

# MODULE 7: VARIABLES & STRINGS

On completion of this module you will be able to work with and manipulate data, both numerical and text, found within single value holders. You will also be able to manipulate numerical values with simple mathematical formulas and principles.

## Subject Outcome 1: Introduction

### Subject Outcome 2: Internal variables

<a href="#">Screen.Height</a>	<a href="#">Screen.Width</a>	<a href="#">Screen.MouseX</a> <a href="#">Screen.MouseY</a>	<a href="#">CurDir</a>	<a href="#">Application.Path</a>
<a href="#">Application.ExeName</a>	<a href="#">Date\$</a>	<a href="#">Time\$</a>		

### Subject Outcome 3: String manipulation

<a href="#">Adding text</a> <a href="#">New Line CHR\$(13)</a>	<a href="#">Insert\$</a>	<a href="#">Replace\$</a>	<a href="#">ReplaceSubStr\$</a>
<a href="#">Field\$</a>	<a href="#">LCase\$</a> <a href="#">UCase\$</a>	<a href="#">Left\$</a> <a href="#">Right\$</a> <a href="#">Mid\$</a>	<a href="#">LTrim</a> <a href="#">RTrim</a>
<a href="#">Reverse\$</a>	<a href="#">Space\$</a>	<a href="#">Delete\$</a>	<a href="#">String\$</a>
<a href="#">Len</a>	<a href="#">Instr</a>	<a href="#">RInstr</a>	<a href="#">Tally</a>
<a href="#">Const</a>	<a href="#">QuickSort</a>	<a href="#">LBound</a> <a href="#">UBound</a>	<a href="#">Data "...":..."</a> <a href="#">Read</a>
<a href="#">Swap</a>	<a href="#">Chr\$(#)</a>	<a href="#">ASC(\$)</a>	<a href="#">STRF\$()</a>
<a href="#">[?]</a>	<a href="#">- [deduct char]</a>		

### Subject Outcome 4: Numerical manipulation (Math's)

<a href="#">Priority settings</a>	<a href="#">Three main groups</a>	<a href="#">Valid arithmetic operators</a>	<a href="#">Relation Operators</a>
<a href="#">Advanced Mathematics</a>	<a href="#">ABS()</a>	<a href="#">Ceil()</a>	<a href="#">Cint()</a>
<a href="#">Fix()</a>	<a href="#">Frac()</a>	<a href="#">Floor()</a>	<a href="#">Round()</a>
<a href="#">Sgn()</a>			

### Subject Outcome 5: Escape Characters

## 7.1 VARIABLES & STRINGS

### 7.1.1 Introduction

Variables are keywords (user defined, internal or reserved by FREEQ's use only) that holds a value, either in text or numerical format that is used by the program to gather or display information. Variables are stored in main memory and stay there for the entire duration of the program (in other words, till the program is closed - END).



#### 7.1.1.1 Creating and type of Variables

##### Syntax:

<b>\$TypeCheck On</b>	All programs must start with this command. This means that all variables must be created before you may use it. It ensures that the correct format (either text or numerical) is assigned and used correctly. It also avoids duplication of reserved words that will lead to your program "crashing".
<b>Const ? = ...</b>	<p>This will fix a variable to one specific value. This value cannot be altered and set specific rules and regulations that the program must follow, for instance:</p> <p><b>Const False = 0</b> <b>Const True = 1</b></p> <p>This means that whenever a confirmation is done with the value of 1, the confirmation must be true and 0 (zero) being always false. It replaces the numerical value with a meaning full word;</p> <p><b>if answer = true then ...</b></p> <p>whereas standard coding would be <b>if answer=1 then</b></p>
<b>Dim ? as ...</b>	<p>You assign an user defined word (your own descriptive word) as a variable, being;</p> <ul style="list-style-type: none"> <li>• String (text)</li> <li>• Byte (numerical value; 0-255)</li> <li>• Word (numerical value; 0-65535)</li> <li>• Short (numerical value; -32768 till 32768 )</li> <li>• Single (numerical value)</li> <li>• Double (numerical value - advised to use)</li> <li>• Integer (numerical -2 billion till +2 billion)</li> </ul>

	<p>There are multiple numerical value holders that will differ in memory size usage. If your program uses too much memory, a program could crash.</p> <p><b>Example of establishing a variable:</b></p> <p><b>DIM age AS single</b></p> <ul style="list-style-type: none"> <li>• This will create a numerical variable.</li> <li>• The age value of the member will be held here.</li> </ul> <p><b>DIM SURNAME as string</b></p> <ul style="list-style-type: none"> <li>• This will create a string variable.</li> <li>• The surname of the member will be held here.</li> </ul>
--	--

The **\$TYPECHECK ON** command is generated automatically by **FREEQ** to be the very first command for your program (when you press **F5**, it will be placed into your program).

The **DIM** command is placed before the **DECLARE SUB** commands as they must be established before the **SUB** programs are verified to be correct and complete.

