IDE Error Messages

Click a message to display Help.

.. not valid outside of class scope

A - E

Cannot forward declare user-defined class or data type

CASE ELSE must be the last CASE in a SELECT statement

Compiler stack overflow at: <token name>

Compiler statement stack overflow at: <token name>

DIM required on declarations in this scope

Duplicate procedure name: <procedure name>

F - J

Illegal character after %INCLUDE directive

Illegal character after continuation character

Illegal directive

Illegal duplicate END statement

Illegal executable code in Declarations

Illegal executable code in Options

Illegal executable code outside procedure

Illegal on declarations in this scope: <keyword>

Illegal range specifier

Illegal statement

Illegal type suffix on keyword: <keyword>

Illegal use of escape character

Illegal use of escape character in identifier

Illegal use of parentheses

Invalid type for procedure

K - O

ME not valid outside of class scope

Name too long

Named product class instance not valid here

Out of memory

P - T

Procedure definitions illegal in this scope

Procedures may not be forward declared

PUBLIC not allowed in this module

SET required on class instance assignment

Statement illegal in CLASS block: <keyword>

Statement illegal in TYPE block: <keyword>

Statement is illegal in this scope

Syntax checking buffer overflow

U - Z

Unexpected: <token>; Expected: <token>

Unmatched block terminator

Unterminated block statement

Unterminated square bracket reference

Unterminated string constant

Unterminated <keyword> block

{button ,AL(`H_IDE_IDE_ERROR_MESSAGES_OVER_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Duplicate procedure name: <procedure name>

You assigned the same name to more than one Sub, Function, Type, Property Set, or Property Get statement assigned to an object or (Globals). No two subs assigned to an object or (Globals) can have the same name, no two functions can have the same name, and so on.

Remove the duplicate name.

{button ,AL(`H_IDE_STR_DUPLICATESECTION_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Illegal executable code in Declarations

In (Declarations), you included a statement that is not allowed in the section.

Only the following are allowed in object (Declarations): comments, the Private keyword, the Declare (external C calls), and the Const, Dim, Type, and class statements.

(Global) (Declarations) can also contain the Public keyword.

Remove the invalid statements from (Declarations).

Illegal statement

You used a colon (:), followed by an underscore (_), to separate and then recombine statements in a line of script. This is not allowed in the IDE.

Use a colon (:) between multiple statements in a line; use an underscore (_) at the end of a line to continue the line.

{button ,AL(`H_IDE_STR_ILLSTMT_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Procedures may not be forward declared

You tried to use the Declare statement to forward declare a sub, function, or property for an object. This is not necessary because the IDE generates forward declares for you.

{button ,AL(`H_IDE_STR_INVFWD_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Cannot forward declare user-defined class or data type

You tried to use the Declare statement to declare a type or class before defining it. This is not allowed in the IDE.

{button ,AL(`H_IDE_STR_INVFWDCL_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

Unterminated block statement

You omitted the ending statement for a block statement that begins with one of the following keywords:

Class

Do

For

ForAll

Function

If...Then...Else...EndIf

Property Get

Property Set

Select

Sub

Type

While

With

Enter the appropriate ending statement.

{button ,AL(`H_IDE_STR_UNMATCHEDBLOCKBEGIN_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Unmatched block terminator

You omitted the statement that begins with one of the following keywords and marks the beginning of a statement block:

Class

Do

For

ForAll

Function

If...Then...Else...EndIf

Property Get

Property Set

Select

Sub

Type

While

With

Enter the appropriate beginning statement.

{button ,AL(`H_IDE_STR_UNMATCHEDBLOCKENDER_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Syntax checking buffer overflow

A statement you entered exceeds the limit of 32K bytes. Split the statement into multiple units, each with fewer than 32,768 bytes of data.

In general:

- Each line of script (including each line in a multiline statement) can contain approximately 1000 characters.
- Each statement can contain approximately 2400 language elements.

{button ,AL(`H_IDE_STR_TOKENOVERFLOW_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

PUBLIC not allowed in this module

You defined a public variable, sub, function, property, or constant for an object other than (Globals). You can do one of the following to correct this problem:

- Move the definition to (Globals).
- Leave the definition where it is, but do one of the following:
  - If you are declaring a variable, replace the Public keyword with Private or Dim.
  - Otherwise, delete the Public keyword or change it to Private.

{button ,AL(`H_IDE_STR_ILLPUBLIC_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

Procedure definitions illegal in this scope

You tried to define a sub, function, or property within a sub, function, or property that defines a class method or property. This is not allowed.

Move the definition outside the scope of the class method or property.

{button ,AL(`H_IDE_STR_ILLSUBPROG_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Illegal duplicate END statement

You added an End Sub, End Function, or End Property statement to a procedure that already contains the statement. Delete the duplicate statement.

{button ,AL(`H_IDE_STR_ALREADYHASENDER_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Illegal executable code outside procedure

You entered an executable statement outside a sub, function, or other procedure. Delete the statement or move it inside a procedure definition.

If you want the statement to be executed when scripts for an object are loaded, move the statement into the Initialize sub. If you want the statement to be executed when scripts for an object are unloaded, move the statement into the Terminate sub.

{button ,AL(`H_IDE_STR_ILLCODEOUTSIDE_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Statement is illegal in this scope

You tried to enter a statement in a scope where it is not allowed. Check LotusScript Help for information about the statement you tried to enter and whether it can be used in module, procedure, user-defined data type, or class scope.

{button ,AL(`H_IDE_STR_ILLSTMTINSCOPE_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Invalid type for procedure

You tried to change a product-defined procedure to another type of procedure. This is not allowed.

You cannot, for example, change a product-defined sub to a function by changing the Sub statement to a Function statement. Similarly, you cannot change a product-defined function or procedure to another procedure type.

{button ,AL(`H_IDE_STR_INVSUBPROGRAMTYPE_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Illegal directive

Any of the following could have caused this error:

- You used an unrecognized directive. For example:

```
%EndRem    ' Illegal
%End Rem   ' Legal
```

- You nested a %Rem...%End Rem block inside another %Rem...%End Rem block.
- You used an %End Rem without a preceding %Rem.
- You entered a %If, %Else, %ElseIf, or %EndIf directive in a script you created in the IDE.

