crystal decisions™

Seagate Crystal Reports 7.0

Passing Parameter Fields at runtime using Crystal Print Engine calls (API)

Overview

This document describes the new API calls for Parameter Fields and Stored Procedure Parameters in Seagate Crystal Reports version 7.0.

Contents

Introduction

The API calls that are utilized in this document are:

- PEOpenEngine
- PEOpenPrintJob
- PEOutputToWindow
- PEStartPrintJob
- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterCurrentValue
- PEAddParameterDefaultValue
- PEAddParameterCurrentRange
- PEClearParameterCurrentValuesAndRanges
- PEClosePrintJob
- PECloseEngine

- PEGetErrorCode for error handling

NOTE 1	This tutorial assumes that: • The report to be previewed to the screen is connecting via a native connection to PC based database • Visual Basic 6 is being used

NOTE 2	• If using VB3 or VB4 16bit, add the GLOBAL.BAS file to your application. • If using VB4 32 bit, VB5 or VB6, add the GLOBAL32.BAS file to your application. *The GLOBAL.BAS and GLOBAL32.BAS are modules that have declared structures for the Crystal Reports Print Engine. These structures are made available to VB that will allow us to make calls to the Print Engine.

Creating an application that will launch a report

It is necessary to declare the essential variables that will be used within this VB project:

1. Add another Module to your VB application.

2. Within the **General Declarations** portion of the Module, type in the code listed below.

General Declarations

```
Global job As Integer
Global Handle As Integer
Global ErrorNum As Integer
Global Result As Integer
Global ParamName As String
Global ParamReport As String
Global ParamNum As Integer
```

Adding menus to the Visual Basic form:

1. Click on the **Tools | Menu Editor** menu option. The **Menu Editor** dialog appears (Figure 1).

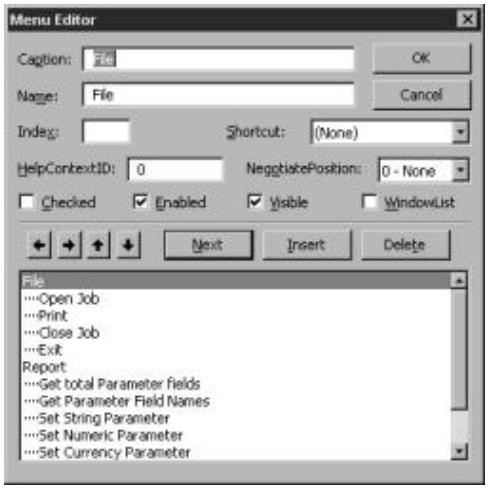

Figure 1

2. Add 'File' to the **Caption** and **Name** textboxes and click on the **Next** button.
3. Click on the right arrow button.
4. Add 'Open Job' to the **Caption** and **Name** textboxes and click on the **Next** button.
5. Add 'Logon Server' to the **Caption** and **Name** textboxes and click on the **Next** button.
6. Add 'Print' to the **Caption** and **Name** textboxes and click on the **Next** button.
7. Add 'Close Job' to the **Caption** and **Name** textboxes and click on the **Next** button.

8. Add 'Exit' to the **Caption** and **Name** textboxes and click on the **Next** button.
9. Click on the left arrow button.
10. Add 'Report' to the **Caption** and **Name** textboxes and click on the **Next** button.
11. Click on the right arrow button.
12. Add 'Get total Parameter fields' to the **Caption** and **Name** textboxes and click on the **Next** button.
13. Add 'Get Parameter Field Names' to the **Caption** and **Name** textboxes and click on the **Next** button.
14. Add 'Add Parameter Current Value' to the **Caption** and **Name** textboxes and click on the **Next** button.
15. Click on the right arrow button.
16. Add 'Add String' to the **Caption** and **Name** textboxes and click on the **Next** button.
17. Add 'Add Numeric' to the **Caption** and **Name** textboxes and click on the **Next** button.
18. Add 'Add Currency' to the **Caption** and **Name** textboxes and click on the **Next** button.
19. Add 'Add Boolean' to the **Caption** and **Name** textboxes and click on the **Next** button.
20. Add 'Add Date Time' to the **Caption** and **Name** textboxes and click on the **Next** button.
21. Add 'Add Time' to the **Caption** and **Name** textboxes and click on the **Next** button.
22. Click on the left arrow button.
23. Add 'Add Parameter Default Value' to **Caption** and **Name** textboxes and click on the **Next** button.
24. Click on the right arrow button.
25. Add 'Add String' to the **Caption** and **Name** textboxes and click on the **Next** button.
26. Add 'Add Numeric' to the **Caption** and **Name** textboxes and click on the **Next** button.
27. Add 'Add Currency' to the **Caption** and **Name** textboxes and click on the **Next** button.
28. Add 'Add Boolean' to the **Caption** and **Name** textboxes and click on the **Next** button.
29. Add 'Add Date Time' to the **Caption** and **Name** textboxes and click on the **Next** button.
30. Add 'Add Time' to the **Caption** and **Name** textboxes and click on the **Next** button.
31. Click on the left arrow button