Commands	Description
<pre> \$TYPECHECK ON  \$INCLUDE &lt;RapidQ2.inc&gt;  dim age as single dim surname as string declare sub callsubnow  CREATE Form AS QFORM   Caption = "Form":Width = 640   Height = 480:Center:onclick=callsubnow END CREATE  SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle)  Form.ShowModal  sub callsubnow   if surname="botha" then end end sub </pre>	<p>Force creation of variables.</p> <p>Establish <b>numerical variable</b> with name <b>AGE</b>. Establish <b>string variable</b> with name <b>SURNAME</b>.</p> <p>If the value of <b>SURNAME</b> is <b>BOTHA</b>, then end the program.</p>



### 7.1.1.2 Methods

#### Methods of using the DIM command to create the variable.

- Individual method:
  - Each method is established individually.
  - **DIM age AS SINGLE**
- Group method:
  - Multiple variable names grouped together as either string or numerical variable.
  - **DIM (age, year, day, month) AS DOUBLE**
  - All the above mentioned variables are DOUBLE numerical variables.
- Multi-Level method:
  - It is having sub levels in multiplication.
  - **DIM member{5,3} AS STRING**

Each cell holds a string value ...				
1.1	2.1	3.1	4.1	5.1
1.2	2.2	3.2	4.2	5.2
1.3	2.3	3.3	4.3	5.3

### 7.1.1.3 Conversion (String = Numerical)

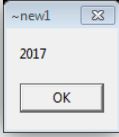
Within the programming environment you cannot just display or read a numerical variable the same as a string variable and other way around. You must convert the variables (string as numerical and other way around) to read, display or copy them amongst each other.

#### 7.1.1.3.1 STR\$()

This command will read and convert a numerical variable as a string.



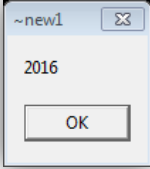
**Syntax:**

Sno	Command	Description
1	<pre>\$TYPECHECK ON dim age as string dim year as single year=2017  age=str\$(year) showmessage age</pre>	<p>Establish age as a string variable. Establish year as a numerical variable.</p> <p>Add value to the numerical variable called <b>YEAR</b>.</p> <p><b>Convert</b> the numerical value of <b>YEAR</b> to the string variable called <b>AGE</b>. Display the result.</p> 

7.1.1.3.2 VAL()

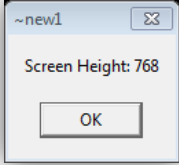
This will read and convert a string variable as a number.

**Syntax:**

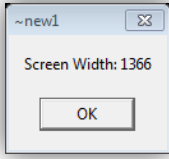
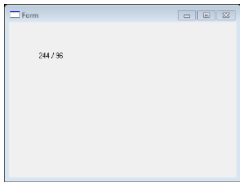
Sno	Command	Description
1	<pre>\$TYPECHECK ON dim age as string age="2016"  dim year as single  year=val(age) showmessage str\$(year)</pre>	<p>Establish age as a string variable. Add value to the string variable. All text values added to a string variable must be indicated between "..." Establish year as a numerical variable.</p> <p><b>Convert</b> the string value within <b>AGE</b> to the numerical variable called <b>YEAR</b>. Display the result – you <b>CANNOT</b> display a numerical variable. You must convert a numerical variable as a string to display it on screen.</p> 

7.1.2 INTERNAL VARIABLES

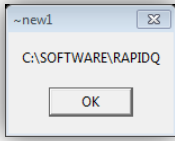
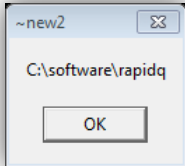
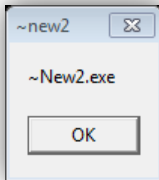
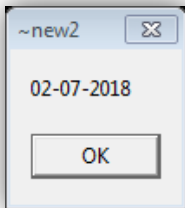
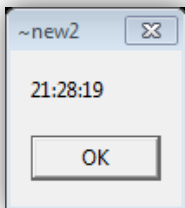
Internal variables are reserved variables within FREEQ that holds value for the programmer to be used as required. They are the following:

Sno	Command	Test Application	Description
1	<pre>Screen.Height</pre> 	<pre>Showmessage "Screen Height: "+str\$(screen.height)</pre>	<p>This is a numerical variable of the entire screen's height. It is a number, so it must be converted to be displayed. Notice how we add two text strings to form a sentence; <b>"Screen.Height: "+str\$(...)</b></p>



Sno	Command	Test Application	Description
2	<p><b>Screen.Width</b></p> 	<p>Showmessage "Screen Width: "+str\$(screen.width)</p>	<p>This is a numerical variable of the entire screen's width. It is a number, so it must be converted to be displayed. Notice how we add two text strings to form a sentence;  <b>"Screen.Width: "+str\$[...]</b></p>
3	<p><b>Screen.MouseX Screen.MouseY</b></p>  <p>This will display the position of the mouse cursor related to the entire screen (not just the QFORM). We will use a timer (TYD) to detect the position of the mouse, every 5<sup>th</sup> of a second (interval=5). Once the timer sub program is called (displaymousepos), then the value of the MOUSEX and MOUSEY will be assigned to the string value WOORD. The WOORD's string value will then be assigned to the LABEL caption.</p>	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; declare sub displaymousepos <b>dim woord as string</b> dim tyd as QTimer     tyd.interval=5     tyd.ontimer=displaymousepos  CREATE Form AS QFORM     Caption = "Form":Width = 400:Height = 300     Center     create label as qlabel         left=50:top=50     end create END CREATE tyd.enabled=1 SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal  sub displaymousepos <b>woord=str\$(screen.mousex)</b> <b>woord=woord+" / "+str\$(screen.mousey)</b> <b>label.caption=woord</b> end sub         </pre>	<p>Start sub-routine  Establish string.  Start timer (no onmousemove command), needs a timer to refresh mouse position.  Property settings for QFORM.  Create QLABEL that will display position of mouse on screen.</p> <p>Start the timer</p> <p>Start sub-program  Assign numerical value of <b>MOUSEX</b> to <b>WOORD (str\$())</b>.  Assign the value of <b>WOORD</b> add, " / " text and then add the value of <b>MOUSEY</b> as a string (<b>str\$()</b>)  Assign the value of <b>WOORD</b> as <b>LABEL .CAPTION</b></p>