  If you want to use %If directives, you must enter them in a file that you call with the %Include directive.

{button ,AL(`H_IDE_STR_BADDIR_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

CASE ELSE must be the last CASE in a SELECT statement

You used a CASE clause after CASE ELSE in a Select Case statement. No other Case clause may follow a CASE ELSE clause.

Make CASE ELSE the last clause in the SELECT Case statement, or omit the keyword Else.

{button ,AL(`H_IDE_STR_CASEELSE_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

DIM required on declarations in this scope

You declared a variable at module level without the Dim, Public, or Private keyword, or you declared a variable inside a procedure without the Dim or Static keyword. One of these is required.

Add the appropriate keyword to the declaration.

{button ,AL(`H_IDE_STR_DIMREQ_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Illegal character after continuation character

The line-continuation character underscore (_) is followed on the same line by a character that is not the comment character ('). The line-continuation character must be the last character on a line, except for an optional comment, beginning with the comment character.

Remove everything following the line-continuation character on the line, or insert a comment character after it to comment out the rest of the line.

{button ,AL(`H_IDE_STR_ILLCONTINUE_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Illegal use of escape character

You included an escape character at the end of a line. This is not allowed. For example:

```
aString$ = "This is a tilde: "
anotherString$ = aString$~
' This is illegal
```

Remove the escape character.

{button ,AL(`H_IDE_STR_ILLESCAPE_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Illegal use of escape character in identifier

You included an escape character in one of the following contexts in which that character is not allowed:

- In a declared name (a variable, constant, procedure, class, or user-defined data type)
- In the name of an implicitly declared variable
- In a label definition or reference
- In the name of the reference variable in a ForAll statement

For example:

```
Dim fo~x As Integer   ' Illegal
```

Remove the escape character.

{button ,AL(`H_IDE_STR_ILLESCAPEID_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

Named product class instance not valid here

In one of the following statements, you used the name of a product object in a context in which it is not allowed:

- An assignment statement (Let or =) in either of the following forms:

  Let *name* = ...

  name = ...

- A Set statement in either of the following forms:

  Set *name* = NEW...

  Set *name* = ...

  Set *name* = Bind...

- A Delete statement
- An Erase statement
- A ForAll statement
- A Get or Put statement
- An Input # or Line Input # statement
- An LSet or RSet statement
- A Mid or MidB statement
- A ReDim statement

Replace the name with an appropriate name, or remove the invalid statement.

Illegal range specifier

You used a Def*type* range in one of the following illegal ways:

- No range was specified.
- The beginning of the range was not a single character between A and Z (ASCII uppercase or lowercase), inclusive.
- The end of the range was not a single character between A and Z (ASCII uppercase or lowercase), inclusive.

Correct the error.

{button ,AL(`H_IDE_STR_ILLRANGE_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

.. not valid outside of class scope

You used "dotdot" syntax outside of a procedure within a class. The "dotdot" syntax is only valid inside procedures within a class. You use "dotdot" notation when referring to a procedure in a base class when the derived class has a procedure of the same name, as in the following example:

```
 CLASS BaseClass
   SUB MySub
    PRINT "In BaseClass's  MySub"
  END SUB
END CLASS

CLASS DerivedClass  AS BaseClass
  SUB MySub
   PRINT "In DerivedClass's MySub "
  END SUB

  SUB MyOtherSub
     CALL MySub                    'Print "In DerivedClass's MySub "
    CALL BaseClass..MySub          'Print "In BaseClass's MySub "
   END SUB
END CLASS
```

Remove the "dotdot" syntax and use an object reference variable in its place.

{button ,AL(`H_IDE_STR_INVALIDDD_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

ME not valid outside of class scope

You used the keyword ME outside of a procedure within a class. Use the keyword ME only inside procedures within a class. You use Me within the definition of a class when referring to members of that class.

Remove the keyword ME. If you are referring to a class member , use an object reference variable instead of Me.

{button ,AL(`H_IDE_STR_INVALIDME_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

SET required on class instance assignment

You attempted to assign an object reference to a variable but omitted the SET keyword. (An object reference can be a reference to an instance of a user-defined class, a product object, an OLE automation object, or the constant NOTHING). The SET keyword is required in object reference assignments. For example:

```
Class MyClass
' ...
End Class
Dim MyObj As New MyClass
Dim varV As Variant
varV = MyObj          ' Illegal syntax
```

Insert the SET keyword in the assignment statement:

```
Class MyClass
' ...
End Class
Dim MyObj As New MyClass
Dim varV As Variant
Set varV = MyObj      ' Legal syntax
```

{button ,AL(`H_IDE_STR_NEEDSET_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

Out of memory

You must free enough memory to perform the operation that caused this error message. To free memory in your computer, do one of the following:

- If you have other programs in memory, end one or more of those programs.
- Reduce the amount or size of PUBLIC data.
- Activate extended memory.

{button ,AL(`H_IDE_STR_OUTOFMEMORY_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Compiler stack overflow at: <token name>

The statement being compiled is too complex. It may contain a complex expression, or deeply nested block statements, such as a Do or For statement.

Reduce the nesting level, or break up the statement into multiple, less complex statements.

{button ,AL(`H_IDE_STR_PARSESTACKOVERFLOW_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Unexpected: <token>; Expected: <token>

The compiler encountered an unexpected language element.

If the unexpected language element is a number appearing inside square brackets, it represents the ASCII code of an unprintable character. For example, if you enter the Backspace character in a statement where a name is expected, the following error message appears when you compile the script:

```
Unexpected: [8]; Expected: Identifier
```

For more information, refer to the list of expected language elements following the unexpected language element in the error message.

