



دروس تعليم LabVIEW

الدرس الثامن

Strings and Files

الفهرس

iii	هدف الدرس
1	مقدمة
2	Strings
2	String Indicator و String Control
3	String Controls خصائص
3	طريقة اظهار النصوص
6	Limit Single Line خاصية
7	Update Value while Type خاصية
9	The Scrollbar شريط الانزلاق
11	Tables (الجدول)
15	Listboxes
16	نمط الاختيار (Selection Mode)
22	Strings الدوال الخاصة بـ
22	String Length الدالة
23	Concatenate Strings الدالة
24	Format Into String الدالة
25	Format String كيفية تحديد
31	Get Date/Time String الدالة
34	Format Date/Time String الدالة
36	Parsing Factions
36	String Subset الدالة
37	Scan from String الدالة
40	Match Pattern الدالة
50	File Input /Output
51	High Level File I/O اولاً
51	Write to Measurement File Express VI
56	Read from Measurement File Express VI

61.....	Write to Spreadsheet File VI
65.....	Read from Spreadsheet File VI
69.....	Text Spreadsheet الفرق بين ملفات Excel Spreadsheet وملفات
71.....	Low Level File I/O ثانيا
71.....	Text Files كيفية التعامل مع
71.....	Open/Create/Replace File الدالة
72.....	Close File الدالة
73.....	Write to Text File الدالة
77.....	Read from Text File الدالة
81.....	Binary كيفية التعامل مع الملفات من النوع
81.....	Write to Binary File الدالة
84.....	Read from Binary File الدالة

هدف الدرس

- التعرف على Strings وخصائصها.
- التعرف على الدوال الخاصة بالتعامل مع Strings.
- التعرف على كيفية تحويل الارقام الى Strings والعكس.
- التعرف على العناصر ListBox وTable.
- التعرف على File Path.
- التعرف على انواع مختلفة من الملفات وكيفية التعامل معها في LabVIEW.

مقدمة

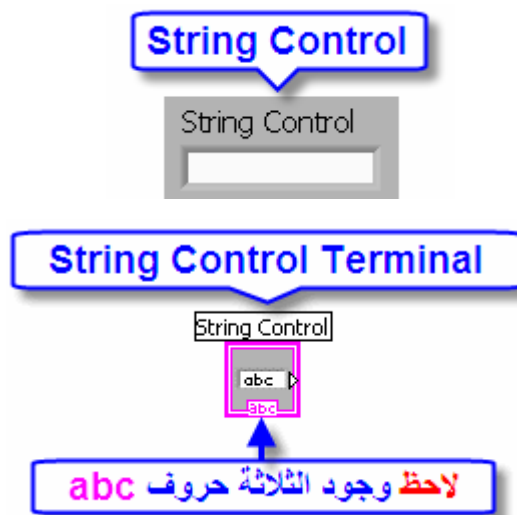
فى هذا الدرس سوف نتعرف على عنصر هام جدا فى لغات البرمجة وهو String (النص) وسوف نتعرف على كيفية التعامل مع Strings والدوال الخاصة بها فى LabVIEW وكيفية معالجتها ووضعها فى الشكل المناسب لارسالها او استقبالها من اجهزة القياس او لاي استخدامات اخرى. كما سنتعرف على كيفية تخزينها واسترجاعها من انواع مختلفة من الملفات مثل ملفات Text وملفات Binary . كما سنتعرف على نوعين جديدين من الملفات لتخزين بيانات القياس وهما LVM وTDM.

Strings

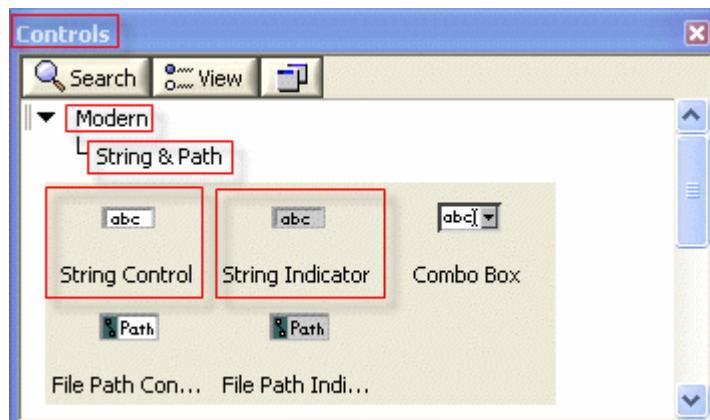
النص أو String هو عبارة عن مجموعة من الحروف (Characters) ويطلق عليها ASCII Characters. ويوجد لكل حرف كود (رقم خاص بكل حرف) يسمى ASCII Code ، فكما نعلم ان كل البيانات داخل الحاسب الالى تكون على هيئة ارقام .
وهذه الحروف تشمل الحروف الابجدية الصغيرة و الكبيرة و الارقام من 0 الى 9 و الرموز وايضا حرف غير قابله للاظهار او الطباعة مثل Space (المسافة) او New Line (سطر جديد) او Carriage Return و الحروف الغير قابلة للطباعة يكون لها تأثير على شكل الطباعة او اظهار النص .

ويوجد String Control و String Indicator و String Constant فى LabVIEW لاطهار و ادخال و التعامل مع Strings كما يوجد دوال كثيرة للتعامل مع Strings وسوف نتعرف على اهمها فى هذا الدرس .

String Indicator و String Control

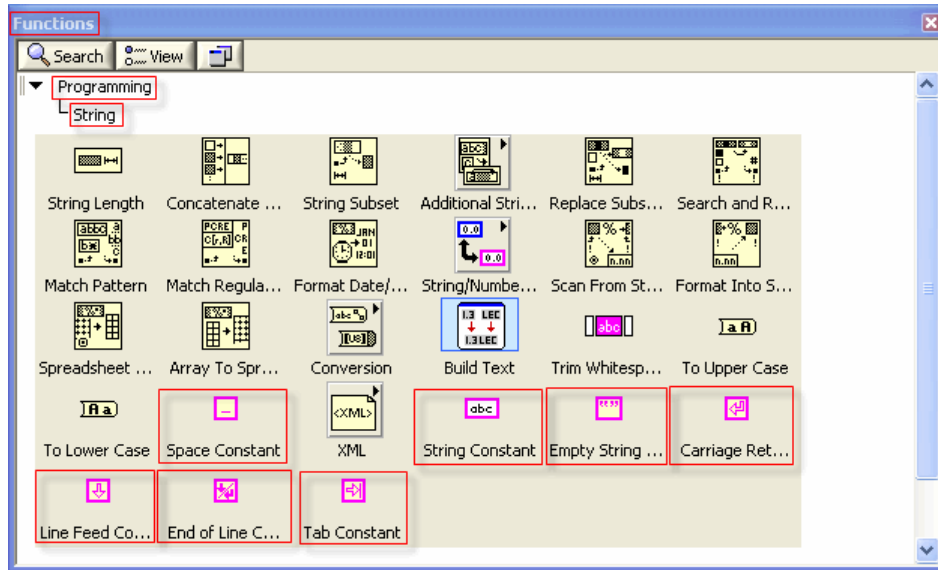


يتم ادراج String Indicator او String Control فى Front Panel من
Controls Palette>>Modern>>Strings & Path



ويتم ادراج String Constant فى Block Diagram من

Functions Palette>>Programming>>String



String Controls خصائص

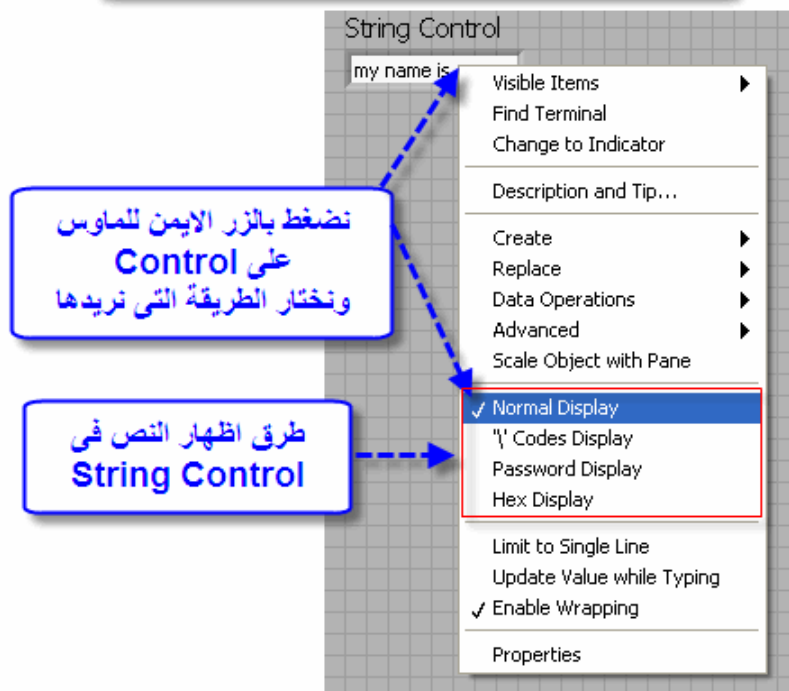
توجد لـ Strings عدة خصائص منها :

طريقة اظهار النصوص

يمكن اظهار النصوص فى String Control باكثر من طريقة .

ولتغيير طريقة اظهار النص فى String Control نضغط عليه بالزر الايمن للماوس و نختار الطريقة التى نريدها.

تغيير طريقة اظهار النص فى String Control



وهذه الطرق هي:

Normal Display

وفيهما يظهر النص بالصورة الطبيعية التي نعتاد عليها وهذه الطريقة هي الطريقة الأساسية.



“\” Code Display

وفيه تظهر الحروف الغير قابلة للطباعة على هيئة العلامة \ يليها رقم ASCII Code بالنظام السداسي عشر (Hex) لكل حرف .

مثال \20 يمثل المسافة (Space) حيث ان ASCII Code للمسافة هي 20 بالنظام السداسي عشر. وتوجد حروف غير قابلة للطباعة كثيرة الاستخدام تظهر في صورة العلامة \ يليها حرف .

مثال المسافة بدلا من ان تظهر بالصورة \20 تظهر بالصورة \s.

وهذا الجدول يوضح كيف تظهر الحروف الغير قابلة للطباعة في هذا النمط

العلامة \ وبعدها ASCII Code للحرف بالنظام السداسي عشر	\00 الى \FF
والحروف المكونة لـ ASCII Code حروف كبيرة مثال (\0C) او (\FF)	
(\08) ASCII Code يكافىء (BS) Backspace	\b
(\0C) ASCII Code يكافىء (FF) Form feed	\f
(\0A) ASCII Code يكافىء (LF) New Line	\n
(\0D) ASCII Code يكافىء (Cr) Carriage Return	\r
(\09) ASCII Code يكافىء (HT) Tab	\t
Space يكافىء (\20) ASCII Code	\s
(\5C) ASCII Code يكافىء (\) Backslash	\\

ملحوظات :

1- الحروف الخاصة مثل حرف s فى \s وحرف t فى \t تكون حرف صغيرة.

واذا كتبت كحروف كبيرة لن يتم تفسيرها كحروف خاصة .

مثال \b تعنى Backspace اما \B فهى لا تعنى Backspace و \B يعتبرها LabVIEW على

انها \0B ASCII Code.

2- حروف النظام السداسى عشر تكون حروف كبيرة مثل \FF او \0A.

مثال

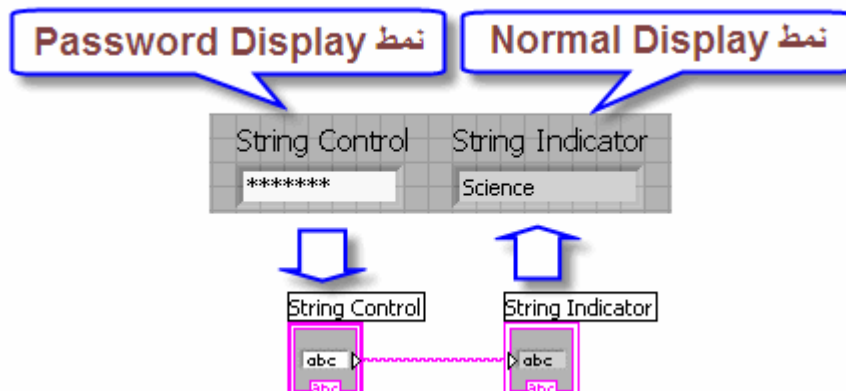


ملحوظات :

- 1- تغيير نمط الاظهار لا يؤثر على محتوى String.
- 2- يستخدم هذا النمط فى اختبار النصوص التى يتم ارسالها او استقبالها من اجهزة القياس عن طريق Serial Port مثلا ، حيث يتم استخدام حروف غير قابلة للطباعة مثل \n New line او \r Carriage Return لانهاى الاوامر و البيانات المرسله عند مخاطبة اجهزة القياس.

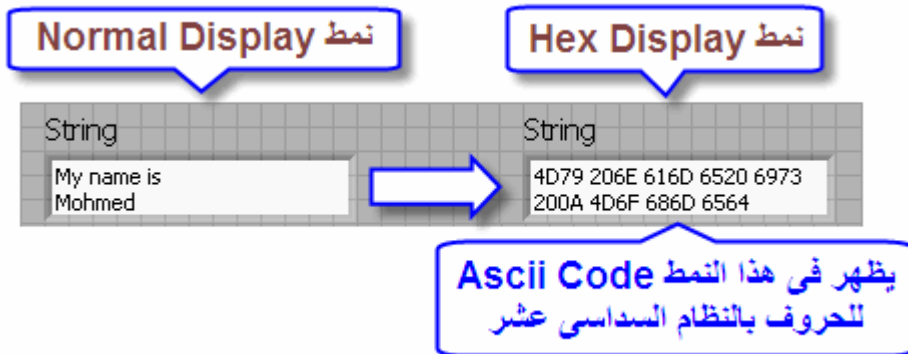
Password Display

وفى هذا النمط لا تظهر الحروف انما تظهر بدلا منها الرمز * .
فعند ادخال حروف فى String Control فى هذا النمط لا تظهر الحروف التى تكتبها وانما يظهر بدلا منها الرمز * ولكن القيمة التى يتم قراءتها من خلال String Control Terminal فى Block Diagram هى مجموعة الحروف التى ادخلتها.
ويستخدم هذا النمط عندما يستخدم String Control فى ادخال كلمات السر.



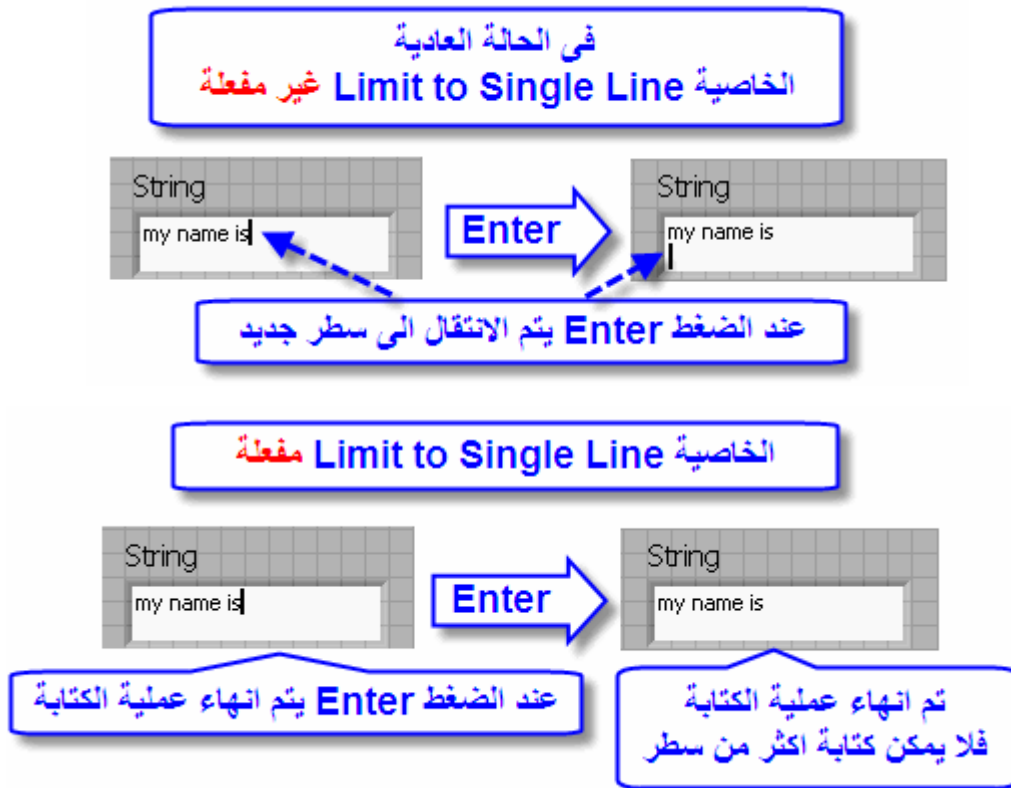
Hex Display

في هذا النمط يتم اظهار ASCII Code لكل الحروف الموجودة في النص بالنظام السداسى عشر بدلا من الحروف.



خاصية Limit Single Line

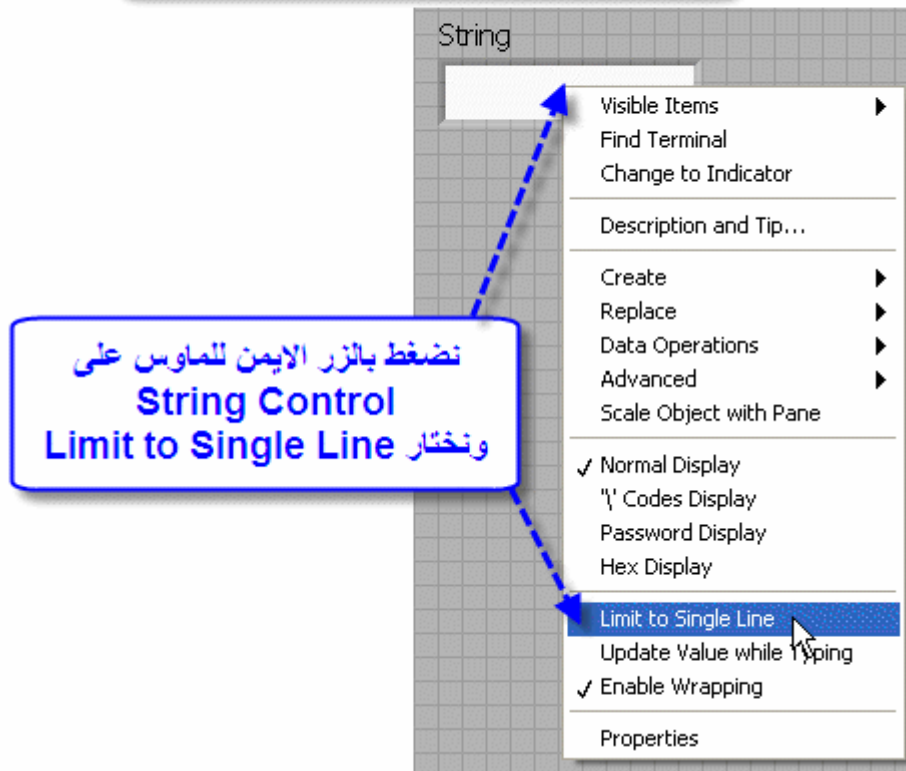
تحدد هذه الخاصية عدد السطور الممكنة في String Control بسطر واحد وعند ادخال الحروف وضغط Enter فهذا يعنى انه تم الانتهاء من ادخال السطر. اما عند عدم اختيار هذه الخاصية —وهذه هي الحالة العادية— و الضغط Enter اثناء ادخال الحروف الى String Control فانه يتم الانتقال الى سطر جديد في Control.



ولتنفيذ هذه الخاصية نضغط بالزر الايمن للماوس على String Control ونختار

Limit to Single Line

تفعيل خاصية Limit to Single Line



خاصية Update Value while Type

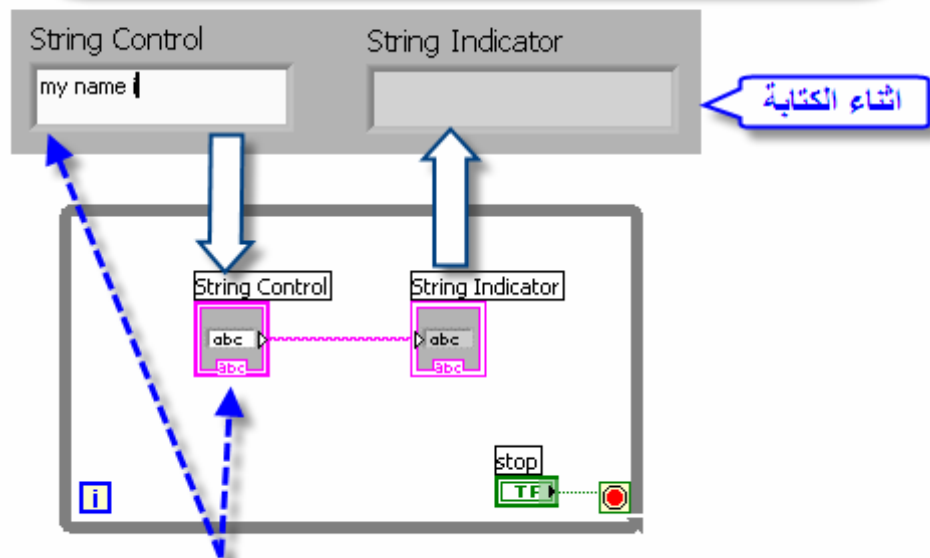
عند الكتابة على String Control لا يتم تغيير قيمة Terminals الخاصة بهم في Block Diagram الا بعد انتهاء عملية الكتابة وذلك سواء بالضغط Enter او بالضغط بالماوس على اى مكان اخر خارج Control او بالضغط على الزرار ☒ الموجود فى شريط الادوات.

وذلك حتى لا يتم قراءة النص غير كامل قبل الانتهاء من كتابته .

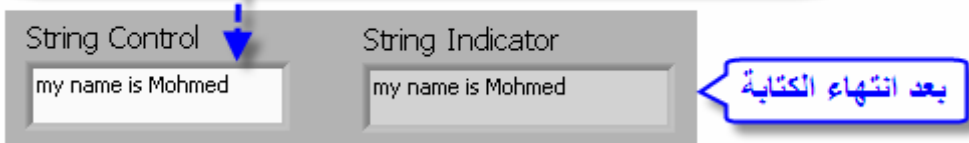
ولكن اذا اردت ان يتم تحديث قيمة Terminal اثناء الكتابة فاننا نفعل الخاصية Update Value

while Type

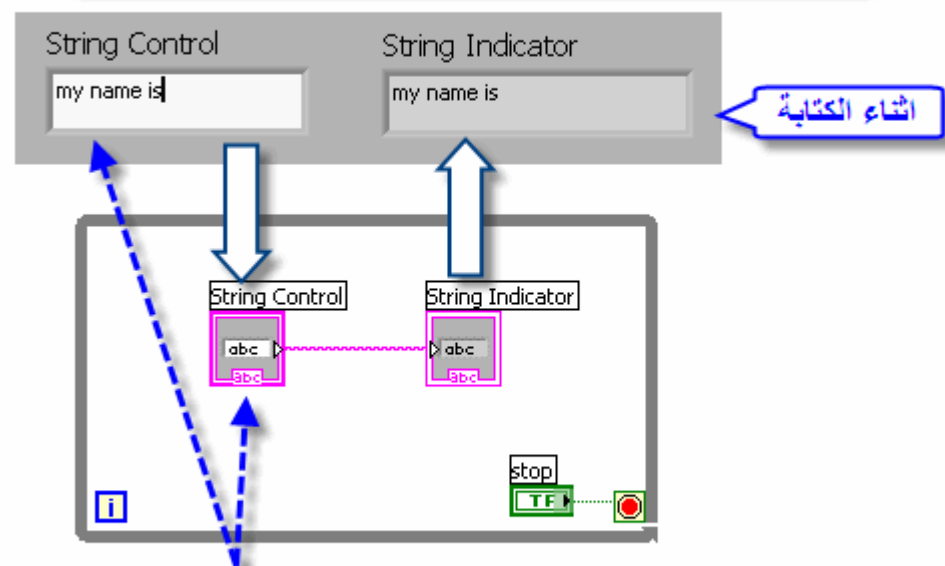
الحالة العادية
خاصية Update Value while Typing غير مفعلة



لا يتم تحديث Terminal الخاصة بـ String Control
الا بعد انتهاء الكتابة



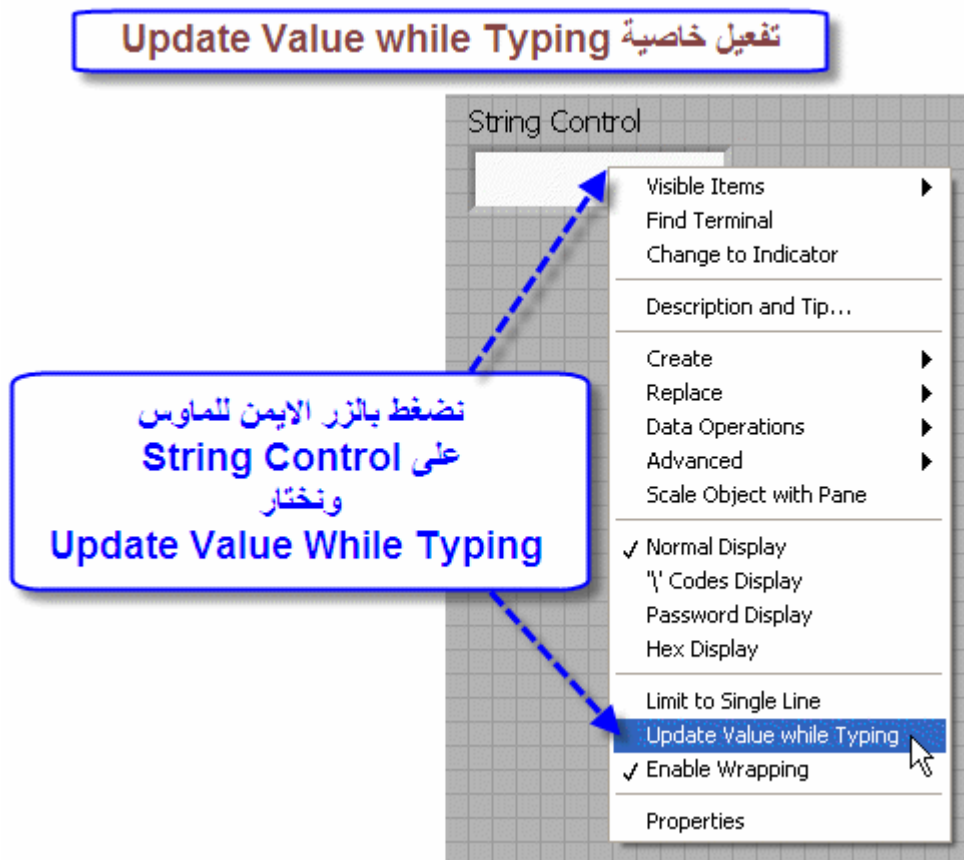
خاصية Update Value while Typing مفعلة



يتم تحديث Terminal الخاصة بـ String Control
بأي تغيير في قيمته أثناء الكتابة

ويتم تفعيل خاصية Update Value while Typing بالضغط بالزر الايمن للماوس على String

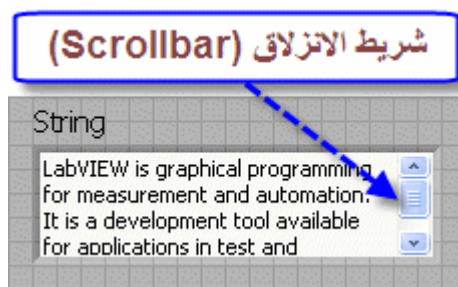
Control واختيار Update Value while Typing



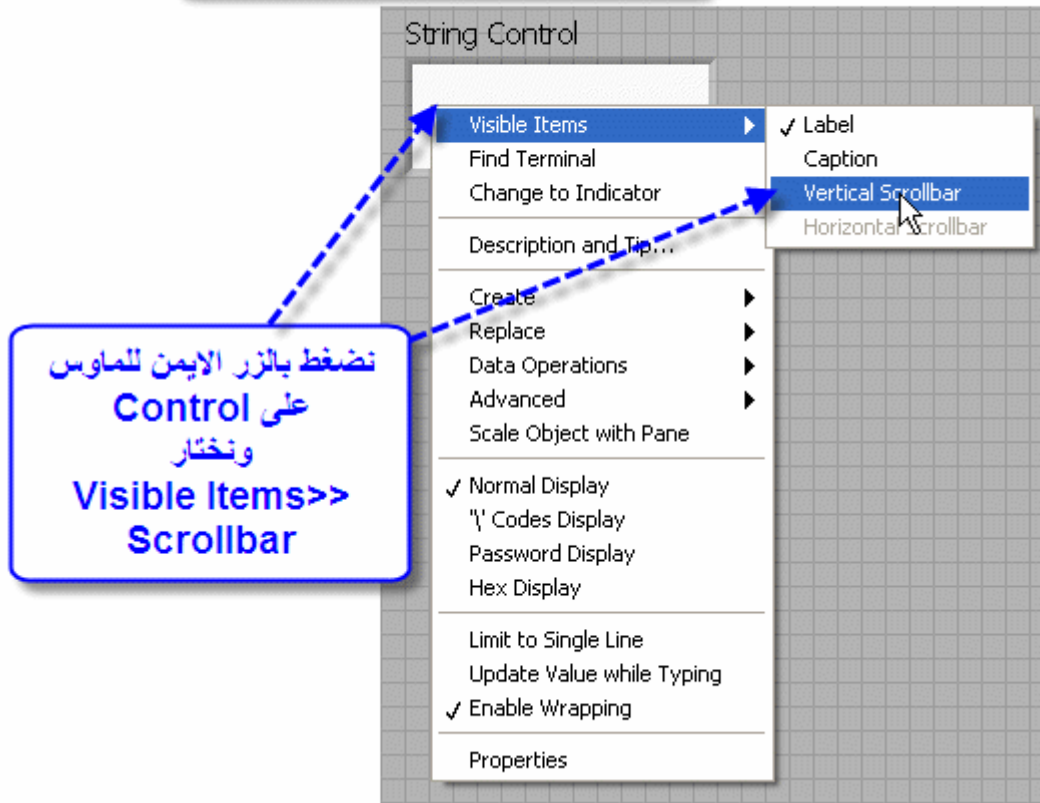
شريط الانزلاق The Scrollbar

يستخدم شريط الانزلاق لعرض النصوص الكبيرة في اقل مساحة من Front Panel. ويمكن اظهار شريط الانزلاق داخل String Control او String Indicator وذلك بالضغط بالزر الايمن للماوس على String واختيار

Visible Items>>Scrollbar



اظهار شريط الانزلاق (Scrollbar)



Tables (الجدول)

Table عبارة عن مصفوفة ذات بعدين من Strings يتم عرضها في هيئة صفوف واعمدة مثل اى مصفوفة.

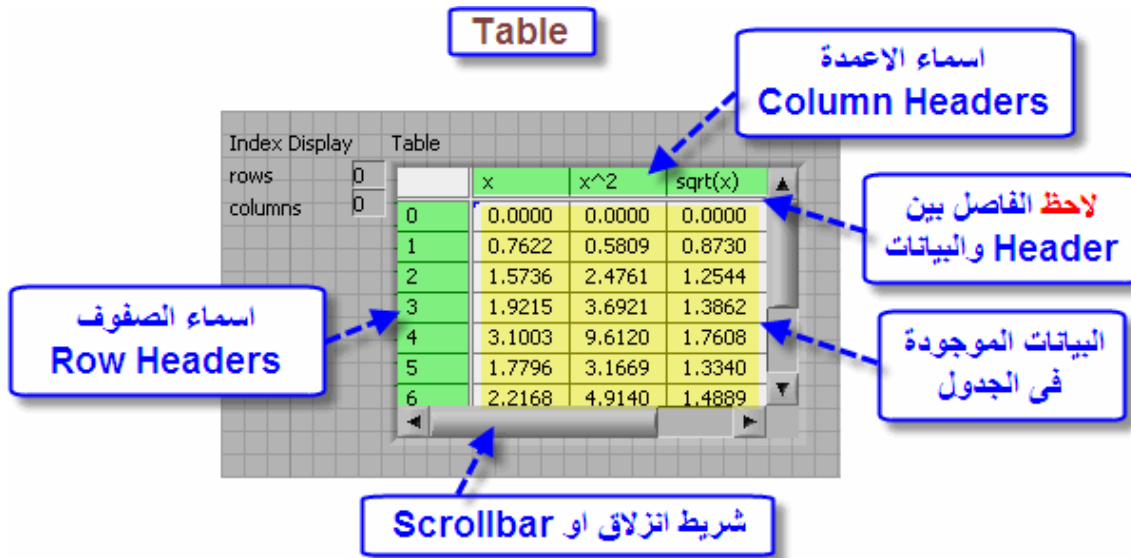
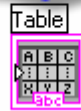
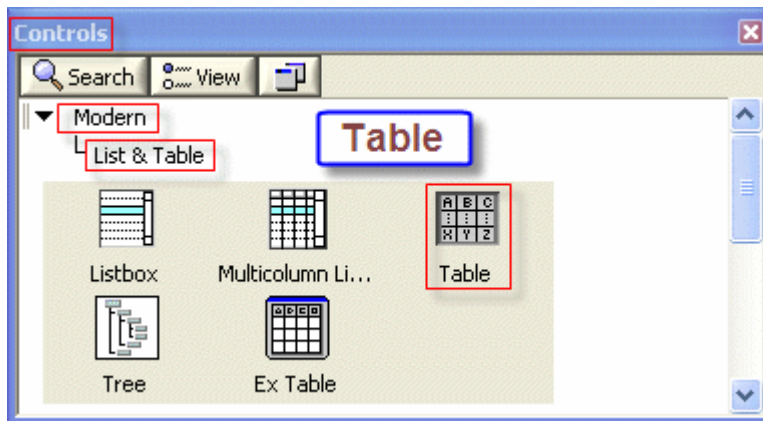


Table Terminal



ويتم ادراج Table Control او Table Indicator من

Controls Palette>>Modern>>List & Table



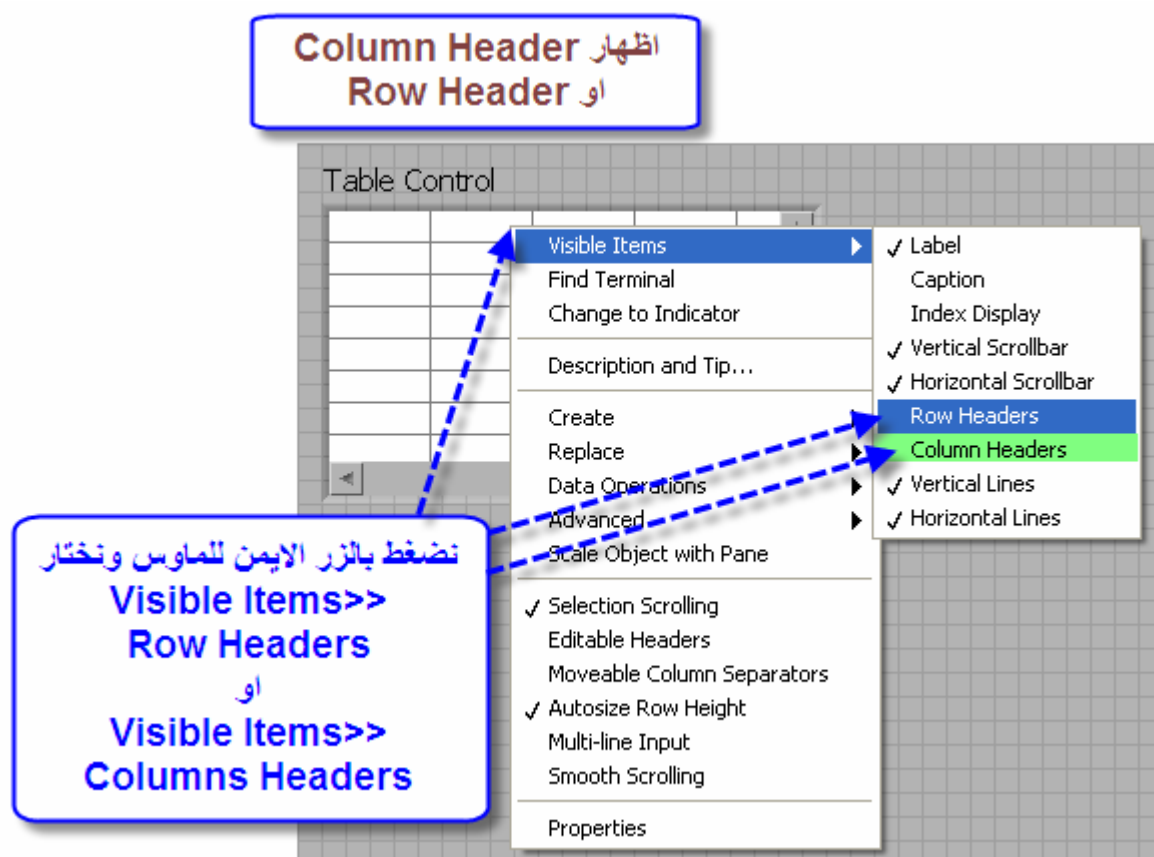
ويمكن اظهار Columns Header (لكتابة اسماء للاعمدة) وايضا Rows Header (لكتابة اسماء الصفوف) لـ Table ويتم ذلك بالضغط بالزر الايمن للماوس على Table واختيار

