http://www.mindspring.com/~jcta

TAwk Component

TAwk is a Delphi component which provides the same functionality as the VsAwk vbx for Visual Basic.   It allows you to quickly scan and parse text files.   In addition it provides a Like method which is similar in function to the Like operator in Visual Basic.

TAwk is shareware and copyrighted by:

John C. Taylor
3475 Holcomb Br Rd
Suite 202
Norcross, GA   USA   30092

TAwk Properties

Action
ConvertCase
EndLine
ErrorNumber
ErrorText
F
Filename
FilterQuotes
FS
LineNumber
MatchQuotes
N
Progress
QuoteChars
StartLine

Technical Support

If you have questions about TAwk you can contact me email at:

Compuserve: 76350,3301
Internet: jcta@mindspring.com **OR** 76350.3301@compuserve.com

If you found TAwk on the internet you may not have the latest version.   To ensure that you have the latest version, visit my web site and download from there.   As I update TAwk, I automatically send notifications to registered users so if you find it useful be sure to register your copy.   Note that TAwk is not crippled in any way and there are no nag screens.   The version you have is fully functional.

If you have suggestions for improvements or you suspect a bug, please email me with the details.

Action Property

Description:
The Action property is an integer which tells TAwk what to do.   The valid values are:

0 - Parse String
1 - Scan File
2 - Stop scanning and close file

Example:
```
Awk1. Filename := 'c:\readme.txt'
Awk1.Action := 1;
Awk1.Execute;
```

The above example scans the file 'c:\readme.txt'.

ConvertCase Property

Description:

The convertcase property tells Awk what type of case conversion to do when parsing a string or scanning a file.   The valid values are:
ccLower - Convert the line to lower case before parsing.
ccNone   - No conversion
ccUpper - Convert the line to upper case before parsing.

Example:

```
Awk1.Filename := 'c:\readme.txt';
Awk1.Action := 1;
Awk1.ConvertCase := ccUpper;
Awk1.Execute;
```

The above example scans the file 'c:\readme.txt' and converts each line to upper case before parsing.

EndLine Property

Description:

EndLine is a long integer property which tell Awk at which line to stop
scanning a file.   If EndLine is zero, Awk scans the file until an EOF marker is reached.

Example:

```
Awk1.Filename := 'c:\readme.txt';
Awk1.Action := 1;
Awk1.ConvertCase := ccUpper;
Awk1.EndLine := 5;
Awk1.Execute;
```

The above example scans the first 5 lines of file 'c:\readme.txt' and converts each line to upper case before parsing.

ErrorNumber Property

Description:

ErrorNumber is an integer property.   It is runtime and read only.
This property indicates the number of the last error that Awk encountered.
The possible values are:
1 - The filename property specifies a non-existent file.
2 - The StartLine property specifies an invalid line number
3 - File I/O Error

See Also:

OnError Event

OnError Event

Description:
The OnError event is fired when Awk encounters an error.   The errnum and errtext paramaters can be used to diagnose the error.

Example:
```
Procedure Awk1Error(Errnum: integer; Errtext: string);
begin
            MessageDlg('Awk Error: ' + Inttostr(Errnum) + ' ' +
Errtext,mtWarning,[mbok],0);
end;
```

ErrorText Property

Description:
ErrorText is a string property containing a description of the last error encountered by Awk.   It is runtime and read only.

F Property

Description:

F is an array of characters which represents the last line read from a file or string parsed.   The maximum length of F is 8,192 bytes.   This represents the maximum length of a line of text that can be read from a file.   It is runtime only.

Example:

```
StrPcopy(Awk1.F,'this is a string');
```

Filename Property

Description:

The filename property is the name of the file to be scanned by Awk.   It should contain the full path name location of the file.   Awk only processes text files.

Example:

```
Awk1.Filename := 'c:\readme.text';
Awk1.Action := 1;
Awk1.Execute;
```

This example scans the file 'c:\readme.txt'.

FilterQuotes Property

Description:

FilterQuotes is a boolean property. If FilterQuotes is true, Awk will strip single or double quotes from the parsed tokens. Setting this property to false has no effect when MatchQuotes is false.

Example:

When MatchQuotes is true and FilterQuotes is true, a string such as:

"this is a Delphi Component"
would return
this is a Delphi Component
as one token without the quotes. Note that when MatchQuotes is true, everything inside the quotes is considered one token regardless of the setting of the FS property. The setting of the FilterQuotes property is ignored when MatchQuotes is false;

See Also:

MatchQuotes property
FS property

MatchQuotes Property

Example:
MatchQuotes is a boolean property.   When true, Awk considers everything within matching quotes as a single token.   This applies to single and double quotes.
See example under FilterQuotes property

FS Property

Description:

FS is a string that specifies the field separators used by Awk in parsing a line of text.   The default is space and tab.   When Awk encounters any character in the FS string, it considers all following characters a new token until a new instance of a FS characters is encountered.