{button ,AL(`H_IDE_STR_SIMPLESYNTAX_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;', 0)} See related topics

Illegal use of parentheses

You called a sub or function and enclosed its argument list in parentheses. You can only do this under the following circumstances:

- The sub or function is the target of a Call statement. For example:

```
Call MySub()                          ' Legal
Call MyOtherSub("ABC", 4)             ' Legal
Call MyFunction()                     ' Legal
Call MyOtherFunction(123, "XXX")      ' Legal
```

- The sub or function has a single parameter that the caller is passing by value. For example:

```
MySub("ABC")                      ' Legal
MyFunction(anInt%)                ' Legal
```

- The target is a function that is included in a statement. For example:

```
X% = MyFunction(123, "XXX")       ' Legal
```

The following are illegal:

```
MySub()                           ' Illegal
MyFunction()                      ' Illegal
MyOtherSub("ABC", 4)              ' Illegal
MyOtherFunction(123, "XXX")       ' Illegal
```

Remove the parentheses from around the argument list or call the sub or function with the Call statement.

Compiler statement stack overflow at: <token>

The statement being compiled is too complex. It may contain deeply nested block statements, or single-line If statements.

Reduce the nesting level, or break up the statement into multiple, less complex statements.

{button ,AL(`H_IDE_STR_STMTSTACKOVERFLOW_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Illegal type suffix on keyword: <keyword>

You included an illegal data type suffix character in the name of a LotusScript built-in function. Certain LotusScript built-in functions can end in the $ type suffix character; no other data type suffix character is valid on these functions. The names of other functions cannot end in a data type suffix character. For example:

```
Print Date()      ' Legal
Print Date$()     ' Legal
Print Date#       ' Illegal
Print CDat(Date)  ' Legal
Print CDat$(Date) ' Illegal
```

Remove the suffix character.

{button ,AL(`H_IDE_STR_SUFFKEYWORD_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;', 0)} See related topics

Unterminated <keyword> block

You omitted the keyword that marks the end of one of the following block statements:

Class

Do

For

ForAll

Function

If...Then...Else...EndIf

Property Get

Property Set

Select Case

Sub

Type

While

Terminate the block with the appropriate statement.

{button ,AL(`H_IDE_STR_UNTERMBLK_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Unterminated string constant

You omitted the double quotation mark that signals the end of a quoted literal on a single line. Double quotation marks must be paired on the same line. For example:

```
Print "Hi,        ' Illegal because end quotation mark is missing
Martin."

Print "Hi, " _    ' Legal because string is properly quoted
"Martin."         ' Legal because string is properly quoted and
                  ' preceded by line-continuation character
' Output: Hi, Martin.
```

Terminate the string with double quotation marks on the same line where it starts.

{button ,AL(`H_IDE_STR_UNTERMSCONST_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Unterminated square bracket reference

A square bracket reference was not terminated by a close square bracket (]) on the same line. Square brackets are used in some cases when referring to the names of product objects.

Terminate the square bracket reference with a close square bracket on the same line. Make sure that the product you are using supports square bracket notation for references.

{button ,AL(`H_IDE_STR_UNTERMSQB_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Illegal character after %INCLUDE directive

The %Include directive is followed on the same line by something other than the name of the file to include. The name of the file to include must be the only thing following %Include on a line, except for an optional comment, beginning with the comment character.

Remove everything following %Include and the name of the file, or insert a comment character after it to comment out the rest of the line.

{button ,AL(`H_IDE_STR_ILLINCLUDE_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

Illegal executable code in Options

In (Options), you included a statement that is not allowed in the section.

Only the following are allowed in (Options): Option, Def*type*, Use, and UseLSX statements and Const statements associated with Use and UseLSX.

Remove the invalid statements from (Options).

Illegal on declarations in this scope: <keyword>

The following conditions could have caused this error:

- You used the keyword Dim, Public, Private, or Static when defining a member variable in a Type statement. For example:

```
Type MyType
  Public X As Integer       'Illegal: Public keyword not allowed here.
End Type
```

  Remove the Dim, Public, Private, or Static keyword.

- You used the Dim keyword when defining a member variable in a Class statement. For example:

```
Class MyClass
  Dim X As Integer      ' Illegal: Dim keyword not allowed here.
End Class
```

Remove the Dim keyword.

{button ,AL(`H_IDE_STR_ILLDECLSCOPE_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;', 0)} See related topics

Name too long

The specified name is too long (it is truncated in the error message). The maximum length of a LotusScript name is 40 characters.

Shorten the name to 40 or fewer characters.

{button ,AL(`H_IDE_STR_NAMETOOLONG_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} See related topics

Statement illegal in CLASS block: <keyword>

You used an illegal statement in a Class...End Class block.

The only legal statements in a Class. ..End Class block are:

- Declarations of variables without the keyword Dim or Static

  A variable may be declared Public or Private, or with no leading keyword.
- Definitions without the keyword Static
- Definitions of the constructor and destructor subs (Sub New and Sub Delete) for the class
- The Rem statement
- The directives %Rem...%End Rem and %Include

By extension, when you use the %Include directive in a Class...End Class block, the file to which it refers must not contain any statements that are illegal inside a Class...End Class block.

Remove the illegal statement from the Class...End Class block.

{button ,AL(`H_IDE_STR_ILLCLSTMT_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

Statement illegal in TYPE block: <keyword>

You used an illegal statement in a Type...End Type block. The only legal statements in a Type...End Type block are declarations of variables without the leading keyword Dim, Public, Private, or Static; the Rem statement; and the directives %Rem...%End Rem and %Include. All other statements are illegal.

By extension, when you use the %Include directive in a Type...End Type block, the file to which it refers must not contain any statements that are illegal inside a Type...End Type block.

Remove the statement from the Type...End Type block.

{button ,AL(`H_IDE_STR_ILLTYSTMT_REF_RT;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)}
See related topics

Überblick: LotusScript IDE

Das LotusScript Integrated Development Environment (IDE) besteht aus einer Reihe von Tools, mit denen Sie Scripts in Lotus-Anwendungen erstellen und debuggen können.