32. Add 'Add Current Range' to the **Caption** and **Name** textboxes and click on the **Next** button.
33. Add 'Clear Current Values and Ranges' to the **Caption** and **Name** textboxes and click on the **Next** button.

Opening the Engine and the Print Job

The two API calls that are used here are:

- PEOpenEngine
- PEOpenPrintJob

Using the PEOpenEngine API call could be considered the equivalent to adding an OCX/ActiveX control to a VB form.

Using the PEOpenPrintJob API call could be considered the equivalent to setting the **ReportFileName** property for the OCX/ActiveX control.

1. On the VB form, click on the **File** menu and click on the **Open Job** option.

2. Type in the following code:

```
Private Sub OpenJob_Click()

'Open the Print Engine
'This is the same as adding an OCX control to a VB 'form
'The Handle variable captures the handle for the 'print
'engine.  On success, it returns 1 for true.

Handle = PEOpenEngine

'If the Print Engine opened successfully, notify the 'user,
else
'let the user know that an error occurred and what 'the
error number is.

If Handle <> 0 Then
   MsgBox "The Print Engine has been opened"
Else
   ErrorNum = PEGetErrorCode(Handle)
   MsgBox "A Failure occurred when opening the Print
Engine"
   MsgBox "The error code is: " & ErrorNum
End If

'Open the Print Job for a particular report to open.  'This
is like the ReportFileName property in the OCX.
```

```
'The Job variable captures the handle of the print 'job.
On success, it returns 1 for true.


Job = PEOpenPrintJob(App.Path & "\shell.rpt")


'If the Print Job opened successfully (i.e. Found the
'report), notify the user with a 'Success' message, 'else
let the user know that an error occurred when 'opening the
Print Job and what the error number is.


If Job <> 0 Then
    MsgBox "The job opened successfully for   SHELL.RPT.rpt"
   MsgBox "The job number is: " & Job
Else
   ErrorNum = PEGetErrorCode(Job)
   MsgBox "The job failed to open for SHELL.RPT"
   MsgBox "The error code is: " & ErrorNum
End If


End Sub
```

Getting the number of Parameters in the report

The API call that is used here is:

- PEGetNParameterFields

The PEGetNParameterFields API call checks the report and returns the number of parameter fields that exist within the report.

1. On the VB form, click on the **`Report`** menu and click on the **`Get total Parameter fields`** option.

2. Type in the following code:

```
Private Sub GetParameters_Click()


ParameterCount = PEGetNParameterFields(Job)
MsgBox "The number of ParameterFields in the report is: " &
ParamNum


End Sub
```

Getting the name(s) of the Parameter(s) in the report

The two API calls that are used here are:

- PEGetNParameterFields
- PEGetNthParameterField

The PEGetNParameterFields API call checks the report and returns the number of parameter fields that exist within the report

