Error checking

- Single logic test

```
If( logic test )
        Exit script( error result)
End if
```

- Single pass loop

Error checking

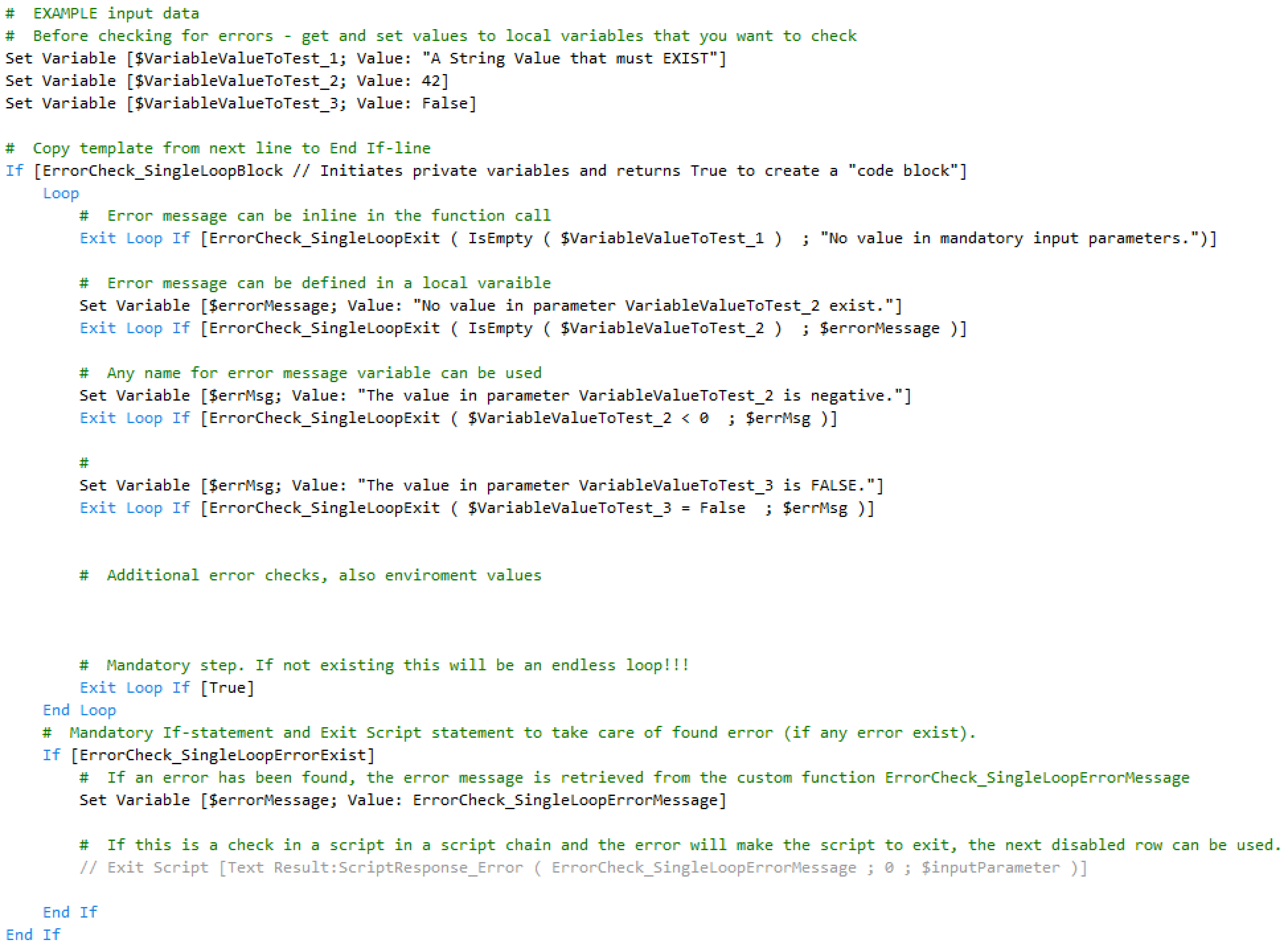
- Single pass loop (part one)

```
Loop
    Exit loop if (logic test 1)
    Exit loop if (logic test 2)
    ….
    Exit loop( True)
End loop
```

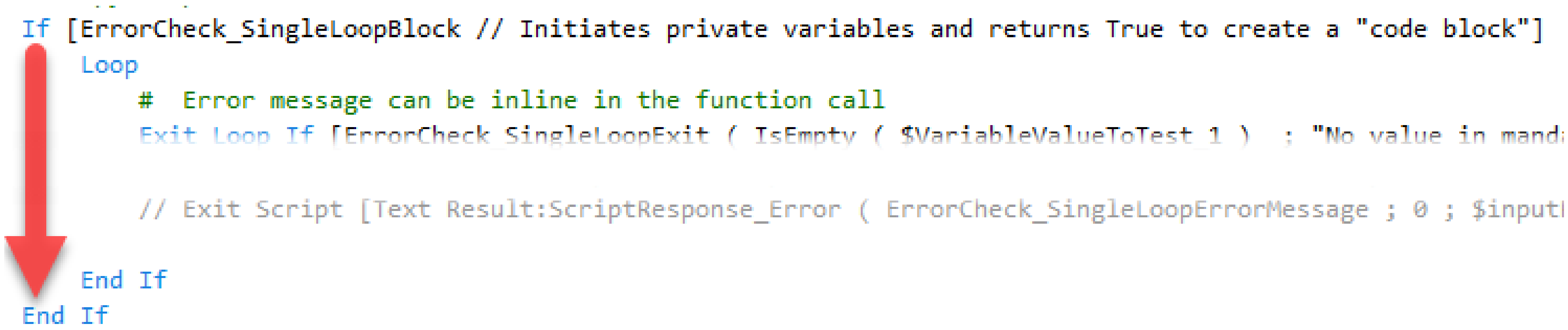
Error checking

- Single pass loop (part two)

```
…
End loop
If(   ThereWasAnError  )
        Exit Script ( ThatError )
End if
```