Example:

```
Awk1.FS := '(' + ')';
StrPcopy(Awk1.F,' (1+2)');
Awk1.Execute;
```

The above example would result in one token = 1+2.

LineNumber Property

Description:
The linenumber property is a long integer and represents the current line in the file being processed.   It is runtime and read only.

Example:
```
Procedure Awk1Scan;
begin
        Panel1.caption := 'Awk Scanning Line: ' + inttostr(Awk1.LineNumber);
end;
```

The above example uses the OnScan event to report the current line number.

N Property

Description:
N is an integer which represents the number of tokens that resulted from parsing the last line of text.   It is runtime and read only.

Example:
```
StrPcopy(Awk1.F, 'one two three');
Awk1.Action := 0;
Awk1.Execute;
NumTokens := Awk1.N;
```

In the above example, N will be 3 after the execute method is called.

Progress Property

Description:
Progress is a longint property that reports the percent of the file which has been scanned.   This property can be monitored in the OnScan event to report the progress of the current file.

See Also:
OnScan Event

OnScan Event

Description:
Awk fires the OnScan event after scanning a line of text from a file.

Example:

```
Procedure TForm1.Awk1Scan;
begin
        Panel1.caption := 'Awk Scanning Line: ' + inttostr(Awk1.LineNumber);
end;
```

The above example uses the OnScan event to report the current line number.   To obtain the entire line that was scanned you can examine the F property. Example:

```
Procedure TForm1.Awk1.Scan;
begin
      Panel1.caption := StrPas(Awk1.F); {shows the last line scanned}
end;
```

See Also:
GetToken method

GetToken Method

Description:

Call the GetToken method to retrieve a particular token which resulted from Awk's parsing a line of text. GetToken takes one integer paramater and returns a string. The value of the integer parameter should be in the range of 0 to Awk.N.

Example:

```
StrPcopy(Awk1.F, 'one two three');
Awk1.Action := 0;
Awk1.Execute;
for i := 0 to Awk1.N - 1 do
begin
    s := Awk1.GetToken(i);
    listbox1.items.add(s);
end;
```

The above example retrieves the tokens parsed and adds them to a list box.

QuoteChars Property

Description:
QuoteChars is a string which represents the characters to be used by TAwk when matching quotes. The default is " (#34). You should use care when changing this property. Consider the following examples:

When QuoteChars is " the string { "O'neal is a Delphi programmer" } would
be parsed into one token:

O'neal is a Delphi programmer

However, if QuoteChars is set to ' and " (#39+#34) the same string would be parsed into six tokens:

O
neal
is
a
Delphi
programmer

If QuoteChars is set to ' (char #39) the string would be parsed into two tokens:
O
neal is a Delphi programmer
Note:
If TAwk encounters the end of the line of text without finding a matching end quote character, the entire string after the first quote character will be included in the token as if there was in fact a matching end of quote.
For example, the string "this is a string
would be parsed into one token - this is a string.

StartLine Property

Description:
The StartLine is of type longint and tells Awk at which line to begin scanning a file.

Example:
```
Awk1.Filename := 'c:\readme.txt';
Awk1.Action := 1;
Awk1.ConvertCase := ccUpper;
Awk1.StartLine := 4;
Awk1.EndLine := 5;
Awk1.Execute;
```

The above example scans the file 'c:\readme.txt' and converts each line to upper case before parsing. Awk starts the scan at line 4 and ends at line 5.

TAwk Events

OnBegin
OnConvertCase
OnEnd
OnError
OnParse
OnScan

OnBegin Event

Description:
The OnBegin event is fired prior to Awk scanning a file.

Example:
```
procedure Awk1Begin;
begin
        MessageDlg('Starting file scan',mtInformation,[mbok],0);
end;
```

OnConvertCase Event

Description:
Awk fires the OnConvertCase event before converting the case of a line of text.   The conversion is done according to the setting of the Convertcase property.

See Also:
ConvertCase Property

OnEnd Event

Description:
The OnEnd event is fired when Awk has completed scanning the file.

Example:
```
procedure Awk1End;
begin
	MessageDlg('File processing completed.',mtInformation,[mbok],0);
end;
```

OnParse Event

Description:
The OnParse event is fired prior to Awk parsing a line of text.