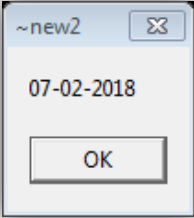
Sno	Command	Test Application	Description
4	<b>CurDir\$</b>  	\$TYPECHECK ON \$include "rapidq2.inc" <b>Showmessage CurDir\$</b>	This string variable will determine and hold the value of the <b>current directory</b> that the computer is <b>focused on</b> (current active system directory for instance <b>C:\MS3</b> This directory may be altered using <b>CD "..."</b>
5	<b>Application.Path</b>  	\$TYPECHECK ON \$include "rapidq2.inc" <b>Showmessage Application.Path</b>	This will indicate the <b>path</b> where the current program ( <b>EXE file</b> ) is <b>running</b> from. Without the EXE program file name.
6	<b>Application.ExeName</b>  	\$TYPECHECK ON \$include "rapidq2.inc" <b>Showmessage application.EXENAME</b>	This will identify and hold the name of the current EXE program (without the path).
7	<b>Date\$</b>  	\$TYPECHECK ON \$include "rapidq2.inc" <b>Showmessage date\$</b>	This will identify and hold the <b>current date</b> (the computer system date) <b>as a string variable (\$</b> represents a string). You will notice it is in the format of <b>mm/dd/yyyy</b> with mm Month dd Day yyyy Year.
8	<b>Time\$</b>  	\$TYPECHECK ON \$include "rapidq2.inc" <b>Showmessage time\$</b>	This will identify and hold the <b>string value of the current system's time</b> . The format is in <b>hh:mm:ss</b>



### 7.1.3 STRING MANIPULATION

String manipulation entails the alteration of a string value so to fit the required string. An example is for instance the **date\$** string.

To swap the day and month so that the **date\$** will read dd-mm-yyyy you will code:

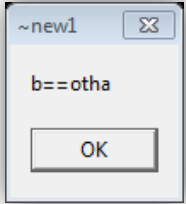
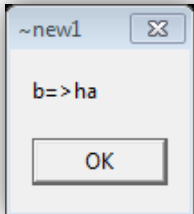
<pre>\$TYPECHECK ON #include "rapidq2.inc" dim woord as string woord=mid\$(date\$,4,2)+"-"+left\$(date\$,2)+"-"+right\$(date\$,4) showmessage woord</pre>	<p>The following order of the string value is shifted to read dd-mm-yyyy instead of mm-dd-yyyy. Remember that reserved variables (7.1.2) cannot be altered, they are read only (can only read the value, not write to or alter it). Therefore we create a string that will hold the new combination of the manipulated string.</p>
	

The following command manipulates a string's value:

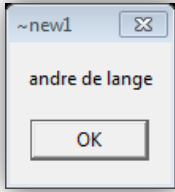
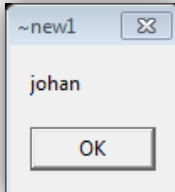
Sno	Command	Test Application	Description
1	<pre>+</pre> <pre>chr\$(13)</pre>	<pre>\$TYPECHECK ON #include "rapidq2.inc" dim woord as string dim ekstra as single ekstra=86 woord="Today is "+date\$+" and the time "+time\$ woord=woord+chr\$(13)+"I am "+str\$(ekstra)+" years old" showmessage woord</pre>	<p>The <b>+</b> is used to add strings to become a "sentence".</p> <p><b>CHR\$(13)</b> is used to start a sentence on a new line.</p> <p>To <b>continue adding to a string</b>, you repeat the string <b>woord=woord+</b></p> <p>The reason for <b>EKSTRA</b> to be within <b>STR\$( )</b> is that it is a <b>numerical value</b>. Remember the <b>spaces</b> between the strings to be added.</p>



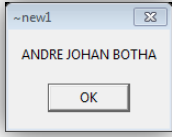
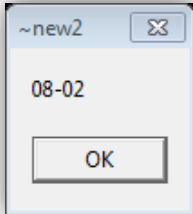
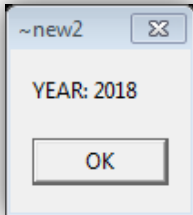


Sno	Command	Test Application	Description
2	<p><b>Insert\$(?,...,#)</b></p>  <p>You may insert a TAB: Insert\$(chr\$(9),</p>	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; declare sub checknow(key as byte) dim woord as string  CREATE Form AS QFORM Caption = "Form".Width = 400:Height = 300 Center create label as qlabel left=50:top=50:caption="SURNAME" end create create edit1 as qedit left=60:top=65:onkeyup=checknow end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal  sub checknow(key as byte) if key=13 then <b>woord=insert\$["==",edit1.text,2]</b> <b>showmessage woord</b> end if end sub                     </pre>	<p>This function is used to <b>insert the indicated ?</b> string within <b>identified</b> string at <b>position #</b>. The <b>first position</b> within the string is <b>0 (zero)</b>. <b>You may insert any text or number and even a sentence.</b> We will now create an <b>QEDIT</b> field for the user to fill in his/her surname. When the user presses <b>(ENTER)</b> the sub-program will be called whereby we will insert <b>==</b> at position 2 (actually position 1 as we count the first position from 0 zero).</p> <p>If enter was pressed <b>(13)</b>. Assign the following value to string <b>WOORD...</b> insert <b>==</b> within the result of <b>edit1.text</b> at <b>position 2</b>.</p>
3	<p><b>Replace\$(...?,#)</b></p>  <p>To replace characters within nothing (delete the characters) use the command "" only (no text between the two “.</p>	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; declare sub checknow(key as byte) dim woord as string  CREATE Form AS QFORM Caption = "Form".Width = 400:Height = 300 Center create label as qlabel left=50:top=50:caption="SURNAME" end create create edit1 as qedit left=60:top=65:onkeyup=checknow end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal  sub checknow(key as byte) if key=13 then <b>woord=replace\${edit1.text,"=&gt;".2}</b> <b>showmessage woord</b> end if end sub                     </pre>	<p>This function will replace a portion of a string as identified at the position # (the first position is 0 zero).</p> <p>Replace the characters within <b>EDIT1.TEXT</b> from position <b>2</b> with the text <b>=&gt;</b></p>