- Mit dem Script-Editor können Sie Scripts schreiben und deren Syntax prüfen. Außerdem können Sie Breakpoints setzen, löschen, aktivieren und deaktivieren, die beim Debuggen von Scripts verwendet werden.
- Mit dem Script-Debugger können Sie Breakpoints setzen, löschen, aktivieren und deaktivieren, und Sie können Schritt für Schritt durch Scripts gehen, um festzustellen, wo das Problem bei der Ausführung von Scripts liegt.
- Mit den Feldern Breakpoints, Browser, Ausgabe und Variablen können Sie Scripts erstellen und debuggen:
  - Im Feld Breakpoints werden Breakpoints aufgeführt, die in Scripts gesetzt wurden. In diesem Feld können Sie zu Breakpoints in Scripts gehen. Außerdem können Sie Breakpoints löschen, aktivieren und deaktivieren.
  - Im Feld Browser werden LotusScript-Schlüsselwörter, Komponenten von Anwendungs-Scripts sowie Typ-Bibliotheken und -Klassen für OLE Automation-Objekte aufgeführt. In diesem Feld können Sie auch Schlüsselwörter, Komponentennamen und OLE Application-Class-Kennungen in ein Script einfügen.
  - Im Feld Ausgabe wird die Ausgabe angezeigt, die von LotusScript Drucken-Befehlen generiert wird, die in Scripts enthalten sind.
  - Beim Debuggen werden in dem Feld Variablen Informationen über Variablen in einem Script angezeigt. In diesem Feld können Sie auch Variablenwerte ändern.

{button ,AL(`H_IDE_THE_LOTUSSCRIPT_IDE_OVER_RT;H_IDE_THE_BREAKPOINTS_BROWSER_OUTPUT_AND_VARIABLES_PANELS_OVER;H_IDE_THE_SCRIPT_DEBUGGER_OVER;H_IDE_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER;H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER;',0)} Siehe Verwandte Themen

Überblick: Im LotusScript IDE-Fenster arbeiten

Jedes Dokument in einer Anwendung verfügt über ein eigenes IDE-Fenster. Wenn das Dokument aktiv ist, können Sie dessen IDE-Fenster öffnen und Scripts für Dokumentobjekte anzeigen, schreiben, kompilieren und debuggen. Sie müssen ein anderes Dokument aktivieren und dessen IDE-Fenster öffnen, wenn Sie mit Scripts in einem anderen Dokument arbeiten möchten.

Wenn Sie das IDE öffnen, werden der Script-Editor und die Felder Breakpoints, Browser und Ausgabe in separaten Ausschnitten des Fensters angezeigt. Beim Debuggen eines Scripts werden die Felder Script-Debugger und Breakpoints, Browser, Ausgabe und Variablen angezeigt.

Sie können die Ausschnitte nacheinander anzeigen, indem Sie den Teiler ziehen, der die Ausschnitte teilt.

Wenn Sie das IDE schließen oder deaktivieren, werden %Rem-Blöcke, mehrzeilige Strings, Objekt-Subroutinen, Funktionen und Eigenschaftenblöcke sowie benutzerdefinierte Typen und Klassenblöcke automatisch abgeschlossen. Außerdem werden Scripts neu formatiert, um Zeilen einzupassen, die nicht folgerichtig eingegeben oder nicht richtig eingerückt wurden. Schließlich werden alle Scripts für das Dokument gespeichert.

Wenn Sie versuchen, das IDE zu schließen, während der Script-Debugger noch ausgeführt wird, wird mit einer Meldung angegeben, daß ein Script ausgeführt wird, und Sie werden gefragt, ob Sie das Script stoppen möchten.

{button ,AL(`H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER_RT;H_IDE_CLOSING_THE_IDE_WINDOW_STEPS;H_IDE_CLOSING_THE_SCRIPT_DEBUGGER_OVER;H_IDE_SHOWING_AND_HIDING_PANES_IN_THE_IDE_WINDOW_STEPS;H_IDE_SWITCHING_BETWEEN_IDE_WINDOWS_STEPS;H_IDE_SWITCHING_BETWEEN_PANES_IN_THE_IDE_WINDOW_STEPS;',0)} Siehe Verwandte Themen

IDE-Fenster schließen

Doppelklicken Sie auf das Systemmenüfeld in der oberen linken Ecke des IDE-Fensters, um das IDE zu schließen.

Hinweis Das IDE wird automatisch geschlossen, wenn Sie das entsprechende Dokument schließen.

{button ,AL(`H_IDE_CLOSING_THE_IDE_WINDOW_STEPS_RT;H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER;H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER;',0)} Siehe Verwandte Themen

Ausschnitte im IDE-Fenster anzeigen und verbergen

Ziehen Sie den Ausschnitt-Teiler im Fenster ganz nach oben oder unten, um Ausschnitte im IDE-Fenster anzuzeigen und zu verbergen. Wenn Sie z. B. nur den Script-Editor anzeigen möchten, ziehen Sie den Ausschnitt-Teiler nach unten, bis der Editor das ganze Fenster belegt.

Umschalten zwischen verborgenen und sichtbaren Ausschnitten
Drücken Sie F6.

Wiederherstellen der vorgegebenen Teilung
Klicken Sie bei gedrückter Strg-Taste auf den Ausschnitt-Teiler.

Zwischen einfachen und doppelten Ausschnitten umschalten
Doppelklicken Sie auf den Ausschnitt-Teiler.

{button ,AL(`H_IDE_SHOWING_AND_HIDING_PANES_IN_THE_IDE_WINDOW_STEPS_RT;H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER;H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER;',0)} Siehe Verwandte Themen

Zwischen Ausschnitten im IDE-Fenster wechseln

Wenn Sie zwischen dem Script-Editor oder Script-Debugger und den Feldern Breakpoints, Browser, Ausgabe und Variablen wechseln möchten, drücken Sie F6 oder verwenden Sie die Menübefehle Script und Debug. Der Ausschnitt bzw. das Feld, zu dem Sie wechseln, wird zum aktiven Ausschnitt oder Feld im IDE-Fenster.

Mit F6 zu den Feldern Breakpoints, Browser, Ausgabe und Variablen wechseln

1. Im Script-Editor oder Script-Debugger drücken Sie F6.

   Damit wird zu dem Feld gegangen, das als letztes den Fokus hatte.

2. Klicken Sie auf das Feld Register, oder drücken Sie Umschalttaste+Strg+→ bzw. Umschalttaste+Strg+←, um einem anderen Feld den Fokus zu geben.

Mit Menübefehlen zu den Feldern Breakpoints, Browser, Ausgabe und Variablen wechseln

1. Im Menü Script wählen Sie Breakpoints, Browser, Ausgabe oder Variablen.

   Damit erhält das angegebene Feld den Fokus.

