>><!-- Start of web page body. -->
    <tr>
        <td>[[TITLE]]</td>
        <td>[[Rating]]</td>
        <td>[[DIRECTOR]]</td>
        <td>[[Str(Year(RELEASED),4)]]</td>
        <td align="right">[[Transform(PRICE, "$9,999,999.99")]]</td>
    </tr>
<</dbBody>><!-- End of web page body. -->

<<dbFooter>><!-- Start of web page footer. -->
</table>
</div>
<br>
Last Updated:   [[DTOC(Date())]]<br>
</body>
</html>
<</dbFooter>><!-- End of web page footer. -->
```

Return to Creating Tables

VideoCat.htm Source

The following is a listing of the VideoCat.htm template file. This template is desgined to create a catalog of video titles. The HTML used to generate web pages is divided into four sections in this template. The start and end of each section is identified by a section declaration that appears in blue.

```
<<dbHeader>><!-- Start of web page header. -->
<!-- Bluestem Technologies -->
<html>
<head>
<meta name="AUTHOR" content="Bluestem Technologies">
<title>dB Web Sample</title>
<body bgcolor="#FFFFFF" text="#000000">
<</dbHeader>><!-- End of web page header. -->

<<dbGroup1Header>><!-- Start of group #1 header. -->

<br>
<H1>[[CATEGORY]]</H1>
<hr>
<</dbGroup1Header>><!-- End of group #1 header. -->

<<dbBody>><!-- Start of web page body. -->

<H3>[[TITLE]]</H3>
{{DESC}}
<br><br>
Starring:   {{STARRING}}<br>
Director:   [[DIRECTOR]]<br>
Released:   [[Str(Year(RELEASED),4)]]<br>
Price:   [[Transform(PRICE, "$9,999,999.99")]]<br>
[[IF(Empty(AWARDS), '', 'Awards:   '+Alltrim(Awards)+'<br>')]]
[[IF(RATING='R', '<IMG SRC="R.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="R">',
'')]]
[[IF(RATING='PG', '<IMG SRC="PG.GIF" WIDTH=30 HEIGHT=30 BORDER=0
ALT="PG">', '')]]
<hr>
<</dbBody>><!-- End of web page body. -->

<<dbFooter>><!-- Start of web page footer. -->

<br>
Last Updated:   [[DTOC(Date())]]<br>
</head>
</html>
<</dbFooter>><!-- End of web page footer. -->
```

Return to Creating a Video Catalog

REC00007.htm Source

The following is a sample web page generate by dB Page Builder.

```
<!-- Created by dB Page Builder.   http://www.pchelponline.com/bluestem -->
<!-- Bluestem Technologies -->
<html>
<head>
<meta name="AUTHOR" content="Bluestem Technologies">
<title>dB Web Sample</title>
<body bgcolor="#FFFFFF" text="#000000">

<br>
<H1>Comedy</H1>
<hr>

<H3>Brazil</H3>
A black comedy set in a dark high tech future.   A bureucrat that is
targeted by the government as the result of a computer glich is saved
by a rebel heating and cooling repairman.
<br><br>
Starring:   Johnathan Pryce, Kim Greist, Robert DeNiro, Katherine Helmond, Bob
Hoskins, Ian Holm<br>
Director:   Terry Gilliam<br>
Released:   1985<br>
Price:   $49.95<br>
<IMG SRC="R.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="R">
<hr>

<H3>Dave</H3>
An unassuming nice guy played by Kevin Kline is asked to act as a
Presidential stand-in when the President becomes ill.   Dave wins the
support of the nation and outperforms his ellected predecessor.
<br><br>
Starring:   Kevin Kline, Sigourney Weaver, Frank Langella, Ben Kingsley, Kevin
Dunn, Charles Grodin, Faith Prince<br>
Director:   Ivan Reitman<br>
Released:   1993<br>
Price:   $29.79<br>
<IMG SRC="PG.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="PG">
<hr>

<H3>Ferris Bueller's Day Off</H3>
Broderick is a high school student playing hookey and avoiding his
principle.   Ferris and his friends make the most of their day off by
exploring the city of Chicago.
<br><br>
Starring:   Matthew Broderick, Jennifer Grey, Alan Ruck, Mia Sara, Jeffery
Jones, Edie McClurg<br>
Director:   John Hughes<br>
Released:   1986<br>
Price:   $27.89<br>
```

```
<IMG SRC="PG.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="PG">
<hr>

