Appendix C

Quick reference to useful features

Standard subroutines

Mathematical functions

The following table contains examples and explanations of some mathematical functions in *Delphi*:

Abs	Returns the absolute value of a number – the result is of the same type as the number. For example: Abs(-4) = 4; Abs(-3.5) = 3.5; Abs(6.8) = 6.8
Ceil *(stored in unit Math)*	This function returns the largest integer value nearest to the number. For example: Ceil(2.1) = 3 ; Ceil(2.9) = 3 ; Ceil(-3.1) = -3
Floor *(stored in unit Math)*	This function returns the smallest integer value nearest to the number. For example: Floor(2.1) = 2; Floor(2.9) = 2; Floor(-3.1) = -4
Frac	This function returns only the decimal part of a real number. For example: Frac(74.89) = 0.89 ; Frac(0.3728) = 0.3278
PI	PI returns a value for the mathematical constant PI. For example: rC:= 2 * PI * iR;
Power *(stored in unit Math)*	Raises the first number to the power of the second number. All variables must be compatible with the data type Extended. For example, to calculate 4^3, use the following statement: rAnswer := Power(4, 3); To calculate $4^{0.5}$ (the same as $4^{1/2}$), use the following statement: rAnswer := Power(4, 0.5);
Random	iVal := Random(iX); *// Returns an integer value in the range [0..(iX-1)].* rVal := Random; *// Returns a real number >=0 but < 1.* *Tip:* Random (100) has 100 possible answers (from 0 to 99). To generate a number between 1 and 100 (both included) do the following: iNo := Random (100) + 1; To make the numbers generated by the Random function less predictable use the Randomize procedure. Randomize simply scrambles the results of the Random function. You only have to use this procedure once in your program, preferably when the program starts (use the OnActivate event of the Form).

RandomRange *(stored in unit Math)*	iVal := RandomRange(1,100); Returns an integer value between 1 and 100 (both can be included).
Round	This function rounds a real value and returns an integer value rounded to the nearest whole number. For example: Round(8.8) = 9 ; Round(8.3) = 8 If the number is halfway between two integers, it will always be rounded to the nearest even number. For example: Round(7.5) = 8 ; Round(8.5) = 8
Sqr	Returns the square of a number – the result is of the same type as the number.
Sqrt	Returns the square root of a number. The result is always of type real. *Note:* The square root of a negative number can not be calculated.
Trunc	This function truncates a real number to an integer. It 'chops' off the decimal part and returns only the integer part. For example: Trunc(8.5) = 8 ; Trunc(8.3) = 8

String manipulation functions

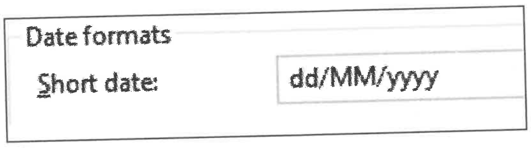
Concat	The Concat function can be used instead of the '+' operator.
	For example:
	lblOut.Caption:=Concat('Hello ', sName);
	provides the same result as
	lblOut.Caption := 'Hello ' + sName;

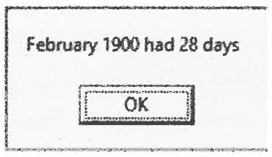
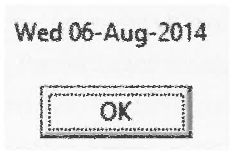
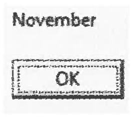
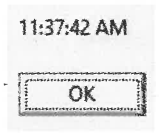
Copy	Copies a substring from sS, from position iStart for iLength characters. For example: sWord := Copy(sS, iStart, iLength);
Length	Returns the number of characters that a string value consist of as an integer. For example: iNumChars := Length(sName);
Pos	Returns the position of the first letter of sString in sName. For example: iPos := Pos(sString, sName);
UpCase	The Upcase function is used to convert one character to upper case. In this example the first character of the string is converted to upper case. For example: sWord[1] := Upcase(sWord[1]);
UpperCase	Converts all letters in sWord to upper case and assigns it to sWord. For example: sWord := Uppercase(sWord);

String manipulation procedures

Delete	Delete(sString, iStart, iLength); Deletes a substring from a string. sString: The string that should be changed. iStart: The position of the first character that should be deleted. iLength: The number of characters to be removed from the string – the starting one included. For example: sS := 'Lets start from scratch'; Delete(sS,12,5); Value of sS is now: 'Lets start scratch'
Insert	Insert(sSub, sString, iPosition); Inserts a substring into a string in a given position. sSub: the string that should be inserted. sString: The string that should be changed. iPosition: The position where the substring should start in the changed string. Example: sS := 'Bread tastes good'; Insert('Hot ', sS, 1); Value of sS is now: 'Hot Bread tastes good'
Str	Str(rNumber, sString:5:2); Converts the value stored in rNumber to a string value and stores the string in sString. In this case the number should consist of 5 places including 2 decimals.
Val	Val(sString, iNumber, iCode); Converts sString to a number and stores the number in variable iNumber. If the variable iCode is 0, the conversion was successful, if not, the value of iCode shows the position of the character in the string that caused the problem.