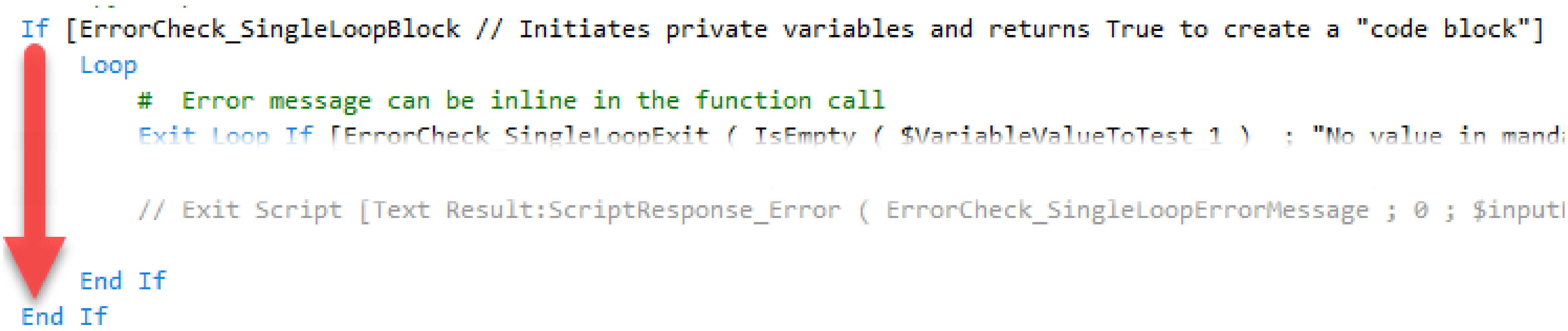


```
#  EXAMPLE input data
#  Before checking for errors - get and set values to local variables that you want to check
Set Variable [$VariableValueToTest_1; Value: "A String Value that must EXIST"]
Set Variable [$VariableValueToTest_2; Value: 42]
Set Variable [$VariableValueToTest_3; Value: False]

#  Copy template from next line to End If-line
If [ErrorCheck_SingleLoopBlock // Initiates private variables and returns True to create a "code block"]
    Loop
        #  Error message can be inline in the function call
        Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $VariableValueToTest_1 )  ; "No value in mandatory input parameters.")]

        #  Error message can be defined in a local varaible
        Set Variable [$errorMessage; Value: "No value in parameter VariableValueToTest_2 exist."]
        Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $VariableValueToTest_2 )  ; $errorMessage )]

        #  Any name for error message variable can be used
        Set Variable [$errMsg; Value: "The value in parameter VariableValueToTest_2 is negative."]
        Exit Loop If [ErrorCheck_SingleLoopExit ( $VariableValueToTest_2 < 0  ; $errMsg )]

        #
        Set Variable [$errMsg; Value: "The value in parameter VariableValueToTest_3 is FALSE."]
        Exit Loop If [ErrorCheck_SingleLoopExit ( $VariableValueToTest_3 = False  ; $errMsg )]


        #  Additional error checks, also enviroment values



        #  Mandatory step. If not existing this will be an endless loop!!!
        Exit Loop If [True]
    End Loop
    #  Mandatory If-statement and Exit Script statement to take care of found error (if any error exist).
    If [ErrorCheck_SingleLoopErrorExist]
        #  If an error has been found, the error message is retrieved from the custom function ErrorCheck_SingleLoopErrorMessage
        Set Variable [$errorMessage; Value: ErrorCheck_SingleLoopErrorMessage]

        #  If this is a check in a script in a script chain and the error will make the script to exit, the next disabled row can be used.
        // Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoopErrorMessage ; 0 ; $inputParameter )]

    End If
End If
```



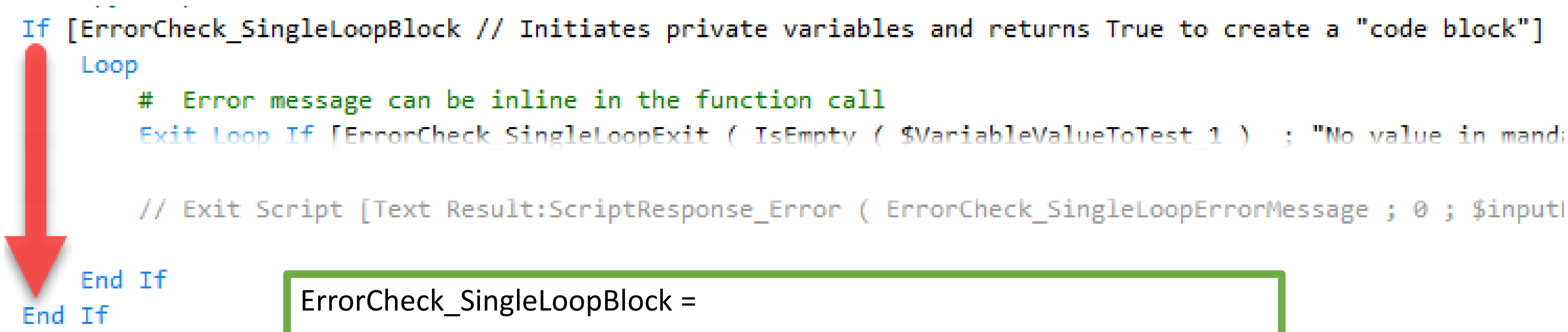
```
If [ErrorCheck_SingleLoopBlock // Initiates private variables and returns True to create a "code block"]
    Loop
        #  Error message can be inline in the function call
        Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $VariableValueToTest_1 )  ; "No value in mand

        // Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoopErrorMessage ; 0 ; $input

    End If
End If
```

= True



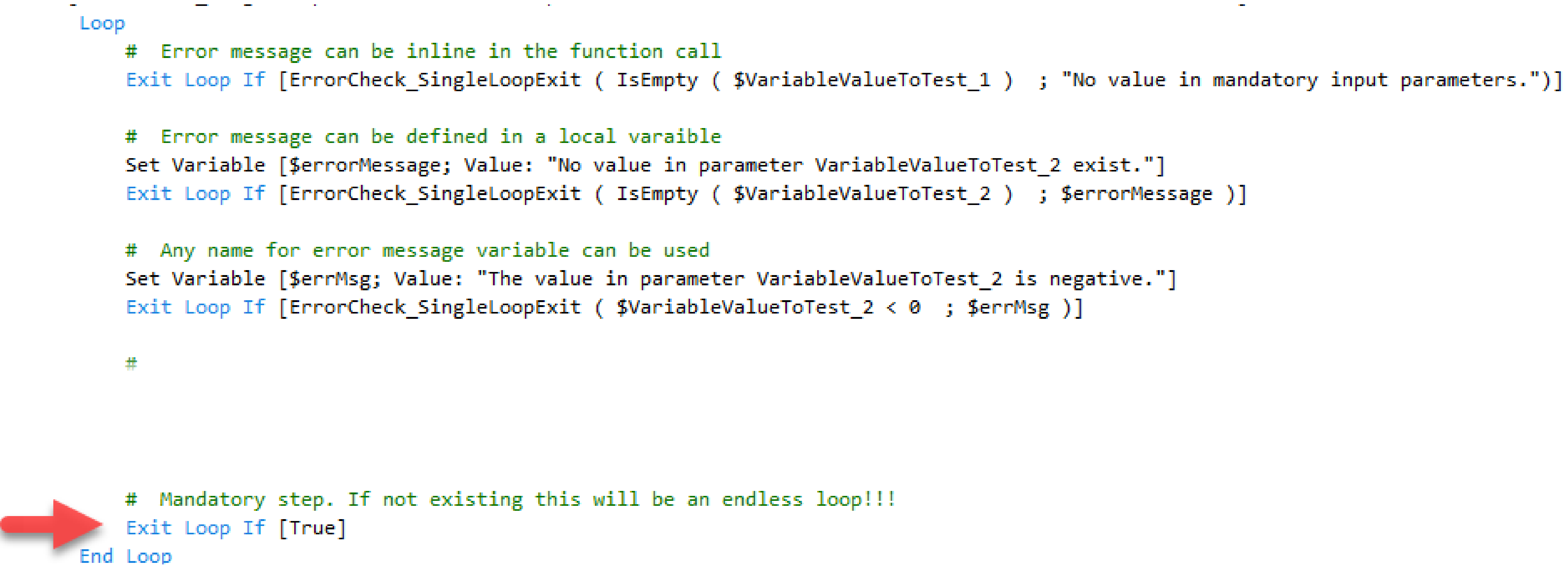
```
If [ErrorCheck_SingleLoopBlock // Initiates private variables and returns True to create a "code block"]
    Loop
        #  Error message can be inline in the function call
        Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $VariableValueToTest_1 )  ; "No value in mand

        // Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoopErrorMessage ; 0 ; $inputI

    End If
End If
```