2. Klicken Sie auf das Feld Register, oder drücken Sie Umschalttaste+Strg+→ bzw. Umschalttaste+Strg+←, um einem anderen Feld den Fokus zu geben.

Von einem Feld zum Script-Editor oder Script-Debugger wechseln

- Drücken Sie F6.
- Klicken Sie in Script-Editor oder Script-Debugger.
- Wählen Sie im Menü Script oder Debug einen Befehl, der im Script-Editor oder Script-Debugger ausgeführt wird.

{button ,AL(`H_IDE_SWITCHING_BETWEEN_PANES_IN_THE_IDE_WINDOW_STEPS_RT;H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER;H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER;',0)} Siehe Verwandte Themen

Zwischen IDE-Fenstern wechseln

Klicken Sie auf ein Fenster, um dem Fenster den Fokus zu geben, oder wählen Sie Nächstes im Systemmenü des jeweiligen Fensters.

Wenn ein IDE-Fenster den Fokus verliert, werden %Rem-Blöcke, mehrzeilige Strings, Objekt-Subroutinen, Funktionen und Eigenschaftenblöcke sowie benutzerdefinierte Typen und Klassenblöcke automatisch abgeschlossen. Außerdem werden Scripts neu formatiert, um Zeilen einzupassen, die nicht folgerichtig eingegeben wurden oder falsch eingerückt sind. Schließlich werden alle Scripts für das Dokument gespeichert.

{button ,AL(`H_IDE_SWITCHING_BETWEEN_IDE_WINDOWS_STEPS_RT;H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER;H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER;',0)} Siehe Verwandte Themen

Überblick: Tastatur und Maus im IDE verwenden

Mit der Tastatur können Sie Befehle ausführen, zwischen Ausschnitten und Feldern im IDE-Fenster wechseln bzw. durch diese gehen und Text in editierbaren Bereichen der Ausschnitte und Felder markieren. Mit der Maus können Sie schnell zu einem beliebigen Teil im IDE-Fenster gehen, Ausschnitte im IDE-Fenster anzeigen oder verbergen, editierbare Elemente markieren, Gliederungslisten einblenden oder ausblenden, Breakpoints verwalten und Schnellmenüs für einzelne Bereiche im Fenster anzeigen.

Allgemeine Verwendungszwecke der Tastatur

- Drücken Sie F6, um zwischen dem Script-Editor oder Script-Debugger und dem als letztes gewählten Feld (Breakpoint, Browser, Ausgabe oder Variablen) zu wechseln.
- Drücken Sie F1, um einen Hilfetext für den aktiven Bereich des IDE oder für ein Script-Schlüsselwort aufzurufen, das Sie im Script-Editor oder Browser wählen.
- Wenn Sie ein Element in einer Gliederungsliste einblenden oder ausblenden möchten, markieren Sie das Element und drücken + (plus) oder - (minus).
- Wenn Sie zwischen den Feldern in einem Ausschnitt hin- und hergehen möchten, drücken Sie die Tabulatortaste, um ein Feld nach vorne zu gehen, bzw. die Umschalttaste+Tabulatortaste, um ein Feld zurückzugehen.
- Wenn Sie die Einfügestelle im Script-Editor zur vorherigen oder nächsten Prozedur versetzen möchten, die ein benutzerdefiniertes Script enthält, drücken Sie Strg+Bild Unten oder Strg+Bild Oben. Die Einfügestelle wird zuerst innerhalb der Scripts eines Objekts und dann von einer Gruppe mit Objektscripts zur nächsten bewegt.

Allgemeine Verwendungszwecke der Maus

- Klicken Sie auf das Symbol Einblenden oder Ausblenden, um Listenelemente ein- oder auszublenden.
- Klicken Sie neben der Zeile in der Breakpoint-Spalte, um einen Breakpoint in einer Zeile zu setzen.
- Klicken Sie auf den Breakpoint in der Spalte, um einen Breakpoint zu löschen.
- Klicken Sie bei gedrückter Strg-Taste auf den Breakpoint, um einen Breakpoint zu deaktivieren oder zu aktivieren.

{button ,AL(`H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER_RT;H_IDE_USING_THE_KEYBOARD_IN_THE_PANELS_STEPS;H_IDE_USING_THE_KEYBOARD_IN_THE_SCRIPT_EDITOR_AND_SCRIPT_DEBUGGER_STEPS;H_IDE_USING_THE_MOUSE_IN_THE_IDE_STEPS;H_IDE_WORKING_WITH_OUTLINE_LISTS_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

Verwendung der Tastatur im Script-Editor und Script-Debugger

Ausführung von Befehlen in den Menüs Script und Debug
Verwenden Sie die Abkürzungstasten, die in den Menüs aufgeführt werden.

Verschieben der Einfügestelle

→ oder ←	Verschiebt die Einfügestelle um ein Zeichen nach rechts oder links.
oder ↓	Verschiebt die Einfügestelle um eine Zeile nach oben oder unten.
Pos1	Setzt die Einfügestelle an den Anfang einer Zeile.
Ende	Setzt die Einfügestelle an das Ende einer Zeile.
Bild Oben oder Bild Unten	Setzt die Einfügestelle an den Anfang oder das Ende eines Script-Blocks.
Strg+→	Setzt die Einfügestelle zum nächsten Wort.
Strg+←	Setzt die Einfügestelle zum vorherigen Wort.
Strg+	Setzt die Einfügestelle an den Anfang der aktuellen Prozedur.
Strg+↓	Setzt die Einfügestelle an das Ende der aktuellen Prozedur.
Strg+Pos1	Setzt die Einfügestelle an den Anfang der aktuellen Prozedur.
Strg+Ende	Setzt die Einfügestelle an das Ende der aktuellen Prozedur.
Strg+Bild Oben	Setzt die Einfügestelle zur vorherigen Prozedur, die ein benutzerdefiniertes Script enthält.
Strg+Bild Unten	Setzt die Einfügestelle zur nächsten Prozedur, die ein benutzerdefiniertes Script enthält.