<H3>Sleepless in Seattle</H3>
Tom Hank's son searched for a new mother with the help of a national
radio show.   Meg Ryan responds the boys request and attempts to
find them.
<br><br>
Starring:   Tom Hanks, Meg Ryan, Bill Pullman, Ross Malinger, Rosie O'Donald,
Rob Reiner, Gaby Hoffman, Rita Wilson<br>
Director:   Nora Ephron<br>
Released:   1993<br>
Price:   $32.67<br>
<IMG SRC="PG.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="PG">
<hr>

<br>
Last Updated:   04/14/97<br>
</head>
</html>
```

Return to Creating a Video Catalog

REC00001.htm Source

The following is a sample web page generate by dB Page Builder.

```
<!-- Created by dB Page Builder.   http://www.pchelponline.com/bluestem -->
<!-- Bluestem Technologies -->
<html>
<head>
<meta name="AUTHOR" content="Bluestem Technologies">
<title>dB Web Sample</title>
<body bgcolor="#FFFFFF" text="#000000">

<br>
<H1>Action</H1>
<hr>

<H3>Apocalypse Now</H3>
A surreal depiction of the Vietnam War inspired by Joseph Conrads's
novel Heart of Darkness.   Charlie Sheen plays a special-op officer
sent to kill a rouge U.S. officer (Marlon Brando.)   Considered to be one
of the best Vietnam movies.
<br><br>
Starring:   Martin Sheen, Marlon Brando, Robert Duvall, Frederic Forrest<br>
Director:   Francis Ford Coppola<br>
Released:   1995<br>
Price:   $39.99<br>
Awards:   Best Cinematography, Sound<br>
<IMG SRC="R.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="R">
<hr>

<H3>Braveheart</H3>
Mel Gibson's adventure epic about 13th-century Scottish rebel William
Wallace's attempt to win independence from Britain.
<br><br>
Starring:   Mel Gibson, Sophia Marceau, Patrick McGoohan, Catherine
McCormack<br>
Director:   Mel Gibson<br>
Released:   1995<br>
Price:   $75.00<br>
Awards:   Best Picture, Director, Cinematography<br>
<IMG SRC="R.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="R">
<hr>

<H3>Butch Cassidy and the Sundance Kid</H3>
Redford and Newman play two outlaws who turn the West on its ear
with bank heists and train robberies, making their way to South
America.
<br><br>
Starring:   Paul Newman, Robert Redford, Katherine Ross, Strother Martin<br>
Director:   George Roy Hill<br>
Released:   1969<br>
Price:   $39.75<br>
```

```
Awards:   Best Story, Screenplay, Cinematography, Song, Original Score<br>
<IMG SRC="PG.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="PG">
<hr>

<H3>Congo</H3>
A loose adaption of Michael Crichton's book Congo.   A group of
scientists search a remote region of the Congo for industrial diamonds.
 The group also dicovers a lost species of dangerous apes.
<br><br>
Starring:   Dylan Walsh, Laura Linney, Ernie Hudson, Joe Don Baker, Tim Curry<br>
Director:   Frank Marshall<br>
Released:   1995<br>
Price:   $19.99<br>
<IMG SRC="PG.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="PG">
<hr>

<H3>Dances With Wolves</H3>
Kevin Costner plays a civil war hero that asks to be stationed in the far
West.   Stationed alone at a remote outpost, Costner is accepted by a
group of Sioux indians.
<br><br>
Starring:   Kevin Costner, Mary McDonnell, Graham Greene, Rodney A. Grant<br>
Director:   Kevin Costner<br>
Released:   1990<br>
Price:   $15.45<br>
Awards:   Best Picture, Director, Screenplay, Adaptation, Score, Cinematography,
Editing<br>
<IMG SRC="PG.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="PG">
<hr>

<H3>Mission Impossible</H3>
A remake of the classic television series.   Tom Cruise leads a team of
operatives into a deadly ambush.   Cruise must then find the double
agent responsible.
<br><br>
Starring:   Tom Cruise, Jon Voight, Emmanuelle Beart, Henry Czerny, Jean Reno<br>
Director:   Brain DePalma<br>
Released:   1996<br>
Price:   $65.00<br>
<IMG SRC="PG.GIF" WIDTH=30 HEIGHT=30 BORDER=0 ALT="PG">
<hr>

