

Crystal Reports

Using VB Variables with the Seagate Crystal VBX/OCX SelectionFormula Property

Overview

This document describes how to use variables when passing a selection formula to a Seagate Crystal Report at runtime with the VBX/OCX control. Several methods are covered including string, numeric, and date data types.

Contents

INTRODUCTION	2
IMPORTANT NOTES	2
EXAMPLES OVERVIEW	3
PASSING A VARIABLE FROM VB THAT RESULTS IN A STRING DATA TYPE IN SEAGATE CRYSTAL REPORTS RECORD SELECTION	4
PASSING A VARIABLE FROM VB THAT RESULTS IN A NUMERIC DATA TYPE IN SEAGATE CRYSTAL REPORTS SELECTION FORMULA	6
PASSING A VARIABLE FROM VB THAT RESULTS IN A DATE DATA TYPE IN SEAGATE CRYSTAL REPORTS RECORD SELECTION FORMULA	9
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	13

Introduction

VB variables can be used with the SelectionFormula property if they are declared As String or converted to string (regardless of the actual data type) before being passed to Seagate Crystal Reports. The value of the variable passed to the report must be in the same format and look exactly as if entered directly into the Seagate Crystal Reports Record Selection Formula Editor

NOTE	The following code also works with the OCX but the Control name defaults to CrystalReport1 instead of Report1 (as it does with the VBX).
-------------	--

String data types must be surrounded by double or single quotes:

`"ABCD"` or

`'ABCD'`

Numbers don't need quotes: `1234`

Dates must look like a Date() function: `Date (YYYY, MM, DD)`

Important Notes

When a selection formula is passed to a report it is appended with a Boolean "AND" to any characters in the Seagate Crystal Reports Record Selection Formula Editor. Any characters include non-printing characters such as spaces and carriage returns, comment lines and existing selection formulas.

When a selection formula is removed from a report by selecting the formula and pressing the spacebar or carriage return, a space character or carriage return remains in the selection formula editor. Whenever a space character or carriage return is left in selection formula editor, passing a selection formula using the SelectionFormula property will append to the space or carriage return as follows:

```
AND {company.LASTYRSALE} > 10000
```

resulting in an "Error in formula." or "20515" message when the application is run.

When a comment line is left in the selection formula editor, passing a selection formula using the SelectionFormula property will append to the comment as follows:

```
// Comment AND {company.LASTYRSALE } > 10000
```

resulting in all records appearing in the report when the application is run.

When an existing formula is left in the selection formula editor, passing a selection formula using the SelectionFormula property will result in the passed formula being appended to the existing formula as follows:

```
{company.STATE } = "WA" AND {company.STATE } = "CA"
```

resulting in a report with no records selected (because the State cannot be both CA and WA at the same time) when the application is run.

Examples overview

The following examples assume

A Seagate Crystal report has been setup with the following three fields, no selection formula, and no spaces or carriage returns in the selection formula editor:

```
{datafile.STRINGFIELD}
```

```
{datafile.NUMFIELD}
```

```
{datafile.DATEFIELD}
```

The variable names used in VB are:

CHARSTR

NUMSTART

NUMEND

DATESTART

DATEEND

YEARDATE

MONTHDATE

DAYDATE

Four TextBoxes have been placed on a VB form:

`StringEntry.Text` - for the string examples

`NumEntryStart.Text` - the start range for the number examples

`NumEntryEnd.Text` - the end range for the number examples

`DateStart.Text` - for the whole date examples

Three additional TextBoxes have been placed on the form as needed for the date examples:

`YearEntry.Text` - for the year

`MonthEntry.Text` - for the month

`DayEntry.Text` - for the day

About syntax:

Ampersand & versus Plus + in the SelectionFormula statement: The Plus + works fine if Variables are declared As String in VB. However if Variables are declared As Integer, using the Plus + will result in a type mismatch error in VB. Operators and Seagate Crystal field names are surrounded by quotes and therefore treated as strings. Strings cannot be concatenated to integers. The Ampersand & actually forces a concatenation by converting an integer to a Variant of VarType 8 which is a string.

The Ampersand & is used in most of the examples in this document as it works for both string and integer variable declarations.

Single quotes are passed to Seagate Crystal Reports String formula fields although double quotes can be used.

Passing a variable from VB that results in a String data type in Seagate Crystal Reports Record Selection

As both VB and Seagate Crystal Reports require string values to be surrounded by quotes, a variable that results in a string data type in Seagate Crystal Reports requires two sets of quotes; one set for VB and one set for Seagate Crystal.

Comparison Operators only need one set of quotes as they are a string value only in VB, not in Seagate Crystal Reports. There are a number of ways to include the extra set of quotes needed by Seagate Crystal Reports. The first two examples show how to pass the variable with the extra quotes when the value of the variable is hard-coded. The third example shows how to pass the variable with the extra quotes when the value of the variable is entered via a TextBox at runtime.