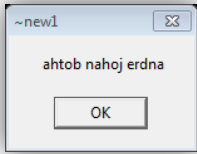


Sno	Command	Test Application	Description
4	<b>ReplaceSubStr\$( )</b> 	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidG2.inc&gt; declare sub checknow(key as byte) dim woord as string CREATE Form AS QFORM   Caption = "Form".Width = 400:Height = 300   Center   create label as qlabel     left=50:top=50:caption="SURNAME"   end create   create edit1 as qedit     left=60:top=65:onkeyup=checknow     <b>text="andre botha"</b>   end create END CREATE  SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal  sub checknow(key as byte) if key=13 then   <b>woord=replacesubstr\$(edit1.text,"botha","de lange")</b>   showmessage woord end if end sub </pre>	<p>This function will replace a specific string combination with the text as identified.</p> <p>Assign some text to the QEDIT as we wish to change the surname of <b>BOTHA</b> to <b>DE LANGE</b>.</p> <p>Change <b>Botha</b> as <b>De Lange</b>.</p>
5	<b>Field\$(...?,#)</b> 	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidG2.inc&gt; declare sub checknow(key as byte) dim woord as string  CREATE Form AS QFORM   Caption = "Form".Width = 400:Height = 300   Center   create label as qlabel     left=50:top=50:caption="SURNAME"   end create   create edit1 as qedit     left=60:top=65:onkeyup=checknow     <b>text="andre johan botha"</b>   end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal  sub checknow(key as byte) if key=13 then   <b>woord=field\$(edit1.text,"",2)</b>   showmessage woord end if end sub </pre>	<p>This function will return a <b>field separated by a specific character</b> (such as a space, etc). Now with the function, it will <b>return the section</b> until the indicated number <b>#</b> of that character has been met. This is an easy method of returning only the middle name and not full names of a person.</p> <p>In reality the following happens: <b>andre-johan-botha</b>. We will search for the second - thus it will indicate johan only as it is between the 1<sup>st</sup> and second character as identified.</p>

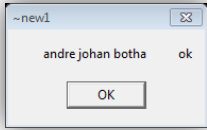
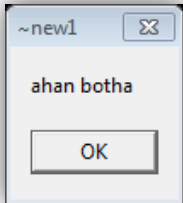


Sno	Command	Test Application	Description
6	<p><b>LCASE\$()</b> <b>UCASE\$()</b></p> 	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; declare sub checknow(key as byte) dim woord as string  CREATE Form AS QFORM Caption = "Form":Width = 400:Height = 300 Center create label as qlabel left=50:top=50:caption="SURNAME" end create create edit1 as qedit left=60:top=65:onkeyup=checknow text="andre johan botha" end create END CREATE SetWindowLong[Form.Handle, -8, 0] SetWindowLong[Application.Handle, -8, Form.Handle] Form.ShowModal  sub checknow(key as byte) if key=13 then woord=ucase\$(edit1.text) showmessage woord end if end sub                     </pre>	<p>The LCASE\$ will alter all text to lower case and UCASE\$ will alter all text as upper case only.</p>
7	<p><b>Left\$(...,#)</b> <b>Right\$(...,#)</b> <b>Mid\$(...,?,#)</b></p>  	<pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" DIM WOORD AS STRING DIM jaar as single WOORD=mid\$(date\$,4,2)+"-"+left\$(date\$,2) SHOWMESSAGE WOORD jaar=val(right\$(date\$,4)) showmessage "YEAR: "+str\$(jaar)                     </pre>	<p>These function will extract text from a string as from the position indicated, from the LEFT, RIGHT or MIDDLE areas.</p> <p>We are determining the current year by reading the DATE\$ value, but only recording the last 4 characters which is the year. We use VAL() as we are assigning it to a numerical variable.</p> <p>With MID\$, it is from position ? for # many characters. Remember that the first position is 0 (zero).</p>

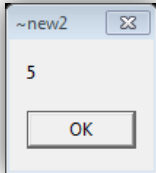
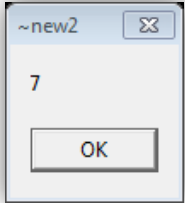
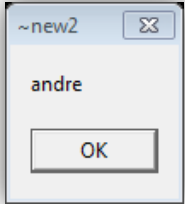