Visible Items>>Column Headers

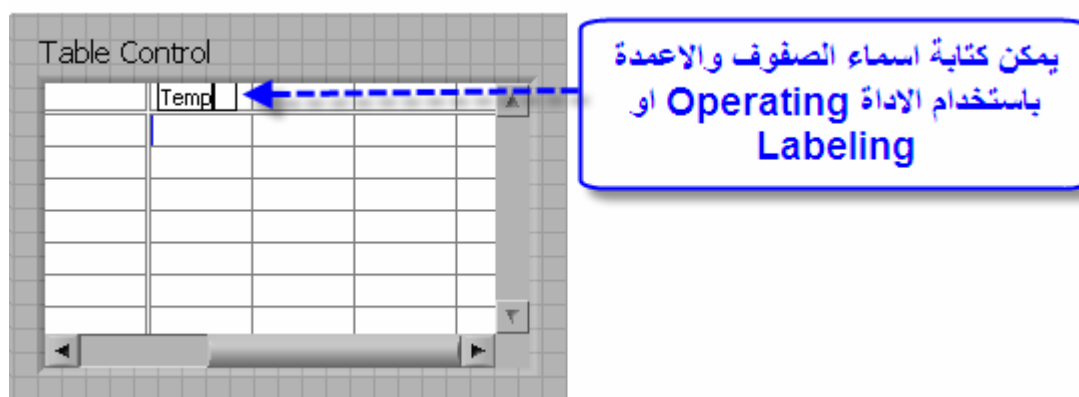
لاظهار Column Header او

Visible Items>>Row Headers

لاظهار Row Headers



وفصل بين Row Headers و Column Headers وباقي البيانات خطين متقاربين. ويمكن الكتابة او تعديل اسماء الصفوف و اسماء الاعمدة في Row Headers و Column Headers مباشرة باستخدام الاداة Operating او Labeling مثل اى Label اخر فى .LabVIEW



كما يمكن تحديد او قراءة اسماء الصفوف والاعمدة من خلال Block Diagram باستخدام .Property Node

تحدد اسماء الاعمدة من خلال Block Diagram

Table Control - Property Node لكتابة اسماء الاعمدة

مصفوفة ذات بعد واحد من Strings
بها اسماء الاعمدة

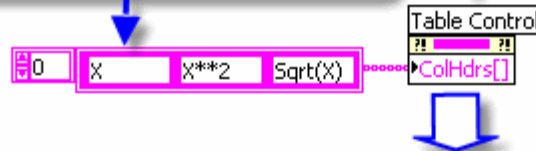


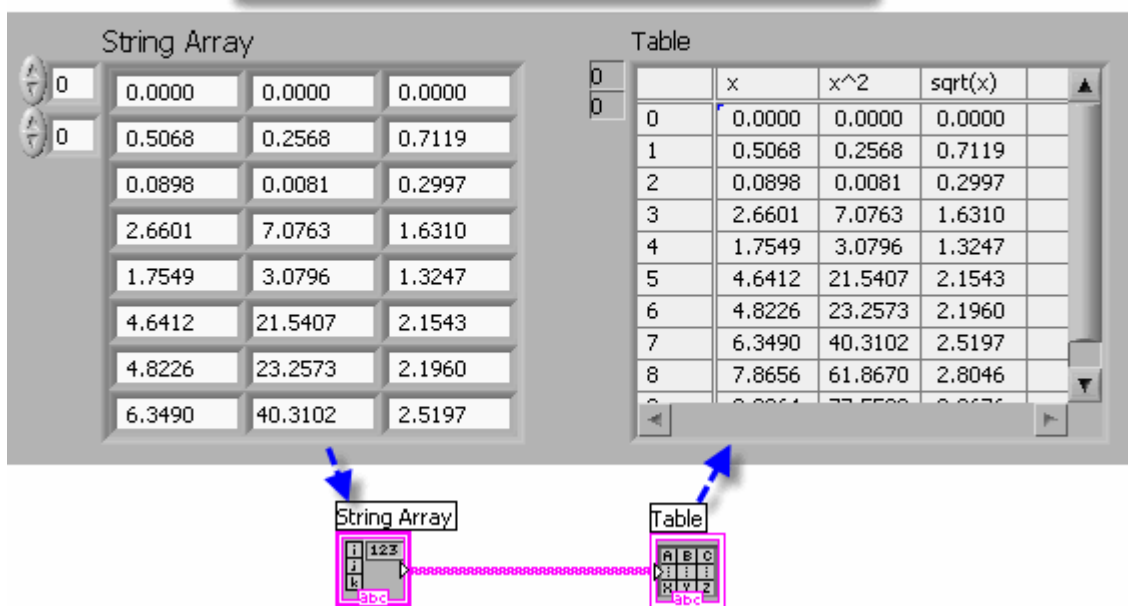
Table Control

	X	X**2	Sqrt(X)

ملحوظة :

كما ذكرنا فان Table عبارة عن مصفوفة ذات بعدين من Strings ويمكن استخدام الدوال الخاصة بالمصفوفات للتعامل مع Table ولمزيد من المعلومات حول هذه الدوال راجع الدرس السادس .Arrays and Clusters

Table عبارة عن مصفوفة من Strings



يوجد Display Index خاص بـ Table مثل المصفوفات وهو يحدد رقم الصف والعمود لأول خلية تظهر في الركن العلوى الايسر لـ Table.

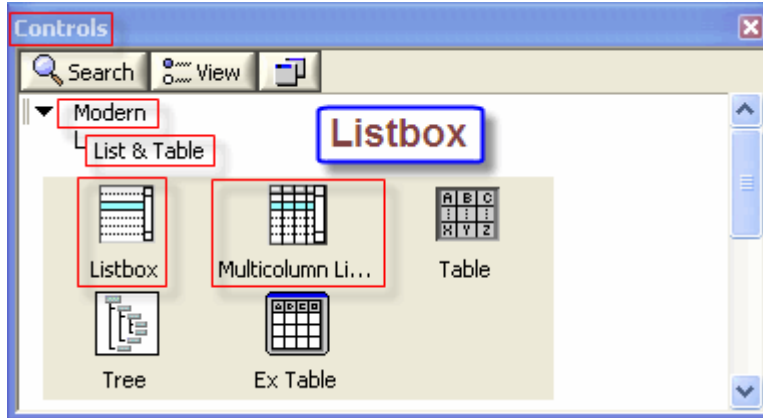
هذه الخلية
رقم صفها=1 و رقم عمودها=2
لاحظ ان الترقيم يبدأ من الصفر

Index Display	Table			
rows	1		x^2	\sqrt{x}
columns	2	2	3.6458	1.3818
		3	7.9441	1.6788
		4	3.7072	1.3876
		5	0.0147	0.3480
		6	0.8974	0.9733
		7	0.2960	0.7376
		8	12.0243	1.8622

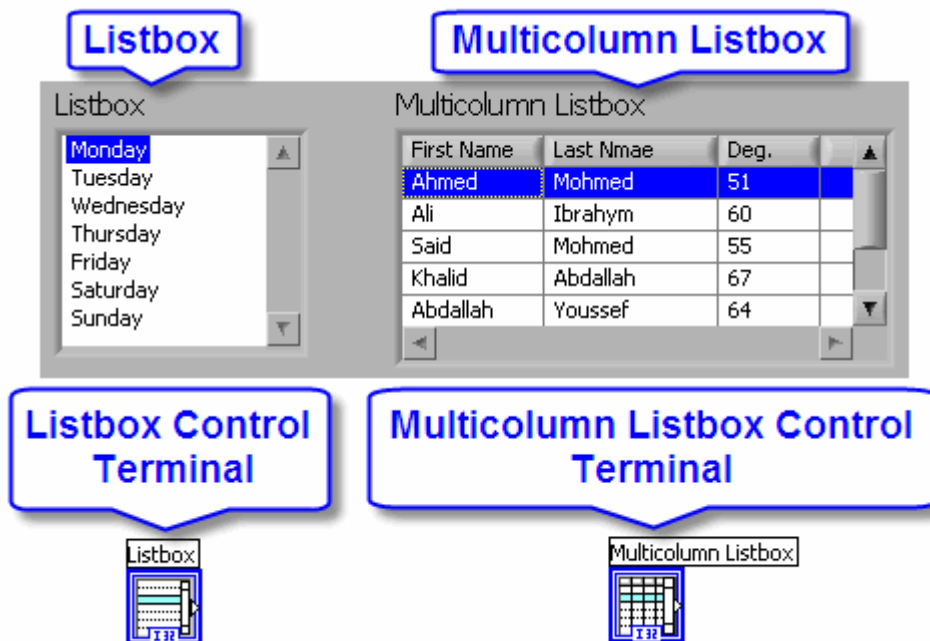
Listboxes

يوجد نوعان من Listboxes في LabVIEW هما Listbox و Multi-Column Listbox ويتم ادراجهما من

Controls Palette>>Modern>> List & Table



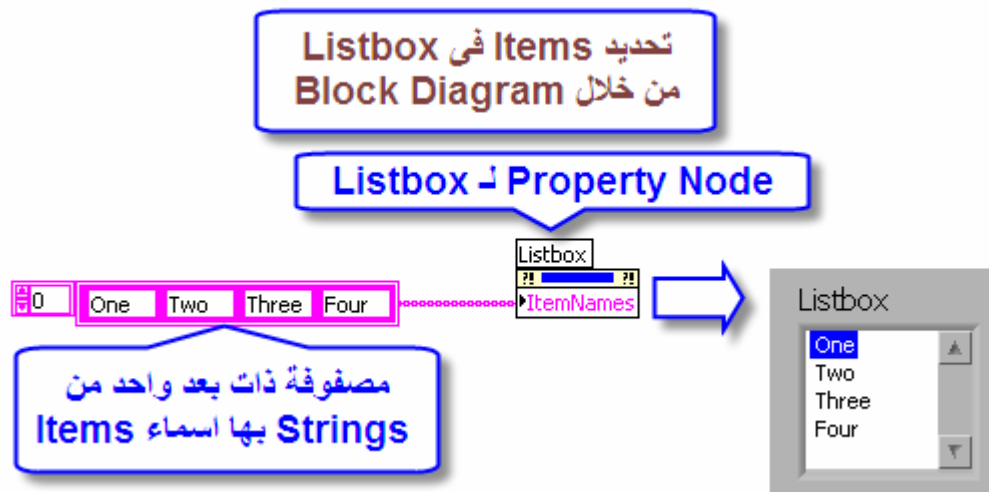
والفرق بينهما ان Listbox به عمود واحد اما Multicolumn Listbox به اكثر من عمود.



وكل سطر في Listbox يسمى Item وفي Edit Mode اثناء تصميم البرنامج يمكن الكتابة Items في Listbox بواسطة الاداة Labeling او Operating .



كما يمكن تحديد Items من خلال Block Diagram باستخدام Property Node.



في اثناء تنفيذ البرنامج يمكن للمستخدم ان يختار من Items في Listbox بالضغط عليها. و قيمة Listbox (Listbox Value) هي رقم الصف الذي اختاره المستخدم او مصفوفة ارقام الصفوف التي اختارها المستخدم كما سيتضح. ويمكن للمستخدم ان يختار Item واحد او اكثر من Item وذلك وفقا لمنط الاختيار (Selection Mode).

نمط الاختيار (Selection Mode)

يحدد كل نمط عدد Items التي يمكن ان يختارها المستخدم في Listbox ولتحديد النمط نضغط بالزر الايمن للماوس ونختار Selection Mode ثم النمط الذي نريده. وهذه الانماط هي :

النمط 0 or 1 Item : يمكن للمستخدم ان لا يختار اي Item او Item واحد فقط.

النمط 1 Item : يستطيع المستخدم ان يختار Item واحد فقط ولا بد ان يكون هناك Item مختار.

النمط 0 or More : يمكن ان لا يتم اختيار اي من Items او ان يتم اختيار اي عدد من Items.

النمط 1 or More : مثل النمط السابق ولكن لا بد ان يكون على الاقل Item واحد مختار.

قيمة Listbox هي ارقام Items التي تم اختيارها .

وفي حالة النمطين الاولين يكون خرج Listbox قيمة وحيدة (Scalar Value) قيمتها هي رقم

Item الذي تم اختياره وفي حالة عدم اختيار اي Item تكون هذه القيمة تساوى -1.

**تحدد نمط الاختيار
(Listbox Selection)**



0 or 1 Item النمط

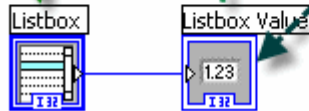


النمط 1 Item

يمكن اختيار item واحد فقط
قيمة Listbox = رقم Item
لاحظ الترقيم يبدأ من الصفر



لاحظ ان قيمة Listbox في
هذا النمط قيمة وحيدة



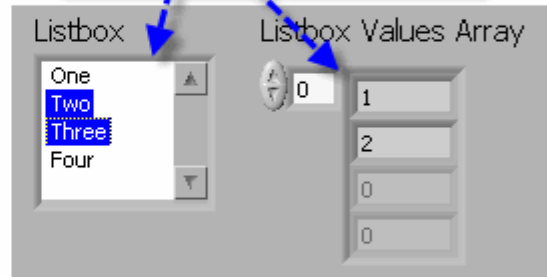
وفي حالة النمطين الآخرين يكون خرج Listbox عبارة عن مصفوفة فيها ارقام Items التي تم اختيارها. وعدد عناصر هذه المصفوفة هي عدد Items التي تم اختيارها.

النمط 0 or More

يمكن عدم اختيار اي Items
فيكون الخرج مصفوفة فارغة

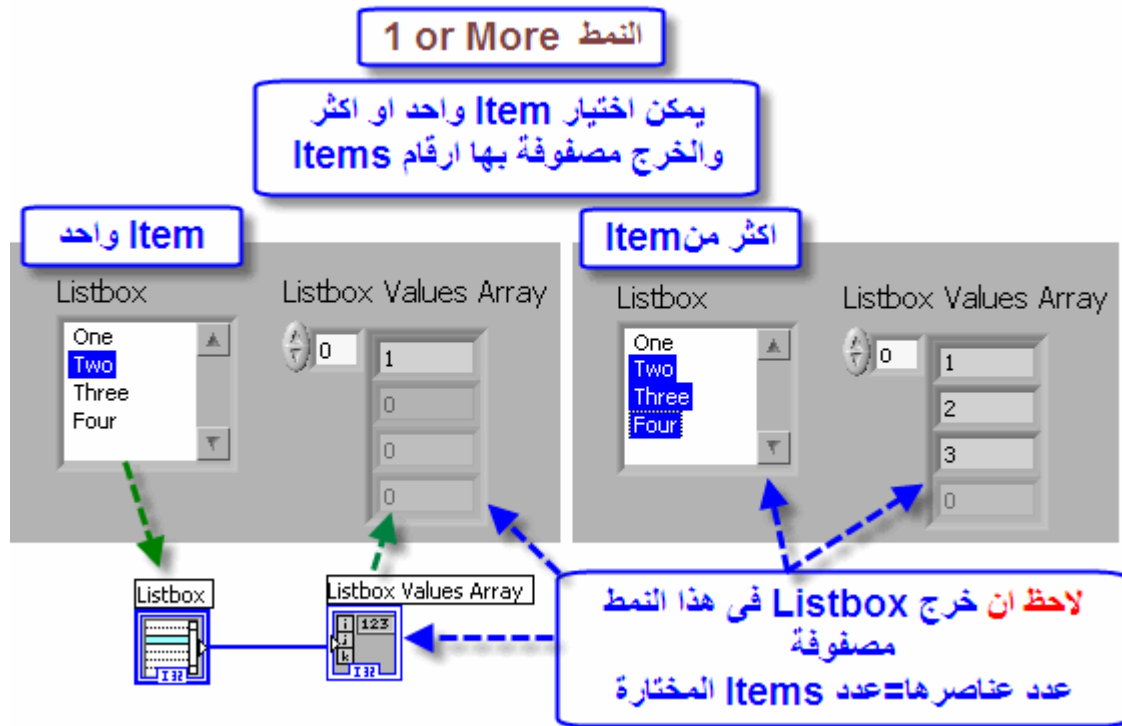


يمكن اختيار اكثر من item
فيكون الخرج مصفوفة
بها ارقام Items المختارة



لاحظ ان خرج Listbox في هذا النمط
مصفوفة
عدد عناصرها = عدد Items المختارة





ملحوظة : لاختيار اكثر من Item: نختار اول Item ثم نضغط المفتاح Ctrl عند اختيار باقي Items.

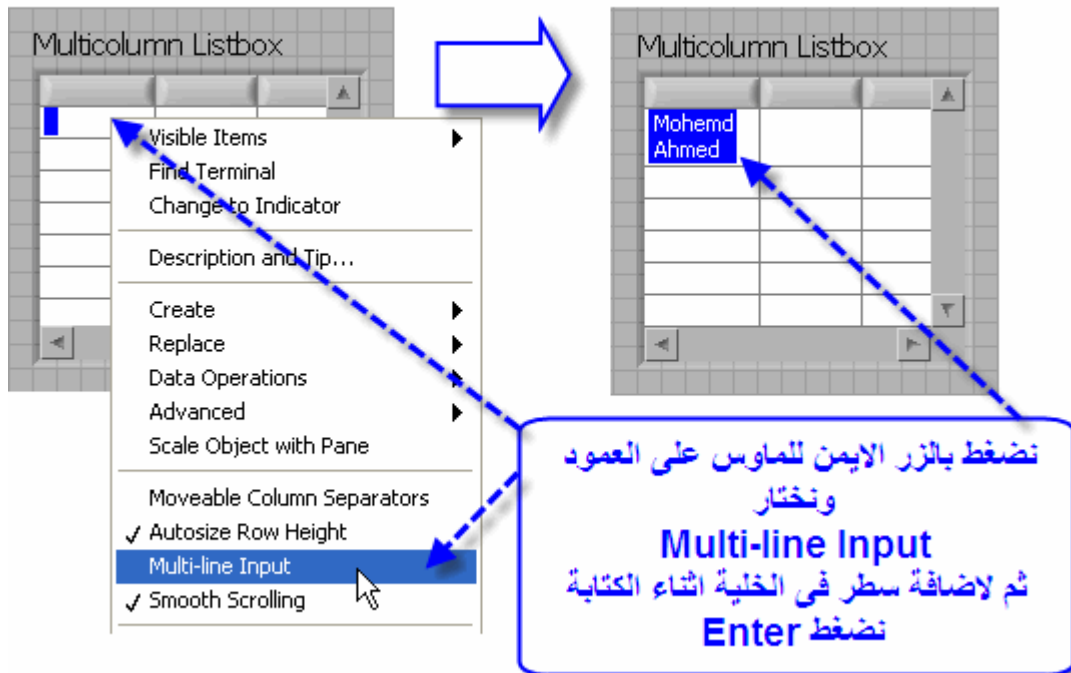
Listboxes مفيدة اذا اردت ان تعرض مجموعة من البيانات في مجموعة صفوف هذه الصفوف يمكن ان تكون في عمود واحد او مقسمة لعدة اعمدة وتريد من المستخدم ان يختار احد هذه البيانات او اكثر.

اذا كانت احد الخلايا في ListBox لا تسع النص الموجود بها فانه عند تنفيذ البرنامج ووضع المؤشر على هذه الخلية فان Tip Strip تظهر بها النص الموجود في الخلية كاملا.



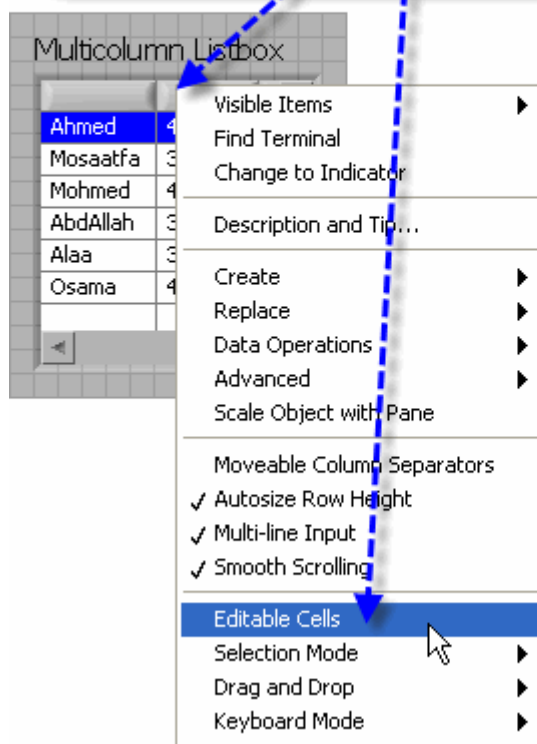
كما يمكن ان نجعل خلايا اي عمود تحتوي على اكثر من سطر وذلك بالضغط على العمود بالزر الايمن للماوس ونختار Multi-line Input ثم نستطيع ان نكتب اكثر من سطر في الخلية بواسطة الضغط على Enter.

كتابة أكثر من سطر في الخلية



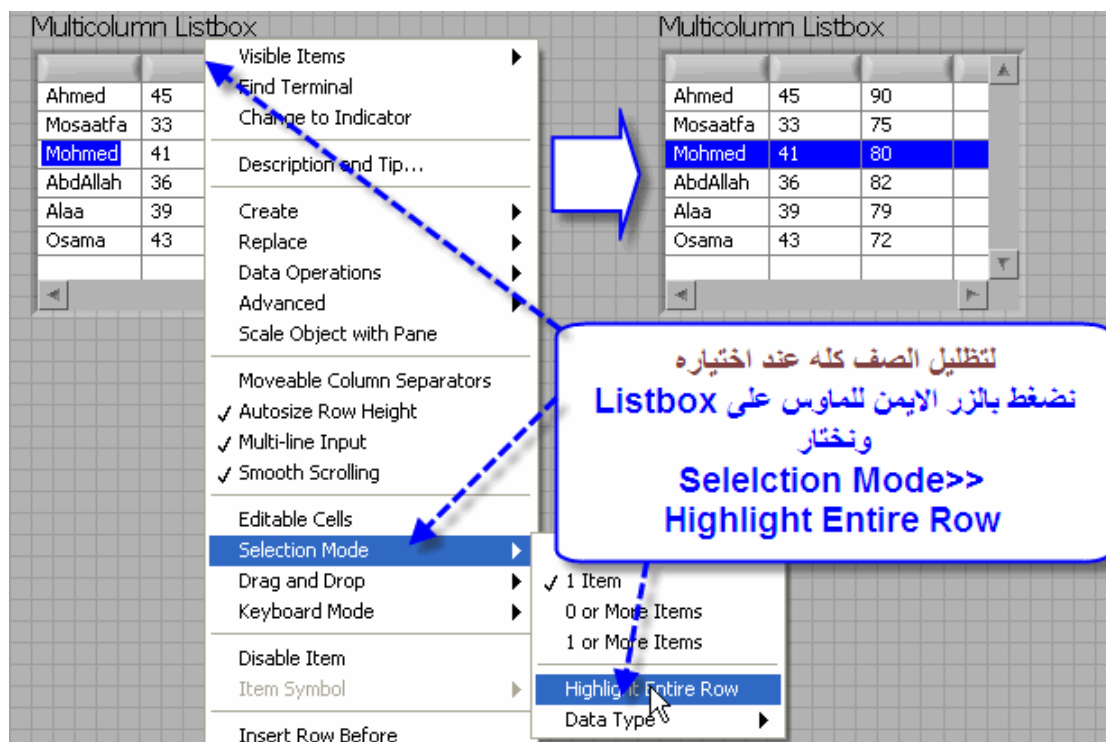
لا يستطيع المستخدم تعديل محتوى خلايا Listbox اثناء تنفيذ البرنامج ولجعل هذا ممكنا نضغط بالزر الايمن للماوس على Listbox ونختار الخاصية Editable Cells.

لتفعيل امكانية تعديل محتوى الخلايا اثناء تنفيذ البرنامج نضغط بالزر الايمن للماوس على Listbox ونختار **Editable Cells**



عند اختيار اى صف فى Listbox يتم تظليل اول خلية من اليسار فى هذا الصف وتظليل الصف كله
نضغط بالزر الايمن للماوس على Listbox ونختار

Selection Mode>>Highlight Entire Row



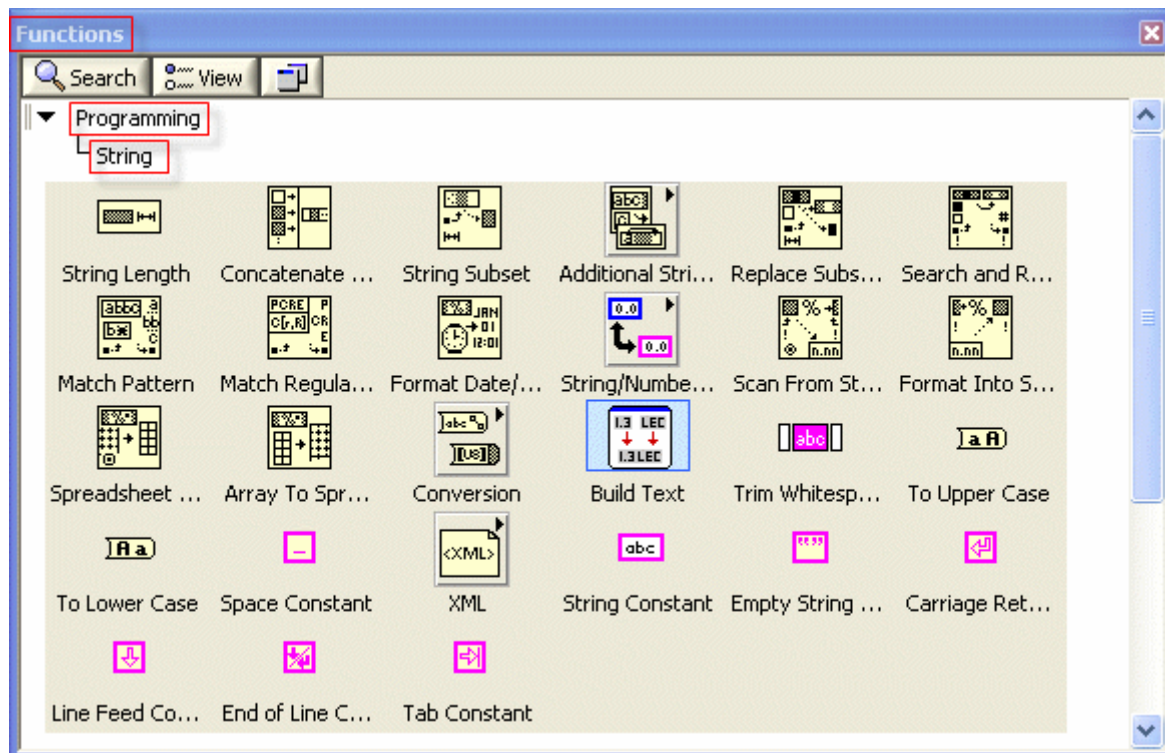
يمكن ضافة صف او عمود او حذف او عمود وذلك من خلال الضغط بالزر الايمن للماوس على اى صف او عمود



الدوال الخاصة بـ Strings

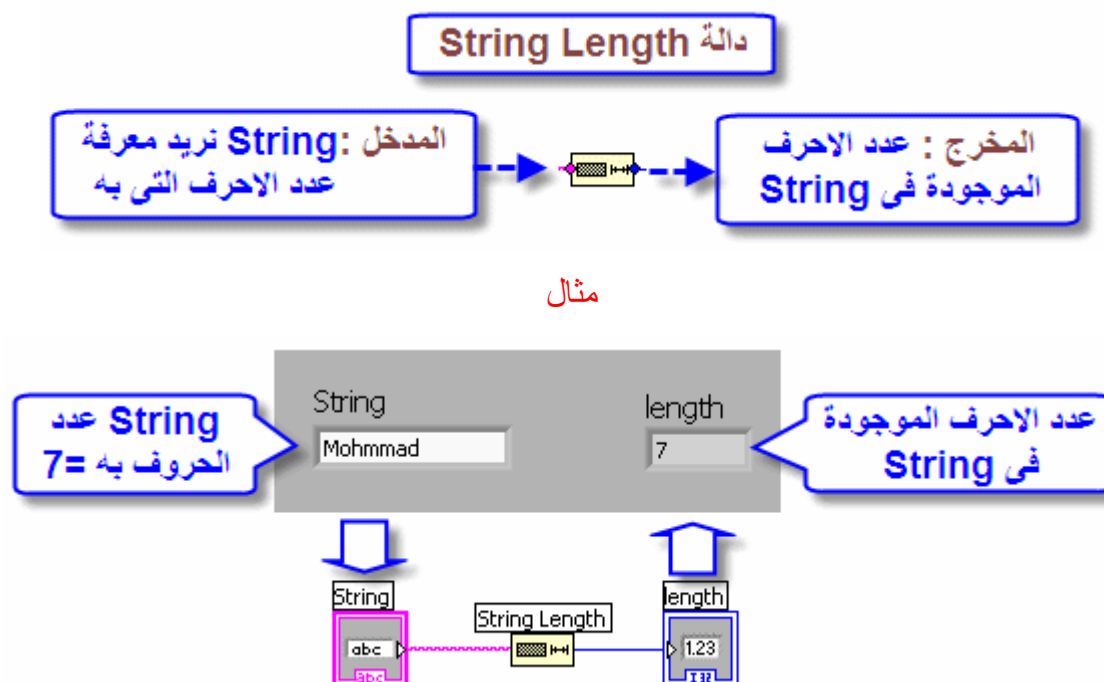
مثل المصفوفات توجد دوال كثيرة في LabVIEW للتعامل مع Strings وسوف نتعرف على اهم هذه الدوال. ويتم ادراج هذه الدوال من

Functions Palette>>Programming>>String



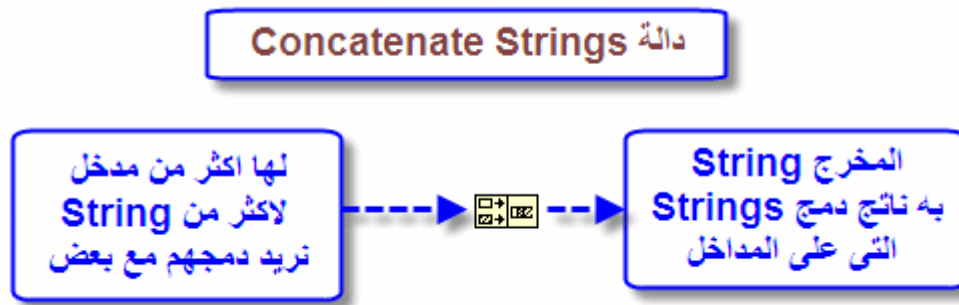
الدالة String Length

تقوم هذه الدالة باخراج عدد الاحرف الموجودة في النص (String).

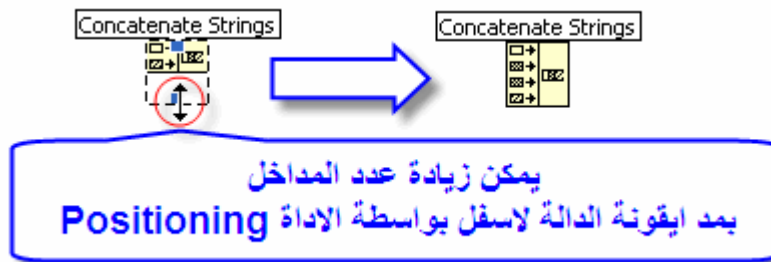


الدالة Concatenate Strings

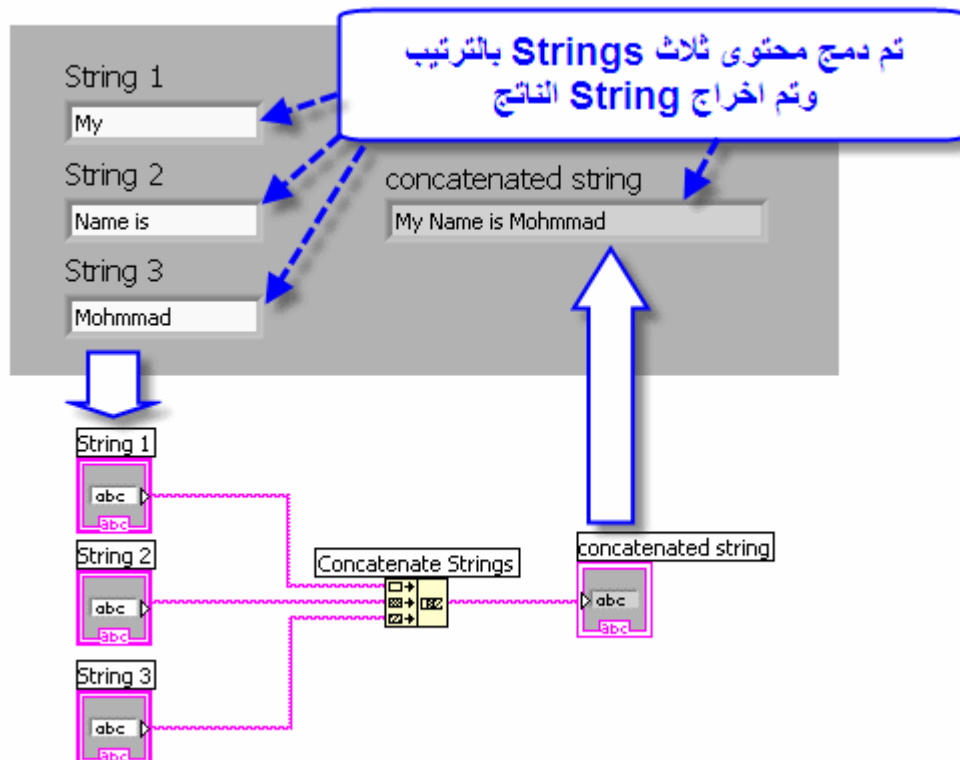
تقوم هذه الدالة بدمج أكثر من String بالترتيب لتكوين String واحد.



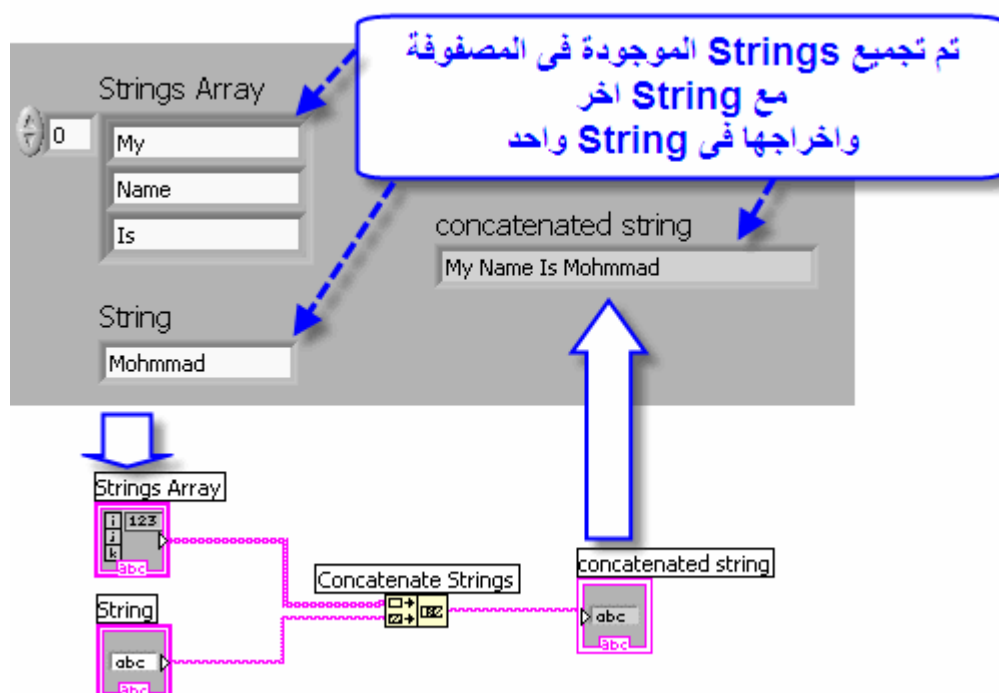
يمكن مد ايقونة الدالة لاسفل بواسطة الاداة Positioning لزيادة عدد المداخل للدالة وزيادة عدد Strings التي يمكن ادماجها.



مثال



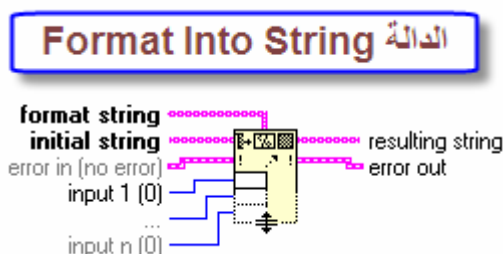
كما يمكن ان يكون المدخل لهذه الدالة مصفوفة ذات بعد واحد من Strings وستقوم الدالة بدمج Strings الموجودة فى المصفوفة بالترتيب واخراجها فى String واحد.