Example 1 has the value hard-coded and includes the extra quotes while assigning the variable a value:

```
Dim CHARSTR As String
'Declare CHARSTR as a string variable.
CHARSTR = "'ABCD'"
'Assign a value to the variable. 'Note the use of two sets
of 'quotes.
Report1.SelectionFormula = "{datafile.STRINGFIELD} = " &
CHARSTR
'Concatenate the Seagate Crystal Reports database
```

```

`field name with the operator and
`the VB variable name.
MsgBox Report1.SelectionFormula
`Display the exact value that is passed to
`Seagate Crystal Reports in a message box.
`This message box will display:
`{datafile.STRINGFIELD} = 'ABCD'
`The Seagate Crystal report includes only records
`where the value in STRINGFIELD is ABCD.

```

Example 2 has the value hard-coded and includes the extra quotes in the SelectionFormula statement:

```

Dim CHARSTR As String
`Declare CHARSTR as a string variable.
CHARSTR = "ABCD"
`Assign a value to the variable. Note the use of one set of
`quotes.
Report1.SelectionFormula = "{datafile.STRINGFIELD} = '" &
_ CHARSTR & "'"
`Concatenate the Seagate Crystal Reports field with
`the operator, an open quote,
`the VB variable name, and a closed quote.
MsgBox Report1.SelectionFormula
`Display the exact value that is passed to
`Seagate Crystal Reports in a message box.
`This message box will display:
`{datafile.STRINGFIELD} = 'ABCD'
`The Seagate Crystal report includes only records
`where the value in STRINGFIELD is ABCD.

```

Example 3 shows how to pass a variable when the value is entered via a TextBox at runtime:

```

Dim CHARSTR As String
`Declare CHARSTR as a string variable.
CHARSTR = StringEntry.Text
`Assign the contents of the StringEntry
`TextBox to `the CHARSTR variable.

```

```

Report1.SelectionFormula = "{datafile.STRINGFIELD} = '" &
_ CHARSTR & "'"
`Concatenate the Seagate Crystal Reports database
`field with the operator, an open quote,
`the VB variable name, and a closed quote.
MsgBox Report1.SelectionFormula
`Display the exact value that is passed to
`Seagate Crystal Reports in a message box.
`This message box will display the Seagate Crystal
`field name, an equal sign, and whatever
`was entered in the StringEntry TextBox at
`runtime surrounded by single quotes.
`The Seagate Crystal report only includes records
`where the value in STRINGFIELD the same as
`the text entered in StringEntry.

```

Passing a variable from VB that results in a Numeric data type in Seagate Crystal Reports Selection Formula

Since the SelectionFormula property passes only string data types, variables declared As Integer must be converted to a string before being passed to Seagate Crystal Reports.

The first example shows a number in a variable declared As String. The second example shows a variable declared As Integer converted to a string simply by using the Ampersand & in the SelectionFormula statement. The third example converts an integer to a string using the str\$() and a Plus + in the SelectionFormula statement. The fourth example shows how to convert and pass two variables as a range when the value of the variables are entered via TextBoxes at runtime.

Example 1 has the value hard-coded and the variable declared As String:

```

Dim NUMSTART As String
`Declare NUMSTART as a string variable.
NUMSTART = "1234"
`Assign a value to the variable. Note the use of one set of
`quotes.
Report1.SelectionFormula = "{datafile.NUMFIELD}=" &
NUMSTART
`Concatenate the Seagate Crystal Reports database
`field with the VB variable name.
`Plus + could be used instead of Ampersand &
`as NUMSTART is as a string

```

```
`and needs no conversion
```

```
MsgBox Report1.SelectionFormula
`Display the exact value that is passed to
`Seagate Crystal Reports in a message box.
`This message box will display:
`{datafile.NUMFIELD}= 1234
`The Seagate Crystal report includes only records
`where the value in NUMFIELD is 1234.
```

Example 2 has the value hard-coded and the variable declared As Integer:

```
Dim NUMSTART As Integer
`Declare NUMSTART as an integer variable.
NUMSTART = 1234
`Assign a value to the variable.
Report1.SelectionFormula= "{datafile.NUMFIELD}=" &
NUMSTART
`Concatenate the Seagate Crystal Reports database
`field with the VB variable name.
`The Ampersand & converts NUMSTART to a
`string.
MsgBox Report1.SelectionFormula
`Display the exact value that is passed to
`Seagate Crystal Reports in a message box.
`This message box will display:
`{datafile.NUMFIELD}= 1234
`The Seagate Crystal report includes only records
`where the value in NUMFIELD is 1234.
```

Example 3 has the value hard-coded and converts the variable with str\$() in the SelectionFormula statement:

```
Dim NUMSTART As Integer
`Declare NUMSTART as a integer variable.
NUMSTART = 1234
`Assign a value to the variable.
Report1.SelectionFormula = "{datafile.NUMFIELD}=" + _
Str$( NUMSTART)
`Convert variable and concatenate the
`Seagate Crystal Reports database field
```

```

`with the VB variable name. Plus + is
`used because the Str$() function
`converts NUMSTART to string
MsgBox Report1.SelectionFormula
`Display the exact value that is passed to
`Seagate Crystal Reports in a message box.
`This message box will display:
`{datafile.NUMFIELD}= 1234
`The Seagate Crystal report includes only records
`where the value in NUMFIELD is 1234.