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

1. On the VB form, click on the **Report** menu and click on the **Get total Parameter fields** option.
2. Type in the following code:

```
Private Sub GetParameterFieldNames_Click()


'Declare an integer to be used within a FOR...NEXT 'loop
Dim i As Integer


'Declare a variable/pointer to the 'PEParameterFieldInfo
structure
Dim ParamInfo As PEParameterFieldInfo


'Set the size of the PEParameterFieldInfo structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

ParamNum = PEGetNParameterFields(Job)


'For the first parameter (starting with 0), perform 'the
following code for the total number of parameter 'fields in
the report - 1 (The total amount of 'Parameter fields in
the report start with 1, not 0)

For I = 0 To (ParamNum - 1)

   Result = PEGetNthParameterField(job, I, ParamInfo)

   MsgBox "The name of Parameter number: " & I & "   is: "
    & ParamInfo.Name + Chr$(0)

Next I


End Sub
```

Passing values to String Parameter fields

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterCurrentValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values are then passed to PEAddParameterCurrentValue.

The PEValueInfo API call contains values defined within VB and stores them in a structure that will be used later by the PEAddParameterCurrentValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEAddParameterCurrentValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Current Value** option. Then click on **Add String**.

2. Type in the following code:

```
Private Sub CurrentString_Click()


'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo
'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO
```

```
'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)
   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
   'Set the datatype to a String parameter
   valueInfo.valueType = PE_VI_STRING

   'Assign the value of the string you are passing.
   valueInfo.viString = "This is my parameter     string."
   + Chr$(0)

   'Set the current value
   Result = PEAddParameterCurrentValue(job,
    ParamInfo.Name, ParamInfo.reportName, valueInfo)

   'Error checking for the PEAddParameterCurrentValue 'API
   call (0 if the call fails)
   If Result <> 0 Then
      MsgBox "The call PEAddParameterCurrentValue was
       successful"
   Else
      ErrorNum = PEGetErrorCode(Job)
      MsgBox "The call PEAddParameterCurrentValue  failed"
      MsgBox "The error code is: " & ErrorNum
   End If
Next I

End Sub
```

NOTE This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters.

Passing values to Numeric Parameter fields

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterCurrentValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values are then passed to PEAddParameterCurrentValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterCurrentValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEAddParameterCurrentValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Current Value** option. Then click on **Add Numeric**.

2. Type in the following code:

```
Private Sub CurrentNumeric_Click()

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
```

```
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)
   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
   'Set the datatype to a Numeric parameter
   valueInfo.valueType = PE_VI_NUMBER

   'Assign the value of the number you are passing.
   valueInfo.viNumber = 201

   'Set the current value
   Result = PEAddParameterCurrentValue(job,
    ParamInfo.Name, ParamInfo.reportName, valueInfo)

   'Error checking for the PEAddParameterCurrentValue  'API
   call (0 if the call fails)
      If Result <> 0 Then
         MsgBox "The call PEAddParameterCurrentValue
          was successful"
      Else
         ErrorNum = PEGetErrorCode(Job)
         MsgBox "The call PEAddParameterCurrentValue
          failed"
         MsgBox "The error code is: " & ErrorNum
      End If
Next I

End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters.

Passing values to Currency Parameter in the report.

The API calls that will be used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterCurrentValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values can then be passed to PEAddParameterCurrentValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterCurrentValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of n parameter fields in the report.

The PEAddParameterCurrentValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Current Value** option. Then click on **Add Currency**.

2. Type in the following code:

```
Private Sub CurrentCurrency_Click()

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)
```

```
For I = 0 To (ParamNum - 1)

   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
   'Set the datatype to a Currency parameter
   valueInfo.valueType = PE_VI_CURRENCY

   'Assign the value of the string you are passing.
   valueInfo.viCurrency = 205.47

   'Set the current value
   Result = PEAddParameterCurrentValue(job,
    ParamInfo.Name, ParamInfo.reportName, valueInfo)

   'Error checking for the PEAddParameterCurrentValue  'API
   call (0 if the call fails)
   If Result <> 0 Then
      MsgBox "The call PEAddParameterCurrentValue was
       successful"
   Else
      ErrorNum = PEGetErrorCode(Job)
      MsgBox "The call PEAddParameterCurrentValue  failed"
      MsgBox "The error code is: " & ErrorNum
   End If
Next I

End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters.

Passing values to Boolean Parameter fields

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields

- PEGetNthParameterField
- PEAddParameterCurrentValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values can then be passed to PEAddParameterCurrentValue.