= True

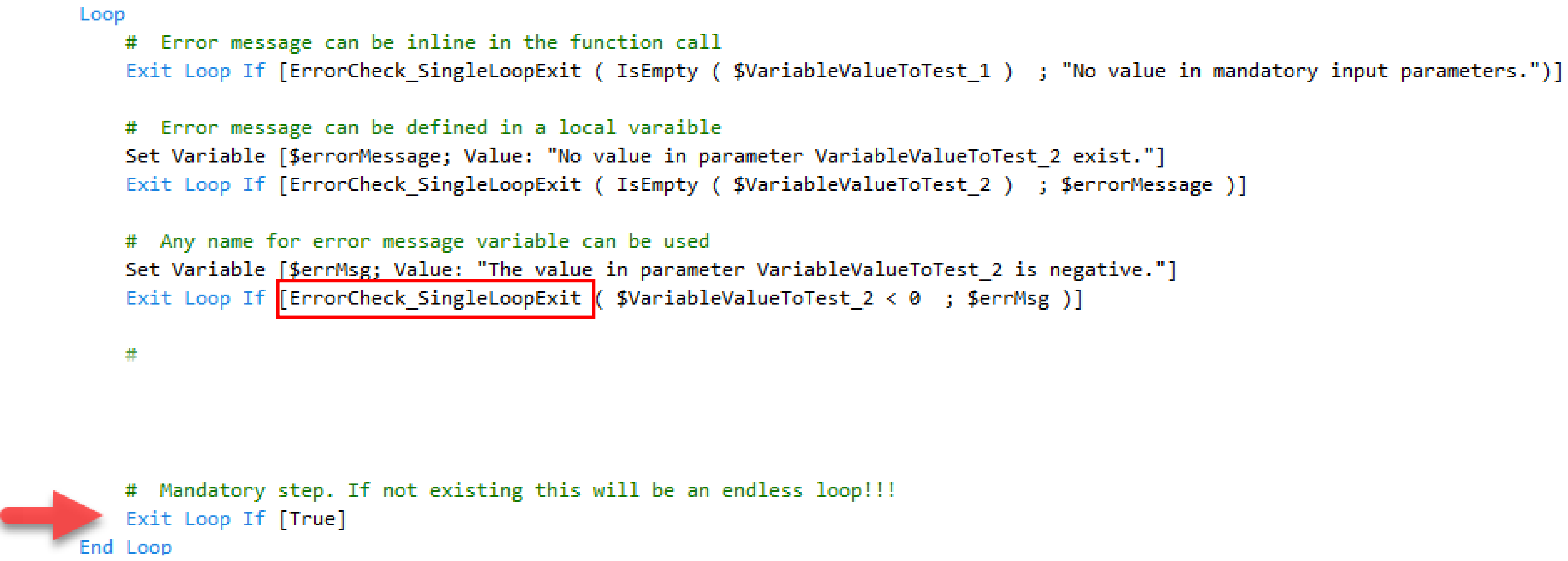


```
If [ErrorCheck_SingleLoopBlock // Initiates private variables and returns True to create a "code block"]
    Loop
        #  Error message can be inline in the function call
        Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $VariableValueToTest 1 )  ; "No value in mand

        // Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoopErrorMessage ; 0 ; $inputI

    End If
End If
```

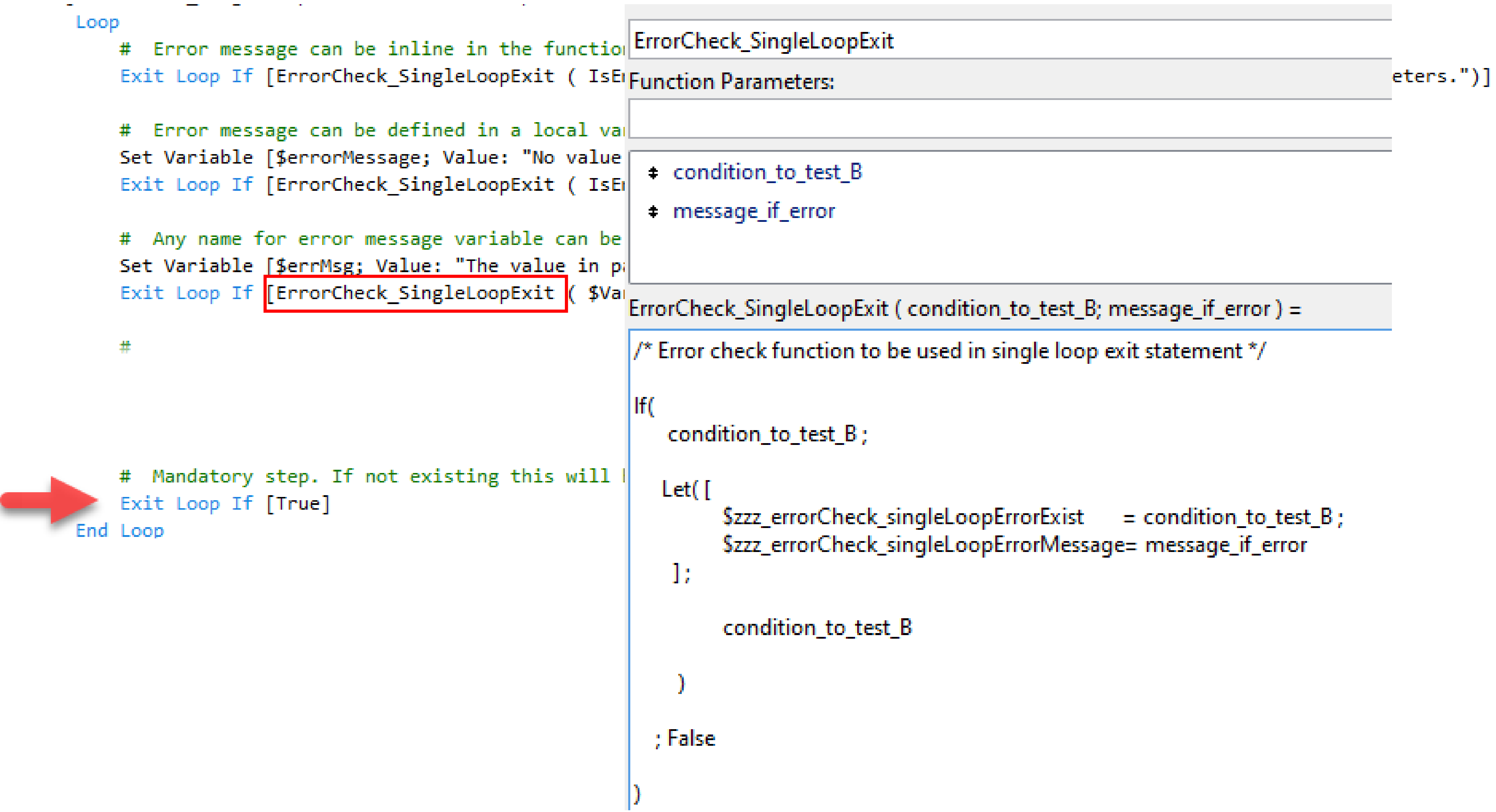
```
ErrorCheck_SingleLoopBlock =



Let( [
            $zzz_errorCheck_singleLoopErrorExist = False ;
            $zzz_errorCheck_singleLoopErrorMessage = ""


      ] ;
            True
   )
```



```
Loop
    #  Error message can be inline in the function call
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $VariableValueToTest_1 )  ; "No value in mandatory input parameters.")]

    #  Error message can be defined in a local varaible
    Set Variable [$errorMessage; Value: "No value in parameter VariableValueToTest_2 exist."]
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $VariableValueToTest_2 )  ; $errorMessage )]

    #  Any name for error message variable can be used
    Set Variable [$errMsg; Value: "The value in parameter VariableValueToTest_2 is negative."]
    Exit Loop If [ErrorCheck_SingleLoopExit ( $VariableValueToTest_2 < 0  ; $errMsg )]

    #



    #  Mandatory step. If not existing this will be an endless loop!!!
    Exit Loop If [True]
End Loop
```



```
Loop
    #  Error message can be inline in the function call
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $VariableValueToTest_1 )  ; "No value in mandatory input parameters.")]

    #  Error message can be defined in a local varaible
    Set Variable [$errorMessage; Value: "No value in parameter VariableValueToTest_2 exist."]
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $VariableValueToTest_2 )  ; $errorMessage )]

    #  Any name for error message variable can be used
    Set Variable [$errMsg; Value: "The value in parameter VariableValueToTest_2 is negative."]
    Exit Loop If [ErrorCheck_SingleLoopExit ( $VariableValueToTest_2 < 0  ; $errMsg )]

    #




    #  Mandatory step. If not existing this will be an endless loop!!!
    Exit Loop If [True]
End Loop
```



```
Loop
    #  Error message can be inline in the functio
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsE                                        eters.")]

    #  Error message can be defined in a local va
    Set Variable [$errorMessage; Value: "No value
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsE

    #  Any name for error message variable can be
    Set Variable [$errMsg; Value: "The value in p
    Exit Loop If [ErrorCheck_SingleLoopExit ( $Va

    #




    #  Mandatory step. If not existing this will
    Exit Loop If [True]
End Loop
```