<br>
Last Updated:   04/14/97<br>
</head>
</html>
```

Return to Creating a Video Catalog

Database Options

Database Formats Supported

dB Page Builder currently supports dBase III+, dBase IV (without memos), dBase V (without memos), FoxPro 2.0, FoxPro 2.5, and FoxPro 2.6 compatible DBF databases. Access and Paradox users should be able to create DBF format databases with a minimum of effort. dB Page Builder does not currently support database indexing or filtering. Users should filter and sort data within their database tool of choice to create databases ready for use by dB Page Builder. Future versions of dB Page Builder may offer enhanced database options as demand warrants.

Opening a Database

To open a database select Open Database from the Database Menu or click on the Open Database button.

Closing a Database

To close a database select Close Database from the Database Menu or click on the Close Database button.

Viewing the Field List

To open a list of database fields click on the View Field List button. This screen can be left open for use as a reference while adding xBase expressions. Double clicking on a field name will insert the field name into the HTML editor.

Viewing the Database

To view the open database select View Database from the Database Menu or click on the View Database button. This option allows the user to view the contents of the open database. Database records cannot be edited in dB Page Builder.

Database Settings

Select Database Settings from the Database Menu to view the settings dialog. The Database Settings Dialog has several options that alter database compatibility and database output. The setting options are as follows:

Hide Deleted - When selected, database records that have been deleted, but not packed, will not appear in output files.

Show Century - When selected, dates appear with century information (e.g. 01/01/2000.)

Translate OEM - When selected, the database engine will convert OEM characters to Ansi characters. This is useful when accessing international databases that are accessed or maintained using DOS database applications. Turning this option on will often correct the problem of unusual characters (usually extended character set) appearing in output database data.

Translate Memos - When selected, data in memo fields will be converted to OEM characters.   This is helpful in correctly displaying international memo fields generated using DOS database applications.

Date Format -   This option allows the user to select a data field display format.   Options include American, Ansi, British, French, German, Italian, and Spanish formats.

REC00001.htm Source

The following is a sample web page generate by dB Page Builder.

```
<!-- Created by dB Page Builder.   http://www.pchelponline.com/bluestem -->
<html>
<head>
<title>SAMPLE</title>
</head>
<body bgcolor="#FFFFFF">
<p><font size="4"><strong>Category: Action</strong></font></p>
<div align="left">
<table border="1" cellpadding="10" cellspacing="0">
    <tr>
        <td align="center" bgcolor="#C0C0C0"><strong>Title</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Rating</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Director</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Released</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Cost</strong></td>
    </tr>
    <tr>
        <td>Apocalypse Now</td>
        <td>R</td>
        <td>Francis Ford Coppola</td>
        <td>1995</td>
        <td align="right">$39.99</td>
    </tr>
    <tr>
        <td>Braveheart</td>
        <td>R</td>
        <td>Mel Gibson</td>
        <td>1995</td>
        <td align="right">$75.00</td>
    </tr>
    <tr>
        <td>Butch Cassidy and the Sundance Kid</td>
        <td>PG</td>
        <td>George Roy Hill</td>
        <td>1969</td>
        <td align="right">$39.75</td>
    </tr>
    <tr>
        <td>Congo</td>
        <td>PG</td>
        <td>Frank Marshall</td>
        <td>1995</td>
        <td align="right">$19.99</td>
    </tr>
    <tr>
        <td>Dances With Wolves</td>
        <td>PG</td>
        <td>Kevin Costner</td>
        <td>1990</td>
        <td align="right">$15.45</td>
    </tr>
    <tr>
        <td>Mission Impossible</td>
        <td>PG</td>
        <td>Brain DePalma</td>
        <td>1996</td>
        <td align="right">$65.00</td>
    </tr>
</table>
</div>
<br>
Last Updated:   04/11/97<br>
</body>
</html>
```

Return to Creating Tables

REC00007.htm Source

The following is a sample web page generate by dB Page Builder.