The PEValueInfo API call contains values defined within VB and stores to a structure that will be used later by the PEAddParameterCurrentValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEAddParameterCurrentValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Current Value** option. Then click on **Add Boolean**.

2. Type in the following code:

```
Private Sub CurrentBoolean_Click()

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)
   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
'Set the datatype to a Boolean parameter
valueInfo.valueType = PE_VI_BOOLEAN

'Assign the value of the boolean you are passing.
valueInfo.viBoolean = 1

'Set the current value
Result = PEAddParameterCurrentValue(job,
 ParamInfo.Name, ParamInfo.reportName, valueInfo)

'Error checking for the PEAddParameterCurrentValue 'API
 call (0 if the call fails)
If Result <> 0 Then
   MsgBox "The call PEAddParameterCurrentValue was
    successful"
Else
   ErrorNum = PEGetErrorCode(Job)
   MsgBox "The call PEAddParameterCurrentValue  failed"
   MsgBox "The error code is: " & ErrorNum
End If
Next I

End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters

Passing values to Date Time Parameter fields

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterCurrentValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values can then be passed to PEAddParameterCurrentValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterCurrentValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEAddParameterCurrentValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Current Value** option. Then click on **Add Currency**.

2. Type in the following code:

```
Private Sub CurrentDateTime_Click()

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Declare an integer to be used within a FOR...NEXT loop
Dim I As Integer

'Set the structure size for PEParameterFieldInfo structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)
   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
   'Set the datatype to a Date Time parameter
   valueInfo.valueType = PE_VI_DATETIME
```

```
        'Assign the value of the number you are passing.
        valueInfo.viDateTime(0) = 1999          'yyyy
        valueInfo.viDateTime(1) = 2                'mm
        valueInfo.viDateTime(2) = 3                'dd
        valueInfo.viDateTime(3) = 3                'hh
        valueInfo.viDateTime(4) = 45               'mm
        valueInfo.viDateTime(5) = 00               'ss


        'Set the current value
        Result = PEAddParameterCurrentValue(job,
        ParamInfo.Name, ParamInfo.reportName, valueInfo)


        'Error checking for the PEAddParameterCurrentValue 'API
        call (0 if the call fails)
        If Result <> 0 Then
           MsgBox "The call PEAddParameterCurrentValue was
            successful"
        Else
           ErrorNum = PEGetErrorCode(Job)
           MsgBox "The call PEAddParameterCurrentValue  failed"
           MsgBox "The error code is: " & ErrorNum
        End If
    Next I


    End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters

Passing values to Time Parameter fields

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterCurrentValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values can then be passed to PEAddParameterCurrentValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterCurrentValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of n parameter fields in the report.

The PEAddParameterCurrentValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Current Value** option. Then click on **Add Time**.

2. Type in the following code:

```
Private Sub CurrentDateTime_Click()

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)

   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
        'Set the datatype to a Time parameter
        valueInfo.valueType = PE_VI_TIME

        'Assign the value of the time you are passing
        valueInfo.viTime(0) = 3              'hh
        valueInfo.viTime(1) = 45             'mm'
        valueInfo.viTime(2) = 00             'ss

        ' Set the current value
        Result = PEAddParameterCurrentValue(job,
         ParamInfo.Name, ParamInfo.reportName, valueInfo)

        'Error checking for the PEAddParameterCurrentValue  'API
        call (0 if the call fails)
        If Result <> 0 Then
           MsgBox "The call PEAddParameterCurrentValue was
            successful"
        Else
           ErrorNum = PEGetErrorCode(Job)
           MsgBox "The call PEAddParameterCurrentValue  failed"
           MsgBox "The error code is: " & ErrorNum
        End If
    Next I

    End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters

Passing values to Default String Parameter fields

The API calls that will be used here are:

- PEParameterFieldInf
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterDefaultValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values are then passed to PEAddParameterDefaultValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterDefaultValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEAddParameterDefaultValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Default Value** option. Then click on **Add String**.

2. Type in the following code:

```
Private Sub DefaultString_Click()

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)