وتفيد دالة Concatenate Strings كثيرا فى تكوين الاوامر التى نريد ارسالها الى اجهزة القياس او تكوين سطور معينة نريد تخزينها فى ملف.

فى كثير من الاحيان نحتاج الى تحويل الارقام الى نصوص (String) وايضا استخلاص الارقام من النصوص وتوجد دوال كثيرة تقوم بهذه المهام ولكن سنتعرف على اهم هذه الدوال وهما الدالة Format into String و الدالة Scan from String وسنتعرف على الدالة String فى هذا الجزء وبعد قليل سنتعرف على الدالة Scan from String.

الدالة Format into String



تقوم هذه الدالة بتحويل انواع مختلفة من البيانات الى نص (String) بالشكل الذى نريده. ويمكن ان تكون هذه البيانات التى نريد تحويلها الى نص عبارة عن :
نص نريده بشكل اخر.
File Path (مسار لملف).

Enumerated Type شبيهه بـ ListBox.

Time Stamp يحتوى على معلومات عن الوقت و التاريخ (راجع الدرس السابق).

Boolean (رقم ثنائى).

رقم صحيح او به كسر عشرى (Integer او Float) .

ويكثر استخدام هذه الدالة لتحويل الارقام الى نصوص

ومداخل هذه الدالة هي

Format String : وهو نص به مجموعة ارقام ورموز لتحديد شكل String الذى نريده وسنوضح

كيفية تحديد هذا المدخل وتأثيره على String الناتج.

Initial String : وهو نص نريده ان يتضمن فى بداية String الناتج.

Input : وهو مدخل البيانات التى نريد تحويلها الى String ويمكن زيادة عدد المدخل من هذا النوع

بمد الايقونة الدالة لاسفل بواسطة الاداة Positioning بحيث يكون لدينا Input 0 و Input 1 و ...

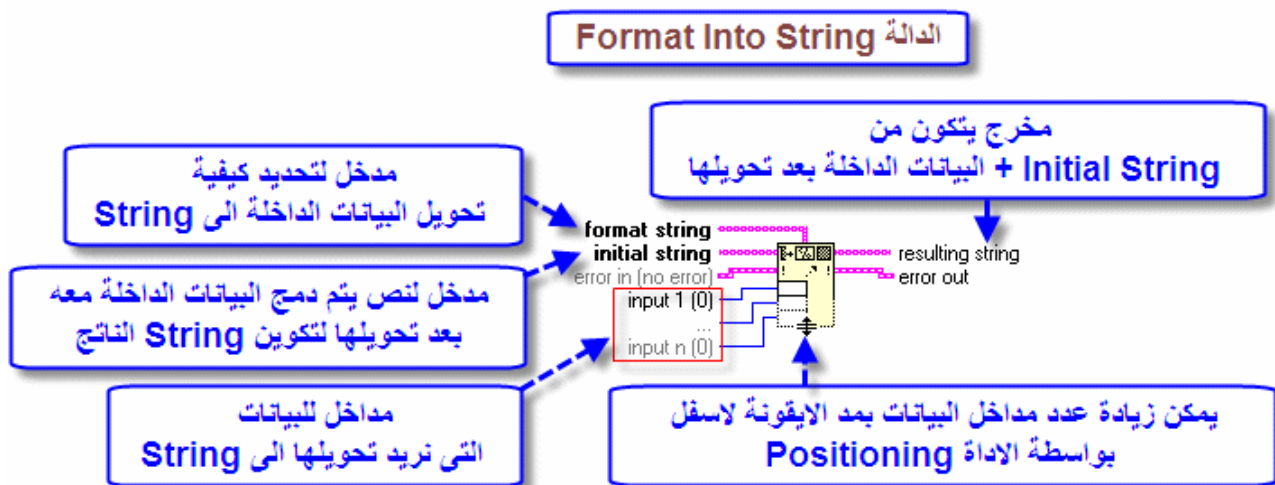
هكذا وبذلك يمكن تحويل اكثر من قيمة الى نفس String الناتج.

ملحوظة : يكون الرمز الموجود على المدخل Input هو **DBL** (مبدئيا ولكنه يتغير تلقائيا وفقا لنوع البيانات الموصل به).

ومخرج هذه الدالة هو

Resulting String : وهو String الناتج بعد تحويل البيانات الداخلة الى نص بالشكل المحدد فى

Format String ودمجها مع النص Initial String.

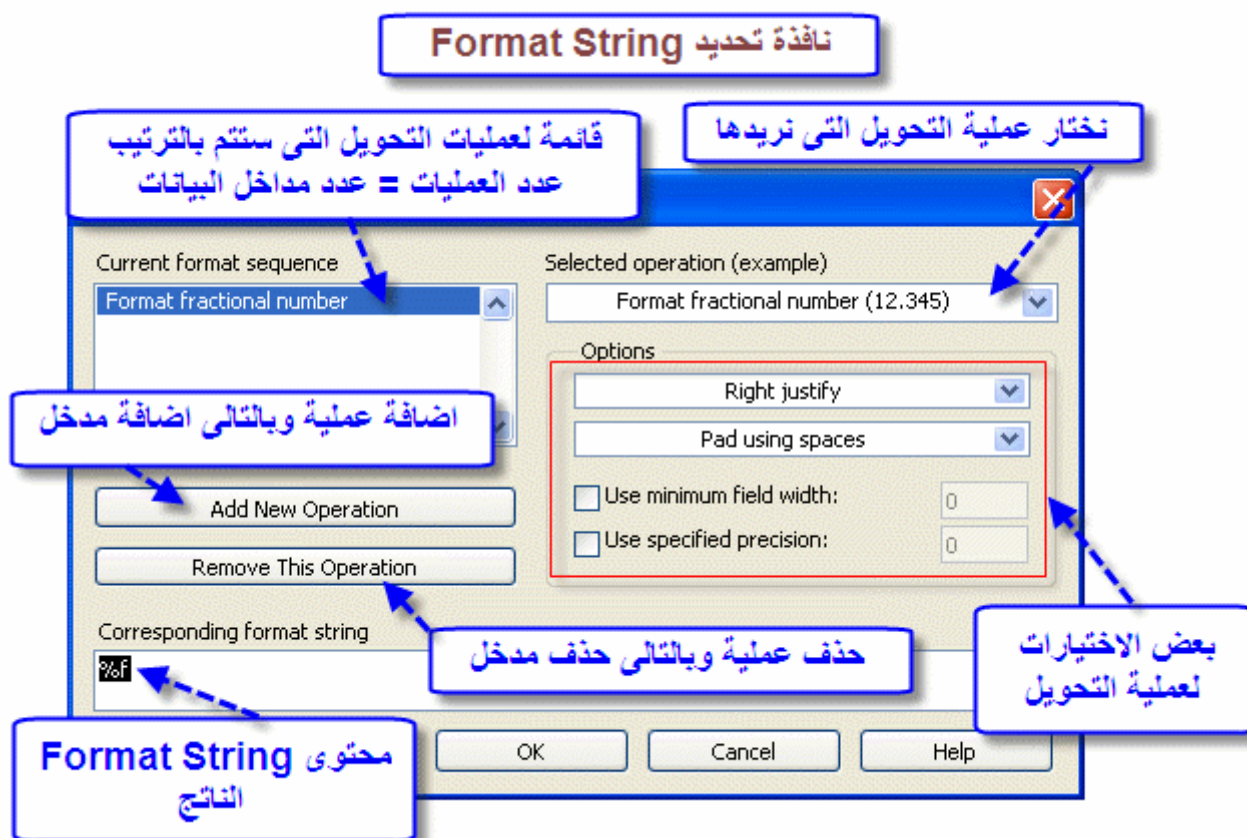


كيفية تحديد Format String

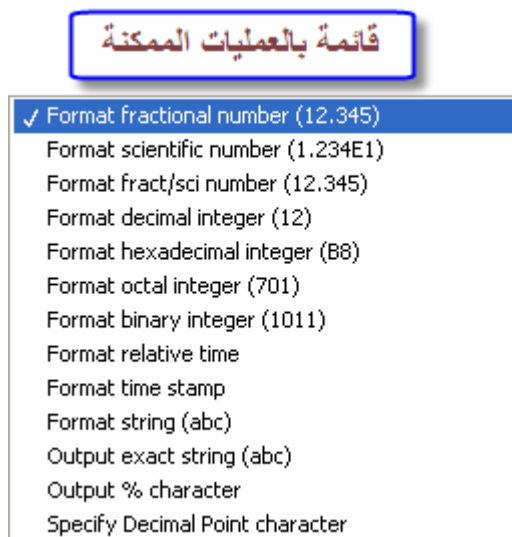
يتكون Format String اساسا من الرمز % ومجموعة اخرى من الارقام والرموز تحدد طريقة

التحويل وتوجد طريقتين لتحديد Format String.

1- يمكن تحديد المدخل Format String بالضغط على ايقونة الدالة مرتين متاليتين او الضغط عليها بالزر الايمن للماوس و اختيار Edit Format String لتظهر لنا نافذة تحديد Format String

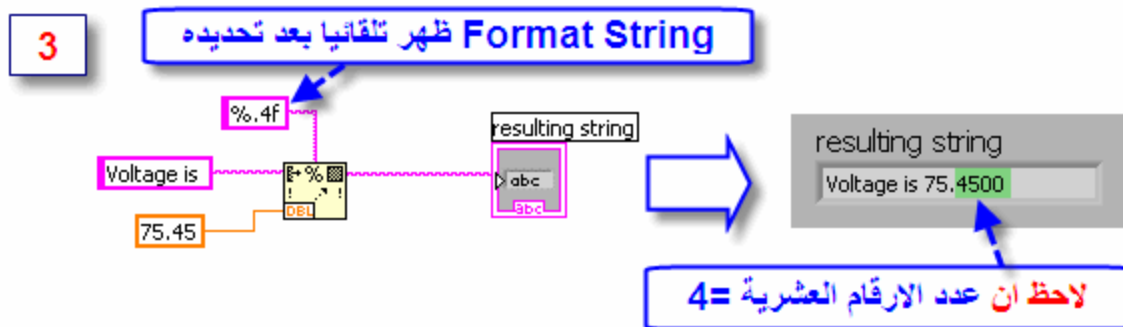
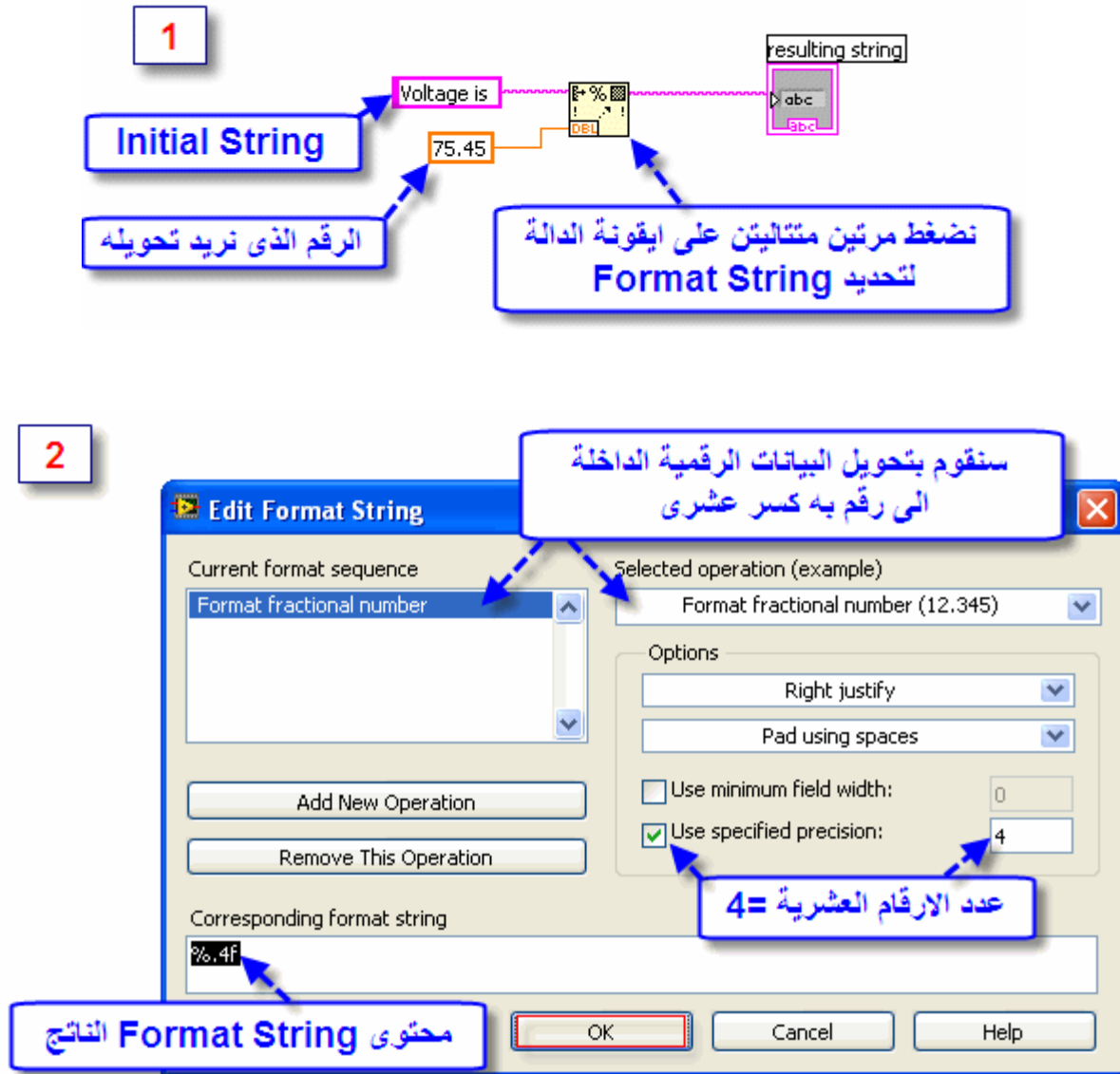


وهذه قائمة بالعمليات التي يمكن اختيارها



ويتوقف نوع العملية على نوع البيانات الداخلة و الشكل المراد تحويلها اليه.

مثال



2- كتابة Format String يدويا

وكما ذكرنا يتكون Format String من العلامة %. وسوف نعرض كيفية كتابة Format String لتحويل بعض البيانات كثيرة الاستخدام الى String.

وسوف نستعرض بعض الاشكال المختلفة لـ Format String

تحويل ارقام بها كسر عشري من النوع Float

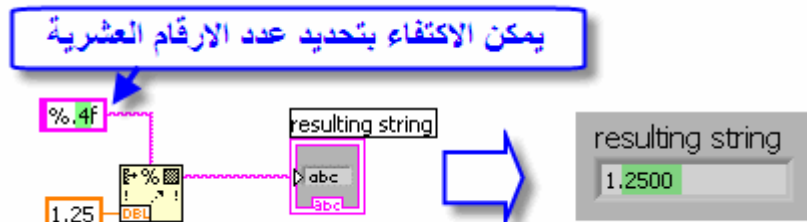
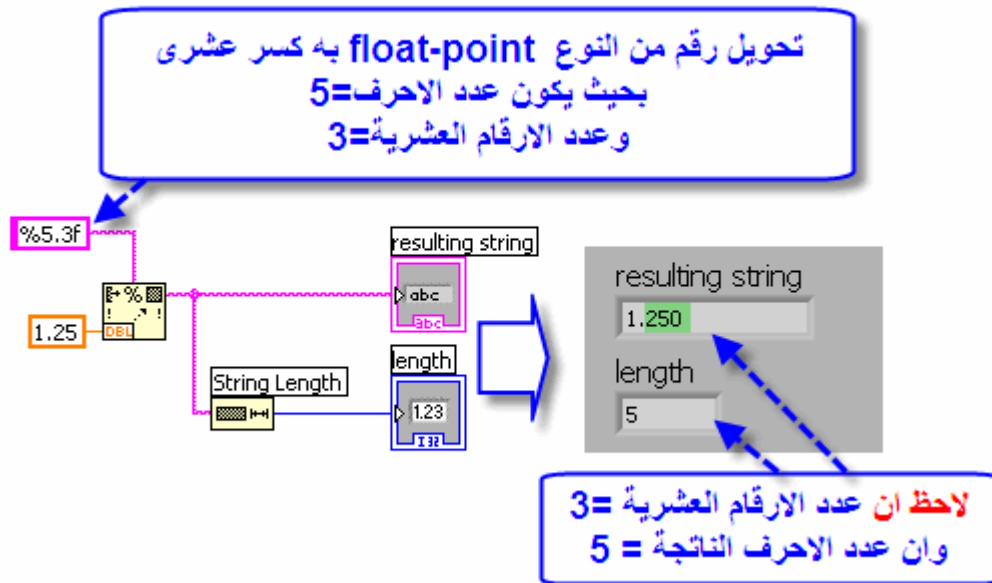
يكون Format String بهذا الشكل % number1.number2 f

حيث Number1 يحدد عدد احرف String الناتج .

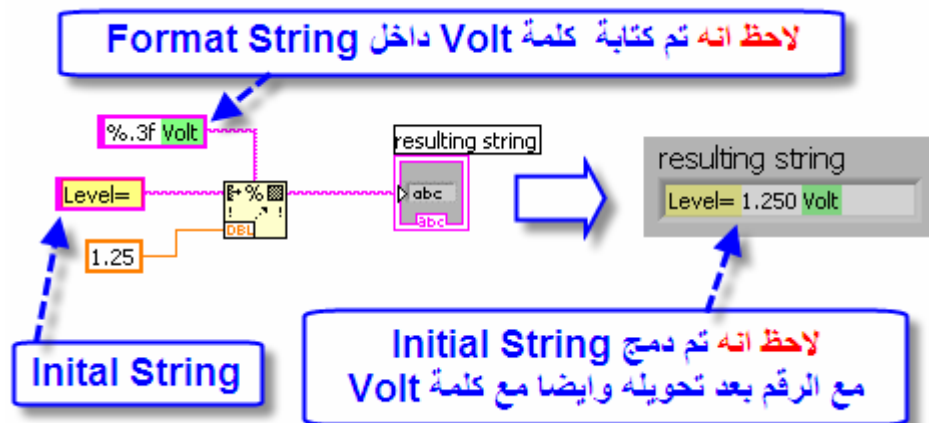
و Number2 يحدد عدد الارقام العشرية .

ملحوظة: يمكن عدم كتابة 1 او Number2 والاكتفاء بـ %f

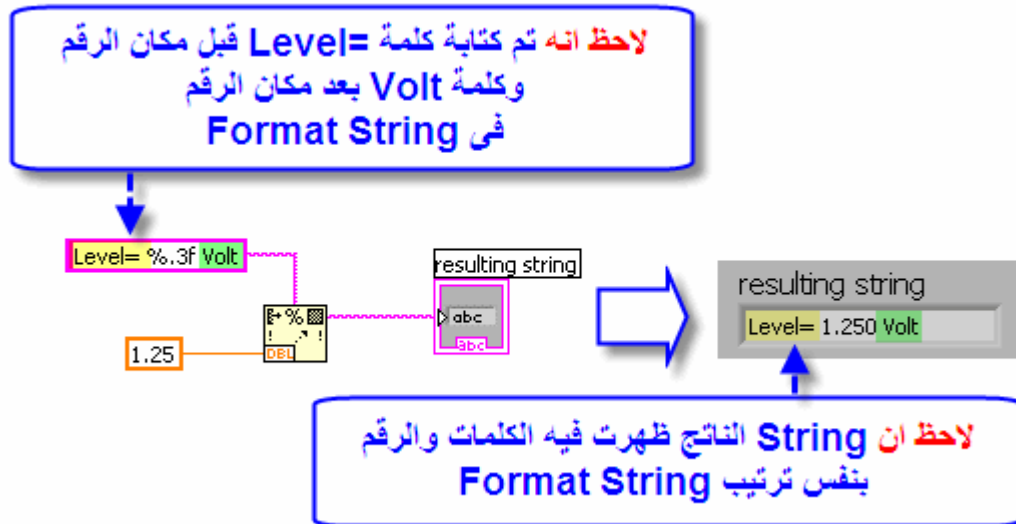
امثلة:



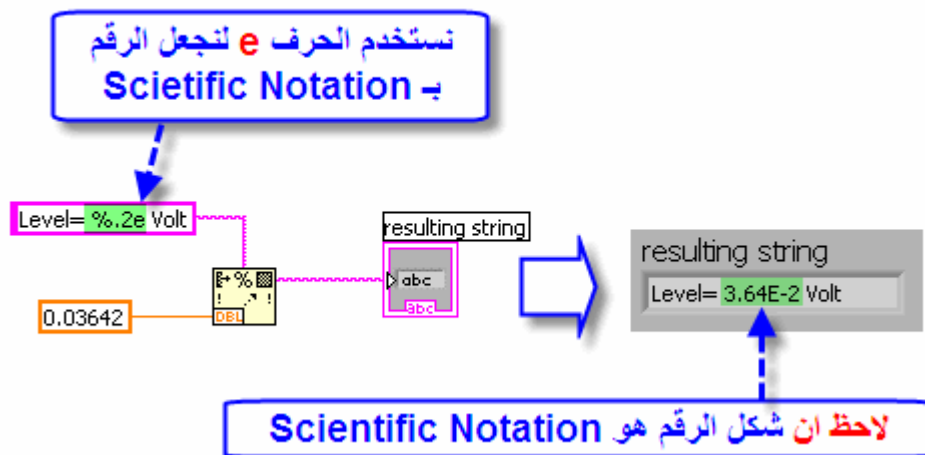
يمكن كتابة نص في Format String وسيظهر هذا النص كما هو String الناتج.



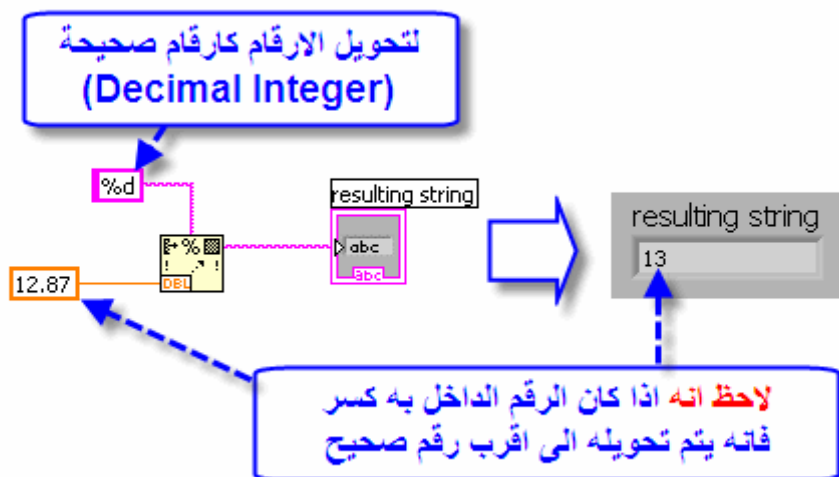
يمكن ايضا ان نكتب Initial String داخل Format String



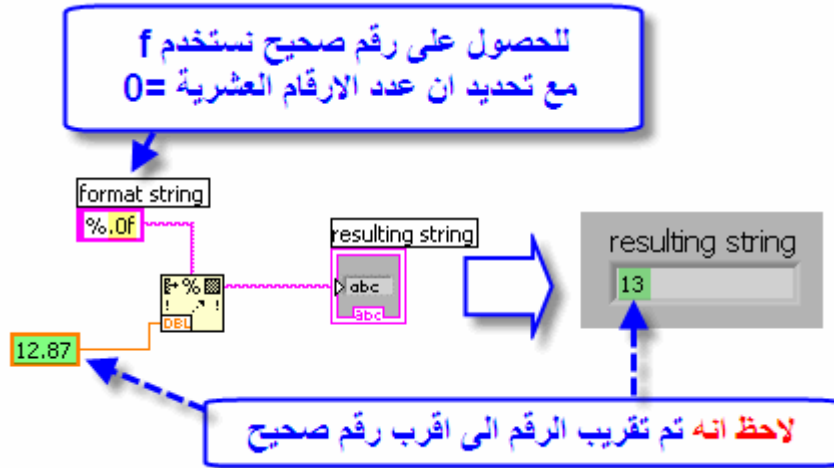
تحويل الارقام من النوع float –point الى Scientific Notation (مثال 3.64E2)
ونستخدم في هذه الحالة الحرف **e** بدلا من **f** ونتبع نفس التعليمات السابقة



تحويل الارقام الى ارقام صحيحة (Decimal Format)
نستخدم الحرف **d** مع العلامة %

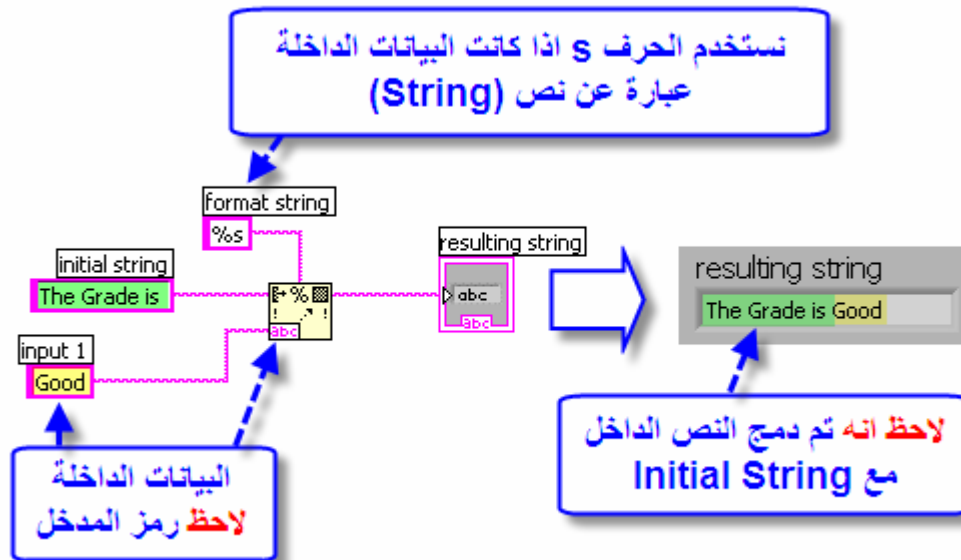


ويمكن ايضا ان نستخدم الحرف f مع تحديد ان عدد الارقام العشرية يساوى 0 .



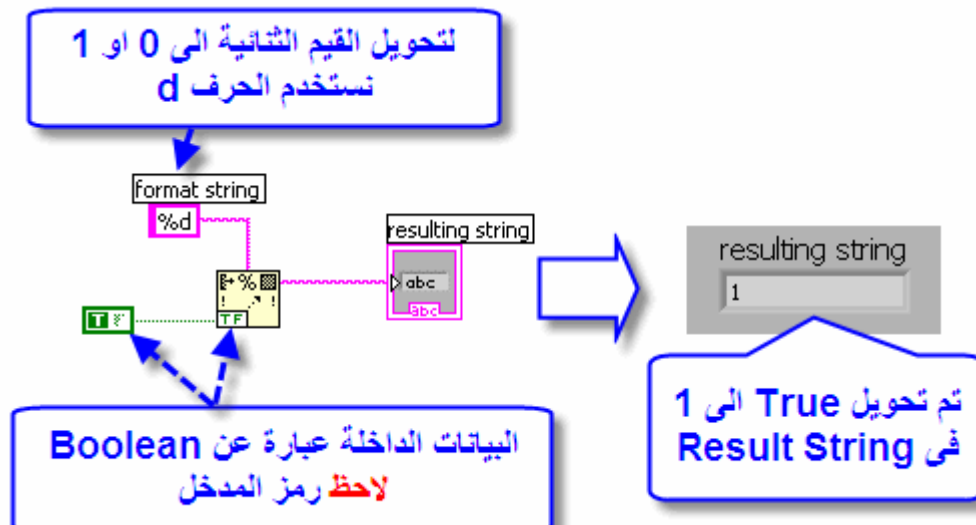
تحويل النصوص

يمكن ان يكون البيانات الداخلة نص فنستخدم الحرف s فى Format String .

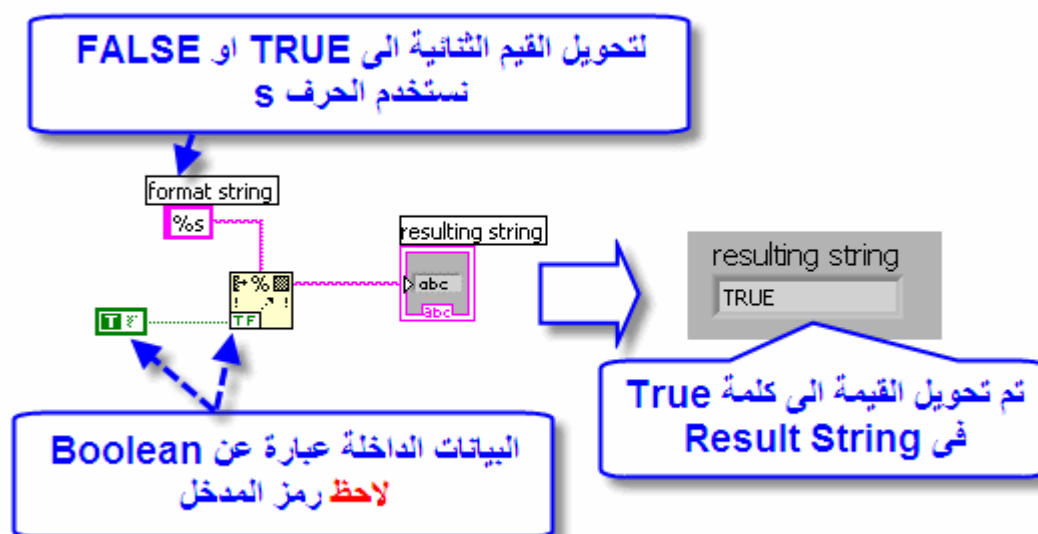


تحويل الارقام الثنائية (Boolean)

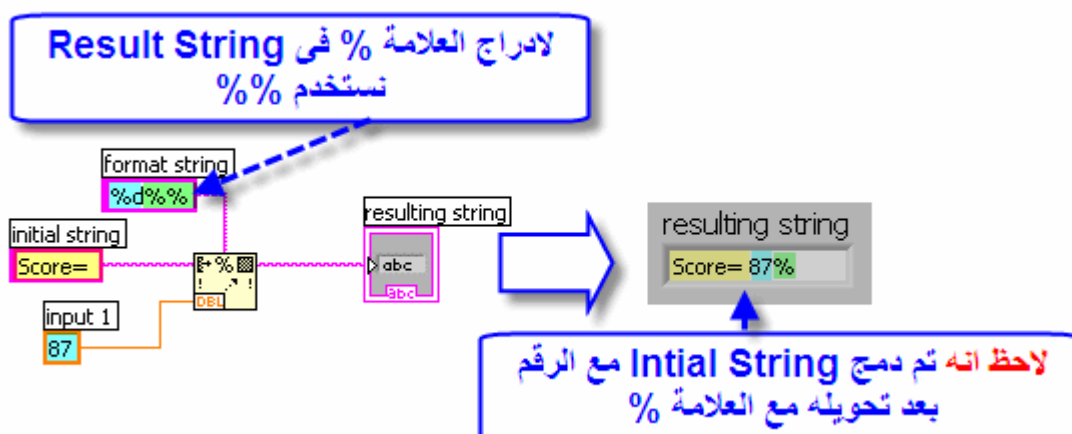
يمكن تحويل الارقام الثنائية الى 0 او 1 (False=>0 و True=>1) وذلك باستخدام الحرف d.



كما يمكن ان تتحول قيمة Boolean الى كلمتى TRUE او FALSE فى Result String وذلك باستخدام الحرف s.



ملحوظة: نستخدم %% فى Format string لظهار العلامة % فى Result String.



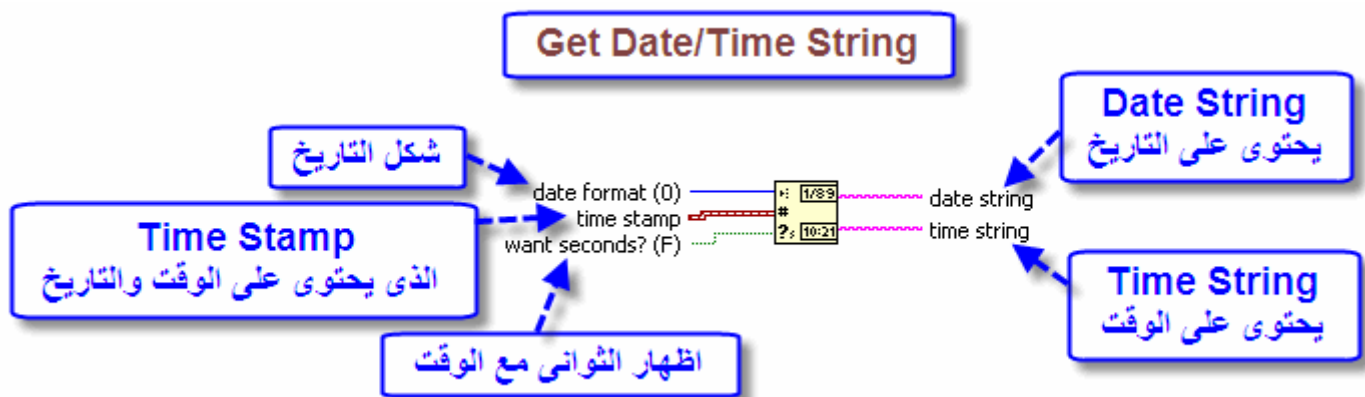
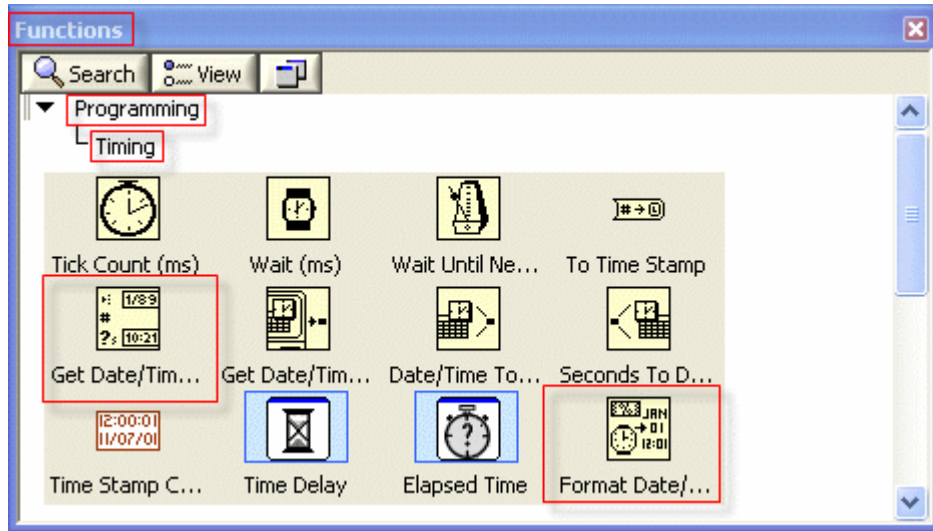
يوجد اشكال اخرى من البيانات يمكن تحويلها ويمكنك استخدام Edit Format String للتعرف عليها اكثر او الاطلاع على Help.

الدالة Get Date/Time String

تخرج هذه الدالة Date String (String) (على التاريخ) و Time String (String) (على الوقت) والوقت والتاريخ يكونان من Time Stamp يتم توصيلها الى مدخل الدالة او يكونان الوقت والتاريخ الحاليين (لحظة تنفيذ الدالة).

ويتم ادراج الدالة من

Functions Palette>>Programming>>Timing



ولهذه الدالة ثلاث مداخل :

Date Format : يحدد شكل التاريخ الذى سنحصل عليه وقيمها كالتالى:

(0) **Short** ويكون التاريخ بهذا الشكل 1/21/94.

(1) **Long** ويكون التاريخ بهذا الشكل Friday, January 21, 1994.

(2) **Abbreviated** ويكون التاريخ بهذا الشكل Fri, Jan 21, 1994.

Time Stamp: وهو Time Stamp الذى يحتوى على الوقت والتاريخ اللذين ستخرجهم الدالة

إذا لم يوصل هذا المدخل تخرج الدالة الوقت والتاريخ الحاليين (وقت تنفيذ الدالة).

Want seconds: يتم توصيل الى هذا المدخل قيمة :

True لاطهار الثواني مع الوقت.