Sno	Command	Test Application	Description
8	LTrim\$(...) RTrim\$(...)	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; declare sub checknow(key as byte) dim woord as string CREATE Form AS QFORM   Caption = "Form":Width = 400:Height = 300   Center   create label as qlabel     left=50:top=50:caption="SURNAME"   end create   create edit1 as qedit     left=60:top=65:onkeyup=checknow     text=" andre johan botha "   end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal  sub checknow(key as byte) if key=13 then   woord=ltrim\$(edit1.text)   woord=rtrim\$(woord)   showmessage woord end if end sub </pre>	<p>These functions will remove all spaces before a string (LTRIM\$) and also trailing spaces (RTRIM\$) – after the string. When you start to work with databases, you will see that there is a lot of leading and trailing spaces.</p> <p>We assign the value of <b>EDIT1.TEXT</b> to <b>WOORD</b> whilst trimming away all <b>leading</b> spaces. Now the result is stored within <b>WOORD</b>, so we will now again remove the <b>trailing</b> spaces from <b>WOORD</b> and not <b>EDIT1.TEXT</b>.</p>
9	Reverse\$(...)	 <pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; declare sub checknow(key as byte) dim woord as string CREATE Form AS QFORM   Caption = "Form":Width = 400:Height = 300   Center   create label as qlabel     left=50:top=50:caption="SURNAME"   end create   create edit1 as qedit     left=60:top=65:onkeyup=checknow     text="andre johan botha"   end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal  sub checknow(key as byte) if key=13 then   woord=reverse\$(edit1.text)   showmessage woord end if end sub </pre>	<p>This function will reverse or mirror the indicated text – very useful when creating encrypted files.</p>

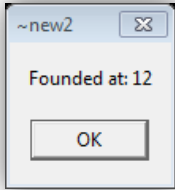
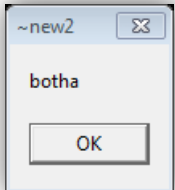
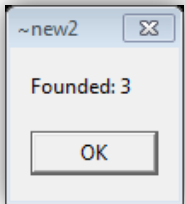


Sno	Command	Test Application	Description
10	<b>Space\$(?)</b> 	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; declare sub checknow(key as byte) dim woord as string CREATE Form AS QFORM Caption = "Form".Width = 400:Height = 300 Center create label as qlabel left=50:top=50:caption="SURNAME" end create create edit1 as qedit left=60:top=65:keyup=checknow text=" andre johan botha " end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal  sub checknow(key as byte) if key=13 then     woord=space\$(3)+(edit1.text)+space\$(5)+"ok"     showmessage woord end if end sub </pre>	<p>This function will <b>add spaces</b> as indicated by <b>?</b> before or after the identified text.</p>
11	<b>Delete\$(...,?,#)</b> 	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; declare sub checknow(key as byte) dim woord as string CREATE Form AS QFORM Caption = "Form".Width = 400:Height = 300 Center create label as qlabel left=50:top=50:caption="SURNAME" end create create edit1 as qedit left=60:top=65:keyup=checknow text="andre johan botha" end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) Form.ShowModal  sub checknow(key as byte) if key=13 then     woord=delete\$(edit1.text,2,7)     showmessage woord end if end sub </pre>	<p>This function will delete a part of a string as identified from to point.</p> <p>It will delete from position 2 till 7.</p>


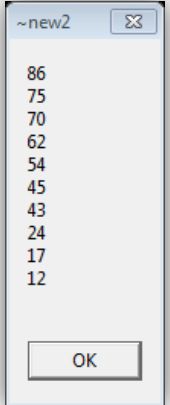


Sno	Command	Test Application	Description
12	String\$(?,...)	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; declare sub checknow(key as byte) dim woord as string CREATE Form AS QFORM   Caption = "Form":Width = 400:Height = 300   Center   create label as qlabel     left=50:top=50:caption="SURNAME"   end create   create edit1 as qedit     left=60:top=65:onkeyup=checknow     text="andre johan botha"   end create END CREATE SetWindowLong[Form.Handle, -8, 0] SetWindowLong[Application.Handle, -8, Form.Handle] Form.ShowModal  sub checknow(key as byte) if key=13 then   woord=string\$(6,"&gt;")+ " "+edit1.text   showmessage woord end if end sub </pre>	<p>This function returns a string <b>consisting out of an indicated repeated amount of a specified character</b>. In other words, it will take the indicated <b>character ?</b> and repeat it as <b>indicated ...</b></p>
13	Len(...)	 <pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" DIM WOORD AS STRING DIM jaar as single WOORD=mid\$(date\$,4,2)+"."+left\$(date\$,2) showmessage str\$(len(woord)) </pre>	<p>This will determine how long is the identified string – how many characters.</p> <p>LEN is a numerical variable therefore if you wish to display it, you need STR\$() the result.</p>
14	Instr(?, "...", "...")	  <pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" DIM WOORD AS STRING dim woord2 as string DIM found as single woord="andre botha" found=instr(0,woord,"botha") showmessage "Founded: "+str\$(found) woord2=left\$(woord,found-1) showmessage woord2 </pre>	<p>This function is used to <b>compare two strings and indicate the position within the main string where it is similar to the comparison string (from the left)</b>. The ? represents the starting point of comparison (from which character)</p> <p><b>Woord2</b> will now be holding the value of only the founded field before the similarity [-1 is required to indicate the character just before the founded match].</p>