    'Get the information for this parameter field
    Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
'Set the datatype to a String parameter
valueInfo.valueType = PE_VI_STRING

'Assign the value of the string you are passing.
valueInfo.viString = "This is my default parameter
 string." + CHR$(0)

'Set the default value
Result = PEAddParameterDefaultValue(job,
 ParamInfo.Name, ParamInfo.reportName, valueInfo)

'Error checking for the PEAddParameterDefaultValue 'API
 call (0 if the call fails)
If Result <> 0 Then
   MsgBox "The call PEAddParameterDefaultValue was
    successful"
Else
   ErrorNum = PEGetErrorCode(Job)
   MsgBox "The call PEAddParameterDefaultValue  failed"
   MsgBox "The error code is: " & ErrorNum
End If
Next I
End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters.

Passing values to Default Numeric Parameter fields

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterDefaultValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values are then passed to PEAddParameterDefaultValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterDefaultValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEAddParameterDefaultValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Default Value** option. Then click on **Add Numeric**.

2. Type in the following code:

```
Private Sub DefaultNumeric_Click()

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Set the structure size for PEParameterFieldInfo structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)

   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
        'Set the datatype to a Numeric parameter
        valueInfo.valueType = PE_VI_NUMBER

        'Assign the value of the number you are passing.
        valueInfo.viNumber = 201

        'Set the default value
        Result = PEAddParameterDefaultValue(job,
         ParamInfo.Name, ParamInfo.reportName, valueInfo)

        'Error checking for the PEAddParameterDefaultValue  'API
        call (0 if the call fails)
        If Result <> 0 Then
           MsgBox "The call PEAddParameterDefaultValue was
            successful"
        Else
           ErrorNum = PEGetErrorCode(Job)
           MsgBox "The call PEAddParameterDefaultValue  failed"
           MsgBox "The error code is: " & ErrorNum
        End If
    Next I

    End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters.

Passing values to Default Currency Parameters

The API calls that are used here are:

- PEParameterFieldInfoPEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterDefaultValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values are passed to PEAddParameterDefaultValue.

The PEValueInfo API call contains values defined within VB and stores them in a structure that will be used later by the PEAddParameterDefaultValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEAddParameterDefaultValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Default Value** option. Then click on **Add Currency**.

2. Type in the following code:

```
Private Sub DefaultCurrency_Click()

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)
   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
   'Set the datatype to a Currency parameter
   valueInfo.valueType = PE_VI_CURRENCY

   'Assign the value of the string you are passing.
```

```
        valueInfo.viCurrency = 205.47


        'Set the default value
        Result = PEAddParameterDefaultValue(job,
        ParamInfo.Name, ParamInfo.reportName, valueInfo)


        'Error checking for the PEAddParameterDefaultValue  'API
        call (0 if the call fails)
        If Result <> 0 Then
           MsgBox "The call PEAddParameterDefaultValue was
            successful"
        Else
           ErrorNum = PEGetErrorCode(Job)
           MsgBox "The call PEAddParameterDefaultValue  failed"
           MsgBox "The error code is: " & ErrorNum
        End If
    Next I


    End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters.

Passing values to Default Boolean Parameter fields

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterDefaultValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values can then be passed to PEAddParameterDefaultValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterDefaultValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of n parameter fields in the report.

The PEAddParameterDefaultValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the Report menu and click on the Add Parameter Default Value option. Then click on Add Boolean.

2. Type in the following code:

```
Private Sub DefaultBoolean_Click()

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO
'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)
   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
   'Set the datatype to a Boolean parameter
   valueInfo.valueType = PE_VI_BOOLEAN

   'Assign the value of the boolean you are passing.
   valueInfo.viBoolean = 1

   'Set the default value
   Result = PEAddParameterDefaultValue(job, ParamInfo.Name,
    ParamInfo.reportName, valueInfo)
```

```
        'Error checking for the PEAddParameterDefaultValue API
        'call (0 if the call fails)
        If Result <> 0 Then
           MsgBox "The call PEAddParameterDefaultValue was
            successful"
        Else
           ErrorNum = PEGetErrorCode(Job)
           MsgBox "The call PEAddParameterDefaultValue  failed"
           MsgBox "The error code is: " & ErrorNum
        End If
    Next I

    End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters

Passing values to Default Date Time Parameter fields

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterDefaultValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values can then be passed to PEAddParameterDefaultValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterDefaultValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of n parameter fields in the report.

The PEAddParameterDefaultValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **`Report`** menu and click on the **`Add Parameter Default Value`** option. Then click on **`Add Currency`**.

2. Type in the following code:

```
Private Sub DefaultDateTime_Click()

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)
   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
   'Set the datatype to a Date Time parameter
   valueInfo.valueType = PE_VI_DATETIME

'Assign the value of the number you are passing.
valueInfo.viDateTime(0) = 1999            'yyyy
valueInfo.viDateTime(1) = 2               'mm
valueInfo.viDateTime(2) = 3               'dd
valueInfo.viDateTime(3) = 3               'hh
valueInfo.viDateTime(4) = 45             'mm
valueInfo.viDateTime(5) = 00              'ss

'Set the default value
Result = PEAddParameterDefaultValue(job, ParamInfo.Name,
ParamInfo.reportName, valueInfo)
```

```
'Error checking for the PEAddParameterDefaultValue API
'call (0 if the call fails)
   If Result <> 0 Then
      MsgBox "The call PEAddParameterDefaultValue was
       successful"
   Else
      ErrorNum = PEGetErrorCode(Job)
      MsgBox "The call PEAddParameterDefaultValue  failed"
      MsgBox "The error code is: " & ErrorNum
   End If
Next I


End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters.

Passing values to Time Parameter fields in the report

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGetNthParameterField
- PEAddParameterDefaultValue

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values can then be passed to PEAddParameterDefaultValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterDefaultValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEAddParameterDefaultValue API call sets a value for a specified parameter field that exists within the report.

1. On the VB form, click on the **Report** menu and click on the **Add Parameter Default Value** option. Then click on **Add Time**.

2. Type in the following code:

```
Private Sub DefaultTime_Click()

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
Dim ParamInfo As PEParameterFieldInfo
Dim valueInfo As PEValueInfo

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
valueInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)
   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
'Set the datatype to a Time parameter
valueInfo.valueType = PE_VI_TIME

'Assign the value of the time you are passing.

valueInfo.viTime(0) = 3              'hh
valueInfo.viTime(1) = 45             'mm
valueInfo.viTime(2) = 00             'ss

'Set the default value
```

```
Result = PEAddParameterDefaultValue(job, ParamInfo.Name,
ParamInfo.reportName, valueInfo)


'Error checking for the PEAddParameterDefaultValue API
'call (0 if the call fails)
   If Result <> 0 Then
      MsgBox "The call PEAddParameterDefaultValue was
       successful"
   Else
      ErrorNum = PEGetErrorCode(Job)
      MsgBox "The call PEAddParameterDefaultValue  failed"
      MsgBox "The error code is: " & ErrorNum
   End If
Next I


End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters

Adding Current Values to a Range

The API calls that are used here are:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGenNthParameterFields
- PEAddParameterCurrentRange

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values are then passed to PEAddParameterDefaultValue.

The PEValueInfo API call contains values defined within VB and stores them to a structure that will be used later by the PEAddParameterDefaultValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEAddParameterCurrentRange API adds a range to the specified parameter field in the report.

1. On the VB form, click on the **Report** menu and click on **Add Current Range** option.

2. Type in the following code:

```
Private Sub AddCurrentRange_Click()

'The range of parameters in this case is defined as a
'number in the report.