Example:
```
Procedure Awk1Parse;
begin
	Panel1.caption := 'Parsing Line Number : ' +			inttostr(Awk1.LineNumber)
end;
```

TAwk Methods

Execute
GetToken
GetItemPos
Like

Execute Method

Description:

The execute method is called initiate the process specified in the Action property.

See Also:

Action Property

Registering by Credit Card

For you convenience I have contracted with *Northstar Solutions* to process any orders that you wish to place with your valid Visa or Mastercard.   They may be contacted ***for orders only*** via any of the following methods:

Voice: 1-800-699-6395 (10:00 am - 10:00 pm Eastern Standard Time.   U. S. callers only)
            1-803-699-6395 (10:00 am - 10:00 pm Eastern Standard Time.)

FAX: 1-803-699-5465 (Available 24 hours. International and business orders encouraged.)

Email: America Online: STARMAIL   Compuserve: 71561,2751   Internet: starmail@aol.com

When you register through Northstar tell them you are registering John Taylor's TAwk component for Delphi.   Have your Visa or Mastercard # and expiration date handy.   ***Also make sure you give them your correct Email address.***

Important note:

Northstart processes registrations only, please contact the author (me) for any product/technical support.   Email and Faxed registrations are encouraged, but all registrations are very must appreciated

GetItemPos Method

Description:

Use the GetItemPos method to retrieve the position (token number) of a given string within a line of text.

Example:

```
StrPcopy(Awk1.F, 'one two three');
Awk1.Action := 0;
Awk1.Execute;
i := Awk1.GetItemPos('two');
```

In the above example, i would result in 2, the second token in the line.
If the string passed to this method does not exist in the line, GetItemPos returns -1.

Like Method

Awk's Like method provides the functionality of the Visual Basic 'Like' statement which is missing in Delphi.   The Like method takes two paramaters and returns true or false.   Both paramaters are string arguments.

example:

```
pattern := '[0-5]*##[a-z]';
expression := '1ABC99z';
Match := Awk1.Like(expression,pattern);
```

In the above example the method will return true.
More about Patterns
Examples

Pattern

Pattern matching provides a useful tool for string comparisons.   Pattern-matching features allow you to use wildcard characters, such as those recognized by DOS.   The following table shows the wildcard characters and what match.

?	Any Single Character
*	Zero or more characters
#	Any single digit
[charlist]	Any single character in charlist
[!charlist]	Any single character not in charlist

A group of one or more characters (charlist) enclosed in brackets ([ ]) can be used to match any single character in expression and can include almost any characters in the ANSI character set, including digits. In fact, the special characters left bracket ([ ), question mark (?), number sign (#), and asterisk (*) can be used to match themselves directly only by enclosing them in brackets.   The right bracket ( ]) cannot be used within a group to match itself, but it can be used outside a group as an individual character.   In addition to a simple list of characters enclosed in brackets, charlist can specify a range of characters by using a hyphen (-) to separate the upper and lower bounds of the range.   For example, [A-Z] in pattern results in a match if the corresponding character position in expression contains any of the uppercase letters in the range A through Z.   Multiple ranges are included within the brackets without any delimiting.
Examples

Pattern Matching Examples

The following examples show how you can use the like method Like to test expressions for different patterns.

Kind of matching	**pattern**	**method returns True**	**method returns False**
Multiple characters	a*a	'aa', 'aBa', 'aBBBa'	'aBC'
Special character	a[*]a	'a*a '	'aaa'
Multiple characters	ab*	'abcdefg','abc'	'cab', 'aab'
Single character	a?a	'aaa', 'a3a', 'aBa'	'aBBBa'
Single digit	a#a	'a0a', 'a1a', 'a2a'	'aaa', 'a10a'
Range of chars	[a-z]	'f', 'p','j'	'2','&'
Outside a range	[!a-z]	'9','&', '%'	'b', 'a'
Not a digit	[!0-9]	'A', 'a', '&', '~'	'0', '1', '9'
Combined	a[!b-m]#	'An9', 'az0', 'a99'	'abc', 'aj0'

Registration

TAwk is shareware. It is not crippled in any way and I have not added any nag screens, etc. You are free to distribute it as long as this help file accompanies any distribution you make. If you use TAwk, please find the time to register it. The registration fee is $ 20 US. You can register by sending me a check. If you are a compuserve member you can register online, GO SWREG #8256. Send checks to:

John Taylor
3475 Holcomb Br Rd
Suite 202
Norcross, GA USA 30092

You can also register by credit card.

If you find any bugs, let me know and I'll fix them. You can contact me
via Compuserve 76350,3301 or the internet at: jcta@mindspring.com or voice (770)-449-6284. If you register your copy you will be entitled to any fixes or updates. If you have any suggestions for improvement or criticisms I'd like to hear from you.