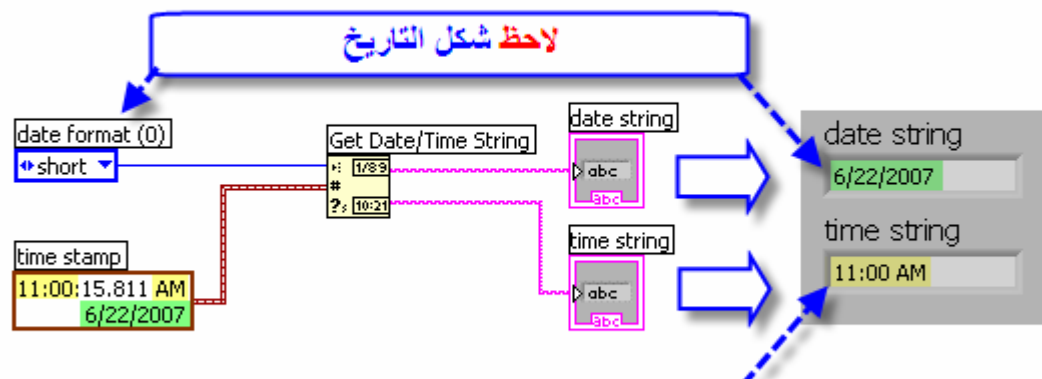
False لعدم اظهار الثواني مع الوقت وهذا هى الحالة العادية.

اى اذا لم يوصل هذا المدخل فان قيمته تكون False

ولا يتم اظهار الثواني مع الوقت.

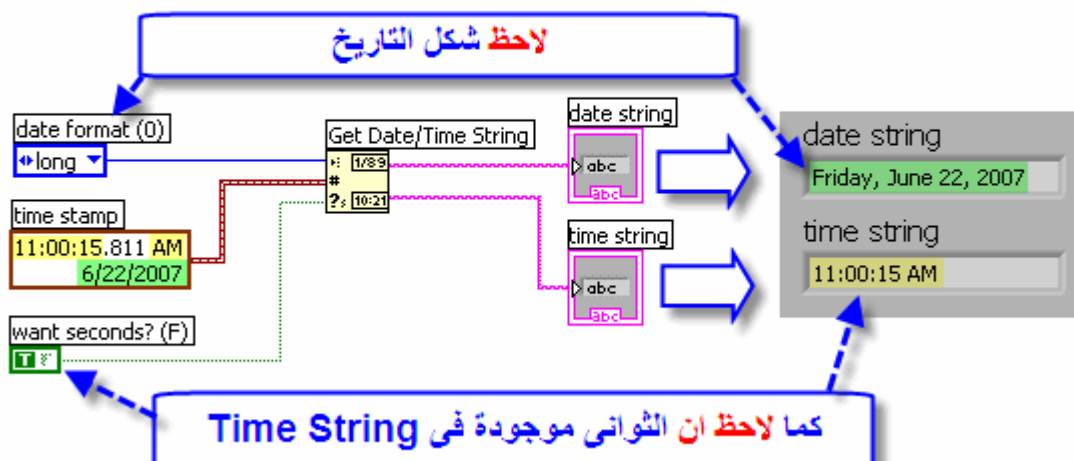
امثلة :

تم الحصول على الوقت والتاريخ في Strings من Time Stamp



كما لاحظ ان الثواني ليست موجودة في Time String

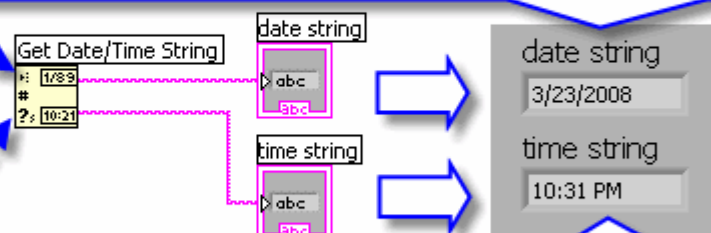
تم الحصول على الوقت والتاريخ في Strings من Time Stamp



كما لاحظ ان الثواني موجودة في Time String

اذا لم يتم توصيل Time Stamp الى مدخل الدالة
نحصل على الوقت والتاريخ الحاليين في Strings

لاحظ ان التاريخ Short لاننا لم نوصل اي قيمة الى المدخل Date Format

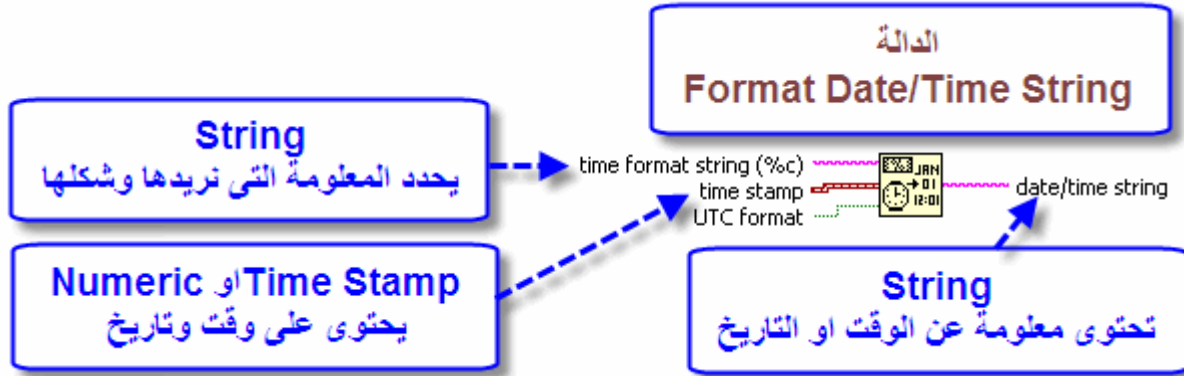


كما لاحظ ان الثواني غير موجودة في Time String
لاننا لم نوصل اي قيمة الى المدخل Want seconds?

الدالة Format Date/Time String

تقوم هذه الدالة باخراج String به معلومات محددة من الوقت والتاريخ مثل اسم الشهر او اسم الاسبوع. والوقت والتاريخ يمكن ان يكونا من Time Stamp او Numeric يوصل بمدخل الدالة او يكونا الوقت والتاريخ الحاليين اذا لم يوصل Time Stamp او Numeric الى مدخل الدالة. وهذه الدالة اكثر شمولية من الدالة Get Date/Time String ويتم ادراج هذه الدالة من

Functions Palette>>Programming>>Timing



ويتم توصيل Time Stamp او Numeric الى المدخل Time Stamp يحتويان على الوقت والتاريخ واذا لم يوصل هذا المدخل يكون الوقت والتاريخ هو الوقت والتاريخ الحاليين. ويحدد المدخل Time Format String المعلومة التي نريدها وشكلها.

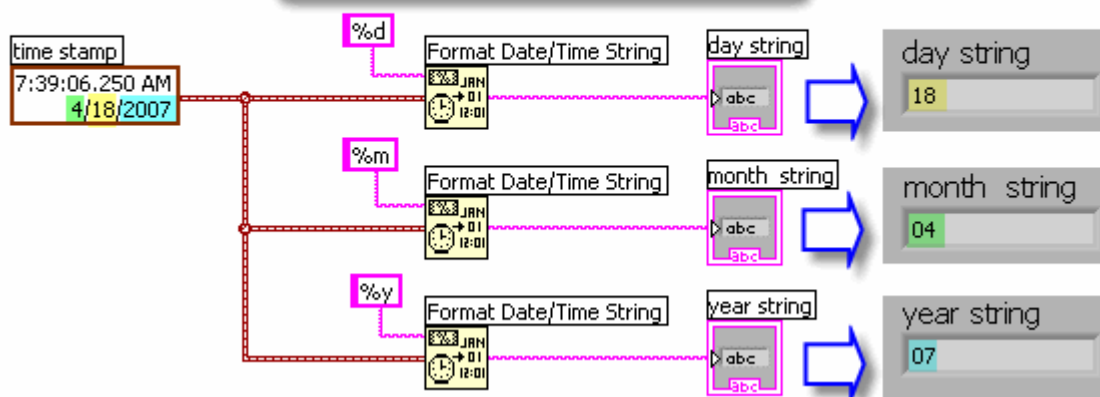
ويحتوى Time Format String على اكواد كل كود يحدد شكل المعلومة التي ستخرج وهذه الاكواد يمكن ان تكون كالتالى:

الكود	معنى الكود
%a	اسم اليوم مختصر مثل Sat
%b	اسم الشهر مختصر مثل Feb
%c	الوقت و التاريخ كما فى اعدادات جهازك
%d	رقم اليوم مكون من خانتين مثال 21
%H	رقم يمثل الساعة بنظام 24 ساعة
%I	رقم يمثل الساعة بنظام 12 ساعة
%m	رقم مكون من خانتين يمثل الشهر مثل 03
%M	رقم يمثل عدد الدقائق
%p	احدى العلامتين AM او PM اللتان تعنيان صباحا او مساء
%S	رقم يمثل عدد الثواني فى الوقت

%x	التاريخ بالنظام المحلي
%X	الوقت بالنظام المحلي
%y	رقم مكون من خانتين يمثل عدد السنين في القرن مثال 08
%Y	رقم يمثل العام متضمنا القرن مثال 2008
%<digit>u	جزء من الثانية حيث يمثل digit عدد الارقام العشرية
مثال %3u	مثال 0.465

ملحوظة: اذا لم يوصل المدخل Time Format String فانه يأخذ القيمة العادية له وهي %c والتي معناها الوقت والتاريخ كما في اعدادات جهازك.

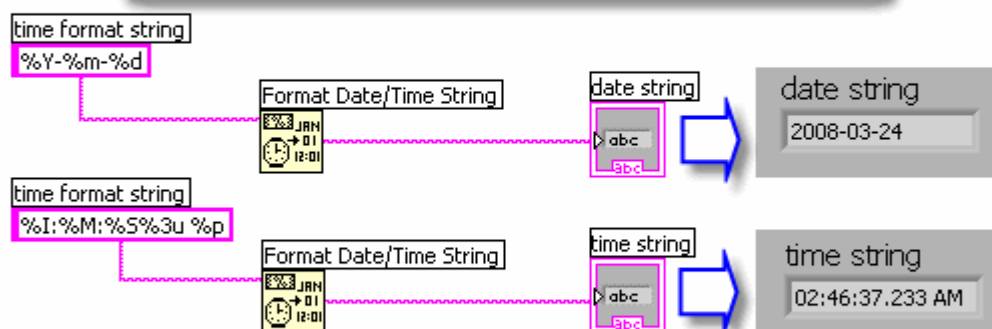
الحصول على Strings تحتوى على رقم اليوم و رقم الشهر ورقم العام



ملحوظة: يمكن ان يحتوى Time Format String على اكثر من كود بجانب ايضا اى حروف او رموز وستظهر هذه الرموز والحروف كما هي فى String الناتج.

الحصول على الوقت والتاريخ الحاليين بالشكل الذى نريده

لاحظ اننا لم نقم بالتوصيل الى المدخل Time Stamp وذلك للحصول على الوقت والتاريخ الحاليين



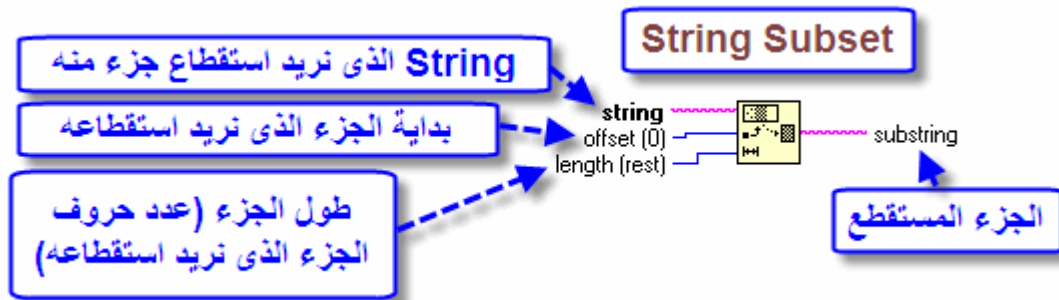
لاحظ ان الرموز (غير الاكواد مثل : و -) التى استخدمنا فى Time Format String ظهرت كما هي فى مواضعها فى String الناتج

Parsing Factions

فى كثير من الاحيان يكون لديك نص تم الحصول عليه مثلا من استقبال بيانات من جهاز قياس او تم قراءته من ملف. وتحتاج الى تفسير هذا النص او استخلاص بعض المعلومات منه مثل رقم يمثل قيمة القياس. وتوجد دوال تساعد فى تفسير النصوص وسنستعرض بعض هذه الدوال.

الدالة String Subset

تقوم هذه الدالة باستقطاع جزء من String .



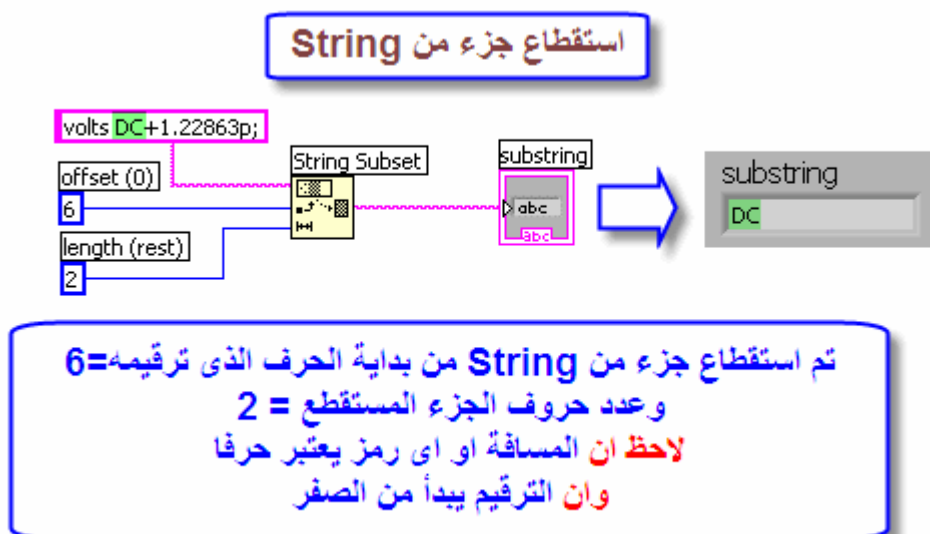
ومداخل هذه الدالة:

String : وهو مدخل لـ String الذى نريد استقطاع جزء منه .

Offset : وهو بداية الجزء الذى نريد استقطاعه والترقيم يبدأ من الصفر. واذا لم يوصل هذا المدخل فان البداية تكون من الصفر.

Length : وهو مدخل لطول او عدد حروف الجزء الذى نريد استقطاعه واذا لم يوصل هذا المدخل فان الجزء المستقطع من البداية (offset) الى نهاية String.

ومخرج الدالة هو String يحتوى على الجزء المستقطع.

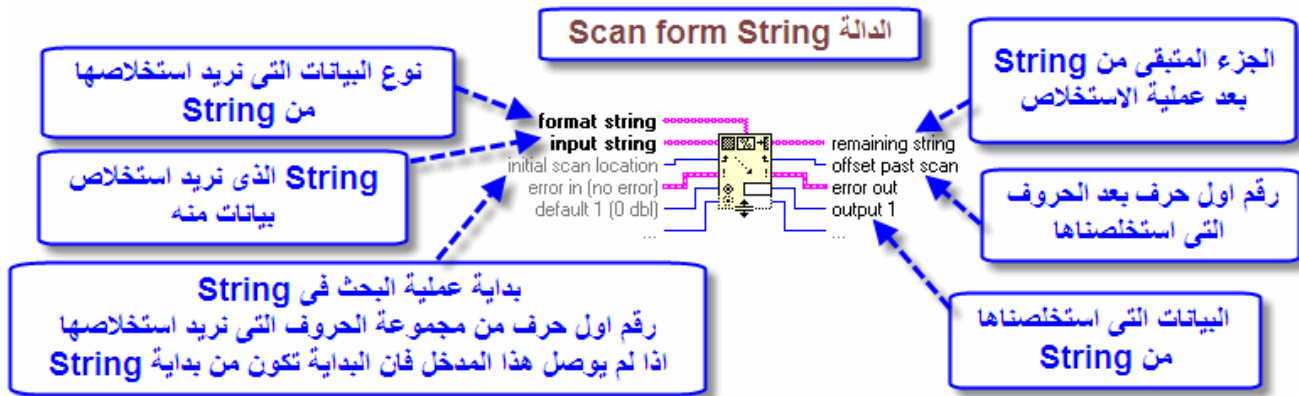


الدالة Scan from String

وظيفة هذه الدالة هي عكس وظيفة الدالة Format into String فهي تقوم باستخلاص الاعداد الرقمية من Strings . فالدالة تقوم بتحويل مجموعة حروف في String تمثل عدد الى قيمة رقمية وهذه الحروف قد تشمل على (0 الى 9 او + او - او e او E او نقطة)

هذا String يحتوى على مجموعة حروف رقمية - وهى الحروف المظلمة -
نريد استخلاصها على هيئة قيمة رقمية وليست String

Temp+45.72deg



ومداخل هذه الدالة هي

Input String : وهو مدخل String الذي نريد استخلاص البيانات منه.
Format String : وهو يحدد نوع البيانات التي نريد استخلاصها ومحتوى Format String مثل المدخل Format String في دالة Format into String وايضا كما سنرى يمكن كتابته يدويا او استخدام النافذة Edit Scan String.
Initial Scan Location : مكان او رقم اول حرف سنقوم باستخلاصه.

ومخارج هذه الدالة هي :

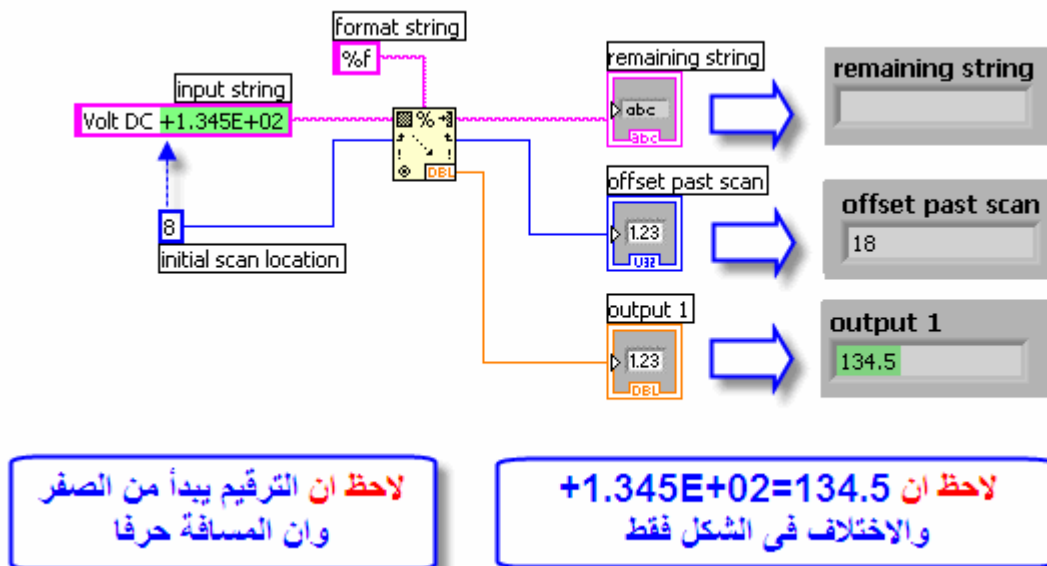
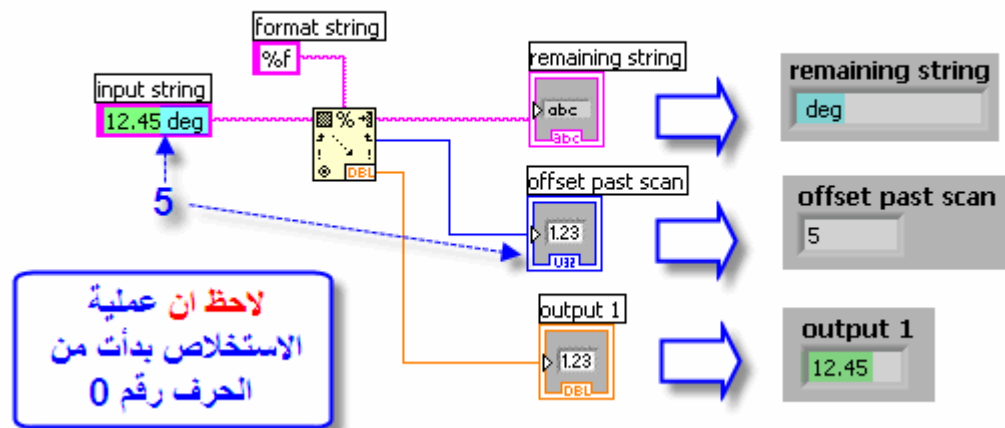
Remaining String : وهو الجزء المتبقى من String بعد عملية البحث.
Offset Past String : وهو رقم اول حرف بعد الجزء الذي تم استخلاصه.
Output : هي البيانات التي استخلصناها ونوعها كما هو موجود في Format String.
ويمكن استخدام هذه الدالة لاستخلاص انواع مختلفة من البيانات ولكن عادة تستخدم لاستخلاص قيم رقمية من Strings.
وكمثال يتم استقبال البيانات من اجهزة القياس من خلال Serial Port على هيئة Strings ونستخدم هذه الدالة لاستخلاص القيم المقاسة من Strings المستقبلية.

ويمكن تحديد Format String بكتابته يدويا ولاستخلاص قيمة عددية من String نكتب داخل Format String العلامة % والحرف f هكذا %f .

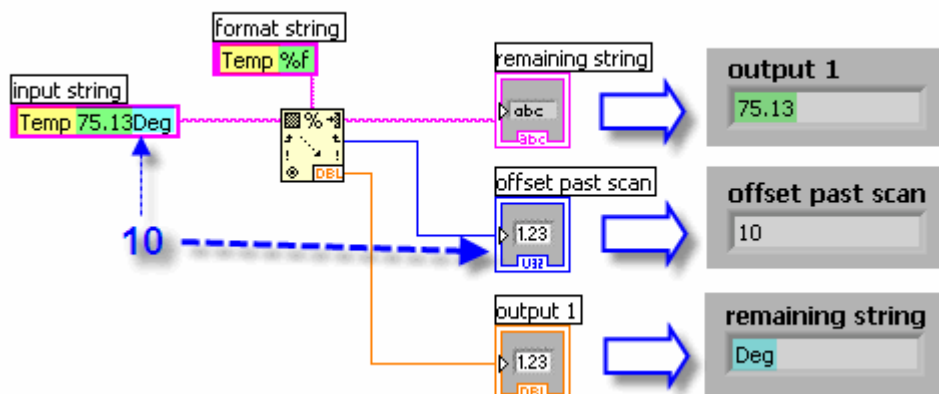
ويمكن استخدام النافذة Edit Scan String لتحديد Format String وذلك بالضغط مرتين متتاليتين بالزر الايسر للماوس على الدالة او الضغط على الدالة بالزر الايمن واختيار Edit Scan String. وبعد تحديد نوع البيانات التى نريد استخلاصها سيتم وضع Format String تلقائيا وسيكون موصلا بالدالة.



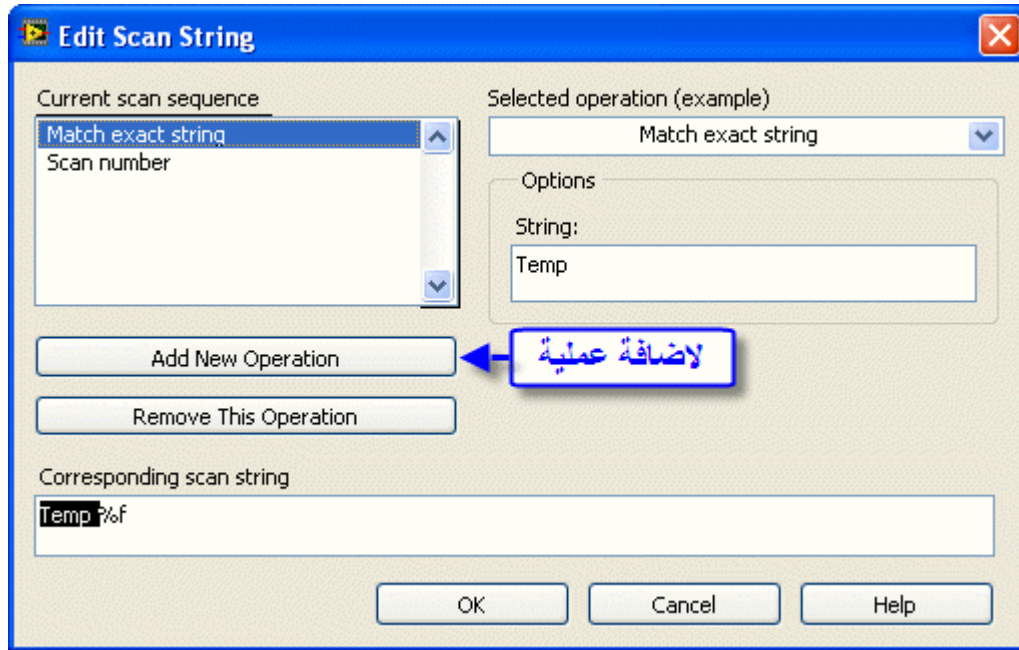
لاحظ ان Format String و Scan String يعبران عن نفس المعنى وهو نوع وشكل البيانات التى نريد استخلاصها.



يمكن كتابة نص ثابت في Format String وسيتم استخلاصه من String ويخرج كما هو وتفيد هذه الطريقة في استخلاص شكل معين مكون من نص و عدد كما يتضح من المثال.

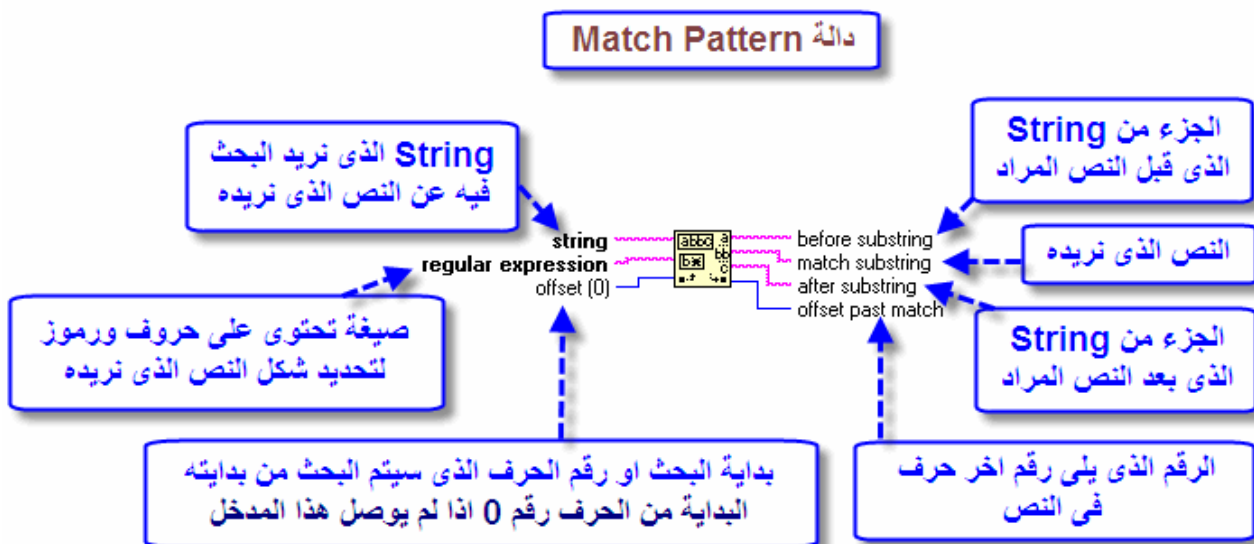


ويمكن عمل ذلك من خلال نافذة Edit Scan String وذلك بتحديد العملية Match Exact String وكتابة Temp ثم اضافة عملية وجعلها العملية Scan Number.



الدالة Match Pattern

تقوم هذه الدالة بالبحث في String عن نص بشكل معين. ويعتمد شكل النص الذي نريده على Regular Expression وهو صيغة تحتوى على حروف ورموز تحدد شكل النص الذي نريده.



ومدخل هذه الدالة هي :

String : وهو String الذي يحتوى على النص الذي نريد البحث عنه.

Regular Expression: وهي صيغة تحتوى على حروف ورموز لتحديد شكل النص الذى نريده
وسنوضح كيفية كتابة هذه الصيغة.

Offset: وهي رقم الحرف فى String الذى سنبدأ منه عملية البحث وقيمة هذا المدخل اذا لم يتم
التوصيل اليه هي 0.

ومخارج هذه الدالة هي :

Before Substring: وهو الجزء من String الذى يسبق النص الذى تم الحصول عليه.

Match String: وهو النص الذى تم الحصول عليه من عملية البحث .

After Substring: وهو الجزء من String الذى يلي النص الذى تم الحصول عليه.

Offset Past Match: هو الرقم الذى يلي رقم اخر حرف من النص الذى تم الحصول عليه.

ونلاحظ انه بعد عملية البحث تم تقسيم String الى ثلاثة اجزاء بناء على صيغة Regular Expression.

واذا لم يتم ايجاد النص المراد داخل String فان المخارج تكون قيمها كالتالى:

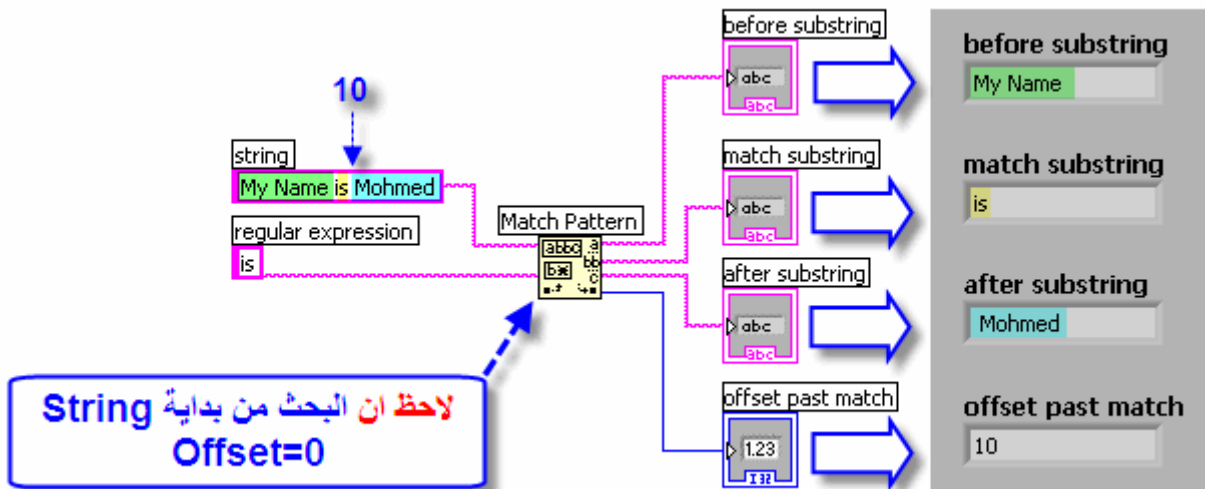
Before Substring: يحتوى على كل محتوى String الذى نبحث فيه.

Match String: تكون فارغة.

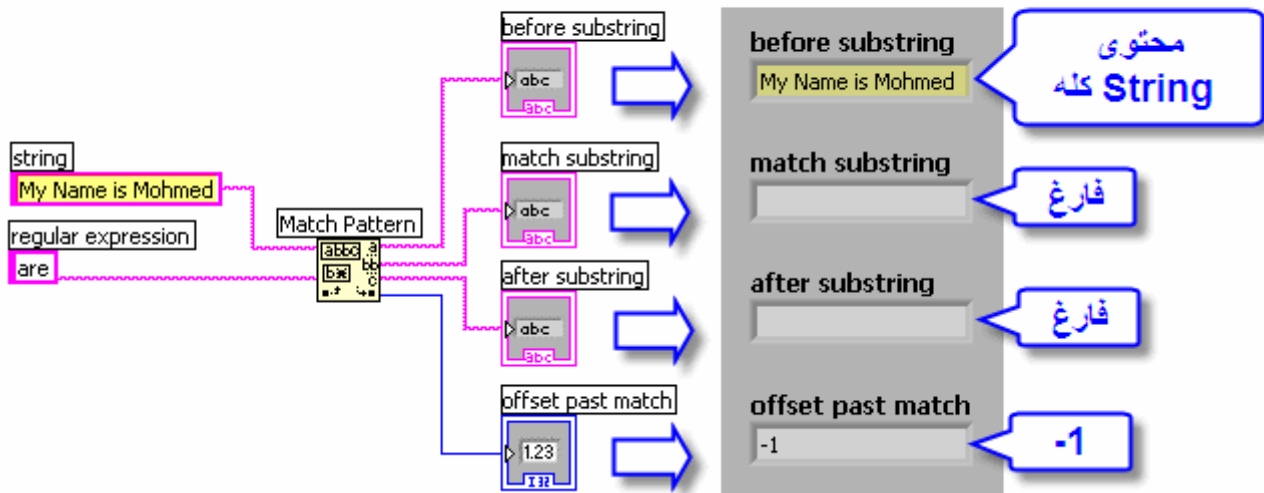
After Substring: تكون فارغة.

Offset Past Match: يكون قيمته ب -1 مما يعنى انه لم يتم ايجاد النص المراد داخل String.

مثال : يمكن ان يكون Regular Expression عبارة عن كلمة نريد ان نبحث عنها فى String.