'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures
'for the start and end of the range

Dim ParamInfo As PEParameterFieldInfo
Dim minInfo As PEValueInfo
Dim maxInfo As PEValueInfo

'Declare an integer to be used within a FOR...NEXT 'loop
Dim I As Integer

'Set the structure size for PEParameterFieldInfo 'structure
ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO

'Set the structure size for PEValueInfo
minInfo.StructSize = PE_SIZEOF_VALUE_INFO
maxInfo.StructSize = PE_SIZEOF_VALUE_INFO

'Get the number of parameters in the report
ParamNum = PEGetNParameterFields(job)

For I = 0 To (ParamNum - 1)
   'Get the information for this parameter field
   Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
   'Set the datatype to a Numeric parameter
   minInfo.valueType = PE_VI_NUMBER
```

```
        maxInfo.valueType = PE_VI_NUMBER


        'Assign the value of the number you are passing.
        minInfo.viNumber = 200
        maxInfo.viNumber = 300


        'Set the current value
        Result = PEAddParameterCurrentRange(job, ParamInfo.Name,
         ParamInfo.reportName, minInfo, maxInfo, 1 Or 2)


        'Error checking for the PEAddParameterCurrentRange API
        call (0 if the call fails)
        If Result <> 0 Then
           MsgBox "The call PEAddParameterCurrentRange was
            successful"
        Else
           ErrorNum = PEGetErrorCode(job)
           MsgBox "The call PEAddParameterCurrentRange failed"
           MsgBox "The error code is: " & ErrorNum
        End If
    Next I


    End Sub
```

NOTE	This assumes that you have already created parameter fields within your report or you have Stored Procedure parameters

Clearing Default Values and Ranges

The API call that are used here is:

- PEParameterFieldInfo
- PEValueInfo
- PEGetNParameterFields
- PEGenNthParameterFields
- PEClearParameterCurrentValuesAndRanges

The PEParameterFieldInfo API structure gets the Parameter name and the Report Name using PEGetNthParameterField. These values to PEAddParameterDefaultValue.

The PEValueInfo API call contains values defined within VB to a structure that will be used later by the PEAddParameterDefaultValue API call.

The PEGetNParameterFields API call checks the report and returns the total number of parameter fields that exist within the report.

The PEGetNthParameterField API call returns information about 1 of N parameter fields in the report.

The PEClearParameterCurrentValuesAndRanges API call clears the specified parameter field of all current values and ranges.

1. On the VB form, click on the **Report** menu and click on the **Clear Current Values** and **Ranges** option.

2. Type in the following code:

```
Private Sub ClearCurrent_Click()


'Declare a variable/pointer to PEParameterFieldInfo 'and
PEValueInfo structures

Dim ParamInfo As PEParameterFieldInfo

Dim valueInfo As PEValueInfo


'Set the structure size for PEParameterFieldInfo 'structure

ParamInfo.StructSize = PE_SIZEOF_PARAMETER_FIELD_INFO


'Set the structure size for PEValueInfo

valueInfo.StructSize = PE_SIZEOF_VALUE_INFO


'Get the information for this parameter field

Result = PEGetNthParameterField(job, I, ParamInfo)
```

NOTE	Insert appropriate looping code for your application that will check or set the value type and pass the parameter value

```
'Clear current values

Result = PEClearParameterCurrentValuesAndRanges(job,
ParamInfo.Name, ParamInfo.reportName)


If Result = 0 Then

   ErrorNum = PEGetErrorCode(Job)

   MsgBox "Could not clear Parameter current values: " &
    ErrorNum

Else

   MsgBox "Parameters cleared"

End If
```

```
End Sub
```

Printing the report to the window

The two API calls that are used here are:

- PEOutputToWindow
- PEStartPrintJob

The PEOutputToWindow API call is considered the same as setting the **Destination** property of the OCX/ActiveX control

The PEStartPrintJob API call is considered as being the same as setting the **Action** property for the OCX/ActiveX control.

1. On the VB form, click on the **File** menu and click on the **Print** option.
2. Type in the following code:

```
Private Sub Print_Click()
'Set the destination of where the report is to be 'printer
(Window, Printer, Export)
'This would be like the Destination property in the 'OCX


Handle = PEOutputToWindow(Job, "REPORT.RPT", 0, 0, 520,
520, 0, 0)


'If there was an error trying to print the report to 'the
window, notify the user of the error with the 'error number
else start the Print Job.
If Handle <> 0 Then
  'Start the print job and print it to the window
  'This would be like the Action property in the OCX
   Handle = PEStartPrintJob(Job, True)


   If Handle <> 0 Then
      MsgBox "The report is printing"
   Else
      ErrorNum = PEGetErrorCode(Job)
      MsgBox "An error occurred while trying to Start the
        Print Job"
      MsgBox "The error code is: " & ErrorNum
   End If