```
<!-- Created by dB Page Builder.   http://www.pchelponline.com/bluestem -->
<html>
<head>
<title>SAMPLE</title>
</head>
<body bgcolor="#FFFFFF">
<p><font size="4"><strong>Category: Comedy</strong></font></p>
<div align="left">
<table border="1" cellpadding="10" cellspacing="0">
    <tr>
        <td align="center" bgcolor="#C0C0C0"><strong>Title</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Rating</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Director</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Released</strong></td>
        <td align="center" bgcolor="#C0C0C0"><strong>Cost</strong></td>
    </tr>
    <tr>
        <td>Brazil</td>
        <td>R</td>
        <td>Terry Gilliam</td>
        <td>1985</td>
        <td align="right">$49.95</td>
    </tr>
    <tr>
        <td>Dave</td>
        <td>PG</td>
        <td>Ivan Reitman</td>
        <td>1993</td>
        <td align="right">$29.79</td>
    </tr>
    <tr>
        <td>Ferris Bueller's Day Off</td>
        <td>PG</td>
        <td>John Hughes</td>
        <td>1986</td>
        <td align="right">$27.89</td>
    </tr>
    <tr>
        <td>Sleepless in Seattle</td>
        <td>PG</td>
        <td>Nora Ephron</td>
        <td>1993</td>
        <td align="right">$32.67</td>
    </tr>
</table>
</div>
<br>
Last Updated:   04/11/97<br>
```

```
</body>
</html>
```

Return to Creating Tables

Using Outlines

This project is designed to introduce you to the process of creating HTML templates. The video database uses a variety of field types that will be used to demonstrate the power of using xBase functions to format output data. The following steps will take you through the entire process of building HTML files.

Step #1 Open the VideoCat.htm template file.
To load the Video Catalog template click on the open file button. Select the "VideoCat.htm" file located in the VideoCat subdirectory of the Tutorial directory. The VideoCat.htm file contains a template to create a simple video catalog.

View Outline.htm Source

Step #2 Open the Video.dbf database.

dB Page Builder supports Clipper, DBase, FoxPro, and Successware style DBF databases. To open the Video.dbf database click on the open database button. Select the "Video.dbf" database located in the VideoCat subdirectory of the Tutorial directory.

Step #3 Viewing Database Data & Database Field List
To view the contents of the Video database click on the View Database button. A spreadsheet style or browse listing of the database will then appear. This screen will not allow users to edit the database.
Click on the View Field List button to display a list of database fields. This screen indicates the name and type of every field in the open database. To insert a field name into the HTML template double click on the field name located on this screen.

Step #4 Set Export Location and File Names
Click on the Export Location button to display the Export Location & HTML File Names dialog. Using the dialog set the export location to the VideoCat subdirectory of the Tutorial directory. Leave the HTML File Names set to Sequentially Numbered Files. Close the dialog when finished.

Step #5 Set Group Order
Click on the Set Group Order button to produce the Group Order dialog. Click on the Use Group checkbox for Group Order #1 to active the group. Click on the Show Outline check box to instruct dB Page Builder to print the Outline sections when the group order value changes. The Group Expression box allows the user to enter a expression to group the data by. Type two quotes (e.g. "") in the Group Expression box. Because the value of the group expression will never change the outline will be printed only once.

Step #6 Check the HTML Template
Click on the Check HTML Template button to perform an analysis of all xBase expressions. If dB Page Builder reports an error, correct the indicated line or index.

Step #7 Generate HTML Files
Click on the Build HTML button to generate web pages and indexes. Don't blink because dB Page Builder will generate the web pages in under a second. Registered users can reasonably expect to process over 2,000 database records a minute on a Pentium computer.

View Generated REC00001.htm

Step #8 View the Results

Click on the Browser button to open the web browser associated with HTML files. A web browser must be associated for this feature to work correctly. When dB Page Builder generates HTML it automatically creates a master index of its completed work. The browser feature will open the master.htm file located in the currently selected export directory. The web page includes links to each HTML file created. Take a look at the 2 web pages generated by the tutorial. Now its time to start thinking about what dB Page Builder can do with your databases.

Return to Tutorial

Outline.htm Source

dB Page Builder is designed to allow users to display database data in two seperate locations per group value.   This feature can be used to create a content outline on one part of a web page followed by more extensive content.   The Outline section is used to define HTML to appear prior to the body section of the HTML template.

```
<<dbHeader>><!-- Start of web page header. -->
<!-- Bluestem Technologies -->
<html>
<head>
<meta name="AUTHOR" content="Bluestem Technologies">
<title>dB Web Sample</title>
<body bgcolor="#FFFFFF" text="#000000">
<</dbHeader>><!-- End of web page header. -->

<<dbGroup1Header>><!-- Start of group #1 header. -->
<H1><center>Outline Example</center></H1>
<</dbGroup1Header>><!-- End of group #1 header. -->

<<dbOutline>><!-- Start of outline body. -->
[['<a href="#'+Alltrim(str(RecNo()))+'">'+title+'</a>']]<br>
<</dbOutline>><!-- End of outline body. -->

<<dbOutlineFooter>><!-- Start of outline footer. -->
<hr>
<</dbOutlineFooter>><!-- End of outline footer. -->

<<dbBody>><!-- Start of web page body. -->
<a name="[[RecNo()]]"></a>
<H3>[[TITLE]]</H3>
{{TEXT}}<br>
<</dbBody>><!-- End of web page body. -->

<<dbFooter>><!-- Start of web page footer. -->
<br>
<hr>
<br>
Last Updated:   [[DTOC(Date())]]<br>
<br>
</head>
</html>
<</dbFooter>><!-- End of web page footer. -->
```

Return to Using Outlines

Using Outlines - REC00001.htm

The following is a sample web page generate by dB Page Builder.