{button ,AL(`H_IDE_USING_THE_KEYBOARD_IN_THE_SCRIPT_EDITOR_AND_SCRIPT_DEBUGGER_STEPS_RT;H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER;H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER;',0)} Siehe Verwandte Themen

Verwendung der Tastatur in den Feldern Breakpoints, Browser, Ausgabe und Variablen

Einen Breakpoint aus dem Feld Breakpoints löschen
Wählen Sie das Breakpoint-Element, und drücken Sie Entf.

An Werten im Feld Variablen vorgenommene Änderungen rückgängig machen
Drücken Sie Esc.

Zum nächsten Feld gehen
Drücken Sie Umschalttaste+Strg+→ bzw. Umschalttaste+Strg+←.

{button ,AL(`H_IDE_USING_THE_KEYBOARD_IN_THE_PANELS_STEPS_RT;H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER;H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER;',0)}
Siehe Verwandte Themen

Verwendung der Maus im IDE

Einfügestelle verschieben
Klicken Sie auf das editierbare Feld, in das Sie die Einfügestelle verschieben möchten.

Text in einem Script markieren
- Doppelklicken Sie auf das Wort, um es zu markieren.
- Drücken Sie die Umschalttaste und klicken Sie, um Text ab der Einfügestelle bis zum Mauszeiger zu markieren.
- Klicken Sie auf den Anfang des Blocks und ziehen Sie die Hervorhebung bis zum Ende des Blocks, um einen Script-Block zu markieren.

Elemente in einer Gliederungsliste anzeigen
- Klicken Sie auf das Symbol Einblenden, um ein Element einzublenden.
- Klicken Sie auf das Symbol Ausblenden, um ein Element auszublenden.

Schnellmenüs anzeigen
- Klicken Sie mit der rechten Maustaste im Script-Editor oder Script-Debugger auf die Breakpoint-Spalte oder den Script-Bereich.
- Klicken Sie mit der rechten Maustaste auf das Feld Breakpoints oder Ausgabe bzw. auf das Textfeld Werte im Feld Variablen.

Breakpoints aus dem Script-Editor und Script-Debugger steuern
- Klicken Sie neben eine Zeile in der Spalte, um einen Breakpoint in dieser Zeile zu setzen.
- Klicken Sie auf einen Breakpoint, um diesen zu löschen.
- Klicken Sie bei gedrückter Strg-Taste auf einen Breakpoint, um diesen zu deaktivieren bzw. zu aktivieren.

Breakpoints aus dem Feld Breakpoints steuern
- Wenn Sie im Ausschnitt Script-Editor oder Script-Debugger einen Bildlauf zu einem Breakpoint vornehmen und den Breakpoint zum aktuellen Breakpoint machen möchten, ohne den Ausschnitt zu aktivieren, klicken Sie auf dieses Breakpoint-Element in Breakpoints.
- Wenn Sie im Ausschnitt Script-Editor oder Script-Debugger einen Bildlauf zu einem Breakpoint vornehmen möchten, machen Sie den Breakpoint zum aktuellen Breakpoint, aktivieren den Ausschnitt und doppelklicken in Breakpoints auf dieses Breakpoint-Element. Die Einfügestelle geht zum Anfang der Breakpoint-Zeile.

{button ,AL(`H_IDE_USING_THE_MOUSE_IN_THE_IDE_STEPS_RT;H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER;H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER;',0)} Siehe Verwandte Themen

Überblick: Mit Gliederungslisten im IDE arbeiten

Gliederungslisten enthalten Elemente, die ein- oder ausgeblendet werden können, um weitere Elementebenen anzuzeigen. Im IDE gibt es verschiedene Gliederungslisten, einschließlich der LotusScript Sprache-Liste im Browser. Diese Liste enthält task-orientierte Kategorien, die Sie einblenden können, um Schlüsselwörter oder andere Unterkategorien anzuzeigen.

Elemente, die Sie einblenden können, werden mit dem Symbol Einblenden gekennzeichnet. Elemente, die Sie ausblenden können, werden mit dem Symbol Ausblenden gekennzeichnet.

Elemente einblenden

- Gehen Sie folgendermaßen vor, um ein Element einzublenden:
  - Klicken Sie auf das Symbol Einblenden für das Element.
  - Markieren Sie das Element, und drücken Sie Return.
  - Markieren Sie das Element, und drücken Sie + (plus).
- Drücken Sie gleichzeitig die Umschalttaste und + (plus), um alle Ebenen im Feld Browser oder die nächste Ebene weiter unten im Feld Variablen einzublenden.

Elemente ausblenden

- Gehen Sie folgendermaßen vor, um ein Element auszublenden:
  - Klicken Sie auf das Symbol Ausblenden für das Element.
  - Klicken Sie auf das Element, und drücken Sie Return.
  - Klicken Sie auf das Element, und drücken Sie - (minus).
- Drücken Sie gleichzeitig die Umschalttaste und - (minus), um alle Elemente auszublenden.

{button ,AL(`H_IDE_WORKING_WITH_OUTLINE_LISTS_IN_THE_IDE_OVER_RT;H_IDE_USING_THE_KEYBOARD_AND_MOUSE_IN_THE_IDE_OVER;H_IDE_WORKING_IN_THE_LOTUSSCRIPT_IDE_WINDOW_OVER;',0)}
Siehe Verwandte Themen

Überblick: Script-Editor

Im IDE Script-Editor können Sie Scripts schreiben und bearbeiten, die Syntax von Scripts überprüfen und Breakpoints für das Debuggen von Scripts setzen. Wenn der Script-Editor geöffnet ist:

- Wird das Menü Script im Hauptmenü angezeigt
- Kann auf die Felder Browser, Ausgabe und Breakpoints zugegriffen werden
- Wird ein Script im Script-Arbeitsbereich angezeigt

Benutzung des Menüs Script

Das Menü Script enthält Befehle, mit denen Sie im IDE

- Neue Subroutinen und Funktionen erstellen können
- Subroutinen und Scripts kompilieren und testen können
- Auf die Felder Breakpoints, Browser, Ausgabe und Variablen zugreifen können

Script-Editor aus dem Hauptmenü öffnen

Wenn der Editor aus dem Hauptmenü geöffnet wird, wird das zuletzt bearbeitete Script angezeigt:

- Wenn kein Script bearbeitet wurde, zeigt der Editor die Subroutine für das Vorgabe-Event des aktuellen Objekts an.
- Wenn kein Vorgabe-Event definiert ist, zeigt der Editor (Declarations) für das Objekt an.