لم يتم ايجاد كلمة "are" داخل String



ويمكن ان تحتوى Regular Expression على رموز خاصة لتحديد شكل النص الذى نريد البحث عنه ولكل رمز دلالة او معنى وسنستعرض هذه الرموز ومعناها:

الرمز	معناه من خلال الدالة Matching Pattern
.	النقطة ومعناها اى حرف فى هذا المكان مثال l.g تعنى ابحث عن اى كلمة تتكون من الحرف l ثم اى حرف ثم حرف g مثل الكلمات leg او lag او lug
?	ومعناها ان ما قبلها مباشرة قد لا يكون موجودا او موجودا مرة واحدة فقط مثال be?t تعنى ابحث عن كلمة تتكون من الحرف b ثم الحرف t مباشرة مثل bt وهنا حرف e الذى هو قبل العلامة ? ليس موجودا. او ان تتكون الكلمة من الحرف b ثم الحرف e ثم الحرف t مثل كلمة bet وهنا حرف e موجودا مرة واحدة. والصيغة be?t لا تنطبق على الكلمة best وذلك لان هناك حرف بين e و t
\	ومعناها الغاء تفسير الرمز الذى بعدها على انه رمز خاص وانما تفسيره على انه حرف عادى مثال \. تعنى نقطة عادية ولن يتم تفسير النقطة على انها اى حرف كما سبق وايضا \? تعنى علامة استفهام ولا يتم تفسير العلامة كما سبق وتستخدم العلامة \ فى تمثيل الحروف الغير قابلة للطباعة فمثلا \b معناها Backspace \f معناها Form Feed \n معناها New Line

ls معناها مسافة	
lr معناها Carriage Return الرجوع لاول السطر	
lt معناها Tab	
\xx (حيث xx رقم بالنظام السداسى عشر) ومعناها الحرف الذى ASCII Code له هو xx مثال 41\ تعنى الحرف A كبير لان ASCII Code له بالنظام السداسى عشر هو 41	
^ اذا جاءت اول حرف فى Regular Expression فمعناها ابحث عن الذى بعدها ولكن فى اول String واذا كان ليس اول String فهو لا يطابق مثال is^ معناها ابحث فى String بحيث تكون is اول String مثل is the exam easy ولكن لن يحدث تطابق مع String التى تحتوى على The exam is easy	^
ومعناها اى حرف مما بداخلها مثال [abc] تعنى الحرف a او الحرف b او الحرف c مثال [Mm]an تتطابق من Man وايضا man اى ابحث عن كلمة man سواء كان حرف m بها كبير او صغير والحرفين الاخرين an صغيرين.	[]
وهناك رموز خاصة تستخدم داخل القوسن [] وهذه الرموز هى	
- الشرطة معناها مدى من الارقام او مدى من الحروف الكبيرة او مدى من الحروف الصغيرة مثال [0-9] معناها اى رقم من 0 الى 9 مثال [a-g] معناها اى حرف صغير من a الى g مثال [L-Q] معناها اى حرف كبير من L الى Q	
الرمزين الاتيين لهما معنى فقط اذا جاؤا كأول حرف داخل القوسين []	
~ معناها اى حرف سواء قابل للكتابة او غير قابل للكتابة ماعدا الحروف التى بعدها او فى المدى الذى بعدها مثال [~0-9] وتعنى اى حرف ماعدا الحروف فى المدى من 0 الى 9	
^ معناها اى حرف من الحروف القابلة للكتابة وايضا المسافة و Tab ماعدا الحروف التى بعدها او فى المدى الذى بعدها مثال [^0-9] تعنى اى حرف قابل للكتابة بما فيها المسافة او Tab ماعدا الحروف من 0 الى 9	^
+ معناها اى عدد من الحرف الذى قبله ولكن لابد ان يوجد على الاقل مرة واحدة مثال be+t تعنى اى كلمة تتكون من الحرف b ثم اى عدد اكبر من الواحد من الحرف e ثم	+

	الحرف t وهى تطابق الكلمات bet و beet و beeet ولا تطابق bt لان الحرف e يجب ان يتواجد على الاقل مرة واحدة
*	معناها اى عدد من الحرف الذى قبله ويمكن الا يكون الحرف موجودا مثال be*t معناها اى كلمة تتكون من الحرف b ثم يليه الحرف t او ان يكون بينهما اى عدد من الحرف e وتطابق الكلمات bt و bet و beet و beeet وهكذا
\$	اذا جاءت كآخر حرف فى Regular Expression فهى تعنى ان يكون ما قبلها يكون موجودا فى نهاية String فمثلا deg\$ تتطابق اذا كانت الكلمة deg فى نهاية String. اذا جاء هذه الرمز ليس فى نهاية Regular Expression فسيكون حرف عادى وليس رمزا خاصا

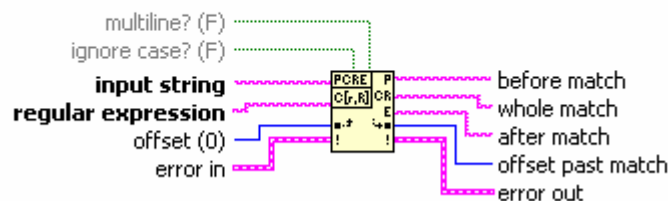
وهذه بعض الامثلة لـ Regular Expression باستخدام الرموز السابقة

الكلمات التى تتطابق معه	Regular Expression
VOLTS	VOLTS
معناها ابحث عن كلمة اول حرف بها V كبير او صغير ثم الحرف الذى يليه الحرف O كبيرا او صغيرا ثم الذى يليه الحرف L صغيرا او كبيرا وهكذا اى ابحث عن كلمة volt ويمكن ان يكون اى حرف فيها كبير او صغير مثل Volts او VOLTS او volts او VOLTs وهكذا	[Vv][Oo][Ll][Tt][Ss]
ابحث عن مسافة او علامة + او علامة -	[+ -]
اى مجموعة من الارقام المتتابة وعدد هذا الارقام واحد او اكثر	[0-9]+
مجموعة تتكون من حرف او كثر من الحروف المسافة و Tab و New Line و Carriage Return	[\t\r\n\s]+
اى مجموعة حروف عددهم على الاقل واحد ولكن هذه المجموعة لا تحتوى على اى رقم من الارقام من 0 الى 9	[~0-9]+
تتطابق اذا كانت كلمة Level موجود فى بداية String مثل Level=10 ولا تتطابق مع The Level=10	^Level
تتطابق اذا كانت كلمة Volt فى نهاية String مثل The Input =10 Volt ولا تتطابق مع The Input =10 Volt Only	Volt\$

(.*)	<p>اى مجموعة حروف بين قوسين او قوسين فارغين</p> <p>لاحظ ان . معناها اى حرف</p> <p>والعلامة * معناها اى عدد من الحرف الذى قبلها او ان يكون غير موجود</p> <p>وهذا كله بين قوسين</p>
([~()]*)	<p>اى مجموعة حروف بين قوسين ولكن هذين القوسين لا يحتويان على اى اقواس .</p> <p>ويمكن ان يكون القوسين فارغين</p> <p>لاحظ ان [~()] تعنى اى حرف ماعدا القوسين</p> <p>وان [~()]* تعنى اى عدد من الحروف ماعدا الاقواس ويمكن ان لا يوجد اى حرف</p> <p>وهذا كله بين قوسين ([~()]*)</p>
[]	الاقواس
[cd][ao][tg]	<p>اى كلمة تبدأ بالحرف c او d ثم الحرف a او o ثم الحرف t او g</p> <p>مثل cat و dog و dot و cot و cog و هكذا</p>

والدالة Match Pattern سريعة وفعالة فى البحث داخل String ولكنها لا تدعم كل امكانيات Regular Expression وتوجد دالة اخرى ذات امكانيات اكبر ولكنها بطيئة نسبية وهى دالة Match Regular Expression

دالة Match Regular Expression



وللاطلاع على امكانيات اكثر لـ Regular Expression واستخدامها مع هذه الدالة يمكن الاستعانة بالبحث فى الانترنت وهناك مصادر كثيرة فى هذه الموضوع.

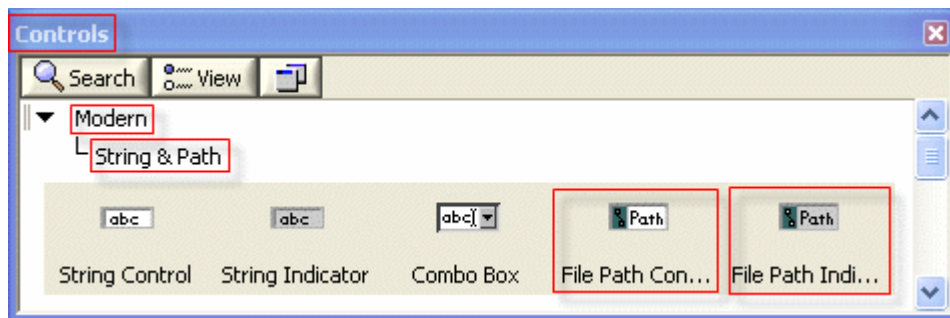
File Paths

قبل ان نتعرف على كيفية التعامل مع الملفات نتعرف على نوع من البيانات لتمثيل مسارات الملفات والمجلدات على وحدات التخزين مثل Hard disk وهو File Path.

ويوجد في LabVIEW لتمثيل مسارات الملفات File Path Control و File Path Indicator و File Path Constant.

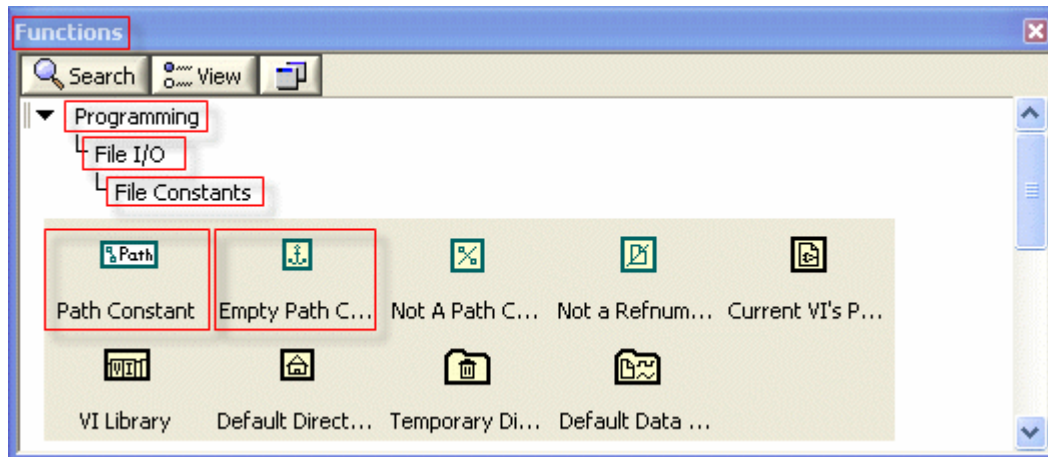
ويتم ادراج File Path Control او File Path Indicator في Front Panel من

Controls Palette>>Modern>>String & Path



ويتم ادراج File Path Constant في Block Diagram من

Functions Palette>>Programming>>File I/O>>File Constants



Path Indicator

Path Indicator

C:\LabVIEW\readfile.vi

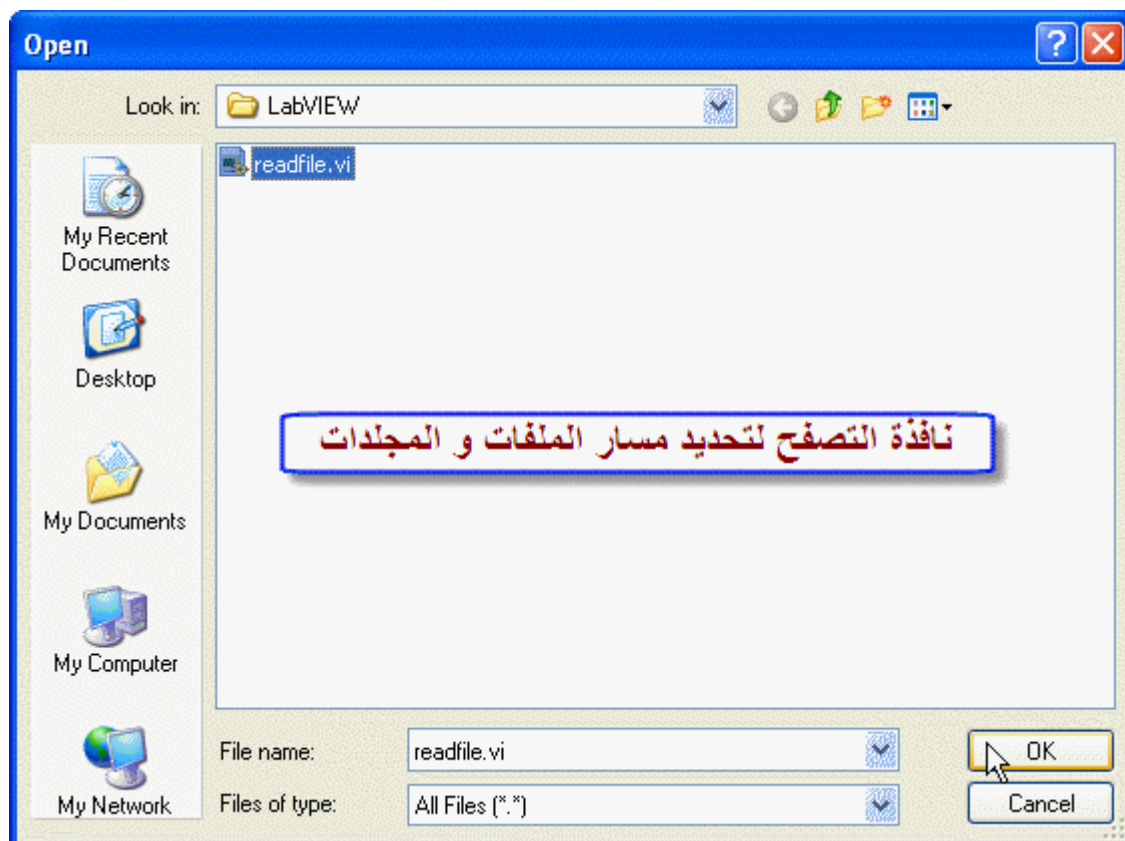
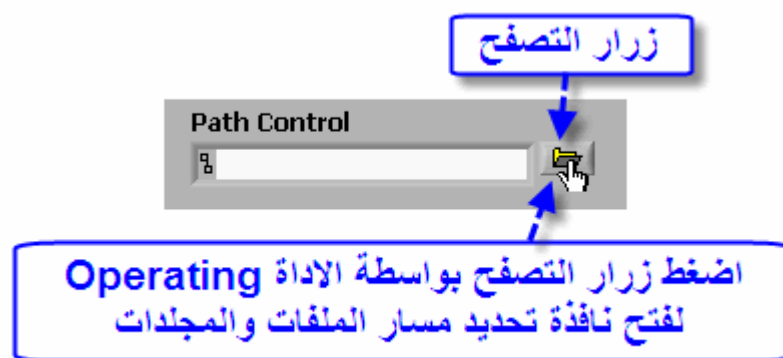
Path Control

Path Control

C:\LabVIEW\readfile.vi



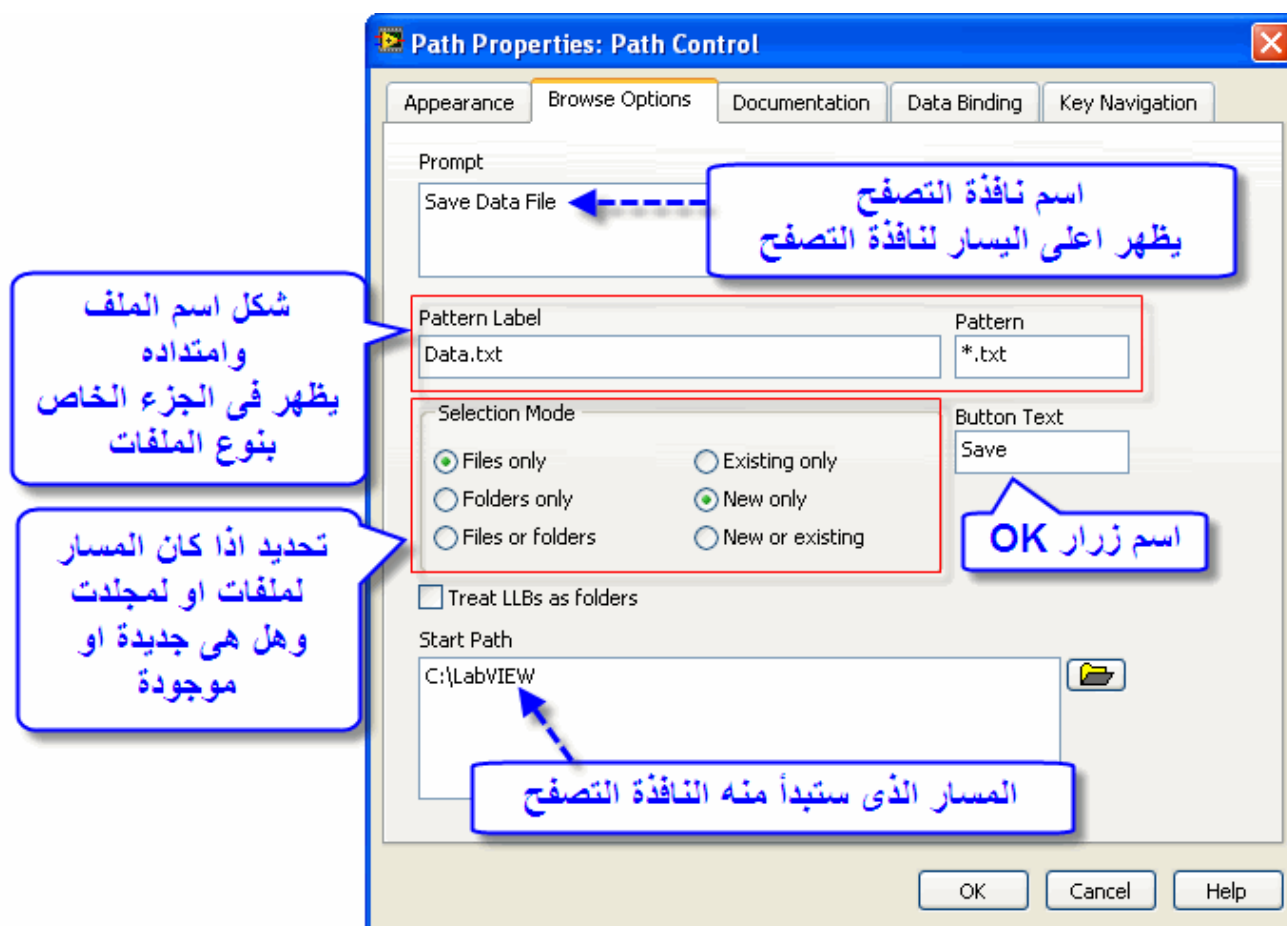
ويوجد مع File Path Control زرار لفتح نافذة لتصفح الملفات والمجلدات الموجودة على وحدات التخزين لتحديد المسار الذي نريده.



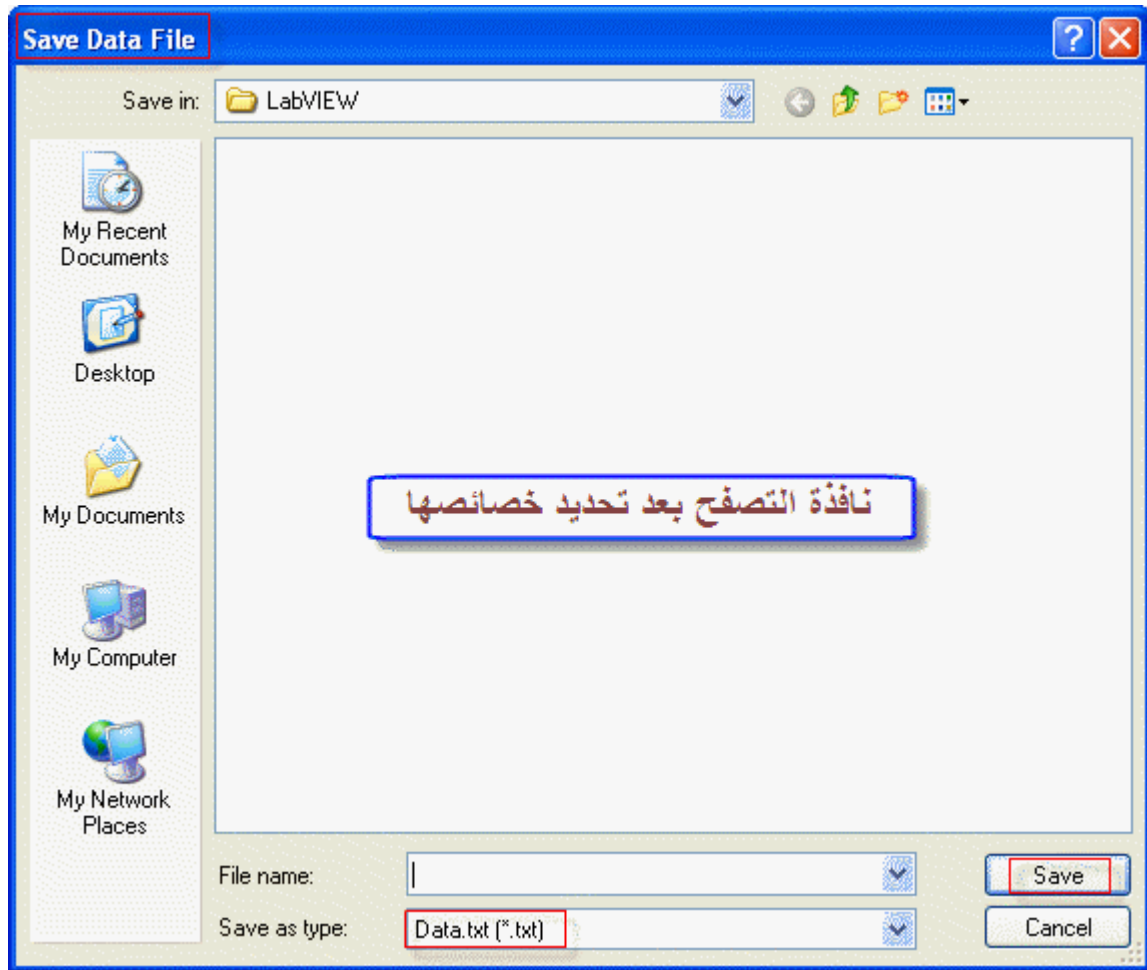
ويمكن تحديد خصائص نافذة التصفح وذلك من خلال الضغط بالزر الايمن للماوس على File Path واختيار Browse Option.



لتظهر لنا نافذة نحدد فيها خصائص نافذة التصفح وكيفية اختيار الملفات.



وهذا شكل نافذة الخصائص بعد تحديد خصائصها



ملحوظة هامة :

✓ تحدد File Path Control و نافذة التصفح الخاصة به فقط مسار الملف لاستخدامه فيما بعد من خلال الدوال الخاصة بالملفات.

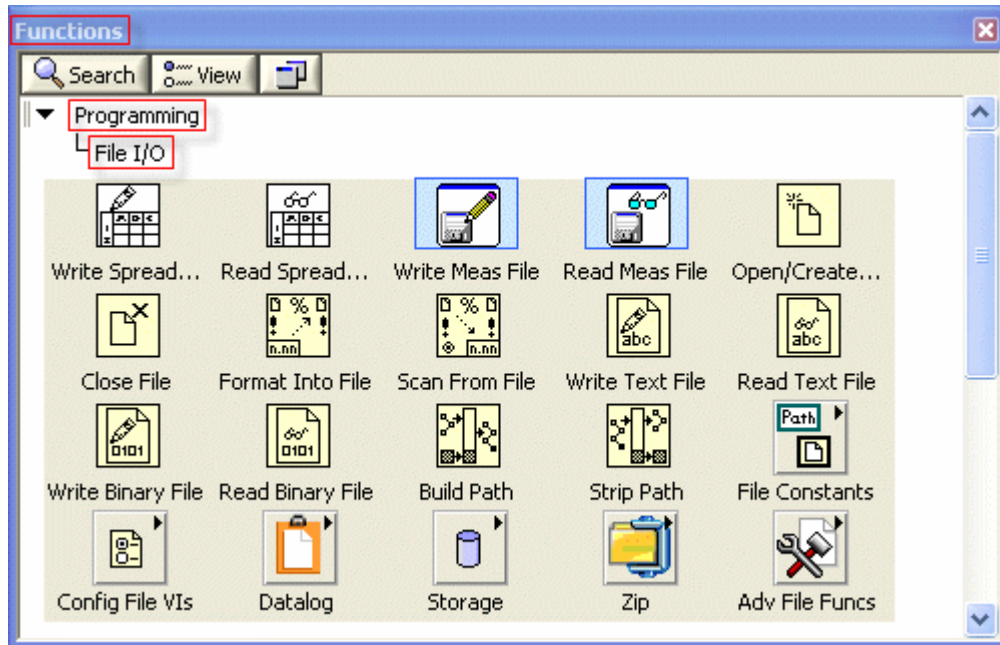
فمثلا اذا حددنا مسار ملف جديد فليس معنى ذلك انه اصبح موجودا ولكن حددنا مساره لكي يتم انشاءه بعد ذلك من خلال دوال التعامل بالملفات.

✓ زررار التصفح يمكن الضغط عليه لفتح نافذة التصفح اثناء تصميم البرنامج او اثناء تنفيذه.

File Input /Output

نظرا للاهمية الكبيرة لتخزين البيانات فى الملفات واستعادتها يوجد فى LabVIEW مكتبة كاملة من الدوال للتعامل مع الملفات وسوف نتعرف فى هذا الدرس على الدوال الاساسية للتعامل مع الملفات. ويمكن ادراج هذه الدوال الاساسية من

Factions Palette>>Programming>>File I/O



ويوجد نوعان من الدوال او VIs التى تتعامل مع الملفات فى LabVIEW وهما :

High-Level File I/O

وهى دوال و VIs سهلة الاستخدام تقوم بجميع العمليات الخاصة بالملفات من فتح او انشاء وكتابة او قراءة وغلق للملفات ، وهى تتعامل مع ملفات ذات فورمات معينة . وهى نسبيا بطيئة ويفضل عدم وضعها فى Loop لتكرار عملية القراءة او الكتابة فى الملفات ومن هذه VIs التى سنتعرف عليها :

Write to Measurement File Express VI

Read from Measurement File Express VI

Write to Spreadsheet File

Read from Spreadsheet File

: Low-Level File I/O

وتتميز هذه الدوال بالمرونة حيث ان كل دالة تقوم بعملية محددة على الملفات فهناك دالة لفتح او انشاء الملف واخرى للكتابة و اخرى للقراءة واخرى لغلق الملف.

وتتميز هذه الملفات بسرعة القراءة والكتابة . ومن هذه الدوال التى سنتعرف عليها

Open/Create/Replace File

Close File
Write to Text File
Read From Text File
Write to Binary File
Read from Binary File

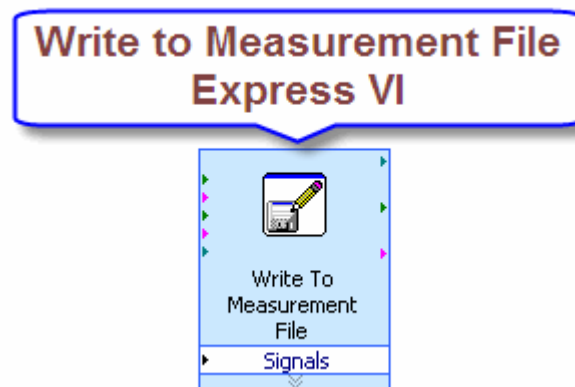
وقبل ان نبدأ التعرف على هذه الدوال نوضح النوعين الاساسيين من الملفات :

النوع الاول : Binary Files وفيه تخزن الملفات على هيئة Bytes. ويتم قراءة هذا النوع من الملفات بنفس الطريقة التي خزن بها والا ستكون البيانات التي تم قراءتها غير صحيحة.

النوع الثاني : Text File وفيه تكون البيانات على هيئة حروف وكل حرف يخزن كرقم وهذا الرقم هو ASCII Code للحرف. لذلك عند التخزين في هذه الملفات نحول ما نريده الى String وهذا ال String يتم تخزينه في الملف بالـ ASCII Code. وما يميز هذه الملفات انه يمكن ان يتم قراءتها بواسطة برامج تحرير النصوص مثل Notepad. ولكنها تأخذ سعة تخزينية اكبر من Binary Files.

اولا High Level File I/O

Write to Measurement File Express VI



تقوم هذه Express VI بتخزين البيانات في ملف ، هذا الملف يمكن ان يكون من النوع :

• Measurement Data File من النوع Text

(text-based measurement file) وهو يكون بالامتداد LVM و يمكن فتحه ببرامج تحرير النصوص مثل Notepad او برامج Spreadsheet مثل Excel. والبيانات في هذا الملف يمكن ان تكون مفصولة بـ Tab او يمكن ان نجعلها مفصولة بـ فاصلة عادية (,). وقد يحتوى الملف LVM ايضا على معلومات عن البيانات مثل وقت وتاريخ الحصول على هذه البيانات.

بيانات فى ملف مفصولة بـ Tab

1	2	3
2	4	6
3	6	9
4	8	12
5	10	15

بيانات فى ملف مفصولة بفاصلة عادية ,

1, 2, 3
2, 4, 6
3, 6, 9
4, 8, 12
5, 10, 15

- Measurement Data File من النوع Binary وهو يكون بالامتداد TDM ويمكن تخزين فيه اى نوع من البيانات. وهو اسرع من النوع LVM فى التعامل وياخذ مساحة تخزينية اصغر ويمكن تخزين الارقام فيه باكثر دقة عن تخزينها فى ملف من النوع LVM.

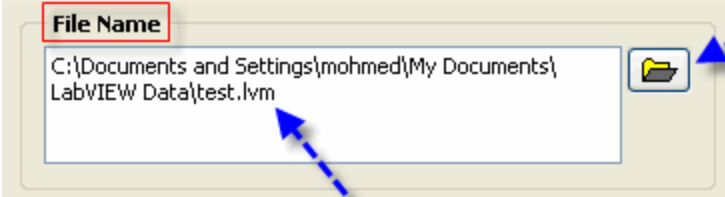
عادة ما تستخدم هذه Express VI لتخزين الاشارات التى يتم قياسها بواسطة كروت DAQ .

عند ادراج هذه Express VI تظهر لنا نافذة اعدادات Express VI لتحديد خصائص الملف الذى سيتم تخزين البيانات فيه. ويمكن فتح هذه النافذة فى اى وقت بالضغط مرتين متتاليتين بالزر الايسر للماوس على ايقون VI.

نافذة اعداد

Write To Measurement File Express VI

نتعرف على الاختيارات في هذه النافذة :



File Name

C:\Documents and Settings\mohmed\My Documents\LabVIEW Data\test.lvm

اضغط زر التصفح لتحديد مسار واسم الملف

مسار واسم الملف الذي سيتم التخزين فيه

ملحوظة: يتم التخزين في هذا الملف إذا لم يمكن اسم الملف على مدخلات VI



File Format

☒ Text (LVM)

☐ Binary (TDM)

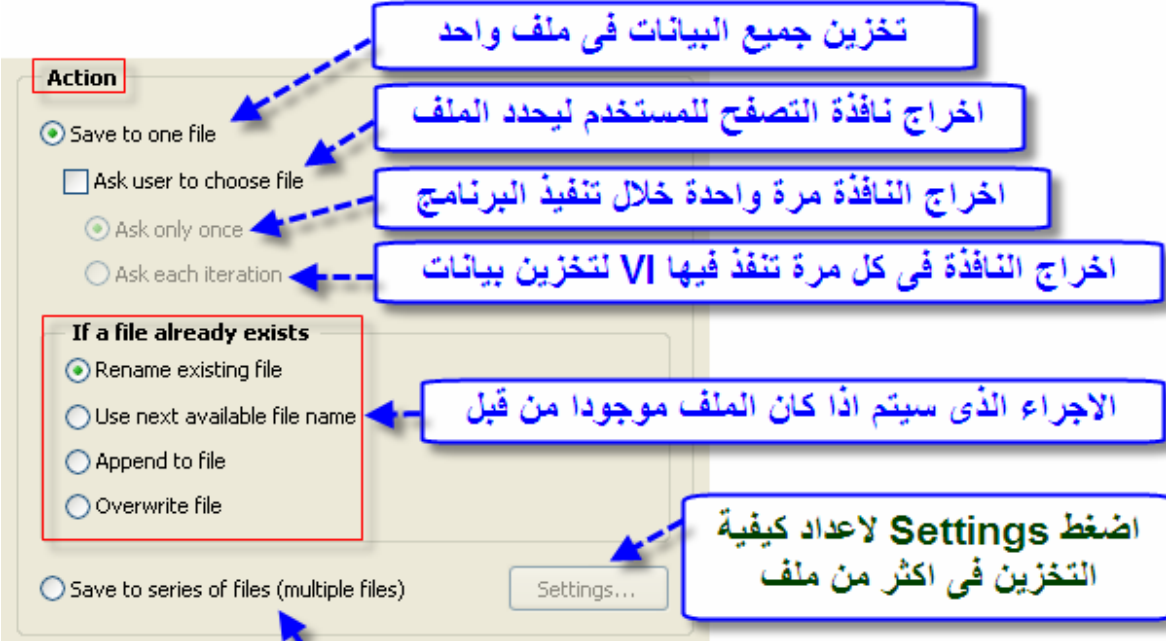
☒ Lock file for faster access

نوع الملف الذي نريده LVM او TDM

في حالة TDM

تقوم هذه الخاصية بتسريع التعامل مع الملفات

وفي هذه الحالة لا يمكن ان يتعامل اثنين Vis مع نفس الملف في نفس الوقت



Action

☒ Save to one file

☐ Ask user to choose file

☒ Ask only once

☐ Ask each iteration

If a file already exists

☒ Rename existing file

☐ Use next available file name

☐ Append to file

☐ Overwrite file

☐ Save to series of files (multiple files)

Settings...

تخزين جميع البيانات في ملف واحد

اخراج نافذة التصفح للمستخدم ليحدد الملف

اخراج النافذة مرة واحدة خلال تنفيذ البرنامج

اخراج النافذة في كل مرة تنفذ فيها VI لتخزين بيانات

الاجراء الذي سيتم اذا كان الملف موجودا من قبل

اضغط Settings لاعداد كيفية التخزين في اكثر من ملف

في كل مرة - اثناء تنفيذ البرنامج - تخزن فيها بيانات يتم تخزينها في ملف جديد مثل test_001.lvm و test_002.lvm و ...test_003.lvm

وضع Header به معلومات عن البيانات
في كل مرة تخزن فيها بيانات
يفضل هذا الاختيار اذا كانت البيانات مختلفة في الخصائص
مثل معدل قراءة هذه البيانات من كارت DAQ

Segment Headers

- ☒ One header per segment
- ☐ One header only
- ☐ No headers

وضع Header واحد لكل البيانات
ويفضل ذلك اذا كانت جميع البيانات لها نفس الخصائص

عدم وضع Header او معلومات عن البيانات
ملحوظة: هذا الاختيار متاح في حالة LVM فقط

لكل عمود من البيانات سيتم وضع عمود X خاص به
ويفضل ذلك اذا كان مثلا البيانات في كل عمود لها قيم للوقت
مختلفة عن البيانات في الاعمدة الاخرى

X Value Columns

- ☐ One column per channel
- ☐ One column only
- ☒ Empty time column

يوجد عمود واحد X لكل اعمدة البيانات
ويفضل ذلك اذا كان مثلا البيانات في كل الاعمدة
لها نفس قيم الوقت

عدم وضع قيم الوقت
هذا الاختيار متاح في حالة LVM فقط

Delimiter

- ☒ Tab
- ☐ Comma

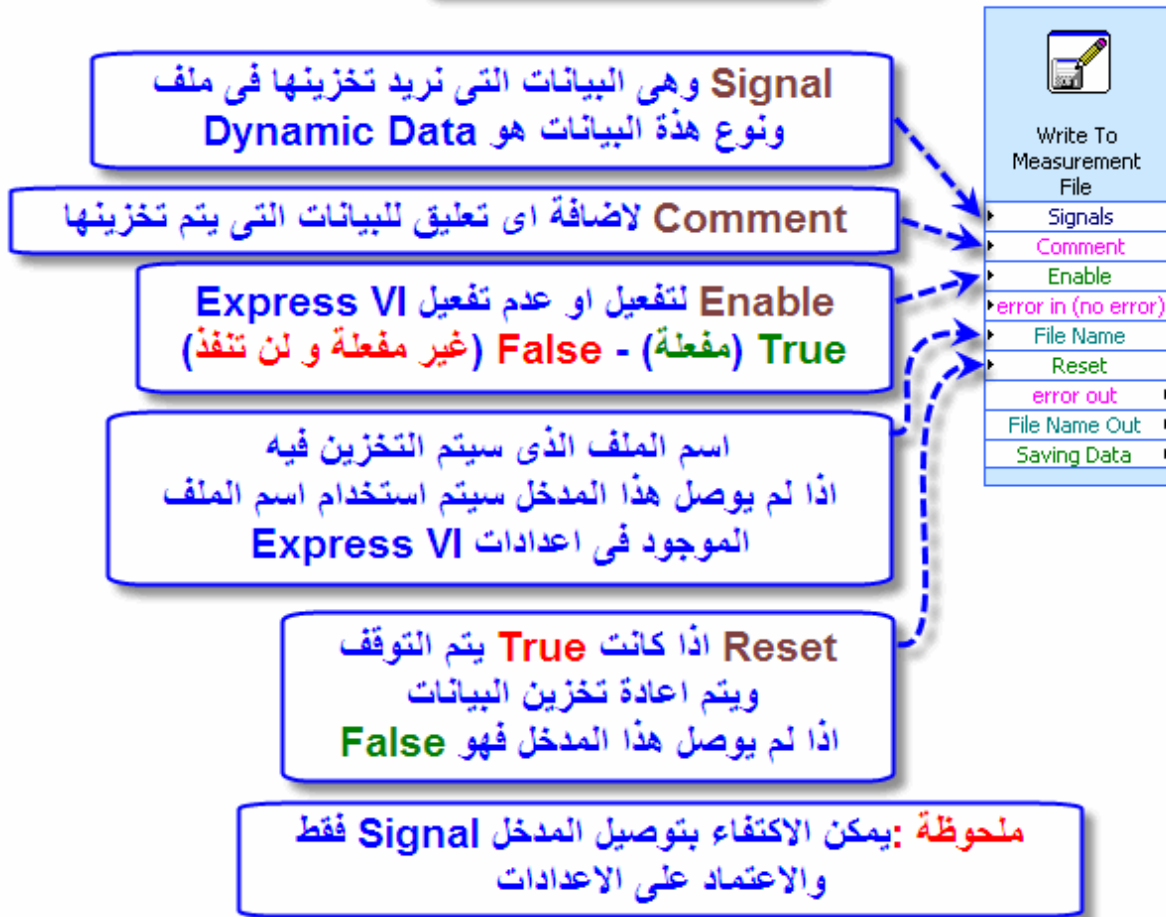
تحديد الفاصل بين البيانات في نفس الصف
Tab او الفاصلة العادية ,

File Description

ادخال وصف للملف وهذا الوصف سيوضع في Header الملف

مداخل هذه Express VI هي :

مداخل Express VI

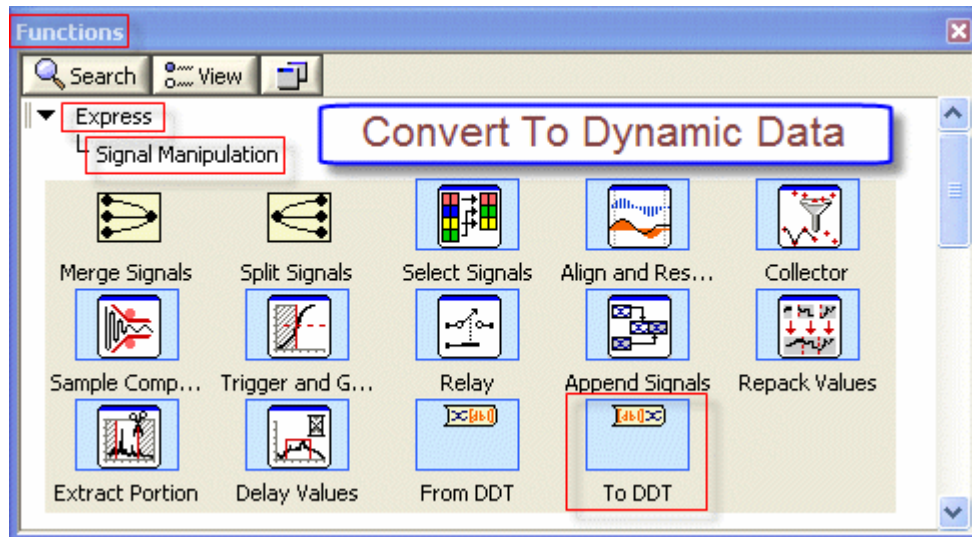


مخارج Express VI

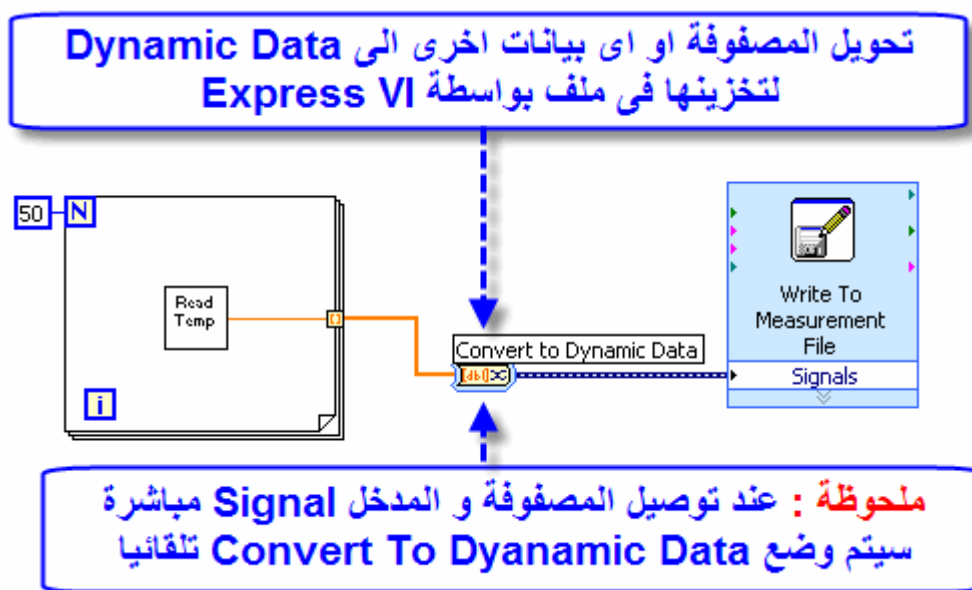


وتدخل البيانات المراد تخزينها الى هذه **Express VI** بحيث تكون من النوع **Dynamic Data** الذى يمكن تجميع فيه اكثر من اشارة. ويمكن تحويل اى نوع من البيانات الى **Dynamic Data** بواسطة **Convert To Dynamic Data** التى يمكن ادراجها من

Functions Palette>>Express>>Signal Manipulation

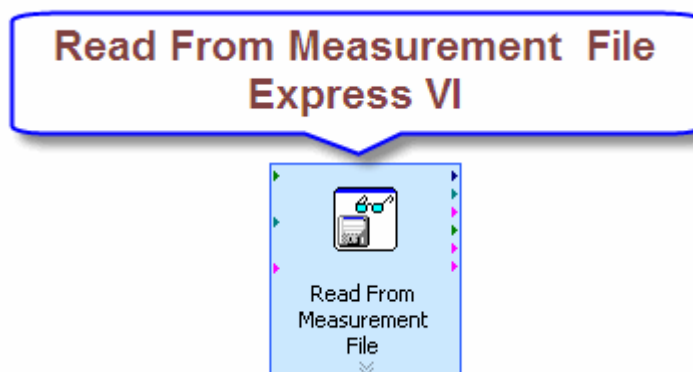


مثال : تخزين مصفوفة



Read from Measurement File Express VI

تقوم هذه Express VI باستدعاء البيانات المخزنة فى ملفات من النوع LVM او Text File او النوع TDM.