```
<!-- Created by dB Page Builder.   http://www.pchelponline.com/bluestem -->
<!-- Bluestem Technologies -->
<html>
<head>
<meta name="AUTHOR" content="Bluestem Technologies">
<title>dB Web Sample</title>
<body bgcolor="#FFFFFF" text="#000000">
<H1><center>Outline Example</center></H1>

<a href="#1">Background                               </a><br>
<a href="#2">Adding Database Fields to HTML           </a><br>
<a href="#3">Generating HTML                          </a><br>
<a href="#4">It Also Makes Indexes                  </a><br>
<a href="#5">What Can I Build With dB Web Builder     </a><br>

<hr>

<a name="1"></a>
<H3>Background</H3>
Over the last several months, one of my friends has labored to produce a 2500 page
web site.   He generated his web site in a well known and highly rated web site
development tool.   As the size of his web site grew, the development software
became progressively slower and more difficult to manage.   He repeatedly
commented to me that he could not find a good tool to generate HTML files from a
database.   Eventually, I decided to attempt to integrate his understanding of web
page development with my database background.   dB Web Builder is the result of
this collaboration.<br>

<a name="2"></a>
<H3>Adding Database Fields to HTML</H3>
dB Web Builder allows the user to open an HTML file much like any other HTML
editing tool.   We recommend that the user create a sample web page in their
favorite graphical HTML editor and then open the sample HTML in dB Web Builder.
Then the user can open a FoxPro compatible DBF database with data to be placed in
web pages.    dB Web Builder   allows the user to insert xBase compatible
expressions into their HTML script.   By surrounding expressions with double brackets
(e.g. [[expression]] ) the user can create an HTML template that dB Web Builder will
use to generate hundreds or thousands of HTML files.   The software supports a wide
variety of xBase expressions that allow the user to format field data, insert date and
time information, and create IF statements that allow for conditional HTML output.
Fields can be evaluated to determine if HTML code should be added to each HTML
output file thus, allowing for database driven HTML customization.   dB Web Builder
allows users to add and format data from character, numeric, date, logical, and
memo fields.<br>

<a name="3"></a>
<H3>Generating HTML</H3>
Once the HTML template has been created the user can than build HTML files.   dB
```

```
Web Builder automatically tests each xBase expression and reports any errors.   dB
Web builder then begins generating a unique HTML file for each database record.
Testing on a 133mHz Pentium produced over 2000 HTML files in under one minute!
dB Web Builder allows the user to select a naming convention for the generated
HTML files.   Database fields can even be used to name output files.<br>

<a name="4"></a>
<H3>It Also Makes Indexes</H3>
If your wondering how to develop links to all 2000 web pages you have just created,
dB Web Builder can help.   Users can create up to 3 sorted   index files.   First, select
the database field to sort the created index .   For example, an index could be created
to sort pages by product name, persons name, or date.   The user can enter an xBase
expression to be used to create an HTML link.   A special variable called "HTMLFILE"
can be inserted into the expression to instruct dB Web Builder to add the file name
into the HTML link.     When dB Web Builder generates HTML files, it also creates the
requested indexes with a sorted list of links to each HTML file.   Cut and paste these
links into your index page and your almost done.   A massive web site in
minutes.<br>

<a name="5"></a>
<H3>What Can I Build With dB Web Builder</H3>
DB Web Builder is designed to allow great flexibility in web page generation without
sacrificing speed and performance.   If you are familiar with HTML and xBase
functions generating, HTML templates will be easy and powerful.   Luckily, xBase
functions are easy to learn and intuitive to use so learning times will be short.   DB
Web Builder makes converting existing databases to web sites simple and fast.
Inventory, personnel, and other databases can be converted to an HTML in minutes.
Even if your data is not currently in a database, dB Web Builder may convince you it
should be. <br>