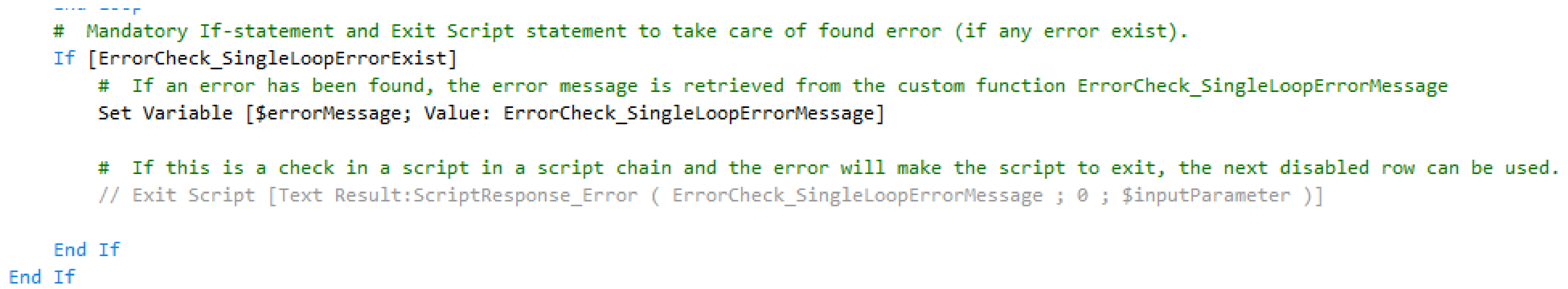
ErrorCheck_SingleLoopExit

Function Parameters:

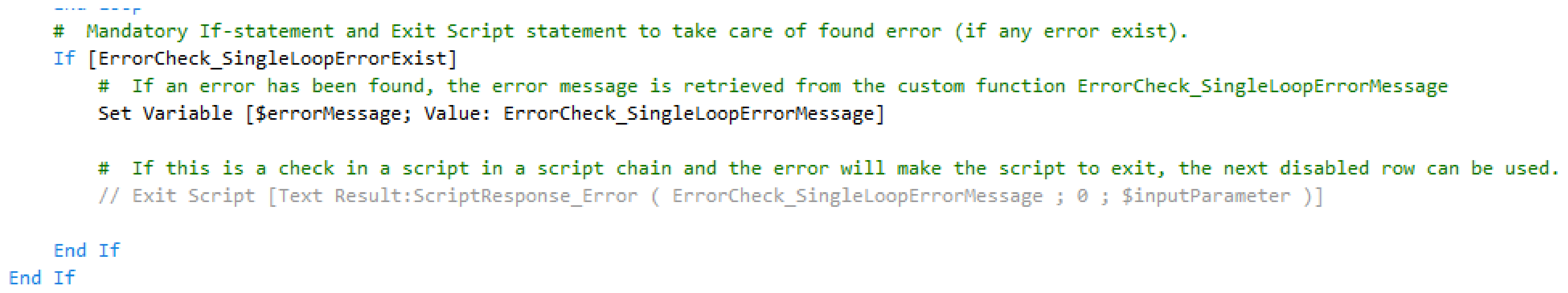
- condition_to_test_B
- message_if_error