وعند ادراج هذه Express VI تظهر لنا نافذة اعدادات Express VI لتحديد خصائص الملف و البيانات التى سيتم استرجاعها. ويمكن فتح هذه النافذة فى اى وقت بالضغط مرتين متتاليتين بالزر الايسر للماوس على ايقونة VI.

نافذة اعداد

Read From Measurement File Express VI

File Name
C:\Documents and Settings\mohmed\My Documents\LabVIEW Data\test.lvm

Time Stamps
☒ Relative to start of measurement
☐ Absolute (date and time)

File Format
☒ Text (LVM)
☐ Binary (TDM)
☒ Lock file for faster access

Segment Size
☒ Retrieve segments of original size
☐ Retrieve segments of specified size (Samples: 100)

Generic Text File
☒ Read generic text files
Delimiter
☒ Tab
☐ Comma
Start row of numeric data: 1
☐ First row is channel names
☐ First column is time channel
Decimal Point
☒ . (dot)
☐ , (comma)

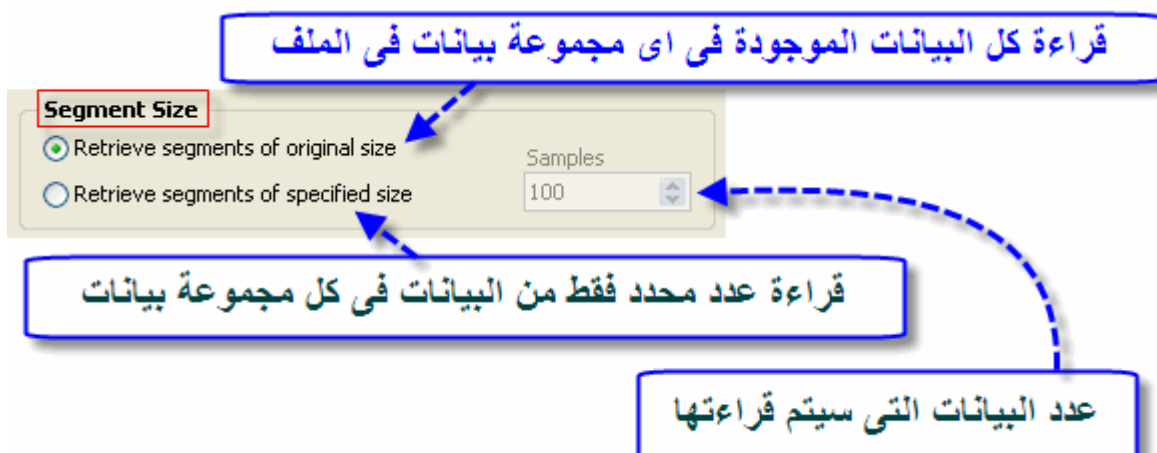
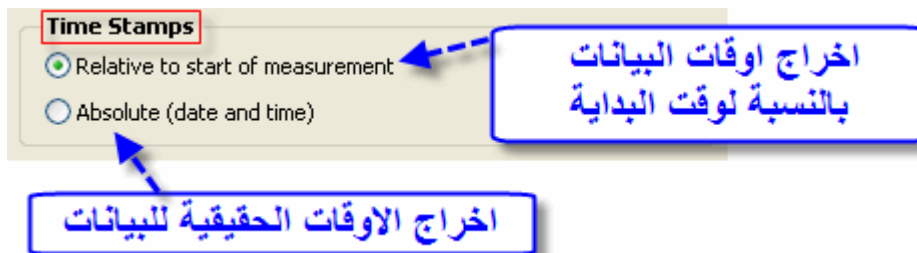
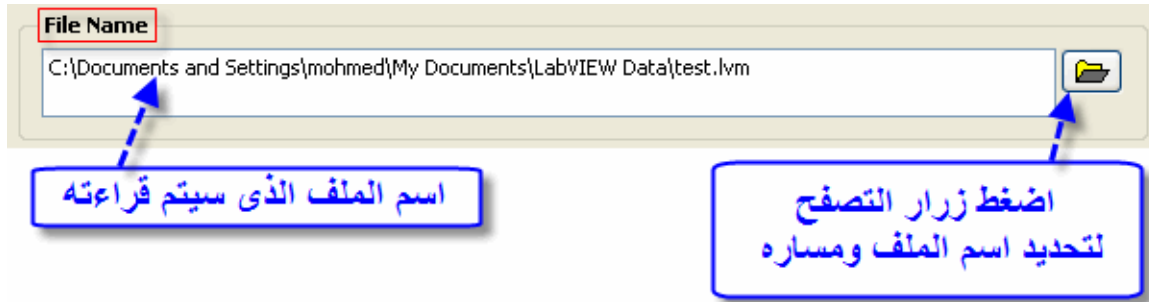
Action
☐ Ask user to choose file

Read File Now

Sample data

OK Cancel Help

و الاختيارات الموجودة فى هذه النافذة هى :



ملحوظة: قد تكون البيانات في الملف على هيئة مجموعات من البيانات كل مجموعة لها Header وفي كل مرة يتم القراءة من الملف يتم قراءة مجموعة من تلك البيانات حتى يتم الانتهاء من قراءة الملف كله. وقد توجد البيانات في مجموعة واحدة.

إذا كان الملف الذي نريد قراءته ملف **Text** عام ولكن ليس بפורمات **LVM** نقوم باختيار هذه الاختيار
 اما اذا كان الملف من النوع **LVM** فأننا نلغي هذه الاختيار

Generic Text File

☒ Read generic text files Read File Now

Delimiter

☒ Tab نوع الفاصل بين البيانات في نفس الصف
☐ Comma Tab او فاصلة عادية ,

Start row of numeric data: 1 رقم صف البيانات الرقمية في الملف

☐ First row is channel names تحديد اذا كان اول صف هو اسماء البيانات الموجودة في كل عمود
☐ First column is time channel تحديد ان اول عمود يعبر عن قيم الزمن

Decimal Point

☒ . (dot) تحديد علامة العلامة العشرية المستخدمة في الملف
☐ , (comma)

Generic Text File

☒ Read generic text files Read File Now

Delimiter

☒ Tab نوع الفاصل بين البيانات في نفس الصف
☐ Comma Tab او فاصلة عادية ,

Start row of numeric data: 1 رقم صف البيانات الرقمية في الملف

☐ First row is channel names تحديد اذا كان اول صف هو اسماء البيانات الموجودة في كل عمود
☐ First column is time channel تحديد ان اول عمود يعبر عن قيم الزمن

Decimal Point

☒ . (dot) تحديد علامة العلامة العشرية المستخدمة في الملف
☐ , (comma)

Text

اضغط الزرار لتشاهد كيف سيتم قراءة الملف

مثال: ملف **Text** يحتوي على بيانات بهذا الشكل

ch1	ch2	ch3
0.000	0.000	0.000
1.000	2.000	3.000
2.000	4.000	6.000
3.000	6.000	9.000
4.000	8.000	12.000

نقوم بعمل الاعدادات كما بالشكل

Generic Text File

☒ Read generic text files

Delimiter

☒ Tab

☐ Comma

Start row of numeric data
2

☒ First row is channel names

☐ First column is time channel

Decimal Point

☒ . (dot)

☐ , (comma)

Sample data

ch1	ch2	ch3
0	0	0
1	2	3
2	4	6
3	6	9
4	8	12

Read File Now

مداخل Express VI

اسم الملف الذي سيتم القراءة منه
إذا لم يوصل هذا المدخل سيتم استخدام اسم الملف
الموجود في اعدادات Express VI

Enable لتفعيل او تعطيل Express VI
True (مفعلة) - **False** (غير مفعلة و لن تنفذ)

Read From Measurement File2

EOF?

error out

File Name

Signals

Comment

Description

Enable

error in (no error)

File Name

مخارج Express VI

End of File?
True يكون هذا الخرج
عندما تصل القراءة الى نهاية الملف

البيانات التي تم قراءتها وتكون على هيئة
Dyanmic Data

اخراج اي تعليق مخزن عن البيانات

اخراج اي وصف مخزن للبيانات

اسم الملف الذي تتم القراءة منه

Read From Measurement File2

EOF?

error out

File Name

Signals

Comment

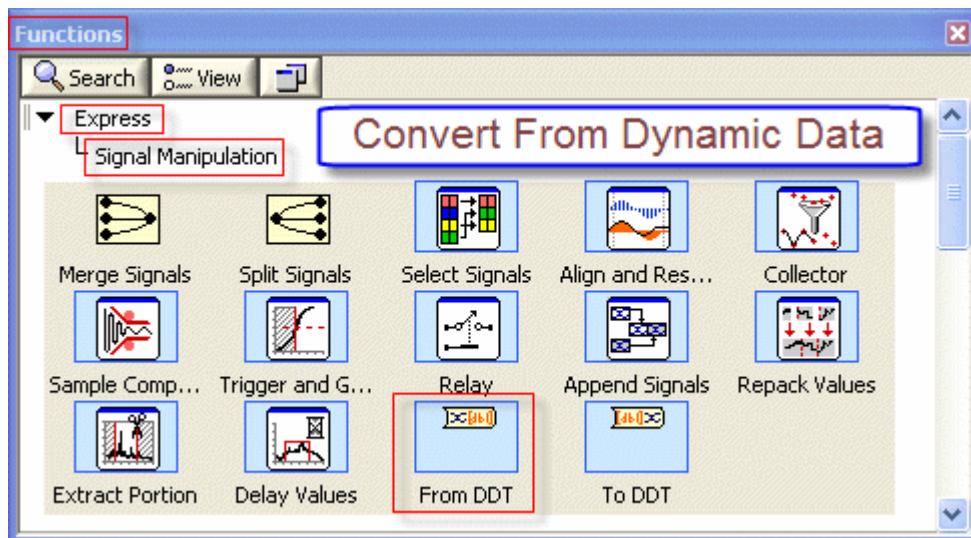
Description

Enable

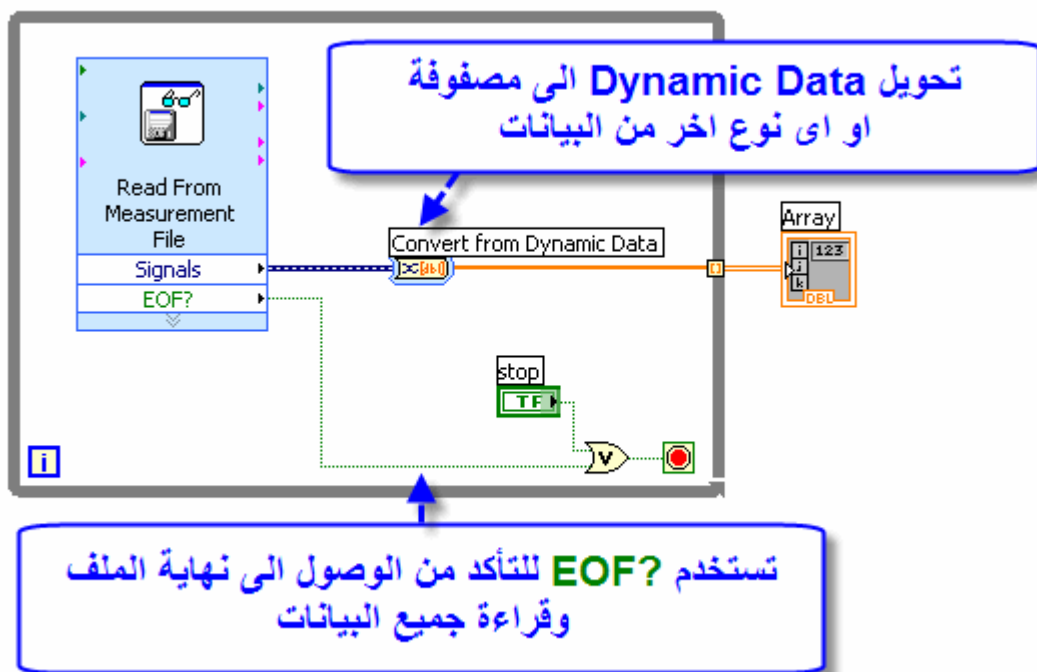
error in (no error)

File Name

تقوم هذه Express VI بقراءة البيانات من الملف وإخراجها على هيئة Dynamic Data وكما نعلم يمكن توصيلها مباشرة الى Graph لرسمها ويمكن تحويلها الى أى صورة أخرى وذلك باستخدام Convert from Dynamic Data التى يمكن إدارجها من Functions Palette>>Express>>Signal Manipulation



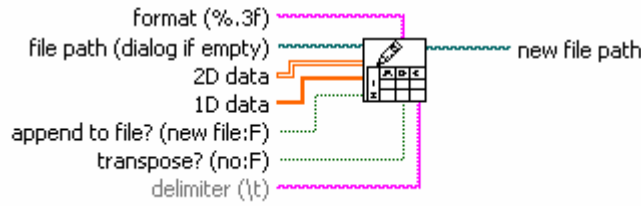
مثال: قراءة مصفوفة من البيانات من ملف



Write to Spreadsheet File VI

تقوم هذه VI بتخزين البيانات الرقمية الموجودة فى مصفوفة الى ملف من النوع Text على هيئة Spreadsheet حيث يمكن فتحه بسهولة ببرامج Spreadsheet مثل Excel. وهذه VI تقوم بتحويل الارقام الموجودة فى المصفوفة الى String ثم تكتب هذا String فى ملف. وهذا الملف يمكن ان يكون ملف جديدا او ان يكون ملف موجود ويتم اضافة البيانات اليه.

Write To SpreadSheet Files VI



ومداخل هذه VI كالتالى:

Format: وهو لتحديد شكل الارقام عند تحويلها الى String وتخزينها فى الملف وصيغة Format هي كما ذكرناها فى هذا الدرس .واذا لم يوصل هذا المدخل فان قيمة Format هي %.3f اى رقم ذو كسر عشرى وعدد الارقام العشرية تساوى 3.

File Path: اسم ومسار الملف ويمكن تحديد مسار الملف من خلال File Path Control او يكون خرج لأى VI اخرى . واذا لم يتم توصيل هذا المدخل او ان قيمة مسار الملف فارغة ستظهر VI عند تنفيذها نافذة التصفح الى المستخدم تطلب منه تحديد مسار واسم الملف المراد التخزين فيه.

2 D Data: مدخل البيانات التى نريد تخزينها اذا كانت فى مصفوفة ذات بعدين.

1 D Data: مدخل البيانات التى نريد تخزينها اذا كانت فى مصفوفة ذات بعد واحد.

ملحوظة: اذا تم التوصيل الى المدخلين 2D Data و 1D Data فسيتم تخزين المصفوفة ذات البعد الواحد ويتم تجاهل المصفوفة ذات البعدين.

Append to File?: لو ان هذا المدخل True تقوم هذه VI باضافة البيانات الى البيانات المخزن بالملف سابقا. واذا كانت False تقوم VI باستبدال البيانات الموجودة فى الملف بالبيانات الجديدة التى سيتم تخزينها. وقيمة هذا المدخل اذا لم يتم التوصيل اليه هي False .

ملحوظة: اذا لم يكن الملف موجود سيتم انشاء الملف وستخزن البيانات به.

Transpose?: اذا كان هذا المدخل True يتم تحويل اعمدة المصفوفة الى صفوف والصفوف الى اعمدة بحيث تخزن الاعمدة على هيئة صفوف فى الملف و الصفوف تخزن على هيئة اعمدة. اما اذا كان False يتم تخزين المصفوفة كما هى بدون تحويل وهذا هو الوضع العادى فى حالة عدم توصيل هذا المدخل.

Delimiter: نوع الفاصل بين البيانات فى الصف فى الملف والقيمة العادية هي Tab ويمكن توصيل اى فاصل مثل الفاصلة العادية (,) او الفاصلة المنقوطة (;).

Write To Spreadsheet File VI

شكل الارقام عند تحويلها الى String وتخزينها

مسار الملف المراد التخزين فيه
إذا كان مسار الملف فارغاً أو المدخل غير موصل
سيتم اخراج نافذة التصفح لاختيار الملف عند التنفيذ

مدخل البيانات المراد تخزينها سواء في صورة
مصفوفة ذات بعدين أو مصفوفة ذات بعد واحد

True: إضافة البيانات الجديدة الى البيانات الموجودة في الملف
False: حذف البيانات المخزنة بالملف واستبدالها بالبيانات الجديدة
إذا كان الملف ليس موجوداً سيتم انشاء ملف جديد

True: تحويل صفوف المصفوفة الى اعمدة
والاعمدة الى صفوف وتخزينها على ذلك
False: تخزين المصفوفة كما هي بدون تحويل
إذا لم يوصل هذا المدخل فان قيمته **False**

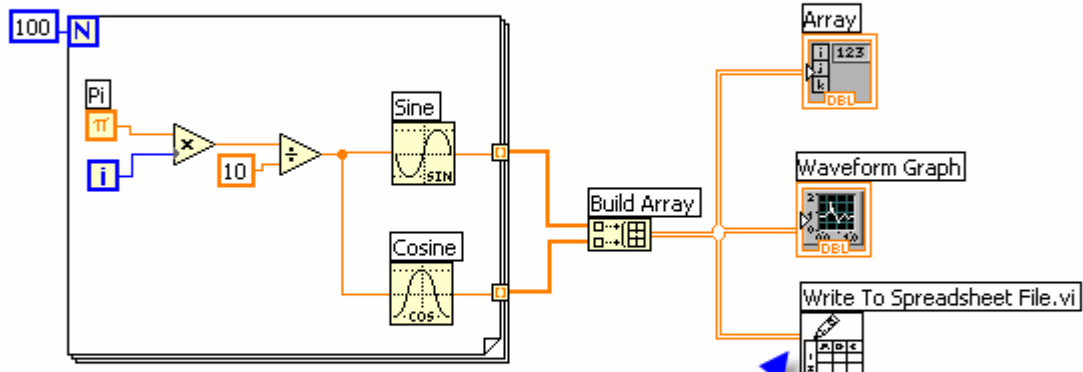
مسار الملف
الذي تم التخزين فيه

format (%.3f)
file path (dialog if empty)
2D data
1D data
append to file? (new file:F)
transpose? (no:F)
delimiter (\t)

الفاصل بين البيانات
في نفس الصف في الملف
إذا لم يوصل هذا المدخل
فان الفاصل **Tab**

مثال: تخزين مصفوفة ذات بعدين

تخزين مصفوفة ذات بعدين في ملف من النوع Text Spreadsheet



ستحل البيانات الجديدة محل البيانات القديمة بالملف
أو يتم انشاء ملف جديد اذا لم يكن الملف موجوداً
ستخزن المصفوفة كما هي بدون تحويل

Format="%.3f"
Append to File=False
Transpose=False
Delimiter=Tab

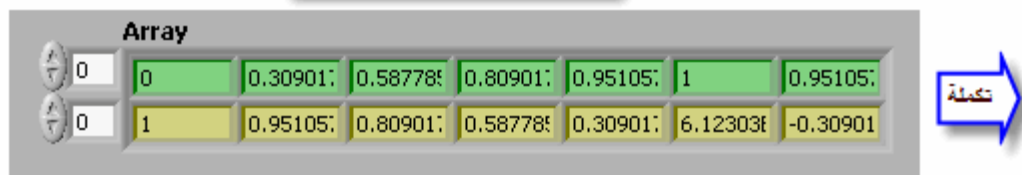
ملحوظة: تم ادراج دالة Cosine و Sine من

Functions Palette>>Mathematics>>Elementary & Spatial Functions>>
Trigonometric Function

وتم ادراج الثابت π من

Functions Palette>>Programming>>Math & Scientific Constants

المصفوفة ذات البعدين

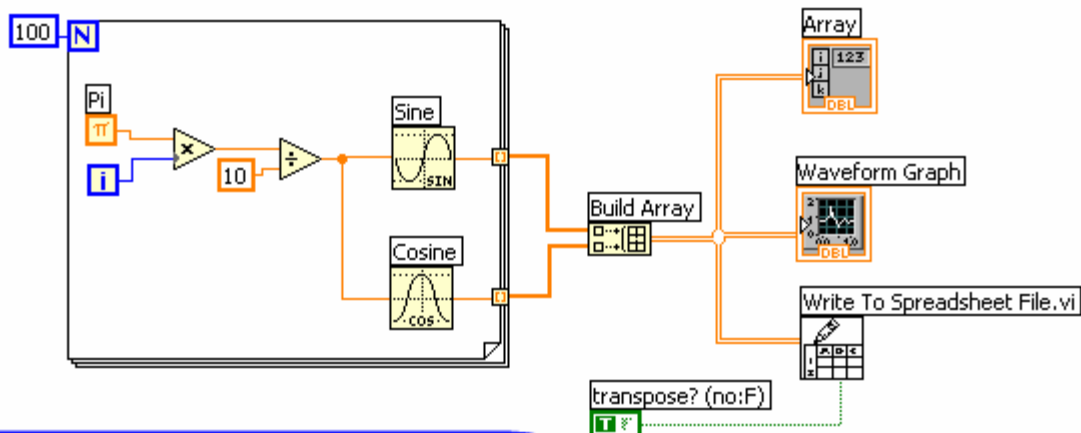


البيانات المخزنة في الملف

0.000	0.309	0.588	0.809	0.951	1.000	0.951
1.000	0.951	0.809	0.588	0.309	0.000	-0.309

لاحظ ان المصفوفة تم تخزينها كما هي بدون تحويل
كما لاحظ ان عدد الارقام العشرية التي تم تخزينها في كل رقم 3=

تخزين مصفوفة ذات بعدين في ملف
من النوع Text Spreadsheet مع عمل Transpose



سيتم تحويل صفوف المصفوفة الى اعمدة
والاعمدة الى صفوف وتخزينها على ذلك

Transpose=True

المصفوفة ذات البعدين

Array	0	0	0.30901	0.58778	0.80901	0.95105	1	0.95105
0	1	0.95105	0.80901	0.58778	0.30901	6.12303	-0.30901	

تكملة

البيانات المخزنة في الملف

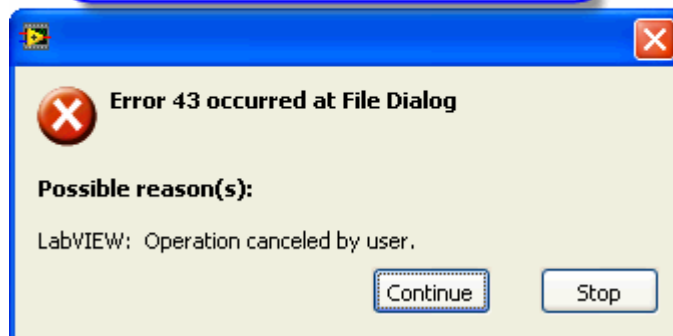
0.000	1.000
0.309	0.951
0.588	0.809
0.809	0.588
0.951	0.309
1.000	0.000
0.951	-0.309

تكملة

لاحظ انه تم تحويل اعمدة المصفوفة الى صفوف والصفوف الى اعمدة وتخزينها على ذلك

ملحوظة: عند عدم توصيل مسار الملف الى مدخل الدالة و اظهرت الدالة عند التنفيذ نافذة التصفح وضغط المستخدم Cancel اى لم يحدد مسار الملف سيحدث الخطأ رقم 43 وتظهر رسالة تفيد بذلك.

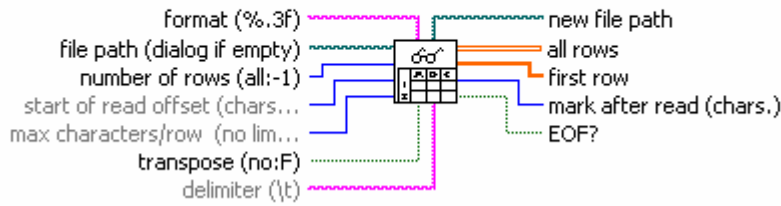
رسالة خطأ لان المستخدم ضغط Cancel فى نافذة التصفح



Read from Spreadsheet File VI

تقوم هذه VI بقراءة عدد معين من السطور او الصفوف من ملف Text يحتوى على بيانات رقميه ثم تحويل هذه السطور او الصفوف الى مصفوفة من الارقام. ويمكن ان تقوم VI بقراءة الملف من اوله او من بداية حرف معين يتم تحديد مكانه (رقمه).

Read From Spreadsheet VI



ويمكن ان تقوم هذه VI بعمل Transpose للمصفوفة قبل اخراجها.

ومداخل هذه VI هي :

Format : صيغة تحدد شكل الارقام عند تحويلها من String الى رقم وقيمة Format العادية "%.3f" اى رقم ذو كسر عشرى وعدد الارقام العشرية =3.

File Path : مسار الملف الذى سيتم القراءة منه . اذا لم يوصل هذا المدخل او كان المسار فارغا سيتم اخراج نافذة التصفح للمستخدم لتحديد مسار الملف.

Number of Rows : لتحديد اقصى عدد من السطور سيتم قراءتها. وعدد السطور التى يتم قراءتها يمكن ان يكون اقل اذا تم الوصول الى نهاية الملف . اذا كان قيمة هذا المدخل اقل من الصفر (-1) يتم قراءة كل السطور او كل الصفوف وهذه هى قيمة المدخل اذا لم يتم التوصيل اليه.

Start of Read Offset : وهو المكان فى الملف الذى ستبدأ منه عملية القراءة وهو يعبر عن رقم الحرف فى الملف الذى ستبدأ منه عملية القراءة .

مثال اذا كان الملف يتكون من Header فيه معلومات عن البيانات (مثل وقت وتاريخ القياس و اسماء البيانات) ثم البيانات الرقمية التى نريد قراءتها وكان عدد حروف هذا Header هى 57 فسوف نبدأ من الحرف رقم 57 (الترقيم يبدأ من الصفر). والقيمة العادية لهذا المدخل هى 0 اى من بداية الملف.

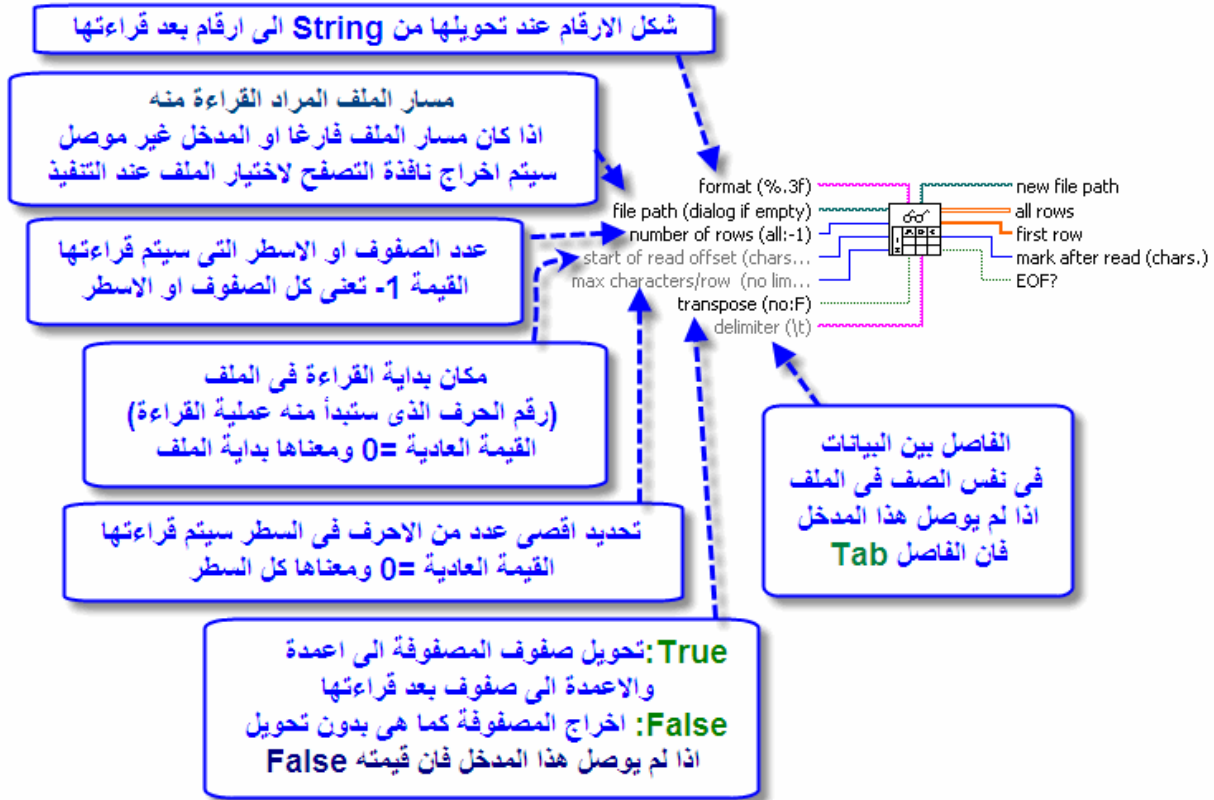
Max Character/Row : اقصى عدد من الحروف فى الصف سيتم قراءتها . واذا لم يوصل هذا المدخل فان القيمة العادية له هى 0 وهى تعنى كل حروف الصف.

Transpose? : اذا كان المدخل True يتم عمل Transpose للمصفوفة الناتجة اى تحويل صفوفها الى اعمد واعمدتها الى صفوف.

اذا كان المدخل False وهى القيمة العادية تخرج المصفوفة كما هى بدون تحويل.

Delimiter : نوع الفاصل المستخدم فى الملف . والقيمة العادية هى Tab.

مداخل Read From Spreadsheet VI



ومخارج VI كالتالي :

New File Path: مسار الملف الذي تمت عملية القراءة منه.

All Rows: مصفوفة أرقام ذات بعدين تحتوي على البيانات التي تم قراءتها.

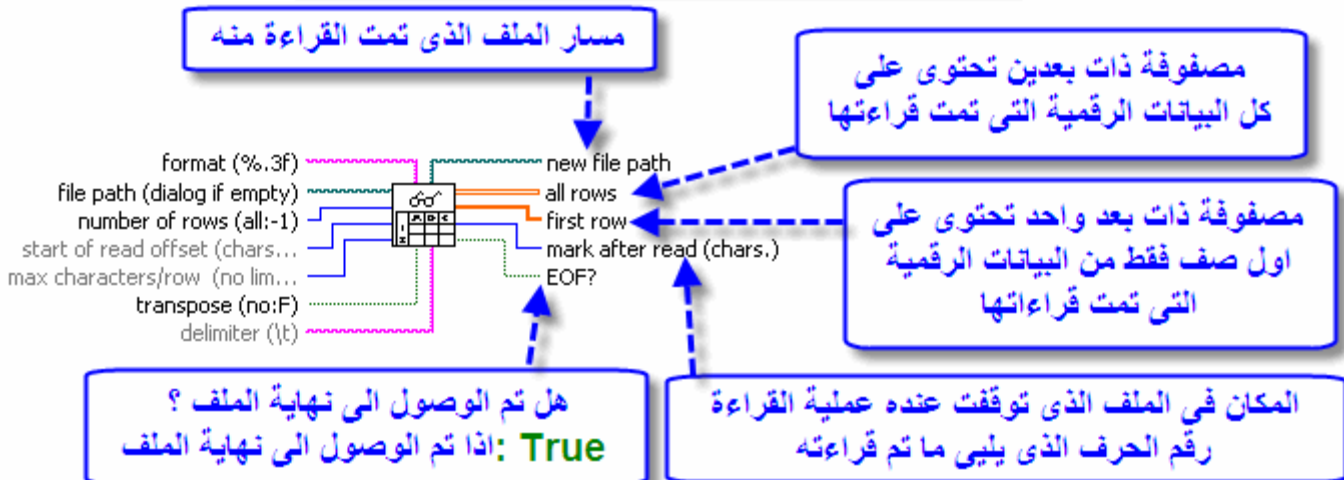
First Row: مصفوفة أرقام ذات بعد واحد تحتوي فقط على أول صف من البيانات التي تم قراءتها.

Mark After Read: وهو المكان في الملف الذي توقفت عنده عملية القراءة. وهو رقم الحرف الذي

يلى ما تم قراءته مباشرة.

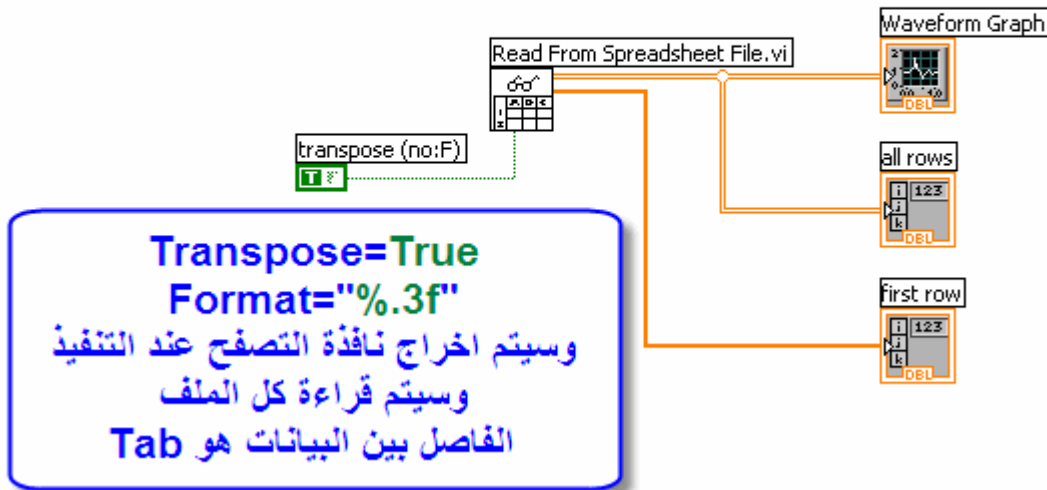
EOF? (End of File): إذا كان قيمته True فهذا يعني أنه تم الوصول إلى نهاية الملف في عملية القراءة.