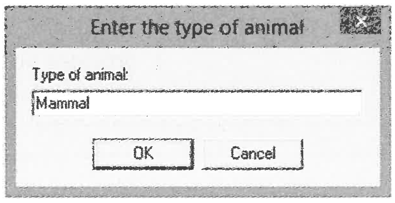
Time and date functions and features

Date	Returns the current system date as a value of data type TDate.
DateTimeToStr	Converts a TDateTime value into a string displaying the date and time. The function call DateTimeToStr(Now) will return a string in the following format: 14/02/2014 08: 38: 00 PM
DateToStr	Converts a value of type TDate to a string. The format will be determined by the Short Date format in the date settings of the operating system. For example, 23 March 2013 will be displayed as 23/03/2013 if the settings is as follows: Date formats Short date: dd/MM/yyyy To display the current system date as a string, you can use the following statement: ShowMessage(DateToStr(Date));

DaysInAMonth *(in the DateUtils unit)*	Returns the number of days in a certain year and month. For example: iNumFebDays := DaysInAMonth(1900, 2); ShowMessage('February 1900 had ' + IntToStr(iNumFebDays) + ' days');  February 1900 had 28 days
FormatDateTime	Formats a TDate or TDateTime value into a string. The format is indicated through the use of a combination of letters and characters as one of the arguments of the function. For example: sDate := FormatDateTime('ddd dd-mmm-yyyy', calDate.Date); ShowMessage(sDate);  Wed 06-Aug-2014
IsLeapYear	The IsLeapYear function returns a value True if a given calendar value is a leap year. The year supplied as the argument can be a value between 0 and 9999.
LongMonthNames *(an array stored in the SysUtils unit)*	This array contains the names of the months of the year. You can use it as follows: sMonthName := LongMonthNames[11]; ShowMessage(sMonthName);  November
Time	Returns the current system time as a value of data type TDateTime.
TimeToStr	Converts the a TDateTIme value to a string. For example: sTime := TimeToStr(Time); ShowMessage(sTime);  11:37:42 AM

Miscellaneous functions

Chr	The function Chr returns the character associated with an ASCII code.
	cChar := Chr(65); *//Returns the character 'A'*
	cChar := Chr(50); *//Returns the character '2'*

InputBox	A function to input a string that creates a pop-up: sAnimal := InputBox('Enter the type of animal', 'Type of animal: ', '');  Enter the type of animal Type of animal: Mammal

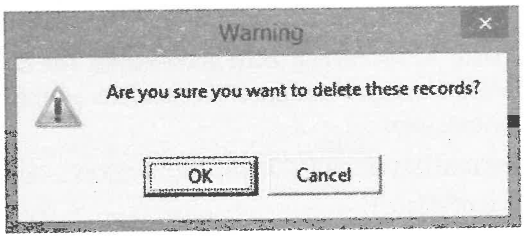
MessageDlg

The function MessageDlg displays a pop-up message to the user. The programmer can specify which type of message it should be (a warning, an error, just information etc.). This determines the symbol and heading which will be displayed in the pop-up message, which Buttons should be displayed (a [Yes] Button, an [OK] Button) etc.

Example code:

```
if MessageDlg('Are you sure you want to delete these records?',
            mtWarning, [mbOk, mbCancel], 0) = mrOK then
```

Example of the pop-up message produced by the MessageDlg function:



Are you sure you want to delete these records?

The parameters of the function MessageDlg are as follows:

```
function MessageDlg(const Message:string; DialogType:TMsgDlgType;
        Buttons:TMsgDlgButtons; HelpContext:Longint ):integer;
```

Arguments may have the following values:

Message	The message can be any string you wish to display.		
DialogType (one of the following)	mtWarning	mtConfirmation	mtInformation
	mtError	mtCustom	
Buttons (one or more of the following)	mbYes	mbNo	mbOK
	mbCancel	mbAbort	mbRetry
	mbIgnore	mbAll	mbNoToAll
	mbYesToAll	mbHelp	
HelpContext	This value is used in conjunction with the Help Button, but it will not be discussed in this Appendix. Use the value 0 (zero).		
The function returns an integer value, but *Delphi* also provides the option to compare the return value to a *Delphi* enumerated value. For example, when the user clicks on the [Yes] Button, the function returns a value mrYes (it is not a string – therefore no quotes should be used).			
Values returned by the function	mrYes	mrNo	mrOK
	mrCancel	mrAbort	mrRetry
	mrIgnore	mrAll	mrNoToAll
	mrYesToAll	The [Help] Button provides no return value.	

Odd	Determines whether the argument is odd or not. Returns a Boolean value. bAnswer := Odd(iValue);

Ord	The function Ord is usually used to determine the ordinal value (ASCII code) of a character.
	iNum := Ord('A'); *// Returns the value 65*
	iNum := Ord('1'); *// Returns the value 49*

RGB	Sets the colour of the screen using a combination of red, green and blue. frmScreen.Color := RGB(iRed, iGreen, iBlue);

Miscellaneous procedures

ShowMessage	A procedure to display a message, without using a specific object on the form.
Randomize	Scrambles the results of the Random function to make the results less predictable.
Beep	Makes a beep noise.
Sleep	Delays the program for a certain number of milli seconds. For example: Sleep(200); *// 200 ms*
Inc	Adds a value to a variable of an ordinal data type. Inc(iNumber, iAdd); *// Increases the value of iNumber by iAdd* Inc(iNumber); *// Increases the value of iNumber by 1* Inc(cChar, 2); Changes the character to two places after this character in the ASCII table. (Increases the ordinal value of the character by 2).
Dec	Subtracts a value from a variable of an ordinal data type. Dec(iNumber,2); *// Decreases the value of iNumber by 2* Dec(iNumber); *// Decreases the value of iNumber by 1* Dec(cChar, 2); Changes the character to two places before this character in the ASCII table. (Decreases the ordinal value of the character by 2).