ErrorCheck_SingleLoopExit (condition_to_test_B; message_if_error) =

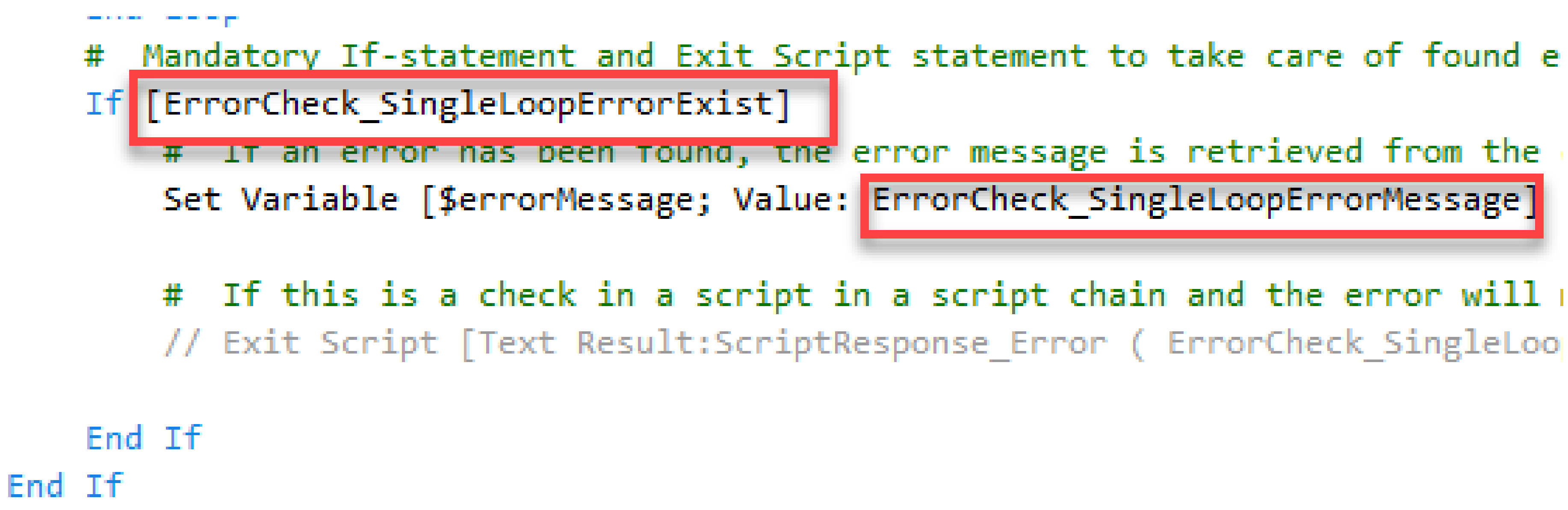
```
/* Error check function to be used in single loop exit statement */

If(
    condition_to_test_B ;

   Let( [
         $zzz_errorCheck_singleLoopErrorExist      = condition_to_test_B ;
         $zzz_errorCheck_singleLoopErrorMessage= message_if_error
    ];

         condition_to_test_B

    )

  ; False

)
```



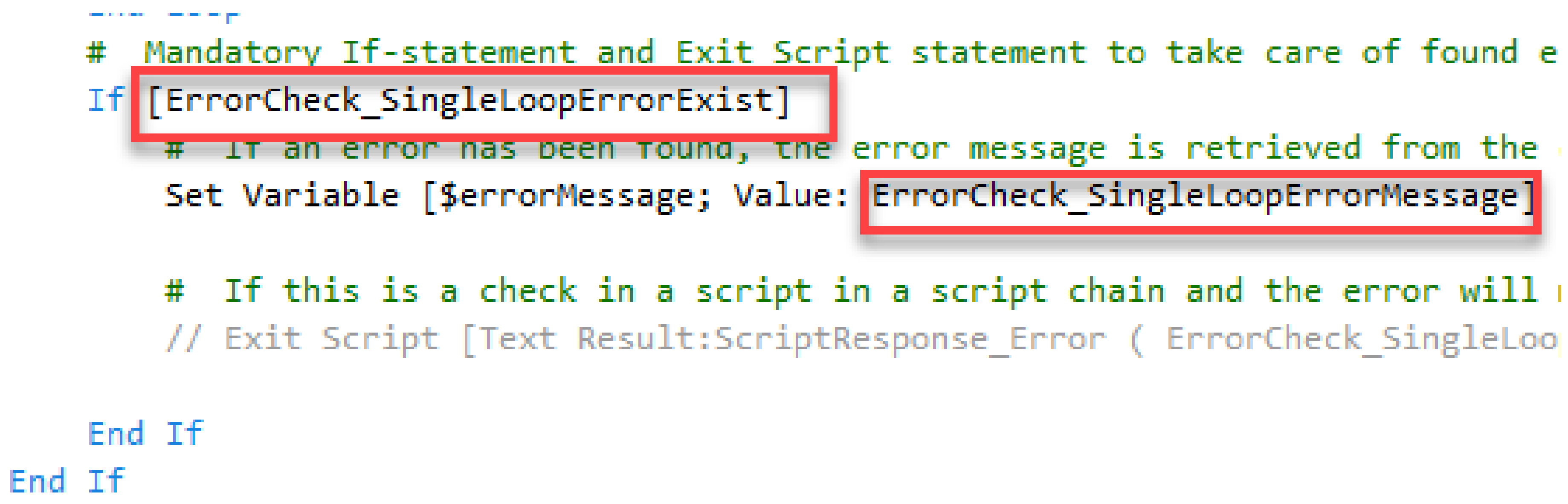
```
    End Loop
    #  Mandatory If-statement and Exit Script statement to take care of found error (if any error exist).
    If [ErrorCheck_SingleLoopErrorExist]
        #  If an error has been found, the error message is retrieved from the custom function ErrorCheck_SingleLoopErrorMessage
        Set Variable [$errorMessage; Value: ErrorCheck_SingleLoopErrorMessage]

        #  If this is a check in a script in a script chain and the error will make the script to exit, the next disabled row can be used.
        // Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoopErrorMessage ; 0 ; $inputParameter )]

    End If
End If
```



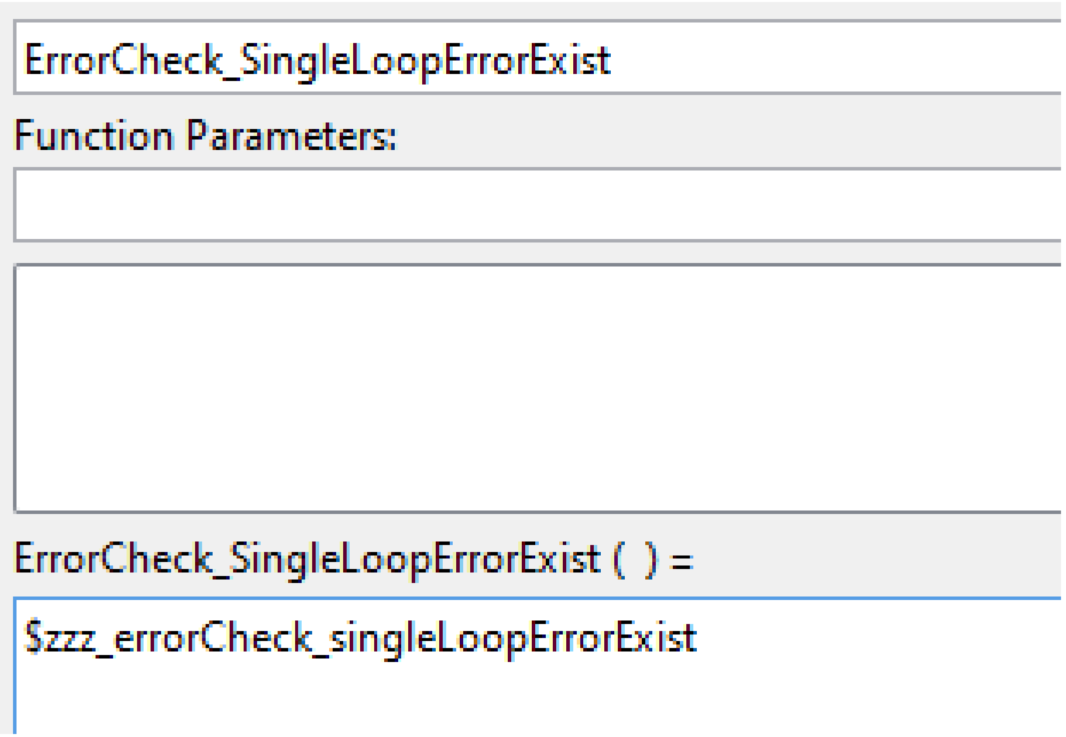
```
    #  Mandatory If-statement and Exit Script statement to take care of found error (if any error exist).
    If [ErrorCheck_SingleLoopErrorExist]
        #  If an error has been found, the error message is retrieved from the custom function ErrorCheck_SingleLoopErrorMessage
        Set Variable [$errorMessage; Value: ErrorCheck_SingleLoopErrorMessage]

        #  If this is a check in a script in a script chain and the error will make the script to exit, the next disabled row can be used.
        // Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoopErrorMessage ; 0 ; $inputParameter )]

    End If
End If
```



```
    #  Mandatory If-statement and Exit Script statement to take care of found e
    If [ErrorCheck_SingleLoopErrorExist]
        #  If an error has been found, the error message is retrieved from the
        Set Variable [$errorMessage; Value: ErrorCheck_SingleLoopErrorMessage]

        #  If this is a check in a script in a script chain and the error will
        // Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoo

    End If
End If
```



```
    End Loop
    #  Mandatory If-statement and Exit Script statement to take care of found e
    If [ErrorCheck_SingleLoopErrorExist]
        #  If an error has been found, the error message is retrieved from the
        Set Variable [$errorMessage; Value: ErrorCheck_SingleLoopErrorMessage]

        #  If this is a check in a script in a script chain and the error will
        // Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoo

    End If
End If
```