مخارج Read From Spreadsheet VI



مثال :

قراءة بيانات رقمية من ملف Text
على شكل صفوف واعمدية



البيانات الرقمية التي تمت قراءتها

all rows

0.000	0.309	0.588	0.809	0.951	1.000	0.951	0.809	0.588
1.000	0.951	0.809	0.588	0.309	0.000	-0.309	-0.588	-0.809

كل الصفوف

first row

0.000	0.309	0.588	0.809	0.951	1.000	0.951	0.809	0.588
-------	-------	-------	-------	-------	-------	-------	-------	-------

الصف الاول فقط

البيانات الرقمية الموجودة في الملف

0.000	1.000
0.309	0.951
0.588	0.809
0.809	0.588
0.951	0.309
1.000	0.000
0.951	-0.309
0.809	-0.588
0.588	-0.809

لاحظ كيف تم عمل Transpose
فالعمودين تم قراءتها على هيئة صفين

ملحوظات :

Read To Spreadsheet File VI و Write to Spreadsheet File VI ليس لهما مدخل

او مخرج لـ Error Cluster فهما يقومان بالتعامل مع الاخطاء داخليا لذا يفضل استخدامهما في

التطبيقات البسيطة وليس التطبيقات المتقدمة.

من المفيد فتح هذين VI ومشاهدة Block Diagram الخاص بهما وذلك يؤدي مزيدا من الفهم لطريقة عملهما .

الفرق بين ملفات Excel Spreadsheet وملفات Text Spreadsheet

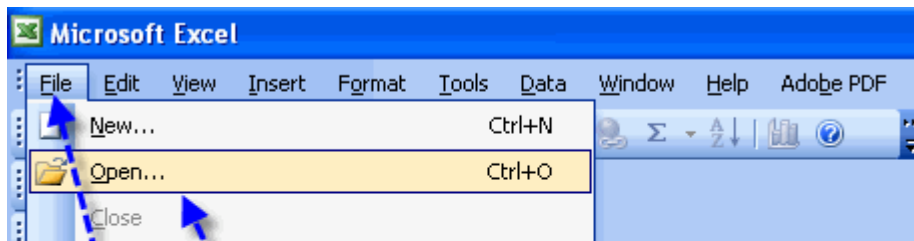
ملفات Excel Spreadsheet ذات الامتداد XLS هي ملفات من النوع Binary لها فورمات خاصة يتم فتحها بواسطة برنامج Excel او البرامج المعدة لذلك ولا يمكن فتحها ببرامج تحرير النصوص مثل Notepad وذلك لانها كما ذكرنا من النوع Binary.

اما الملفات Text Spreadsheet هي ملفات من النوع Text اى يمكن فتحها ببرامج تحرير النصوص مثل Notepad .

وعادة تخزن البيانات فى هذه الملفات على هيئة صفوف او سطور ويفصل بين البيانات فى نفس السطر Tab او الفاصلة العادية (,) او اى فاصلة اخرى. ويفصل بين كل سطر والذى يليه بـ Carriage Return و LineFeed (\r\n).

وهذه الملفات يمكن ان نضع لها اى امتداد مثل txt او csv.

ولفتح هذه الملفات بواسطة برنامج Excel نتبع الخطوات الاتية :



من برنامج Excel نضغط File>>Open



- 1- نحدد نوع الملف طبقا لامتداد
- 2- نختار الملف ثم نضغط Open

Text Import Wizard - Step 1 of 3

The Text Wizard has...
If this is correct, choose the file type that best describes your data.

Original data type:
Choose the file type that best describes your data:

☒ Delimited - Characters such as commas or tabs separate the data.
☐ Fixed width - Fields are aligned in columns with spaces.

Start import at row: File origin:

Preview of file C:\LabVIEW_Data\data.txt.

1	ch1	ch2	ch3
2	0.0000	0.0000	0.0000
3	1.0002	0.0003	0.0000
4	2.0004	0.0006	0.0000
5	3.0006	0.0009	0.0000

1- نحدد ان البيانات مفصولة

2- كما يمكن ان نحدد اول سطر في البيانات ان اردنا ذلك

3- نضغط Next

Buttons: Cancel, < Back, Next >, Finish

Text Import Wizard - Step 2 of 3

This screen lets you select how your text is affected.

Delimiters:
☒ Tab ☐ Semicolon ☐ Comma
☐ Space ☐ Other:

☐ Treat consecutive delimiters as one

Text qualifier:

Data preview:

ch1	ch2	ch3
0.000	0.000	0.000
1.000	2.000	3.000
2.000	4.000	6.000
3.000	6.000	9.000

1- نحدد نوع الفاصل بين البيانات

معينة للبيانات

2- نضغط Finish

Buttons: Cancel, < Back, Next >, Finish

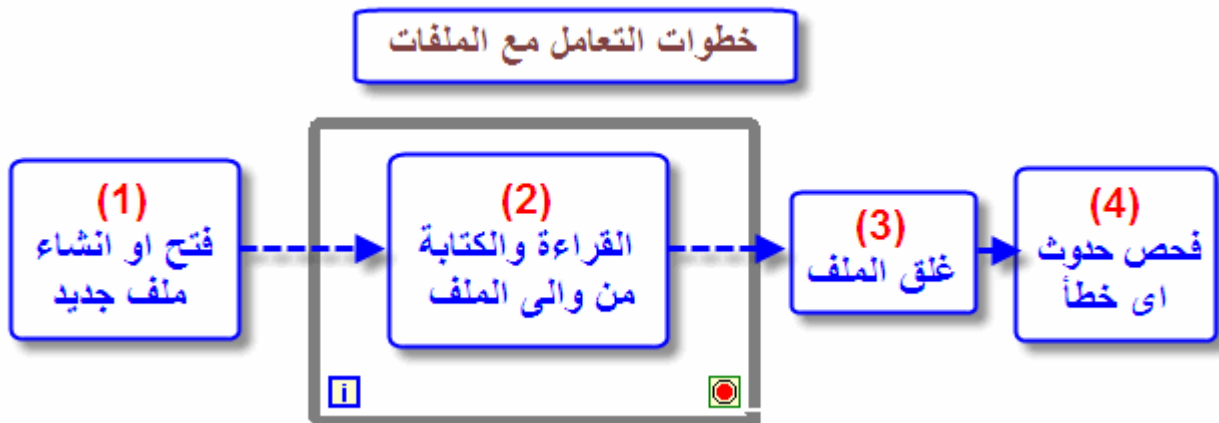
البيانات في برنامج Excel

	A	B	C	D
1	ch1	ch2	ch3	
2	0	0	0	
3	1	2	3	
4	2	4	6	
5	3	6	9	
6	4	8	12	
7				

ثانيا Low Level File I/O

لكي نتعامل مع الملفات هناك ثلاث خطوات متتالية نقوم بعملهم هم :

- 1-فتح او انشاء الملف المراد التعامل معه وعند فتحه نحصل على (Rufnum Reference Number) من خلاله يتم تحديد الملف فى باقى الخطوات.
- 2- القراءة او الكتابة من و الى الملف.
- 3- غلق الملف.
- 4- فحص حدوث اى خطأ وهذه ليست خطوة اساسيه فى التعامل مع الملفات .



وعمليتا فتح و غلق الملف تبطنان من سرعة الكتابة او القراءة من الملف ولذلك يفضل فتح الملف وعند انتهاء كل عمليات الكتابة والقراءة نغلقه.

Read and Write to Measurement File Express VIs و Read and Write to Spreadsheet File تقوم بهذه الخطوات من فتح وكتاب او قراءة و غلق. لذلك لايفضل استخدام هذه VI داخل Loops.

وسنتعرف على الدوال الاساسية لعمل هذه الخطوات.

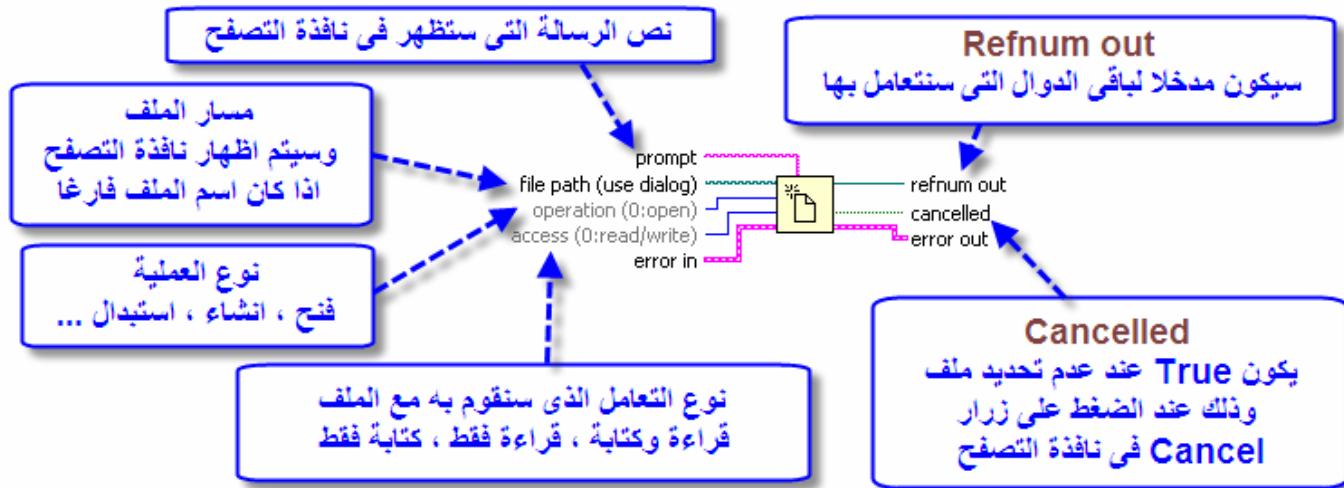
وكما ذكرنا ان هناك نوعان اساسيان من الملفات وهما Text File و Binary File وسنتعرف اولا على كيفية التعامل مع Text File ثم كيفية التعامل مع Binary File.

كيفية التعامل مع Text Files

الدالة Open/Create/Replace File

تقوم هذه الدالة بفتح او انشاء او استبدال ملف وتخرج Refnum للملف وهو رقم يدل على الملف لكي يتم التعامل معه خلال باقى الدوال.

دالة Open/Create/Replace File

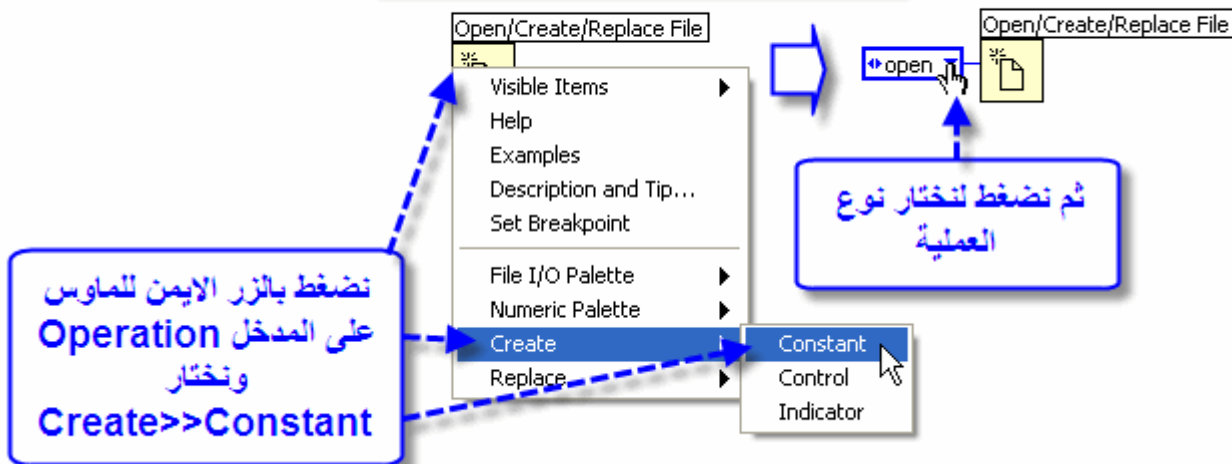


وتقوم هذه الدالة بالخطوة الاولى لفتح الملف او انشاءه ويجب ان نختار نوع العملية Operation الصحيح والا قد يحدث خطأ مثل : اختيار العملية انشاء (Create) ويكون الملف موجودا فسينتج خطأ ويمكن الاطلاع على Help لمعرفة الاخطاء التي يمكن ان تحدث.

ولتحديد نوع العملية يفضل ان نضغط بالزر الايمن للماوس على المدخل Operation ونختار Create

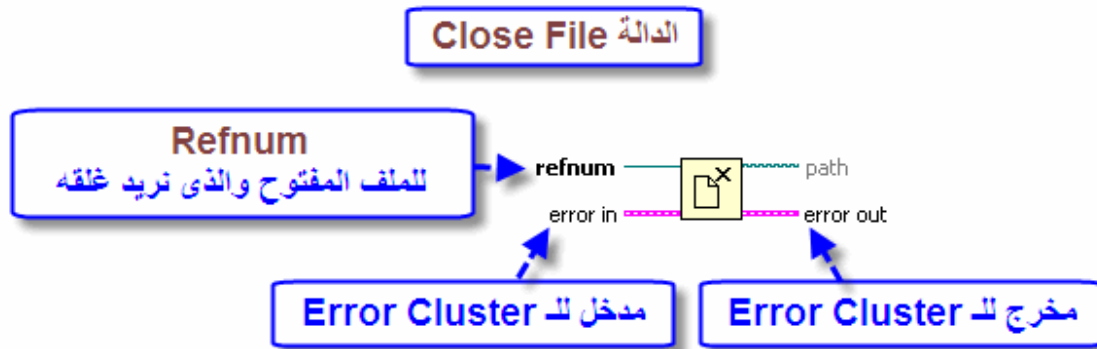
Constant

تحديد قيمة المدخل Operation

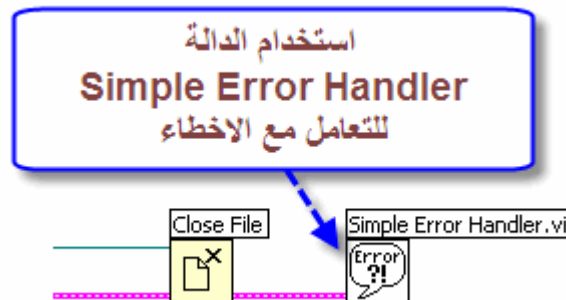


الدالة Close File

تقوم هذه الدالة بغلق الملف الذي تم فتحه من قبل.



ويمكن ان نستخدم Error Cluster الخارج من هذه الدالة لمعرفة هل حدث خطأ ام لا بواسطة VIs الخاصة بالتعامل مع الاخطاء مثل Simple Error Handler.vi.

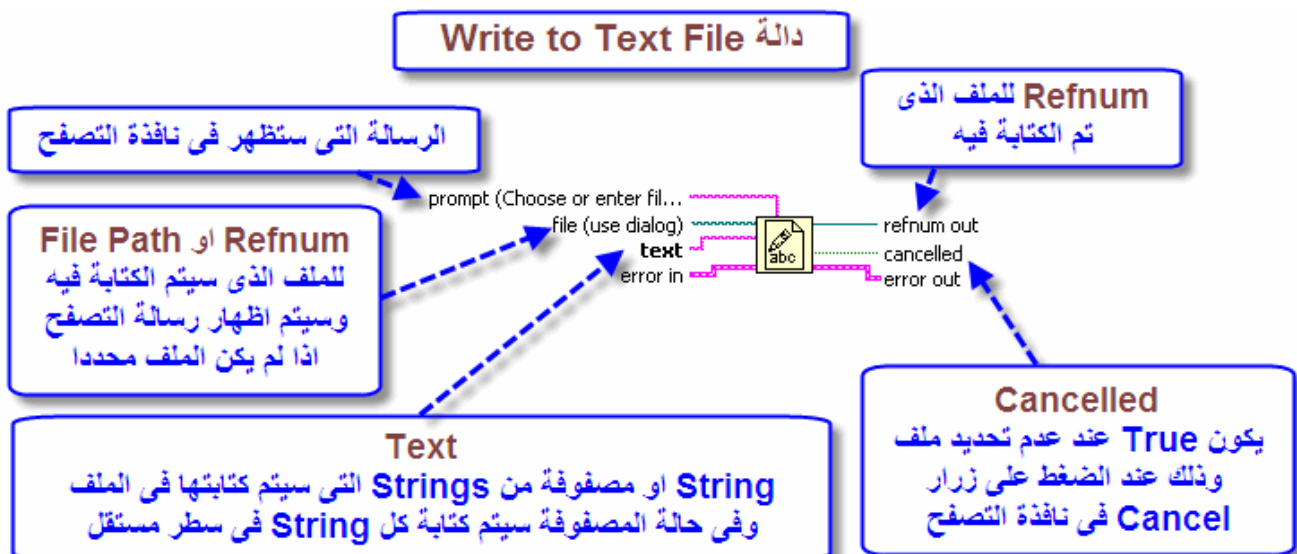


ملحوظات :

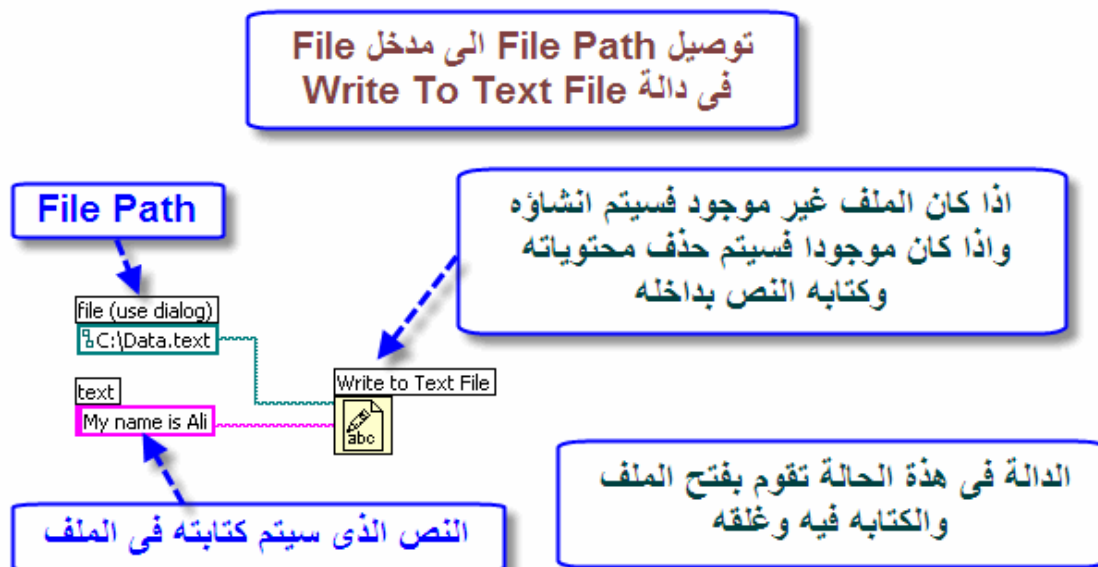
- عند فتح اى ملف يجب ان نقوم بغلقه قبل نهاية البرنامج.
- الدالتين Open/Create/Replace File و Close File ليست خاصيتين لـ Text Files فقط وانما يستخدمان مع جميع انواع الملفات.

الدالة Write to Text File

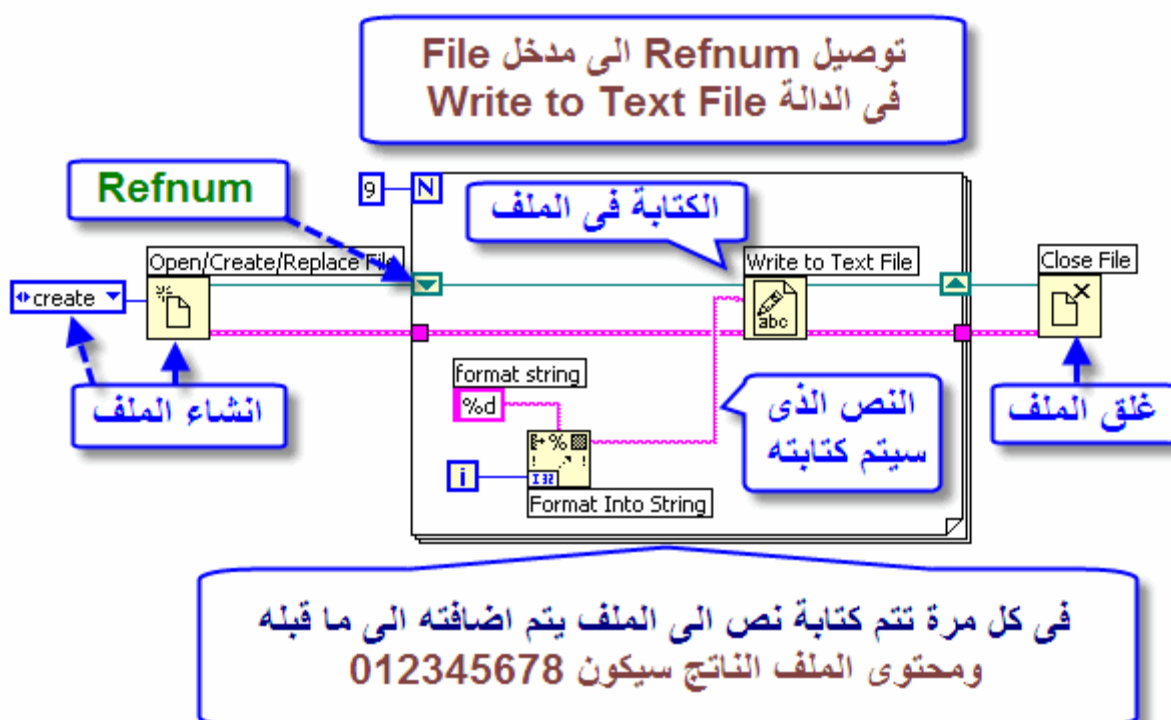
تقوم هذه الدالة بكتابة نص (String) الى ملف او كتابة مصفوفة من النصوص (Array of Strings) الى ملف بحيث سيكون كل نص فى المصفوفة عبارة عن سطر فى الملف.



ويمكن تحديد الملف لهذه الدالة بواسطة توصيل File Path أو Refnum للملف الى المدخل File. وفي حالة توصيل File Path تقوم الدالة بانشاء الملف اذا لم يكن موجودا واذا كان موجودا تقوم بحذف محتوياته وتكتب النص الجديد في الملف ثم تقوم بغلق الملف وبالتالي ليس هناك حاجة لغلق الملف بواسطة الدالة Close File.

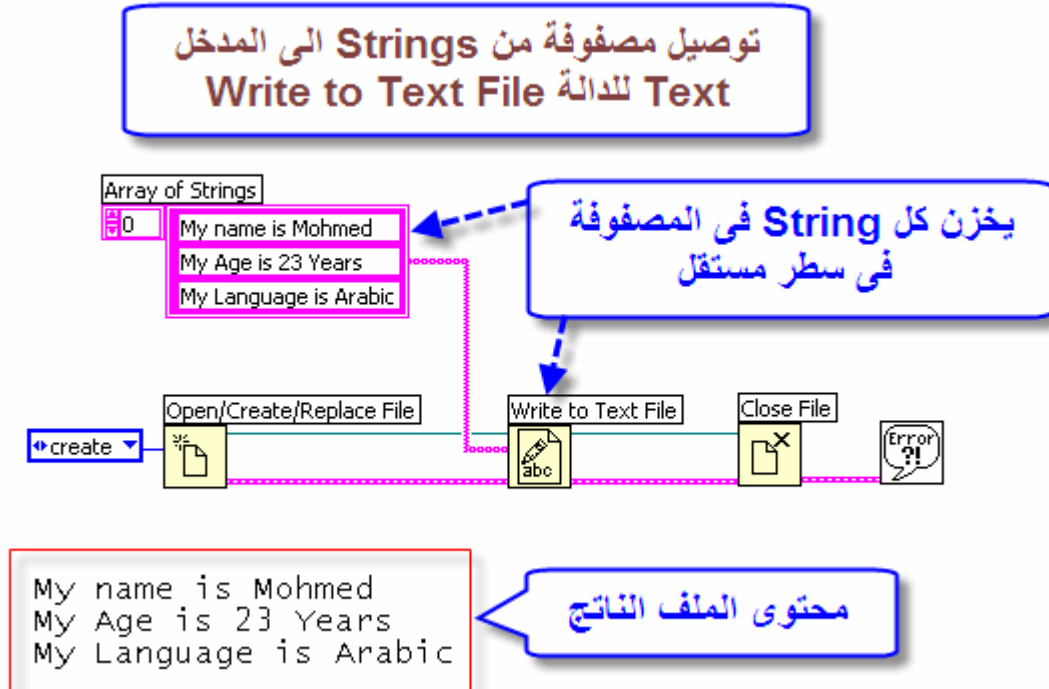


وعند توصيل Refnum فان الدالة تقوم بكتابة النص في الملف بالتتابع بمعنى عند كتابة نص في الملف ثم كتابة نص اخر فانه سوف يلي النص الذي كتبناه من قبله في الملف وهكذا عند كتابه نصوص اخرى. وحقيقا في هذه الحالة تتم الكتابة في المكان الذي يحدده File Marker او File Position وهو الذي سنوضحه بعد الامثلة.

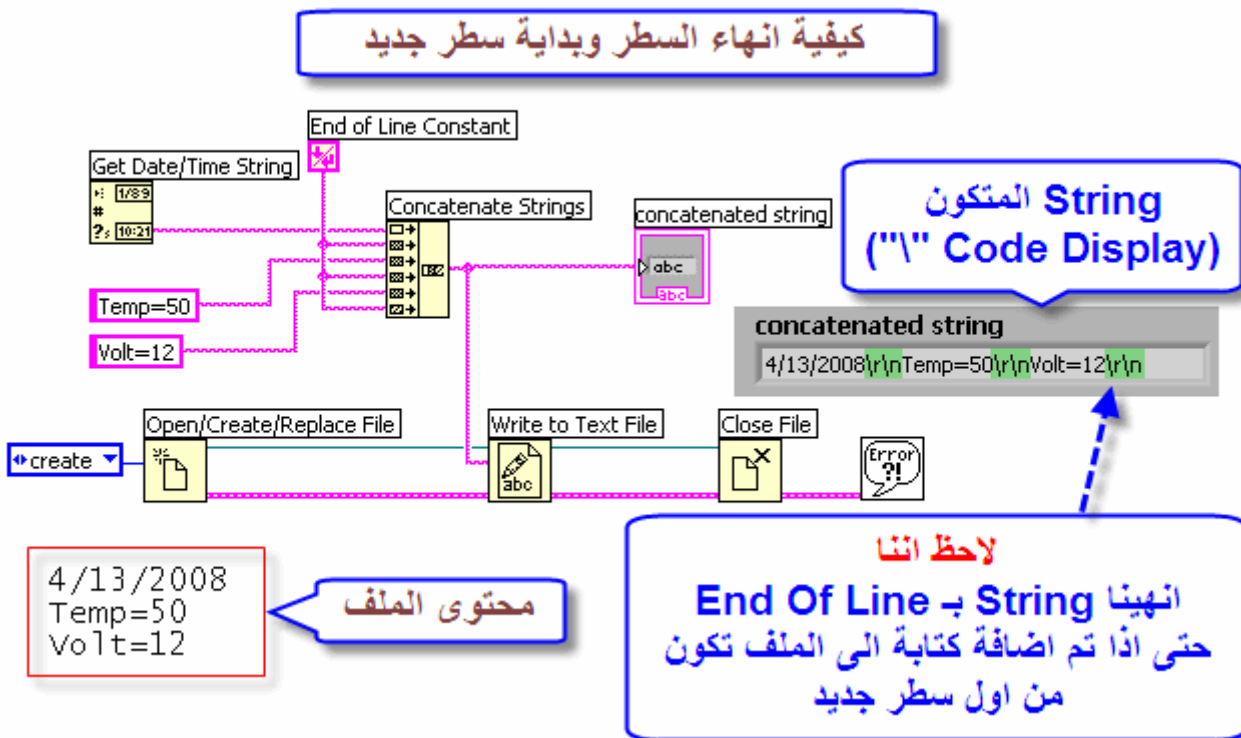


عند توصيل مصفوفة من Strings الى الدالة Write to Text File يتم كتابة كل String في سطر مستقل.

مثال



لانتهاء الكتابة في سطر في الملف و كتابة سطر جديد نستخدم الثابت End of line الذي يتكون من Carriage Return و Line Feed \r\n.



لا حظ انه كما في المثال السابق يمكن انشاء Header للملف قبل تخزين البيانات الرقمية.

File Marker او File Position

File Marker او File Position هو رقم يحدد المكان الذى نقف فيه فى الملف وهو يمثل عدد الحروف من بداية الملف حتى المكان الذى نقف فيه. وهو يحدد مكان الكتابة فى او القراءة من الملف.

وعند بداية فتح الملف تكون قيمته صفر وعند الكتابة فى الملف او القراءة منه يتغير قيمة File Position بمقدار عدد الحروف التى تم كتابتها او قراءتها.

مثال :إذا فتحنا ملف فان File Position سيكون يساوى صفر فعند الكتابة ستتم الكتابة من بداية الملف فاذا كتبنا فى الملف مثلا عدد 100 حرف فان قيمة File Position بعد الكتابة ستساوى 100 وسيكون هذا مكان الكتابة التالية .فاذا كتبنا عدد 50 حرف فان الكتابة ستبدأ من المكان 100 وبعد تمام الكتابة سيكون File Position يساوى 150 وهكذا .

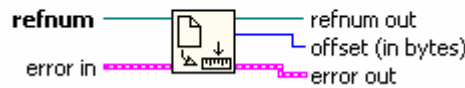
ويحدث نفس الشئ فى حالة القراءة.

وإذا كان File Position يساوى صفر فان الكتابة او القراءة ستكون فى بداية الملف.

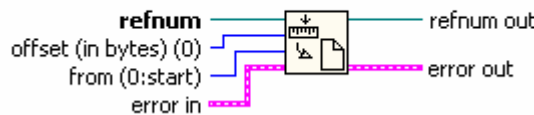
وإذا كان يساوى عدد حروف الملف فان القراءة والكتابة ستكون فى نهاية الملف.

وتوجد دوال لمعرفة File Position وايضا تغيير قيمة File Position وبالتالي المكان الذى ستتم منه القراءة وتتم فيه الكتابة. وايضا دوال لمعرفة حجم الملف وهذه الدوال هى

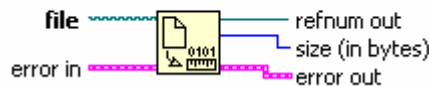
Get File Position



Set File Position

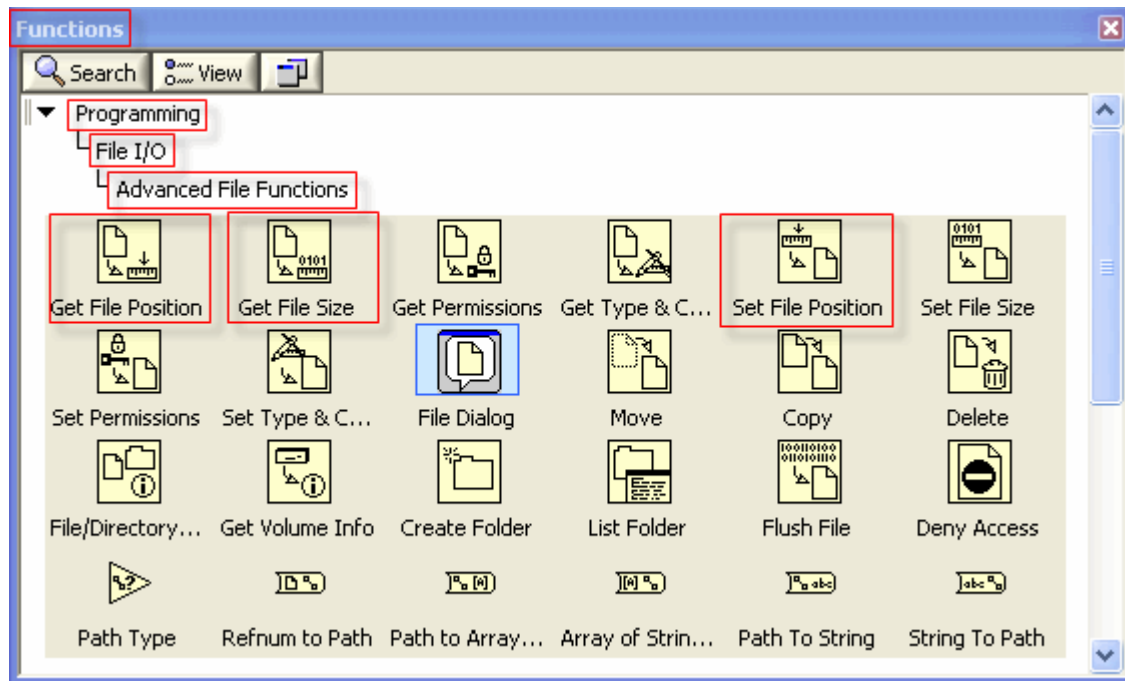


Get File Size

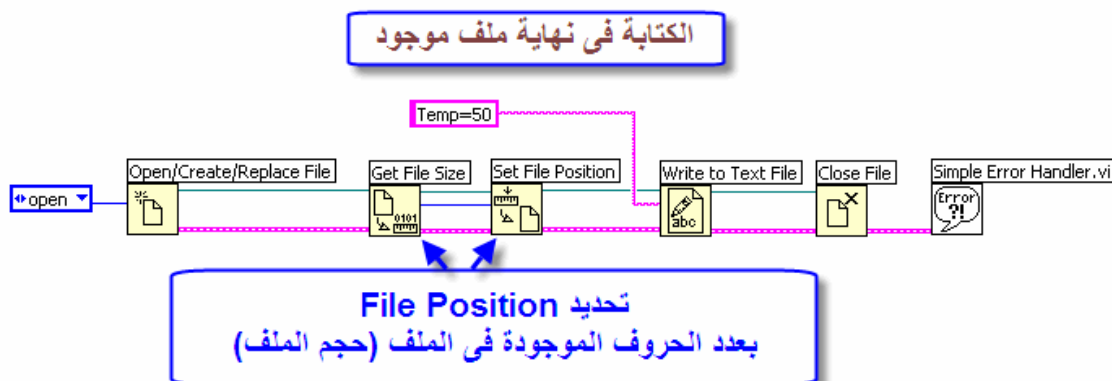


ويمكن ادراج هذه الدوال من

Functions Palette >>Programming>>I/O File>>Advanced File Functions

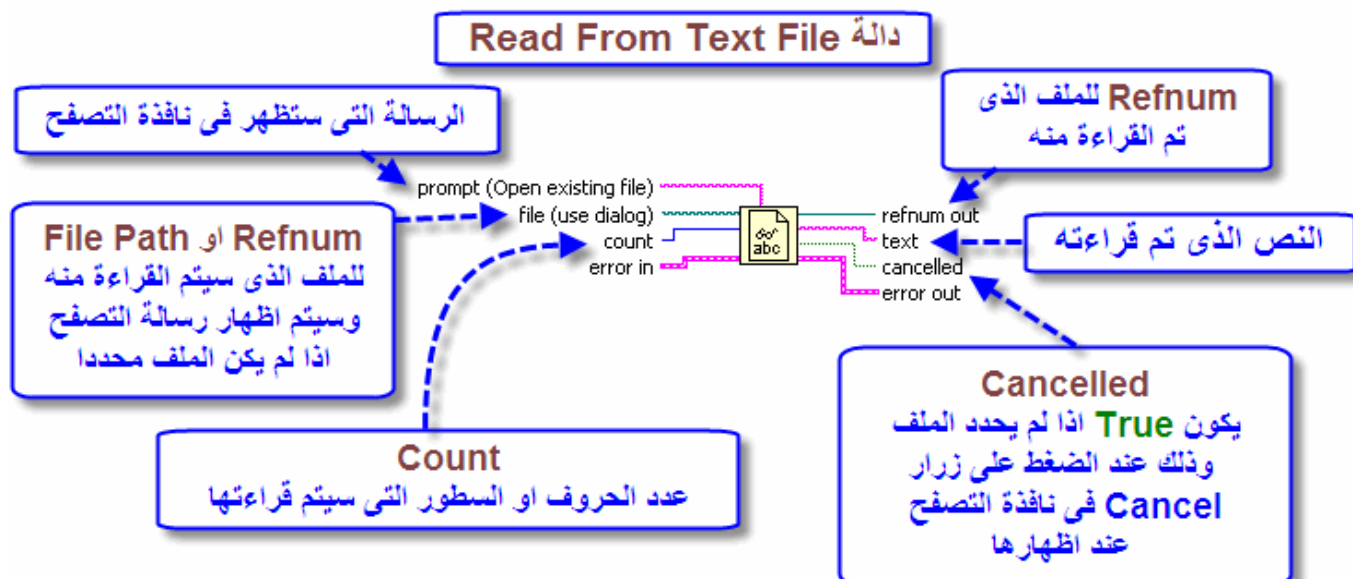


مثال: تحديد الكتابة في نهاية ملف موجود

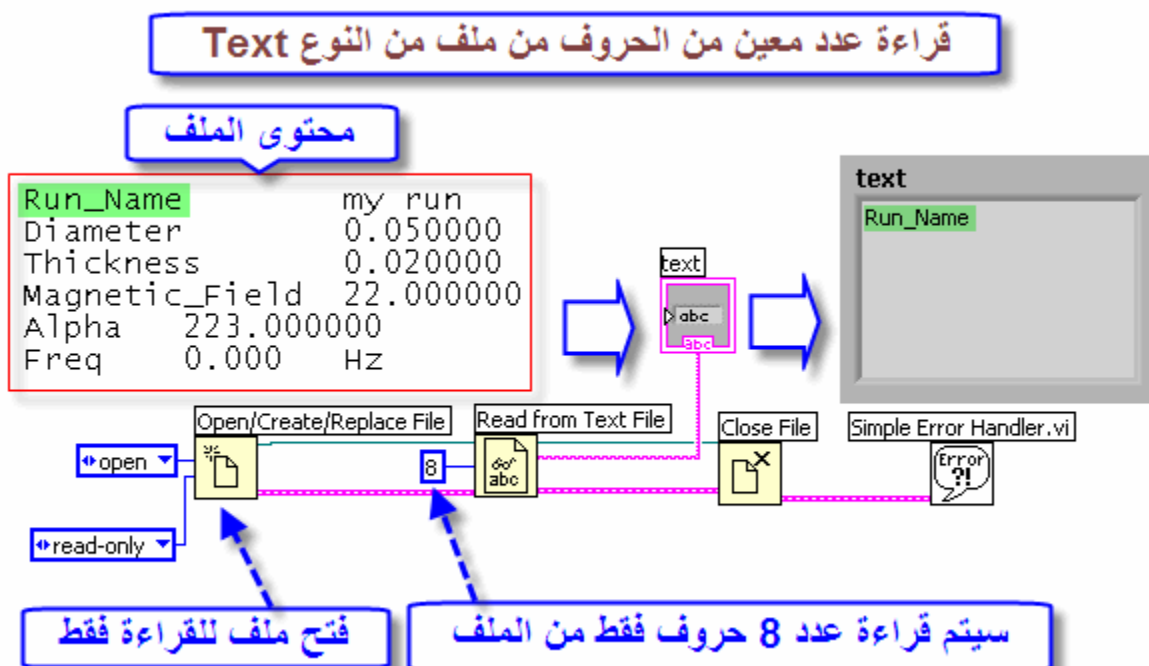
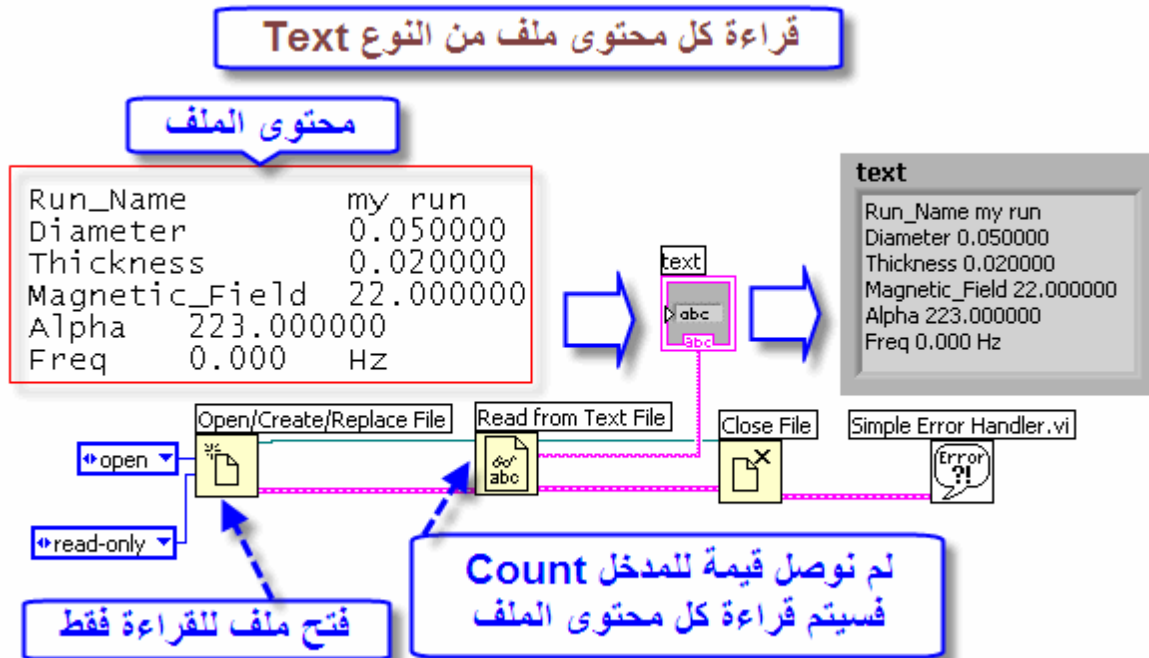


دالة Read from Text File

تقوم هذه الدالة بالقراءة من ملف من النوع Text.