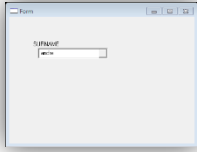
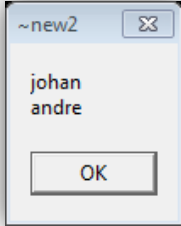
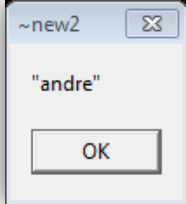
Sno	Command	Test Application	Description
15	RInstr(?, "...", "...")    	<pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" DIM WOORD AS STRING DIM found as single woord="andre johan=botha" found=rinstr(len(woord),woord,"=") showmessage "Founded at: "+str\$(found) showmessage right\$(woord,len(woord)-found)                     </pre>	<p>This is the same as INSTR, but instead of finding the position from the left, it will find the position of the matching character/text from the right. It is however important to indicate the starting position (?) <b>as from the right</b>. A simple method to do this is to use the <b>LEN</b> function. Notice the use of the <b>RIGHT\$</b> function and how we deduct the total <b>LEN</b> from the founded position to only display the matched <b>RINSTR</b>.</p>
16	Tally("...", "...")  	<pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" DIM WOORD AS STRING dim woord2 as string DIM found as single woord="andre botha, johan botha, leonie botha" woord2="botha" found=tally(lcase\$(woord),"botha") showmessage "Founded: "+str\$(found)                     </pre>	<p>This function will count the number of occurrences of two strings. Be careful however as this is case sensitive, so if you don't want the font's case to have an influence, make use of the <b>LCASE\$()</b> function to standardize both strings. You could also have said; <b>Tally(woord,woord2)</b></p>
17	Const as integer Const as string	<pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" const woord3 as string="Hallo" const numbme as single=5 DIM WOORD AS STRING dim woord2 as string DIM found as single woord="andre botha, johan botha, leonie botha" woord2="botha" showmessage woord3+" "+woord2 showmessage str\$(numbme)+" times"                     </pre>	<p>This will <b>create a fixed variable</b> (either numerical or string). It <b>cannot be altered</b> later on. It is therefore key words that will evaluate conditions without being able to be altered such as <b>TRUE=1</b> and <b>FALSE=0</b></p>



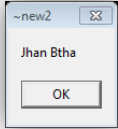
Sno	Command	Test Application	Description
18	<b>QuickSort(?,?,?,#)</b>  # ascend #descend  ? from position ? till position  	<pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" dim i as single dim woord as string <b>dim name(10) as string</b> name(0)="Johan":name(1)="Andre" name(2)="Milda":name(3)="Zorro" name(4)="Leonie":name(5)="Aniska" name(6)="Ricardo":name(7)="Chloe" name(8)="Joyce":name(9)="Neels" <b>quicksort name(0),name(9),ascend</b> woord="" for i = 0 to 9     <b>woord=woord+name(i)+chr\$(13)</b> next i showmessage woord                     </pre>	This function <b>will sort multi-level variables</b> in <b>ascending</b> or <b>descending</b> . <b>Create</b> multi-level variable. <b>Name</b> each variable [no alphabetical following order] <b>Sort</b> alphabetical following ascending. Clear <b>WOORD</b> . <b>Loop</b> the <b>9</b> entries and add to the <b>WOORD</b> string with an <b>ENTER (next line)</b> Display value of <b>WOORD</b> .  Remember multi-level variables? <b>Dim A(50)</b> as string means that <b>A</b> has <b>50</b> values <b>A1, A2</b> , etc. It may either be numerical or string (alphabetical).
		<pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" dim i as single dim woord as string <b>dim age(10) as single</b> age(0)=54:age(1)=43 age(2)=24:age(3)=12 age(4)=17:age(5)=45 age(6)=62:age(7)=75 age(8)=70:age(9)=86 <b>quicksort age(0),age(9),descend</b> woord="" for i = 0 to 9     <b>woord=woord+str\$(age(i))+chr\$(13)</b> next i showmessage woord                     </pre>	<b>Quicksort Numerical Variables</b>
19	<b>LBound (?)</b> <b>HBound (?)</b>	<pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" dim i as single dim woord as string <b>dim name(-50 to 100) as string</b> <b>showmessage str\$(lbound(name))</b> <b>showmessage str\$(ubound(name))</b>                     </pre>	This will determine the <b>lowest</b> and <b>highest dimension settings</b> for a <b>multi-level variable</b> . <b>Not the actual</b> values of the variables, but the <b>lowest and highest settings</b> . This is important for any <b>looping</b> conditions.





Sno	Command	Test Application	Description
20	<b>Data "...", "..."</b> <b>Read ?</b> 	<pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; dim i as single dim name as string data "andre","milda","aniksa","ricardo","chloe" CREATE Form AS QFORM Caption = "Form".Width = 400:Height = 300 Center create label as qlabel left=50:top=50:caption="SURNAME" end create create combo1 as qcombobox left=60:top=65:style=3 end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) for i = 0 to 4 read name combo1.additems name next i combo1.itemindex=0 Form.ShowDialog                     </pre>	<p>Data is a non-executional statement that stores pre-determined (fixed) information to be held by the variable as read in by the READ command. This is usually configuration data used by your program as fixed information or set rules and regulations regarding the execution of the program. The data will be read in following order as they were entered.</p> <p>This program will read the data and add to the ComboBox.</p>
21	<b>Swap ???</b> 	<pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" dim i as single dim woord as string woord="andre" dim woord2 as string woord2="johan" swap woord,woord2 showmessage woord+chr\$(13)+woord2                     </pre>	<p>This function will <b>swap the values of the two identified variables</b>. Take note they must be of the <b>same format</b> (numerical or string).</p>
22	<b>Chr\$(#)</b> 	<pre> \$TYPECHECK ON \$INCLUDE "RAPIDQ2.INC" showmessage chr\$(34)+"andre"+chr\$(34)                     </pre> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>This program will indicate all CHR\$ character with their respective code:</b></p> <pre> \$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; dim i as single CREATE Form AS QFORM Caption = "CHR\$ codes MS3 (Pty)" Width = 250:Height = 550 Center:color=0 create list1 as qlistbox left=20:top=10:width=200:height=490 end create END CREATE SetWindowLong(Form.Handle, -8, 0) SetWindowLong(Application.Handle, -8, Form.Handle) for i = 0 to 255 list1.additems str\$(i)+" "+chr\$(i) next i Form.ShowDialog                     </pre> </div>	<p>This will display a character from the character (ASCII) map.</p>