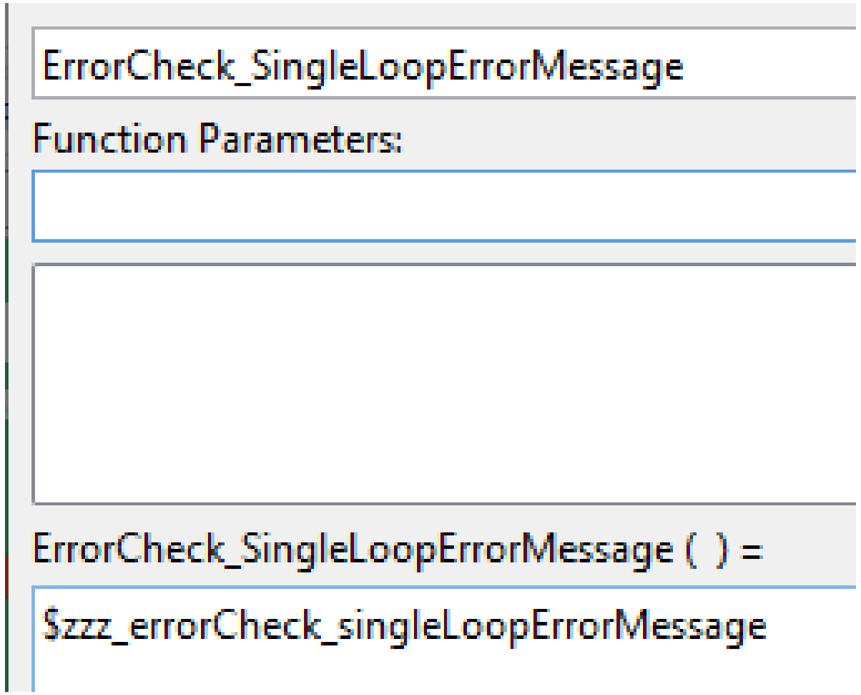


ErrorCheck_SingleLoopErrorExist

Function Parameters:

ErrorCheck_SingleLoopErrorExist ( ) =

```
$zzz_errorCheck_singleLoopErrorExist
```

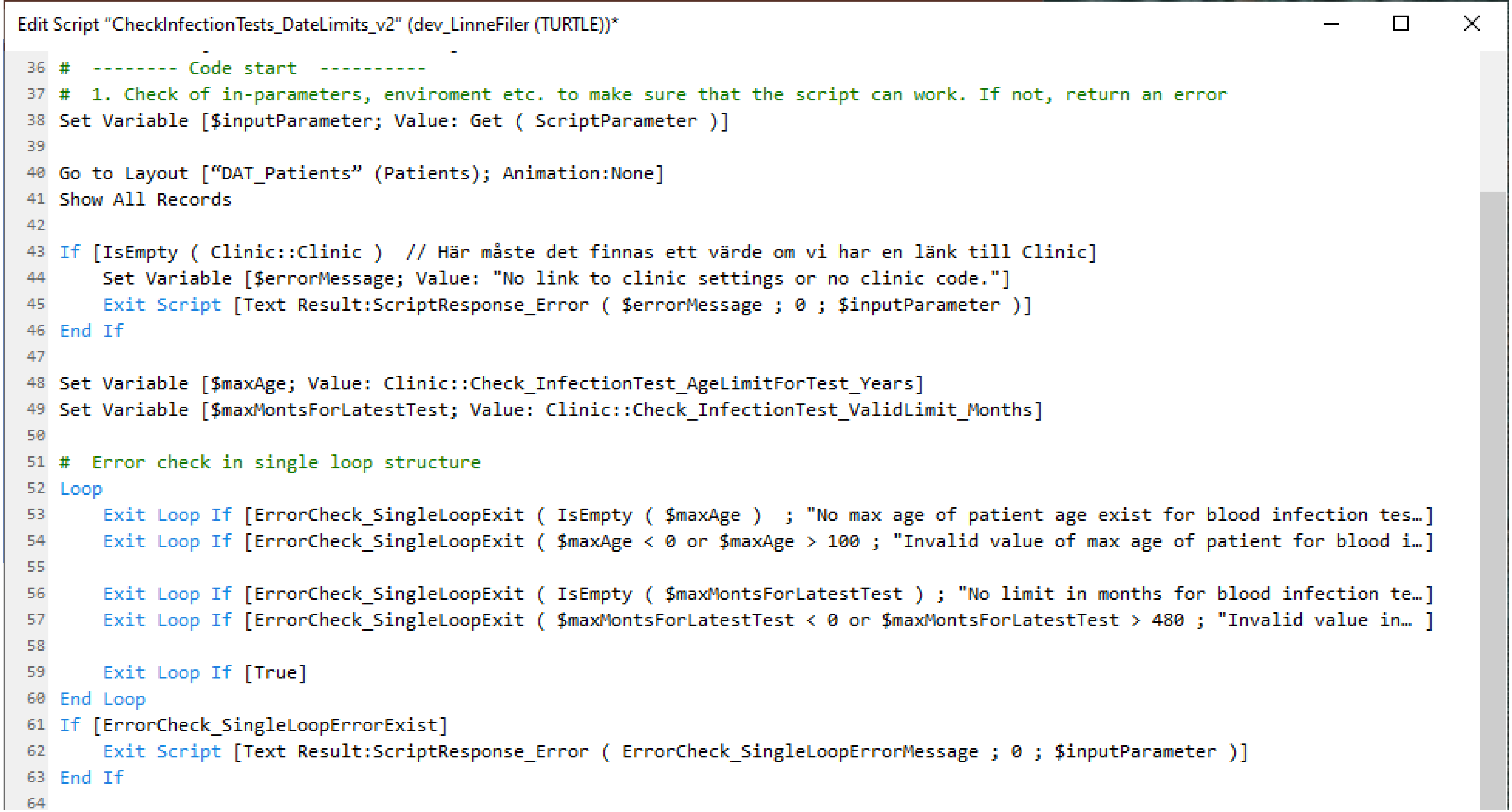


ErrorCheck_SingleLoopErrorMessage

Function Parameters:

ErrorCheck_SingleLoopErrorMessage ( ) =

```
$zzz_errorCheck_singleLoopErrorMessage
```



Edit Script "CheckInfectionTests_DateLimits_v2" (dev_LinneFiler (TURTLE))*

```
#  -------- Code start  ----------
#  1. Check of in-parameters, enviroment etc. to make sure that the script can work. If not, return an error
Set Variable [$inputParameter; Value: Get ( ScriptParameter )]

Go to Layout [“DAT_Patients” (Patients); Animation:None]
Show All Records

If [IsEmpty ( Clinic::Clinic )  // Här måste det finnas ett värde om vi har en länk till Clinic]
    Set Variable [$errorMessage; Value: "No link to clinic settings or no clinic code."]
    Exit Script [Text Result:ScriptResponse_Error ( $errorMessage ; 0 ; $inputParameter )]
End If

Set Variable [$maxAge; Value: Clinic::Check_InfectionTest_AgeLimitForTest_Years]
Set Variable [$maxMontsForLatestTest; Value: Clinic::Check_InfectionTest_ValidLimit_Months]

#  Error check in single loop structure
Loop
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $maxAge )  ; "No max age of patient age exist for blood infection tes…]
    Exit Loop If [ErrorCheck_SingleLoopExit ( $maxAge < 0 or $maxAge > 100 ; "Invalid value of max age of patient for blood i…]

    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $maxMontsForLatestTest ) ; "No limit in months for blood infection te…]
    Exit Loop If [ErrorCheck_SingleLoopExit ( $maxMontsForLatestTest < 0 or $maxMontsForLatestTest > 480 ; "Invalid value in… ]

    Exit Loop If [True]
End Loop
If [ErrorCheck_SingleLoopErrorExist]
    Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoopErrorMessage ; 0 ; $inputParameter )]
End If

```

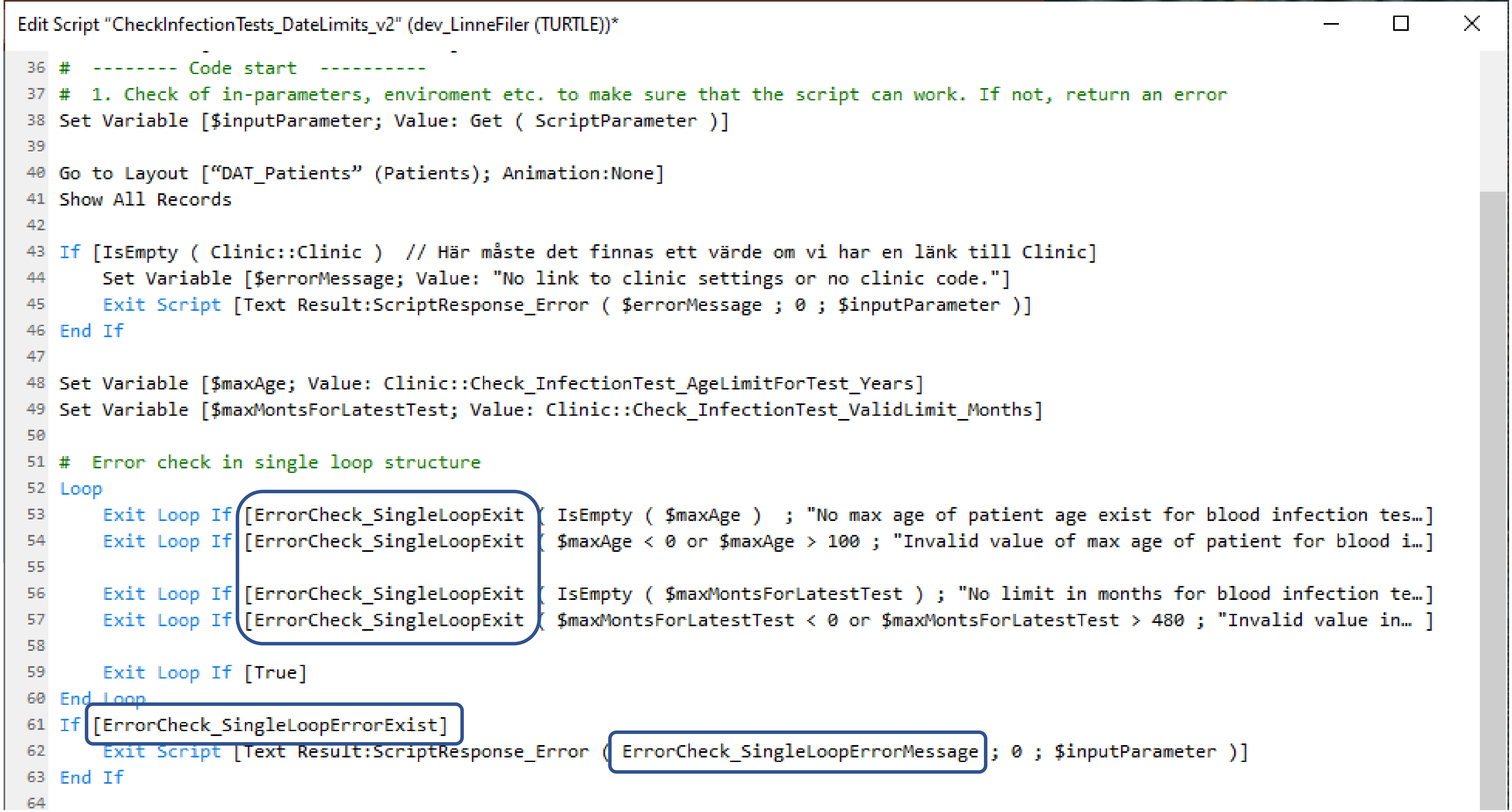
Error checking – Single loop

Custom Functions:

```
ErrorCheck_SingleLoopExit(
    condition_to_test_B;
    message_if_error              )
```