<br>
<hr>
<br>
Last Updated:   04/11/97<br>
<br>
</head>
</html>
```

Return to Using Outlines

Set Group Order

Accessing the Group Order Dialog

To view the Group Order dialog select Set Group Order from the Database Menu or click the Set Group Order button.

About dB Page Builders Group Order Options

dB Page Builder allows users to define up to 3 group order expressions. As dB Page Builder accesses each record in the database the group order expressions are evaluated to determine if the expression value has changed. When a group order value changes, the group header and group footer are added to the HTML. Users can also instruct dB Page Builder to create a new HTML file for each group order value.

Activating a Group

To activate a group order click on the Use Group check box located in the upper right corner of the Group Order box.

Setting the Group Expression

The Group Expression box allows the user to enter a xBase expression. This expression will be evaluated before each database record to determine if the group value has changed. When the group value changes the group footer is added for the previous record and the group header is added for the new database record. If the New Page or Show Outline options are selected they will be activated as well.

The most common form of Group Expression is the use of a database field. Users only need to type the name of the database field to group output by into the Group Expression box. Advanced users can also use xBase functions to modify the field value. Some possible expressions are:

Expression:	Upper(Left(Title,1))
Result:	Creates one group per letter of the alphabet.
Expression:	Year(DateField)
Result:	Creates one group per year.

New Page

Checking the New Page checkbox will instruct dB Page Builder to create one HTML file for each group value in the database.

Show Outline

Users can create HTML using a special template sections called the Outline Body and the Outline Footer. The Outline Body acts exactly like the Body section of the template allowing users to display two complete lists of content within a group. This feature can be used to create a content outline with hyperlinks to expanded content later in the page. For an example of how to use this feature refer to the Using Outlines sample project located in the Tutorial.

Group Headers

The Group Header section is added to the output file whenever a group value changes.   The header is added prior to the body section.   If database field data is printed in the group header it will be the data from the first record with the new group value.   The group header can be used to print content headings and to add space between data.

Group Footers

The Group Footer section is added to the output file whenever a group value changes.   The group footer follows the body section for the previous group value.   If database field data is printed in the group footer the data will come from the last record from the previous group value.

Template Sections

dB Page Builder requires users to place all HTML and xBase expressions within Template Section declarations. Each section is bounded by a start section and end section markers. All section markers are appear in bright blue text.

Sample Section Markers

```
<<dbHeader>><!-- Start of web page header. -->
<</dbHeader>><!-- End of web page header. -->

<<dbBody>><!-- Start of web page body. -->
<</dbBody>><!-- End of web page body. -->

<<dbGroup1Footer>><!-- Start of group #1 footer. -->
<</dbGroup1Footer>><!-- End of group #1 footer. -->
```

To instruct dB Page Builder to add HTML in a template section, place the desired HTML or xBase expression within the appropriate section markers.

```
<<dbHeader>><!-- Start of web page header. -->
<html>
<head>
<meta name="AUTHOR" content="Bluestem Technologies">
<title>dB Web Sample</title>
<body bgcolor="#FFFFFF" text="#000000">
<</dbHeader>><!-- End of web page header. -->
```

To add section markers to a template click on the Insert Template Section button. The Insert Section Markers dialog will then appear. Click on the desired section type to add the section markers to the template. If the user highlights a series of HTML lines in the template and then adds section markers the section markers will automatically be added before and after the highlighted text.

Do not add HTML to the section marker lines.
The section markers will not appear in HTML generated by dB Page Builder.

Template Sections

Header - The header section is added at the top of each HTML file.
Body - The body section is added for each database record.
Footer - The footer section is added to the bottom of each HTML file.
Group Headers - Added prior to the body section when the associated group value changes.
Group Footers - Added following the body section when the associated group value changes.
Outline Body - Acts like the body section when Show Outline is selected for a Group Order.
Outline Footer - Added following the Outline Body and prior to the Body section.

Project Files

Project files allow the user to save and reopen an HTML template, database file, file export location, file naming convention, and group order settings all at one time. The File Menu includes options to Open Project, Save Project, and Save Project As. These options can be used to save the current settings as well as restore those settings at a later time. The project information is maintained in project files using a .dbp file extension.

Open Project - Allows the user to select a project file and then restores the settings stored in the selected file. Any currently opened files will be closed.

Save Project - Saves the current settings to a project file. This option will overwrite the existing file without alerting the user.

Save Project As - Save the current settings to a project file. The user will have the opportunity to select a project file name.