Sno	Command	Test Application	Description
23	<b>ASC(\$)</b>  It is a numerical variable so you need the STR\$( ) to display the value.	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim i as single dim woord as string  <b>showmessage "ASCII code: "+str\$(asc("a"))</b>	This is a string processing function that returns a numeric value that is the ASCII code for the first character in a string expression. <b>This is actually only the reverse of the CHR\$ function.</b> It indicates the ASCII value of a character.
24	<b>STRF\$(#,%,&amp;,!)</b>  <i>What is important is that this function is actually a <b>string variable</b> even though a number is converted. Therefore to allocate the result you need to <b>assign a numerical variable using VAL()</b></i>	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim a as string dim b as double <b>a=strf\$(150.3563,ffgeneral,3,0)</b> <b>b=val(a)</b> <b>showmessage str\$(b)</b>  \$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim a as string dim b as double <b>a=strf\$(150.3563,ffgeneral,5,0)</b> <b>b=val(a)</b> <b>showmessage str\$(b)</b>	This is a <b>conversion function that returns a formatted string representation of the value of a numerical expression.</b>  <b>The layout of the command (#,%,&amp;,!)</b> entails: <ul style="list-style-type: none"><li>• # This represents the number to be converted.</li><li>• % Represents the format (below)</li><li>• &amp; Represents the precision that specifies how many decimal places to be calculated (total characters after .)</li><li>• ! Represents the digits for instance 200.99</li></ul> <b>The possible formats:</b> <ul style="list-style-type: none"><li>• <b>Ffgeneral</b> Converts to the shortest possible decimal string where trailing zeros are removed.</li><li>• <b>Ffexponent</b> Converts to a scientific notation of the form.</li><li>• <b>Fffixed</b> to fixed point format of the form.</li><li>• <b>Ffnumber</b> Converts to a number format of the form.</li></ul>
25	<b>[?]</b>	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim woord as string woord="andre" <b>showmessage woord[2]</b>	This will determine and indicate the <b>character</b> at the position ? <b>within the assigned string.</b>
26	-  	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim woord as string woord="Johan Botha" <b>showmessage (woord-"o")</b>	This will remove the character as indicated from a string.



## 7.1.4 NUMERICAL MANIPULATION (Math's)

Numerical manipulation entails the alteration of a numerical value so complete a mathematical solution. Operators perform mathematical or logical operations on values. They are usually encompassed by expression for example;  $2 * 8$  is a valid expression, and  $*$  is an operator operating on values 2 and 8.

The highest precedence operator will always execute before any lower precedence operator.

Here are the prioritizing levels:

- Brackets [ ... ]
- Division [ / ]
- Multiplication [ \* ]
- Adding [ + ]
- Deducting [ - ]

Mathematical procedures can be divided into three main groups:

- Simple (as mentioned above);  $*/+-$
- Advanced (scientific and graphical programs); *sin, cos, etc*
- Formulas (databases and games); *speed=d/v*

Valid Arithmetic Operators:

Sno	Operator	Character Used	Example	Description
1	Exponentiation	^	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(2^6)</b>	<b>Calculates the power of a number.</b>
2	Negation	-	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(-91+12)</b>	This is a <b>negative number</b> .
3	Multiplication	*	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(2*6)</b>	<b>Multiply numbers.</b>
4	Division	\	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(6/2)</b>	<b>Divide numbers.</b> Amount is rounded off.
5	Floating Point Division	./.	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(6.5/2.3)</b>	<b>Divide floating points.</b> – Amount is not rounded off - has decimal numbers.
6	Left Bit Shift	SHL	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(10 shl 2)</b>	Shift bits left by amount specified. <b>It is to duplicate every result with itself</b> (10 = 20 = 40 =80 = 160 ...)



Sno	Operator	Character Used	Example	Description
7	Right Bit Shift	SHR	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> showmessage str\$(100 shr 3)	Shifts bits right by amount specified (it is like dividing by 2 with every results ... 100 = 50 = 25 = 12 = 6 ...)
8	Modulus	Mod	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> showmessage str\$(55 mod 10)	This will return the remainder of the division. You would have used the \ to divide 55 by 10 (5.5) however now you use the mod function to have the result of the remainder (5).
9	Inverse Modulus	Inv	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> showmessage str\$(3 inv 26)	Returns the inverse of a number in modulus.
10	Addition	+	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> showmessage str\$(3+6)	Adding numbers together.

Relation Operators:

Sno	Operator	Character Used	Example	Description
1	Equality	=	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim age as single dim age2 as single dim woord as string dim woord2 as string woord="andre":woord2="johan" age=75:age2=54 if woord= "andre" then showmessage "its him" end if	Tests for equality between 2 operands (strings included)
2	Inequality	<>	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim age as single dim age2 as single dim woord as string dim woord2 as string woord="andre":woord2="johan" age=75:age2=54 if woord<>woord2 then showmessage "its not him" end if	Test for inequality between 2 operands (strings included)
3	Less than	<	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim age as single dim age2 as single dim woord as string dim woord2 as string woord="andre":woord2="johan" age=75:age2=54 if age2<age then showmessage "she is younger" end if	Tests if operand is less than another (strings included)



Sno	Operator	Character Used	Example	Description
4	Greater than	>	<pre>\$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; dim age as single dim age2 as single dim woord as string dim woord2 as string woord="andre":woord2="johan" age=75:age2=54 if age&gt;age2 then   showmessage "she is older" end if</pre>	Tests if operand is greater than another (strings included)
5	Less than or equal	<=	<pre>If age &lt;= age2 then End If</pre>	Tests if operand is less than or equal to another (strings included).
6	Greater than or equal	>=	<pre>If age &gt;= age2 then End if</pre>	Tests if operand is greater than or equal to another (strings included).
7	And	AND	<pre>\$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; dim age as single dim age2 as single dim woord as string dim woord2 as string woord="andre":woord2="johan" age=75:age2=54 if age&gt;age2 and woord&lt;woord2 then   showmessage "this is a match" end if</pre>	Compare corresponding bits in 2 operands - when you compare values of multiple variables.
8	Or	OR	<pre>\$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; dim age as single dim age2 as single dim woord as string dim woord2 as string woord="andre":woord2="johan" age=75:age2=54 if age&gt;age2 or woord=woord2 then   showmessage "this is a match" end if</pre>	Compare corresponding bits in 2 operands - if condition is valid with and or another condition.