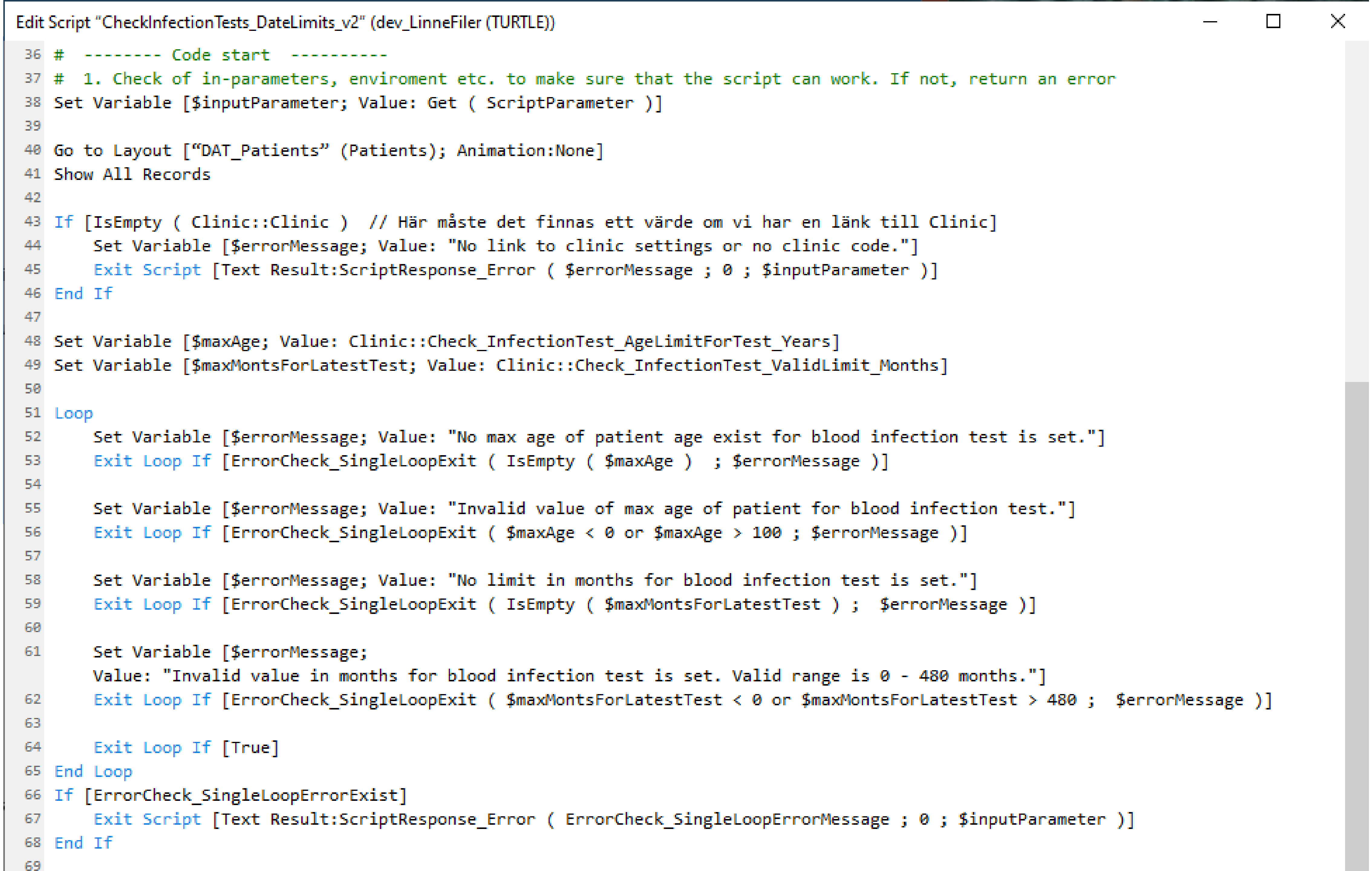
ErrorCheck_SingleLoopErrorExist

ErrorCheck_SingleLoopErrorMessage



Edit Script "CheckInfectionTests_DateLimits_v2" (dev_LinneFiler (TURTLE))*

```
#  -------- Code start  ----------
#  1. Check of in-parameters, enviroment etc. to make sure that the script can work. If not, return an error
Set Variable [$inputParameter; Value: Get ( ScriptParameter )]

Go to Layout [“DAT_Patients” (Patients); Animation:None]
Show All Records

If [IsEmpty ( Clinic::Clinic )  // Här måste det finnas ett värde om vi har en länk till Clinic]
    Set Variable [$errorMessage; Value: "No link to clinic settings or no clinic code."]
    Exit Script [Text Result:ScriptResponse_Error ( $errorMessage ; 0 ; $inputParameter )]
End If

Set Variable [$maxAge; Value: Clinic::Check_InfectionTest_AgeLimitForTest_Years]
Set Variable [$maxMontsForLatestTest; Value: Clinic::Check_InfectionTest_ValidLimit_Months]

#  Error check in single loop structure
Loop
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $maxAge )  ; "No max age of patient age exist for blood infection tes…]
    Exit Loop If [ErrorCheck_SingleLoopExit ( $maxAge < 0 or $maxAge > 100 ; "Invalid value of max age of patient for blood i…]

    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $maxMontsForLatestTest ) ; "No limit in months for blood infection te…]
    Exit Loop If [ErrorCheck_SingleLoopExit ( $maxMontsForLatestTest < 0 or $maxMontsForLatestTest > 480 ; "Invalid value in… ]

    Exit Loop If [True]
End Loop
If [ErrorCheck_SingleLoopErrorExist]
    Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoopErrorMessage ; 0 ; $inputParameter )]
End If
```


Edit Script "CheckInfectionTests_DateLimits_v2" (dev_LinneFiler (TURTLE))

```
#  -------- Code start  ----------
#  1. Check of in-parameters, enviroment etc. to make sure that the script can work. If not, return an error
Set Variable [$inputParameter; Value: Get ( ScriptParameter )]

Go to Layout [“DAT_Patients” (Patients); Animation:None]
Show All Records

If [IsEmpty ( Clinic::Clinic )  // Här måste det finnas ett värde om vi har en länk till Clinic]
    Set Variable [$errorMessage; Value: "No link to clinic settings or no clinic code."]
    Exit Script [Text Result:ScriptResponse_Error ( $errorMessage ; 0 ; $inputParameter )]
End If

Set Variable [$maxAge; Value: Clinic::Check_InfectionTest_AgeLimitForTest_Years]
Set Variable [$maxMontsForLatestTest; Value: Clinic::Check_InfectionTest_ValidLimit_Months]

Loop
    Set Variable [$errorMessage; Value: "No max age of patient age exist for blood infection test is set."]
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $maxAge )  ; $errorMessage )]

    Set Variable [$errorMessage; Value: "Invalid value of max age of patient for blood infection test."]
    Exit Loop If [ErrorCheck_SingleLoopExit ( $maxAge < 0 or $maxAge > 100 ; $errorMessage )]

    Set Variable [$errorMessage; Value: "No limit in months for blood infection test is set."]
    Exit Loop If [ErrorCheck_SingleLoopExit ( IsEmpty ( $maxMontsForLatestTest ) ;  $errorMessage )]

    Set Variable [$errorMessage;
    Value: "Invalid value in months for blood infection test is set. Valid range is 0 - 480 months."]
    Exit Loop If [ErrorCheck_SingleLoopExit ( $maxMontsForLatestTest < 0 or $maxMontsForLatestTest > 480 ;  $errorMessage )]

    Exit Loop If [True]
End Loop
If [ErrorCheck_SingleLoopErrorExist]
    Exit Script [Text Result:ScriptResponse_Error ( ErrorCheck_SingleLoopErrorMessage ; 0 ; $inputParameter )]
End If
```