   Else
```

```
        ErrorNum = PEGetErrorCode(Job)
        MsgBox "The output cannot be sent to the window"
        MsgBox "The error code is: " & ErrorNum
    End If
End Sub
```

Closing the Print Job

The API call used here is:

- PEClosePrintJob

The PEClosePrintJob API call releases the handle of the current report.

1. On the VB form, click on the **File** menu and click onto the **Close Job** option.
2. Place in the following code:

```
Private Sub CloseJob_Click()
'Close the print job after the report has been 'previewed
    PEClosePrintJob (Job)
End Sub
```

Closing the Print Engine

The API call used here is:

- PECloseEngine

The PECloseEngine API releases the handle of the Print Engine (similar to closing a VB form that contains an OCX/ActiveX control).

1. On the VB form, click on the **File** menu and click on the **Close Job'** option.
2. Type in the following code:

```
Private Sub Exit_Click()

'Close the print engine before exiting the 'application.
    PECloseEngine
    End
End Sub
```

Related Knowledge Base Articles

How to you use PEAddParameterCurrentValue to pass a Stored Procedure Parameter?

The API call PESetParameterCurrentValue is defined as:

```
Bool = PEAddParameterCurrentValue(JobHandle, ParameterName,
ReportName, PEValueInfo.currentValue)
```

In practice there are things to note. The Stored Procedure parameter name will have an "@" sign appended before the actual name.

e.g. A stored procedure parameter called "PurchDate" showing up in the Parameter field list as ?@PurchDate will be passed as "@PurchDate"

A parameter field will stay as it has been named.

e.g. A parameter field called ?ParamDate will be seen as "ParamDate".

In both cases, the Report name should be passed as an empty string.

```
Result = PEAddParameterCurrentValue(job, "ParamDate"
+ Chr$(0), "" + Chr$(0), valueInfo)
```

Here is a sample of the code in Microsoft Visual Basic:

```
Private Sub CurrentDateTime_Click()

Dim Result As Integer

Dim valueInfo As PEValueInfo

valueInfo.StructSize = PE_SIZEOF_VALUE_INFO
valueInfo.valueType = PE_VI_DATETIME     'Date Time
parameter value
valueInfo.viDateTime(0) = "1999"                  'yyyy
valueInfo.viDateTime(1) = "2"                        'mm
valueInfo.viDateTime(2) = "3"                        'dd
valueInfo.viDateTime(3) = "3"                        'hh
valueInfo.viDateTime(4) = "45"                      'mm
valueInfo.viDateTime(5) = "00"                       'ss

'Set the current value for the parameter ?PuchDate. At
'this time we are passing 'an empty string for report
'name. The name of a Stored Procedure parameter 'will 'have
a "@" ' in front of it. The Stored Procedure 'parameter
"PurchDate" is 'seen as in "@PurchDate".

Result = PEAddParameterCurrentValue(job, "@PurchDate" +
Chr$(0), "" + Chr$(0), valueInfo)

If Result = 0 Then

    ErrorNum = PEGetErrorCode(Result)

   MsgBox "Could not pass Parameter current values: " &
ErrorNum

Else
```

```
   MsgBox "Parameters passed successfully"
End If
End Sub
```

C2003181

PEAddParameterCurrentRange fails with Error 672

A report contains a single parameter field which will store a range of values.

When trying to set the range parameter at runtime using PEAddParameterCurrentRange. the Minimum and Maximum values are correctly set but PEAddParameterCurrentRange returns an Error 672 and the parameter range is not passed.

This is a known issue and has been tracked (track id 19517).

The GLOBAL.BAS file for Microsoft Visual Baiic has an error in the PEAddParameterCurrentRange declaration. The rangeInfo should be ByVal not ByRef:

```
Declare Function PEAddParameterCurrentRange Lib
"crpe32.dll" (ByVal printJob%, ByVal parameterFieldName$,
ByVal reportName$, rangeStart As PEValueInfo, rangeEnd As
PEValueInfo, ByVal rangeInfo%) As Integer
```

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