```

Example 4 shows how to pass variables as a range when the values are entered via TextBoxes at runtime:

```

Dim NUMSTART As Integer
`Declare NUMSTART as an integer variable.
Dim NUMEND As Integer
`Declare NUMEND as an integer variable.
NUMSTART = NumEntryStart.Text
`Assign the contents of the NumEntryStart
`TextBox to the NUMSTART variable.
NUMEND = NumEntryEnd.Text
`Assign the contents of the NumEntryEnd
`TextBox to the NUMEND variable.
Report1.SelectionFormula = "{datafile.NUMFIELD} >= " & _
NUMSTART & "AND {datafile.NUMFIELD}<=" & NUMEND
`Concatenate the Seagate Crystal Reports database
`field with the Comparison Operators and
`the VB variables.
MsgBox Report1.SelectionFormula
`Display the exact value that is passed to
`Seagate Crystal Reports in a message box.
`This message box will display:
`{datafile.NUMFIELD} >= whatever was
`entered in the NumEntryStart TextBox at
`runtime, the word AND,
`{datafile.NUMFIELD} <= whatever was
`entered in the NumEntryEnd TextBox at
`runtime.
`The Seagate Crystal report only includes records

```



```
'where the value in NUMFIELD is within the
'range entered in the NumEntryStart and
'NumEntryEnd TextBoxes.
```

Passing a variable from VB that results in a Date data type in Seagate Crystal Reports Record Selection Formula

Seagate Crystal Reports accepts dates only as a string data type in the Date(yyyy,mm,dd) format. The characters that make up Date(yyyy,mm,dd) must be concatenated with the Seagate Crystal Reports database field name and the date variable.

The first example shows a range of dates hard-coded and assigned to two variables as strings. The second example shows the date hard-coded and assigned to three variables as strings. The third example shows how to format when the date is entered via three TextBoxes at runtime. The fourth example uses the VB CDate() and Format() functions to convert and format a hard-coded short or long date. The fifth example uses the VB CDate() and Format() functions to convert and format a short or long date entered via a TextBox at runtime.

Example 1 shows a date range hard-coded and assigned to two variables as strings. Formatting is done in the SelectionFormula statement:

```
Dim DATESTART
'Declare DATESTART as a variable
Dim DATEEND
'Declare DATEEND as a variable
DATESTART = "1995, 01, 31"
'Assign a value to the variable. Note that
'the year is 4 digits, the month and day
'are 2 digits. The year, month and day are
'in the order Seagate Crystal expects and are
'delimited by commas, and the entire string
'is surrounded by quotes.
DATEEND = "1995, 06, 30"
'Assign a value to the variable. Note that
'the year is 4 digits, the month and day
'are 2 digits. The year, month and day are
'in the order Seagate Crystal expects and are
'delimited by commas, and the entire string
'is surrounded by quotes.
Report1.SelectionFormula = "{datafile.DATEFIELD} IN _
```

```
Date(" & DATESTART & ") TO Date(" & DATEEND & ")"
```

`Concatenate the Seagate Crystal Reports database
 `field with the words IN Date, an open
 `parenthesis, the first VB variable name,
 `and a closed parenthesis and the words
 `TO Date, an open parenthesis, the second
 `VB variable name and a closed parenthesis.
 MsgBox Report1.SelectionFormula
 `Display the exact value that is passed to
 `Seagate Crystal Reports in a message box.
 `This message box will display:
 `{datafile.DATEFIELD} IN `Date(1995,01,31)
 ` TO Date(1995, 06, 30)
 `The Seagate Crystal report only includes records
 `where the value in DATEFIELD
 `is between Jan. 31, 95 and June 6, 95.