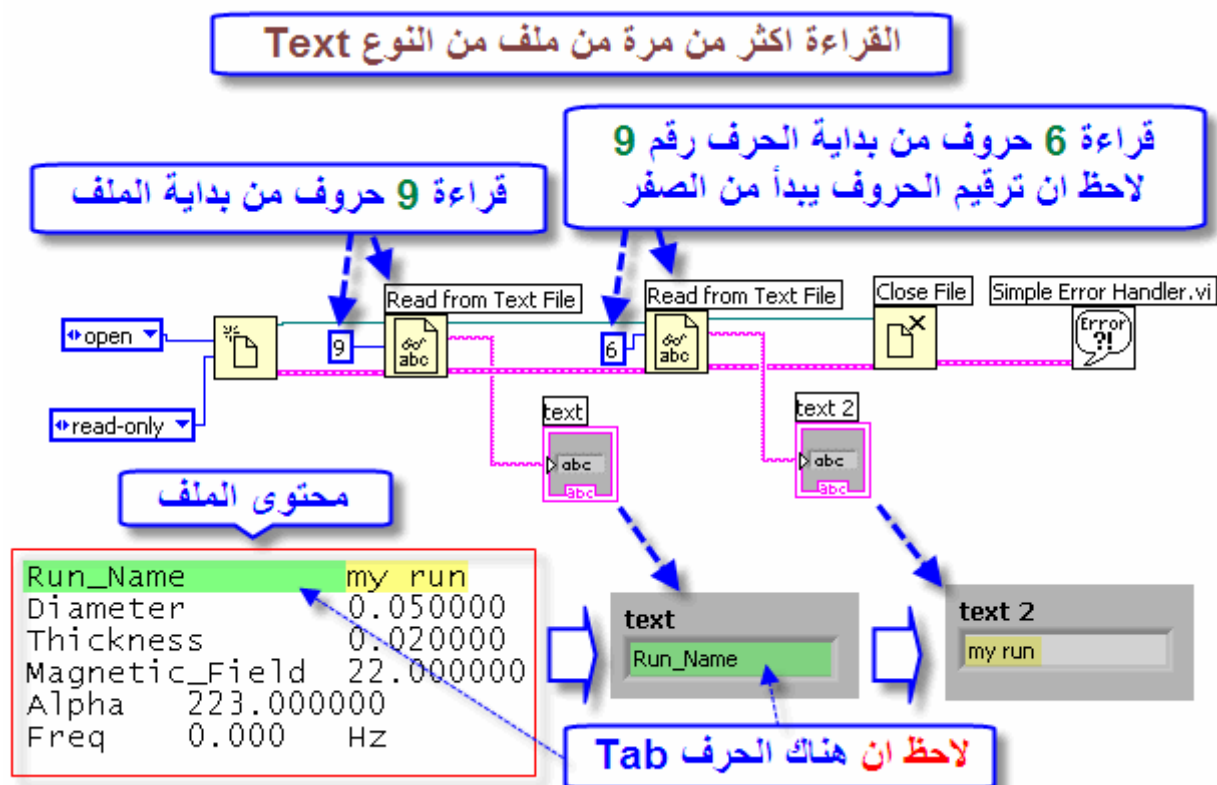


وتقوم هذه الدالة بقراءة عدد معين من الحروف من الملف. وهذا العدد يتم تحديده من خلال المدخل Count وإذا كانت قيمته اصغر من الصفر (-1) فان الدالة تقوم بقراءة الملف كله وهذه الحالة العادية اى انه اذا لم نوصل قيمة الى المدخل Count يتم قراءة الملف كله حتى نهايته.
امثلة:



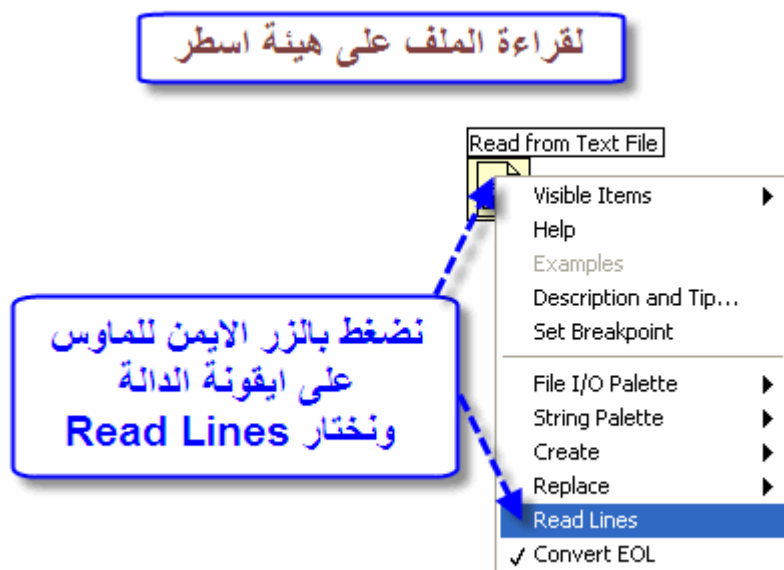
عند فتح ملف يكون قيمة File Position تساوى صفر وعند القراءة تتم القراءة من اول الملف ثم بعد عملية القراءة تكون قيمة File Position هى رقم اخر حرف تم قراءته فعند القراءة مرة اخرى تتم من مكان File Position الجديد وهكذا حتى يتم غلق الملف

مثال :



ملحوظة : عند قراءة الحروف من الملفات يجب الاخذ في الاعتبار المسافات و الحرف Tab وايضا الحروف الغير قابلة للطباعة مثل نهاية الاسطر والبداية من اول السطر Line Feed (n) و Carriage Return (r). ولمشاهدة هذه الحروف يمكن ان نجعل String Indicator في الحالة "Code Display".

يمكن ان تقوم هذه الدالة بقراءة الاسطر من الملف سطر سطر وذلك بالضغط على ايقونة الدالة ونختار Read Lines.



وفى هذه الحالة المدخل Count يحدد عدد الاسطر التى سيتم قراءتها وعند توصيل القيمة (-1) اليه يتم قراءة كل اسطر الملف .

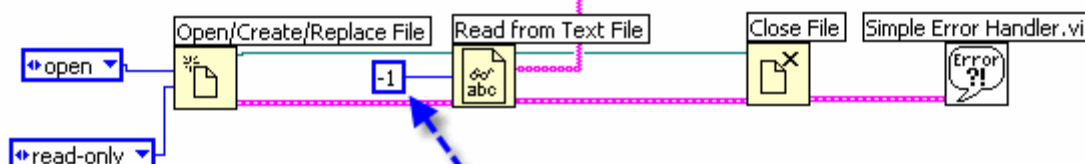
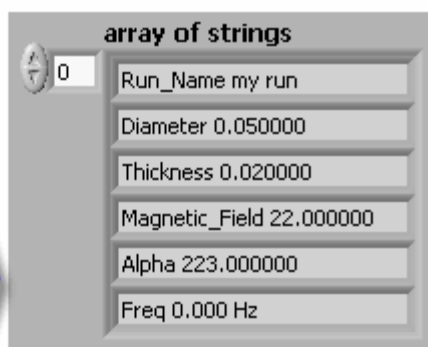
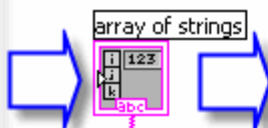
تقوم الدالة باخراج مصفوفة من Strings كل String بها فيه محتوى سطر كامل من الملف.
وعدد Strings الموجودة فى المصفوفة هو عدد الاسطر التى تم قراءتها.

امثلة :

قراءة كل الاسطر الموجودة فى الملف

محتوى الملف

```
Run_Name      my run
Diameter      0.050000
Thickness     0.020000
Magnetic_Field 22.000000
Alpha        223.000000
Freq         0.000 Hz
```

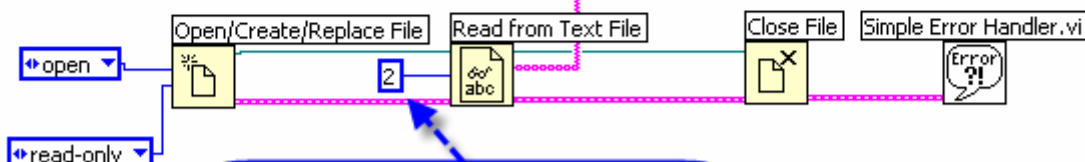
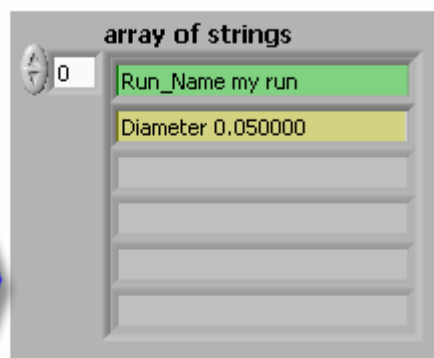
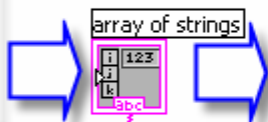


القيمة -1 تعنى قراءة كل اسطر الملف

قراءة عدد معين من الاسطر من الملف

محتوى الملف

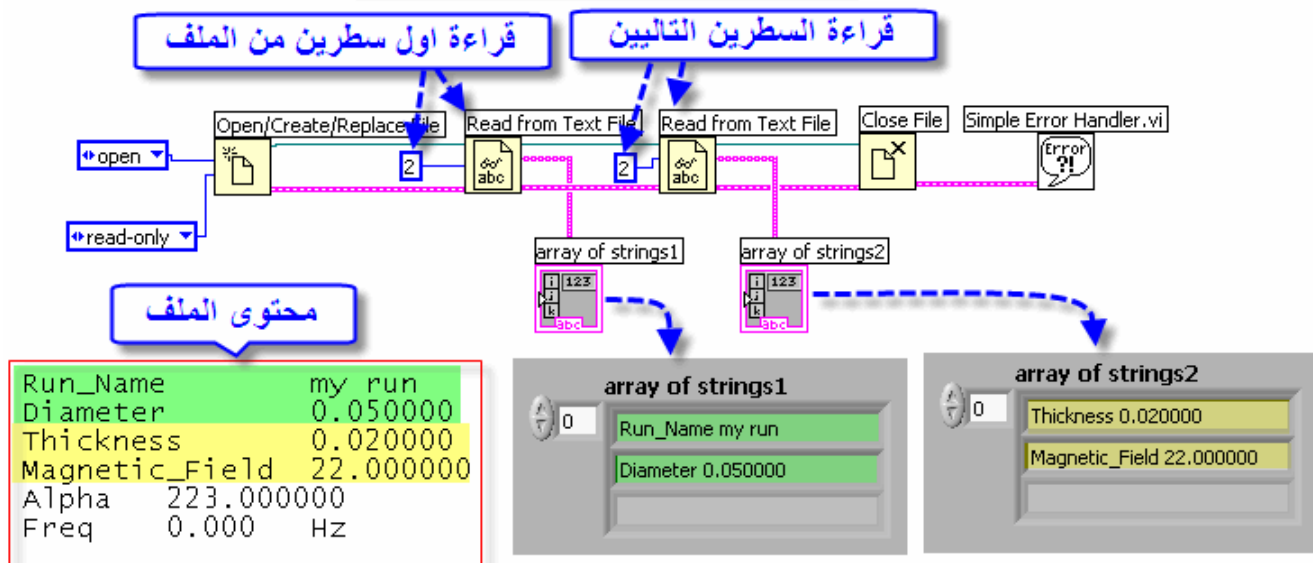
```
Run_Name      my run
Diameter      0.050000
Thickness     0.020000
Magnetic_Field 22.000000
Alpha        223.000000
Freq         0.000 Hz
```



سيتم قراءة سطرين من الملف

فى حالة قراءة الاسطر ايضا يتم قراءة الملف بشكل تتابعى

تتم القراءة من الملفات بشكل متتابع



كيفية التعامل مع الملفات من النوع Binary

يتم تخزين البيانات في الملفات من النوع Binary كما هي دون تحويلها الى ASCII Code لذلك فهي اسرع في التعامل وتأخذ مساحة اصغر وهذه الملفات لايمكن فتحها بالبرامج تحرير النصوص مثل Notepad. ويتم تخزين اى نوع من البيانات في هذه النوعية من الملفات ولكي نقرأ محتوى هذه الملفات يجب ان نعلم نوع البيانات الموجودة في الملف وكيفية تخزينها. لذلك عادة نقوم بعمل برنامج لتخزين الملفات واخر لفتح هذه الملفات او ان يقوم نفس البرنامج بعمل الوظيفتين معا الكتابة والقراءة.

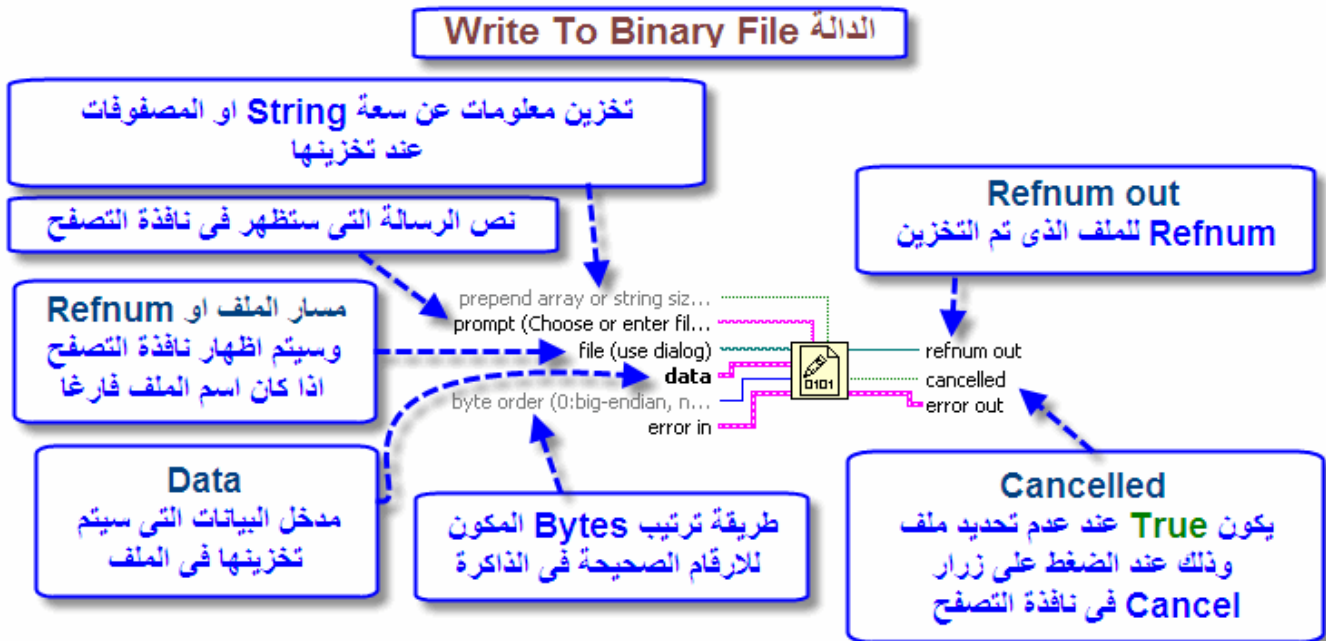
ويمكن استخدام الدوال Open/Create/Replace File لانشاء ملف او فتح ملف

والدالة Close File لغلق الملف مع الملفات من النوع Binary كما استخدمناها مع الملفات من النوع Text.

وتوجد الدالة Write To Binary File للكتابة في الملفات و الدالة Read From Binary File للقراءة من الملف.

الدالة Write to Binary File

تستخدم هذه الدالة لكتابة بيانات في الملفات من النوع Binary. وهذه البيانات يمكن ان توجد في اى صورة (Array ، Cluster ، Waveform ...)



ومداخل هذه الدالة هي :

Prepend Array or string Size? : ونجعل قيمته

True : اذا اردنا ان يرفق LabVIEW معلومات عن حجم String او Array اذا كنا نخزن اى

من النوعين. و True هي قيمة المدخل اذا لم نوصل له اى قيمة.

False: اذا اردنا تخزين البيانات بدون اى معلومات اضافية عن الحجم.

Prompt: مدخل الرسالة التي ستظهر للمستخدم فى حالة اظهار نافذة التصفح للمستخدم لتحديد الملف المراد

التخزين فيه. نص الرسالة سيكون "Choose or enter file path" اذا لم يتم التوصيل لهذا

المدخل.

File: يتم توصيل الى هذا المدخل مسار او Refnum للملف المراد التخزين فيه. وفى حالة عدم التوصيل لهذا

المدخل يتم اظهار نافذة التصفح للمستخدم عند تنفيذ الدالة.

وفى حالة تم توصيل مسار للملف لمدخل File او اظهار نافذة التصفح فان الدالة تقوم بانشاء الملف اذا لم يكن

موجودا او تقوم بحذف محتويات الملف وتخزين البيانات الجديدة اذا كان موجودا

اما فى حالة توصيل Refnum فان الدالة تقوم بالكتابة فى الملف فى المكان الذى يحدده File Position

والذى تكون قيمته فى بداية فتح الملف بصفر (اى بداية الملف)

ولتغيير مكان File Position نستخدم الدوال Get File Position و Set File Position.

وللتخزين فى نهاية الملف نجعل File Position يساوى سعة الملف. وللحصول على سعة الملف نستخدم الدالة

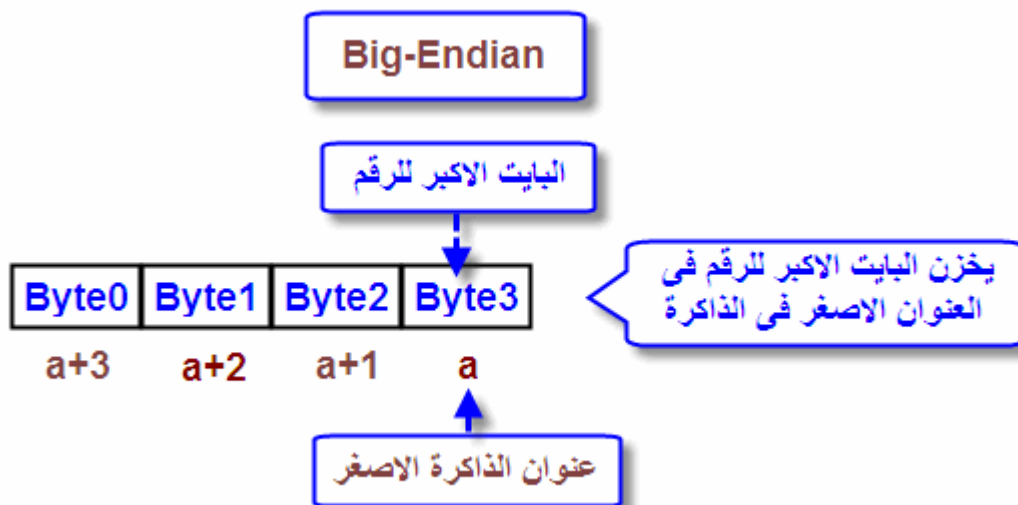
Get File Size كما رأينا فى حالة التعامل مع الملفات من النوع Text.

Data: ويتم توصيل البيانات التي نريد تخزينها الى هذا المدخل وهذه البيانات يمكن ان تكون في اى صورة (مصفوفة ، Cluster ، String ...)

Byte Order: رقم يحدد طريقة تمثيل الارقام الصحيحة في الذاكرة فكما نعلم ان الارقام الصحيحة يمكن ان تتكون من اكثر من Byte مثل 4Bytes (32 Bit).

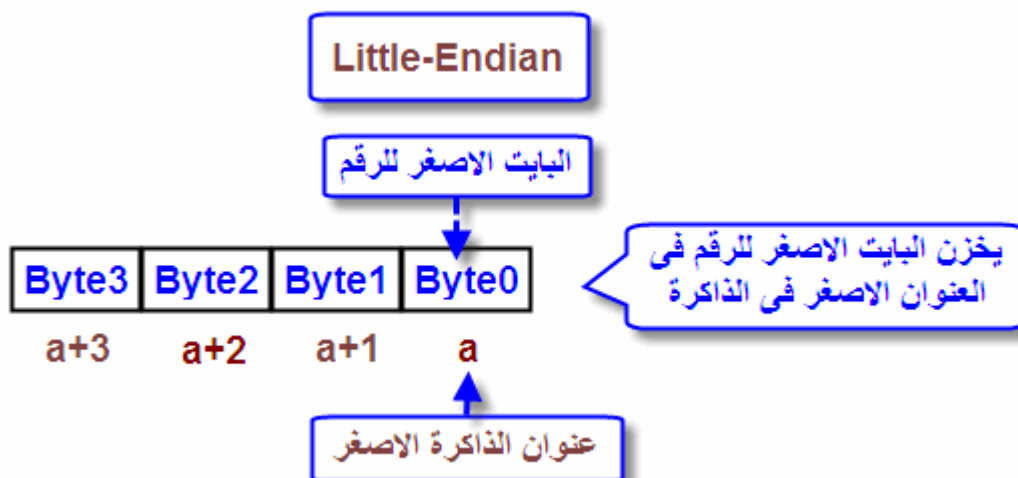
و **Byte Order** يحدد ترتيب تخزين Bytes المكون للرقم في الذاكرة.

(0) Big-Endian: هو يعنى ان البايت الاكبر للرقم يخزن في العنوان الاصغر في الذاكرة ويستخدم هذا الترتيب في انظمة التشغيل مثل Mac. وهذه هي قيمة المدخل اذا لم يتم التوصيل اليه.



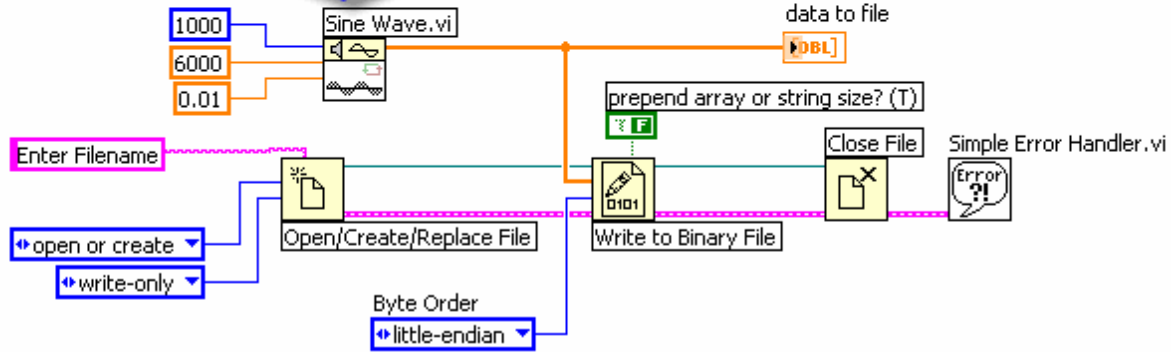
(1) Host Order ، Native: وهو يعنى اتباع الترتيب المتبع في النظام التشغيل الموجود .

(2) Little-Endian: و يعنى ان البايت الاصغر للرقم يخزن في العنوان الاصغر في الذاكرة ويستخدم هذه الترتيب في انظمة التشغيل مثل Windows و Linux.



مثال على التخزين في ملف Binary

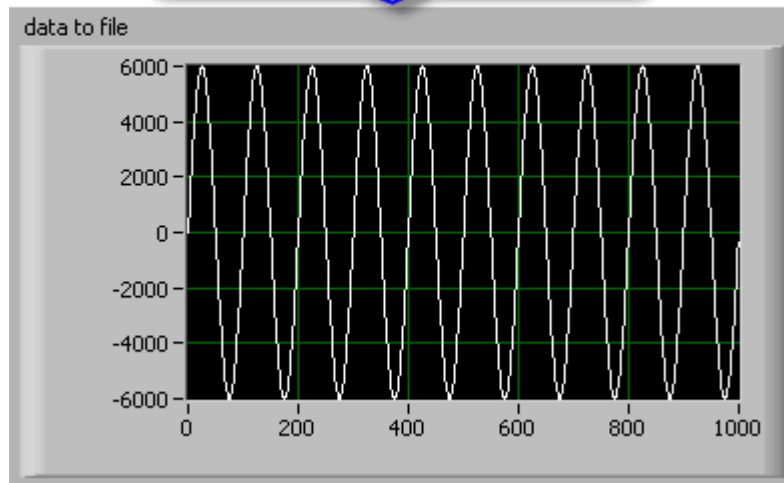
تخرج هذه الدالة قيم Sin Wave في مصفوفة
عدد هذه القيم = 1000
وستتم تخزين هذه المصفوفة في ملف Binary



ملحوظة: يتم ادراج Sin Wave.vi من

Functions Palette>>Programming>>Signal Generation

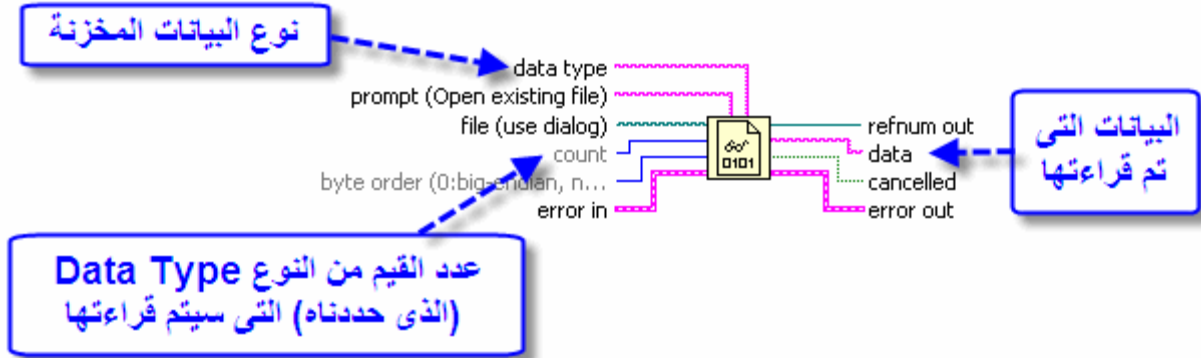
شكل Sin Wave التي تم تخزينها



الدالة Read from Binary File

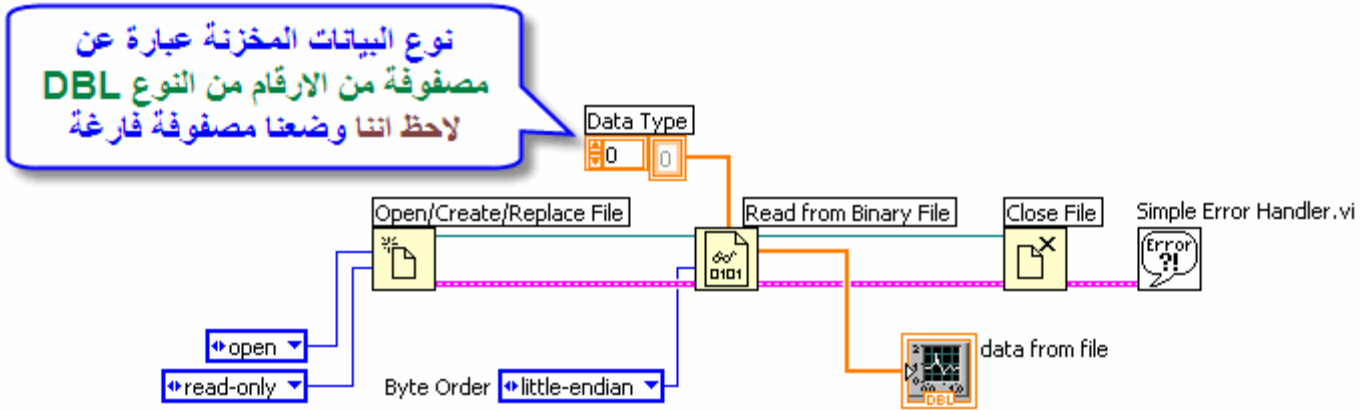
تقوم هذه الدالة بقراءة البيانات من Binary File . لاستخدام هذه الدالة لابد من تحديد نوع البيانات المخزن في الملف للدالة حتى تتم عملية القراءة بالشكل الصحيح. كما يجب ان تكون ترتيب تخزين Bytes (Byte Order) هو نفس ترتيب القراءة.

دالة Read From Binary



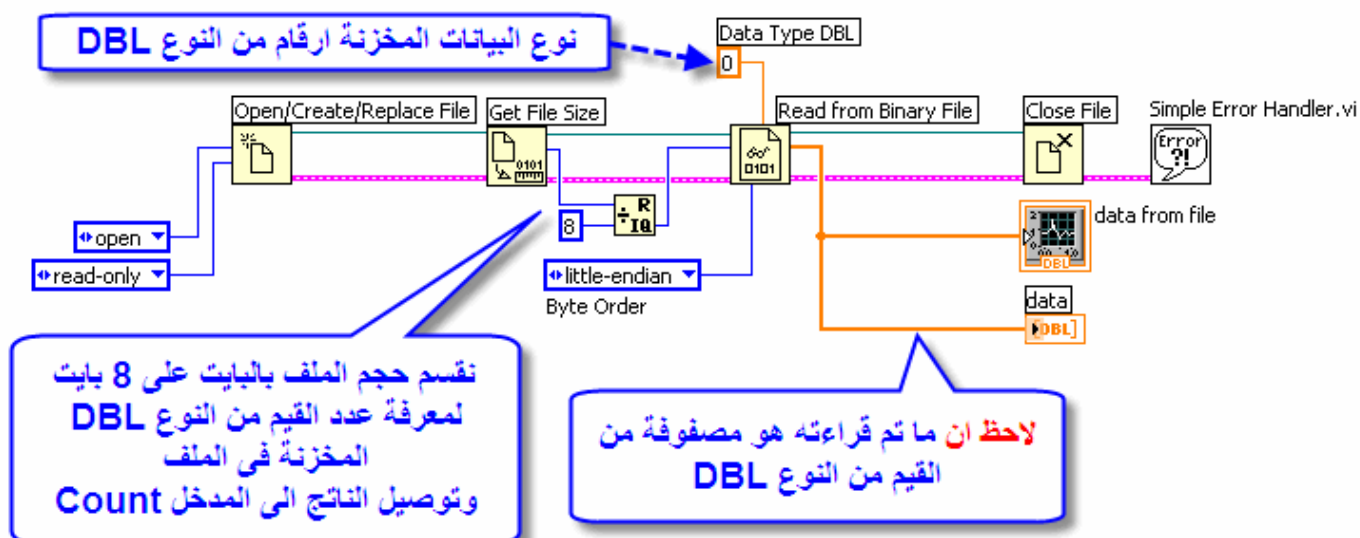
والمدخل **Data type** : يحدد نوع البيانات التي سيتم قراءتها ولا بد ان تتوافق مع نوع البيانات المخزنة.
و المدخل **Count** : يحدد عدد القيم من نوع البيانات الذي حددناه واذا لم يوصل هذه المدخل يتم قراءة قيمة واحدة من نوع البيانات.

قراءة الملف الذي تم تخزينه بواسطة الدالة Write To Binary File



كما يمكن قراءة الملف بطريقة اخرى حيث اننا نعلم ان القيم المخزنة هي ارقام من النوع DBL وان هذه النوع من البيانات يشغل سعة تخزينية تساوى 8 بايت .
بالتالى بمعرفة حجم الملف بالبايت سيكون عدد الارقام المخزنة تساوى خارج قسمة حجم الملف على 8.
ويمكن معرفة حجم الملف باستخدام الدالة Get File size الموجودة فى

Functions Palette >>Programming>>I/O File>>Advanced File Functions



تم بحمد الله الانتهاء من الدرس
الى اللقاء باذن الله مع الدرس القادم

photonlv@gmail.com