Script-Editor aus einem Objekt öffnen

Wenn der Editor aus der InfoBox eines Objekts oder mit einem Schnellmenü geöffnet wird, wird eines der folgenden Scripts angezeigt:

- Wenn das Objekt während der Programmierung gewählt wurde, zeigt der Editor das letzte für dieses Objekt gewählte Script an, unabhängig davon, ob das Script während der Programmierung bearbeitet wurde oder nicht.
- Wenn das Objekt während der Programmierung nicht gewählt wurde, jedoch ein benutzerdefiniertes Script enthält, zeigt der Editor (für das Objekt) die erste Subroutine, Funktion oder eine andere Gruppe mit Anweisungen an, die ein benutzerdefiniertes Script enthält.
- Wenn das Objekt nicht gewählt wurde und kein benutzerdefiniertes Script enthält, zeigt der Editor die Subroutine für das Vorgabe-Event des Objekts an.
- Wenn für das Objekt kein Vorgabe-Event definiert ist, zeigt der Editor (Declarations) für das Objekt an.

{button ,AL(`H_IDE_THE_SCRIPT_EDITOR_OVER_RT;H_IDE_COMPILING_AND_TESTING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_MANAGING_TEXT_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_SETTING_SCRIPT_PROPERTIES_FOR_THE_IDE_OVER;H_IDE_WORKING_WITH_GLOBALS_AND_OBJECTS_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

Überblick: Script-Eigenschaften für das IDE festlegen

Im Script-Editor oder Script-Debugger können Sie Tabulatorstopps setzen, die Funktion Smart Indenting aktivieren und deaktivieren und Farbe bzw. Schriftart für den Text ändern. Sie können auch ein Meldungsfeld aktivieren und deaktivieren, in dem die Anzahl von Fehlern angezeigt wird, die bei der Kompilierung des Scripts aufgetreten sind.

Wählen Sie Script-Editor Vorgaben im Menü Datei und ändern die Einstellungen in dem Dialogfeld, um diese Script-Eigenschaften festzulegen.

Tabulatorstopps und Smart Indenting festlegen

Mit der Einstellung für den Tabulatorstopp wird festgelegt, wie breit ein Tabulator (in Leerschritten) ist. Mit dem Smart Indenting werden Anweisungen innerhalb eines Script-Blocks um einen Tabulatorstopp eingerückt.

Anzahl von Kompilierungsfehlern anzeigen

Vorgabegemäß wird die Anzahl von Fehlern bei der Kompilierung in einem Meldungsfeld angezeigt. Wenn Sie z. B. einen der Befehle Scripts durchsuchen im Menü Script wählen oder wenn ein Event-Handler ausgeführt wird, wird die Anzahl von Fehlern für das Objekt oder Dokument angezeigt. Sie können dieses Meldungsfeld deaktivieren, so daß kein Fehlerzähler angezeigt wird.

Farben für Script-Text festlegen

Vorgabegemäß werden Kommentare, Schlüsselwörter, Anweisungen, Namen und Text, der Fehler enthält, im IDE andersfarbig angezeigt. So werden z. B. Schlüsselwörter blau und Text mit Fehlern rot angezeigt. Sie können diese Vorgabefarben ändern.

Schriftarten für Script-Text festlegen

Die vorgegebene Schriftart für Text im IDE ist ein schwarzer System-Font auf weißem Hintergrund mit 9 Punkten. Sie können Schriftart und Schriftgröße ändern.

{button ,AL(`H_IDE_SETTING_SCRIPT_PROPERTIES_FOR_THE_IDE_OVER_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_FORMATTING_CODE_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_MANAGING_TEXT_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Überblick: Mit (Globals) und Objekten im IDE arbeiten

In dem Drop-down-Listenfeld Objekt im Script-Editor werden die Namen von Objekten im aktuellen Dokument aufgeführt, mit denen Sie Scripts verknüpfen können. Sie wählen ein Objekt, wenn Sie Scripts für Objekt-Events eingeben möchten. Nachdem Sie das Objekt gewählt haben, geben Sie Deklarationen in (Declarations) ein und definieren die Prozeduren, die von den Events benutzt werden. Sie können auch Variablen angeben, die in (Declarations) in der Subroutine Initialize des Objekts deklariert werden, die Variablen in der Subroutine Terminate löschen und Optionen für die Objekt-Scripts in (Options) eingeben.

Wählen Sie (Globals) im Drop-down-Listenfeld Objekt, wenn Sie Deklarationen und Prozeduren eingeben möchten, die von allen Scripts in Ihrem Dokument benutzt werden. Sie können Variablen angeben, die in (Declarations) in der Subroutine Initialize in (Globals) deklariert werden, können Variablen in der Subroutine Terminate löschen und können Optionen für Ihre globalen Scripts in (Options) eingeben.

Objekt-Scripts anzeigen

Wenn Sie (Globals) wählen, wird folgendes angezeigt:

- Das letzte (Globals)-Script, auf das Sie zugegriffen haben, oder
- (Globals)-Deklarationen

Wenn Sie ein Objekt wählen, wird folgendes angezeigt:

- Das letzte Objekt-Script, auf das Sie zugegriffen haben
- Die Subroutine für das Vorgabe-Event des Objekts oder
- Objekt-Deklarationen

Wenn das Symbol Einblenden für ein Objekt in der Liste angezeigt wird, können Sie auf das Symbol klicken, um eine Gruppe mit verwandten Objekten anzuzeigen. Wenn es sich bei dem einblendbaren Element ebenfalls um ein Objekt mit Scripts handelt, können Sie auf das Element klicken, um dessen Scripts anzuzeigen.

Objekte umbenennen

Sie können ein Produktobjekt im IDE nicht umbenennen. Wenn das Objekt durch das Produkt umbenannt wird, wird den Scripts des Objekts automatisch der neue Name zugeordnet. Sie müssen jedoch Referenzen auf den alten Namen überall dort ändern, wo dieser im Script-Text auftaucht.