Example 2 shows the date hard-coded and assigned to three variables. Formatting is done in the SelectionFormula statement:

```
Dim YEARDATE, MONTHDATE, DAYDATE
```

`Declare YEARDATE, MONTHDATE and DAYDATE
 `as variables.
 YEARDATE = "1995"
 `Assign a value to the YEARDATE variable.
 `Note that the year is 4 digits.
 MONTHDATE = "01"
 `Assign a value to the MONTHDATE variable.
 `Note that the month is 2 digits.
 DAYDATE = "31"
 `Assign a value to the DAYDATE variable.
 `Note that the day is 2 digits.
 Report1.SelectionFormula = "{datafile.DATEFIELD}= _
 Date(" & YEARDATE & "," & MONTHDATE & "," & DAYDATE & ")"
 `Concatenate the Seagate Crystal Reports database
 `field with the word Date, an open
 `parenthesis, the YEARDATE variable,
 `a comma, the MONTHDATE variable, a comma,
 `the DAYDATE variable, a comma, and a
 `closed parenthesis.

```

MsgBox Report1.SelectionFormula
'Display the exact value that is passed to
'Seagate Crystal Reports in a message box.
'This message box will display:
'{datafile.DATEFIELD}= Date(1995,01,31)
'The Seagate Crystal report includes only records
'where the value in DATEFIELD is Jan 31, 95.

```

Example 3 shows the date entered via three TextBoxes at runtime. The formatting is included while assigning the variable its value:

```

Dim YEARDATE, MONTHDATE, DAYDATE, DATESTART
'Declare YEARDATE, MONTHDATE, DAYDATE, and
'DATESTART as variables.
YEARDATE = YearEntry.Text
'Assign the YearEntry TextBox value to the YEARDATE
variable.
MONTHDATE = MonthEntry.Text
'Assign the MonthEntry TextBox value to the MONTHDATE
variable.
DAYDATE = DayEntry.Text
'Assign the DayEntry TextBox value to the DAYDATE variable.
DATESTART = "Date(" & YEARDATE & "," & MONTHDATE & "," & _
DAYDATE & ")"
'Concatenate the word Date, an open
'parenthesis, the YEARDATE variable,
'a comma, the MONTHDATE variable, a comma,
'the DAYDATE variable, a comma, and a
'closed parenthesis and assign the value
'to the DATESTART variable.
Report1.SelectionFormula = "{datafile.DATEFIELD}=" &
DATESTART
'Concatenate the Seagate Crystal database field name
'with the DATESTART variable.
MsgBox Report1.SelectionFormula
'Display the exact value that is passed to
'Seagate Crystal Reports in a message box.
'This message box will display the Seagate Crystal
'database field ,an equal sign, the date
'formatting and whatever was entered in
'the YearEntry, MonthEntry, and DayEntry
'TextBoxes at runtime.

```

'The Seagate Crystal report includes only records
'where the value in DATEFIELD is Jan 31, 95.

Example 4 uses the VB CDate() and Format() functions to convert and format a hard-coded short or long date. The formatting is included in the SelectionFormula statement:

```
Dim DATESTART
'Declare DATESTART as a variable.
DATESTART = CDate("January 31, 1995")
'Use the VB CDate() function to convert a
'long or short date (no day of week)
'and assign it to the DATESTART variable.
'Note that the date is quoted.

Report1.SelectionFormula = "{datafile.DATEFIELD}= " _
& Format(DATESTART, " \D\a\t\e\(\yyyy\,mm\,dd\)")
'Concatenate the Seagate Crystal database field with
'the DATESTART variable and use the
'VB Format() function to insert Date(, ,)
'as needed by Seagate Crystal.
MsgBox Report1.SelectionFormula
'Display the exact value that is passed to
'Seagate Crystal Reports in a message box.
'This message box will display:
'{datafile.DATEFIELD}= Date(1995,01,31)
'The Seagate Crystal report includes only records
'where the value in DATEFIELD is Jan 31, 95.
```

Example 5 uses the VB CDate() and Format() functions to convert and format a short or long date entered via a TextBox at runtime. The formatting is included while assigning the variable its value:

```
Dim DATESTART
'Declare DATESTART as a variable.

DATESTART= Format(CDate(DateStart.Text), " _
\D\a\t\e\(\yyyy\,mm\,dd\)")
'Assign the (long or short day, no day of
'week) contents of the DateStart TextBox
'to the DATESTART variable and use the
'VB CDate() and Format() functions to
'convert it and insert Date(, ,)
'as needed by Seagate Crystal.
```

```
Report1.SelectionFormula = "{datafile.DATEFIELD}= " &  
DATESTART  
`Concatenate the Seagate Crystal database field with  
`the DATESTART variable.  
MsgBox Report1.SelectionFormula  
`Display the exact value that is passed to  
`Seagate Crystal Reports in a message box.  
`This message box will display the Seagate Crystal  
`database field ,an equal sign, the date  
`formatting and whatever was entered in the  
`DateStart TextBox at runtime.  
`The Seagate Crystal report will include only  
`records with the Order Date of whatever  
`was entered in the DateStart TextBox
```

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

<http://support.crystaldecisions.com/>

Email Support:

<http://support.crystaldecisions.com/support/answers.asp>

Telephone Support:

<http://www.crystaldecisions.com/contact/support.asp>