**Advanced Mathematics (all numerical variables used with these functions must be dimmed as DOUBLE):**

Sno	Operator	Character Used	Example	Description
1	ACOS	ACOS()	<pre>\$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; showmessage str\$(acos(0.55))</pre>	Function that returns the arccosine of a numeric expression.
2	ASIN	ASIN()	<pre>\$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; showmessage str\$(asin(0.55))</pre>	Function that returns the arcsine of a numeric expression.
3	ATN/ATAN	ATN()	<pre>\$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; showmessage str\$(atn(0.9))</pre>	Numeric expression is the angle expressed in radians.
4	COS	COS(pi)	<pre>\$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; showmessage str\$(cos(3.14153))</pre>	Numeric expression is the angle expressed in radians. <i>PI=3.14153</i>



Sno	Operator	Character Used	Example	Description
5	Log	LOG()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(log(10))</b>	Returns the natural logarithm of a numeric expression.
6	SIN	SIN(pi)	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(sin(3.14153))</b>	Numeric expression is the angle expressed in radians.
7	TAN	TAN()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(tan(90))</b>	Numeric expression is the angle expressed in radians.
8	EXP	EXP()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(exp(1))</b>	Returns the exponential function (raised to the power of n - ... = exp[n])
9	SQR	SQR()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> <b>showmessage str\$(Sqr(9))</b>	Returns the square root of a number.

The following command manipulates a numerical value:

Sno	Command	Test Application	Description
1	ABS()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim nommer as single nommer=abs(-125) showmessage str\$(nommer)	This math function will return the absolute value of a numeric expression. <b>It will therefore turn any negative (-) number and make it a positive expression.</b>
2	CEIL()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim nommer as single nommer=ceil(66.2) showmessage str\$(nommer)	This math function rounds a numeric expression up towards positive infinity.
3	CINT()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim nommer as single nommer=cint(41.2) showmessage str\$(nommer)	Conversion function that converts a numeric expression (single or double) to an INTEGER by rounding the fractional part of the expression.
4	FIX()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim nommer as single nommer=fix(342.97) showmessage str\$(nommer)	<b>A function that removes the fractional part of a number</b> (the fractional part of a number is the numbers beyond the . for example 397.23 – the fraction is .23) It only removes it, not round it off.
5	FRAC()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim nommer as single nommer=frac(342.97) showmessage str\$(nommer)	Returns only the fractional part of a number (397.23 = .23)
6	FLOOR()	\$TYPECHECK ON \$INCLUDE <RapidQ2.inc> dim nommer as single nommer=floor(342.77) showmessage str\$(nommer)	A math function that rounds a numeric expression down towards negative infinity. It will always round off to the lower value.



Sno	Command	Test Application	Description
7	ROUND()	<pre>\$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; dim nommer as single nommer=round(342.77) showmessage str\$(nommer)</pre>	This function will convert a number to an integer by rounding the fractional part of the expression (3.49 = 3 and 3.51 = 4)
8	SGN()	<pre>\$TYPECHECK ON \$INCLUDE &lt;RapidQ2.inc&gt; dim nommer as single nommer=sgn(-4) showmessage str\$(nommer)</pre>	<p>This function will indicate the sign of a numeric expression.</p> <ul style="list-style-type: none"> <li>• Positive = 1</li> <li>• Zero = 0</li> <li>• Negative = -1</li> </ul> <p>Basically it is to determine the value level, either positive, zero or negative.</p>

### 7.1.5 ESCAPE Characters

The command may be set to either **ON** or **OFF**. If it is turned **ON**, you can use escape sequences in your strings for instance instead of **CHR\$(13)** for a new line, you will use **\n**.

Escape sequences can either be a character or numeric escapes. Note that case is sensitive (caps on/off). Usage: **\$ESCAPECHARS ON**

The disadvantage of this method of coding is that all your text strings and assignments will be influenced as the program will assume you are coding escape sequences, such as if you state; **"Andre & Johan"**, it will display onscreen **ANDRE\_JOHAN**. To cancel that error, you must duplicate the character to avoid escape sequences **"ANDRE && JOHAN"**. This complicates coding. The following are escape sequences:

- \a                alarm bell
- \b                backspace created                                chr\$(8)
- \f                form feed
- \n                new line (enter)                                chr\$(13)
- \r                carriage return (hard enter)
- \t                horizontal tab                                    chr\$(9)
- \v                vertical tab
- \\                backslash
- \"                double quote                                    chr\$(34)
- \###             a character between key codes (ASNII) 0-255

Example	Description
<pre>\$TYPECHECK ON \$escapechars on \$INCLUDE &lt;RapidQ2.inc&gt; showmessage \"\" showmessage \"\t\65\66\67\t\70"</pre>	<p>Activate ESCAPE CHARS on.</p> <p>Display "</p> <p>Display a TAB with some letters and then a TAB again.</p>