{button ,AL(`H_IDE_WORKING_WITH_GLOBALS_AND_OBJECTS_IN_THE_IDE_OVER_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_GLOBALS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_SELECTING_AN_OBJECT_TO_WORK_WITH_IN_THE_SCRIPT_EDITOR_STEPS;',0)} Siehe Verwandte Themen

Objekte zur Bearbeitung im Script-Editor wählen

Klicken Sie auf ein Element im Drop-down-Listenfeld Objekt.

Wenn neben einem Element das Symbol Einblenden angezeigt wird, klicken Sie auf das Symbol, um eine Gruppe von zugehörigen Objekten anzuzeigen; danach klicken Sie auf eines der Objekte.

Eingeben oder Ändern von globalen (Declarations) oder Prozeduren bzw. Eingeben oder Ändern von (Options) für Ihre globalen Scripts

Klicken Sie auf (Globals).

Eingeben oder Ändern von (Declarations), (Options) oder Prozeduren für ein Objekt

Klicken Sie auf den Objektnamen.

{button ,AL(`H_IDE_SELECTING_AN_OBJECT_TO_WORK_WITH_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_WORKING_WITH_GLOBALS_AND_OBJECTS_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

(Globals) im Script-Editor eingeben

1. Klicken Sie auf (Globals) im Drop-down-Listenfeld Objekt.
2. In einem beliebigen Script für (Globals) geben Sie Optionen, Deklarationen sowie Subroutinen, Funktionen und andere Prozeduren ein.

   Option-, Def*typ*-, Use- und UseLSX-Anweisungen werden in (Options) verschoben.

   Folgendes wird in (Declarations) verschoben: Dim-Anweisungen, die nicht ausdrücklich in Sub...End Sub-Anweisungen eingegeben werden; Public-, Private-, Type-, Class- und Declare Lib-Anweisungen sowie Const-Anweisungen, die nicht ausdrücklich in (Options) eingegeben werden.

   Sub-, Function-, Property Set- und Property Get-Anweisungen, die nicht in Class...End Class-Anweisungen eingegeben werden, werden in ihren eigenen Abschnitt verschoben, und ihre Namen werden zu der Script-Liste für (Globals) hinzugefügt.

{button ,AL(`H_IDE_ENTERING_GLOBALS_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_WORKING_WITH_GLOBALS_AND_OBJECTS_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

Überblick: Scripts im Script-Editor erstellen

Im Drop-down-Listenfeld Script im Script-Editor werden die Optionen, Deklarationen und Subroutinen aufgeführt, die mit dem Objekt verknüpft sind, das im Drop-down-Listenfeld Objekt gewählt wurde.

Vorgabegemäß sind (Options) und (Declarations) sowie die Event-Subroutinen eines Objekts leer. (Globals) (Options) ist eine Ausnahme; dort wird vorgabegemäß die Option Public-Anweisung angezeigt, mit der alle (Declarations) und Prozeduren, die Sie für (Globals) erstellen, allen Scripts im Dokument zur Verfügung gestellt werden.

Wenn Sie eine neue Subroutine, Funktion oder eine andere Prozedur erstellen, fügt das IDE deren Namen in die Liste mit benutzerdefinierten Prozeduren am Ende der Script-Liste ein.

Fett und kursiv geschriebene Listenelemente wählen

Elemente im Drop-down-Listenfeld Script werden in verschiedenen Schriftarten angezeigt, damit Sie die Auswahl schneller treffen können:

- Vordefinierte Elemente, einschließlich (Options), (Declarations) sowie Initialize-, Terminate- und Event-Subroutinen eines Objekts werden mit einer normalen Schriftart angezeigt. Diese Elemente werden fett angezeigt, nachdem sie bearbeitet wurden.
- Von Ihnen erstellte Prozeduren werden fett und kursiv geschrieben.

Anweisungen für Objekte und (Globals) eingeben

(Options)

Setzen Sie Option-, Def*typ*-, Use- und UseLSX-Anweisungen sowie Const-Anweisungen, die mit Use und UseLSX verknüpft sind, in (Options).

(Declarations)

Setzen Sie Private-, Type-, Class- und Declare- (externe C-Calls)-Anweisungen sowie Const-Anweisungen, die nicht mit Use- oder UseLSX-Anweisungen verknüpft sind, in (Declarations). Setzen Sie Public-Anweisungen in (Globals) (Declarations). Setzen Sie Dim-Anweisungen auf Modulebene in (Declarations).

Subroutinen Initialize und Terminate

Verwenden Sie die Subroutine (Globals) Initialize, um Daten anzugeben, die in (Globals) (Declarations) definiert sind. Diese Subroutine wird ausgeführt, bevor auf andere Prozeduren in (Globals) und auf Variablen in (Globals) (Declarations) zugegriffen wird.

Verwenden Sie die Subroutine Initialize eines Objekts, um Variablen einzurichten, die in (Declarations) des Objekts definiert sind. Diese Subroutine wird ausgeführt, bevor die Objekt-Events ausgeführt werden.

Gleichermaßen verwenden Sie die Subroutinen (Globals) und Terminate für ein Objekt, um Variablen zu löschen, die in den entsprechenden (Declarations) definiert sind. Die Subroutine Terminate wird ausgeführt, wenn ein Script entfernt wird. Ein Objekt-Script wird entfernt, wenn es bearbeitet bzw. wenn (Globals) bearbeitet wird. Ein (Globals)-Script wird entfernt, wenn es bearbeitet wird. Scripts werden auch entfernt, wenn die Anwendung, in der sie ausgeführt werden, geschlossen wird.

Subroutine Event

Setzen Sie Anweisungen, die ausgeführt werden sollen, wenn das Event für das Objekt gestartet wird, in eine Event-Subroutine für ein Objekt.

Prozedur

Setzen Sie Anweisungen, die ausgeführt werden sollen, wenn die Prozedur aus einem anderen Script aufgerufen wird, in eine Subroutine, Funktion oder eine andere Prozedur, die Sie erstellen.

{button ,AL(`H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER_RT;H_IDE_CREATING_DECLARATIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_OPTIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_BROWSER_PANEL_OVER;H_IDE_WORKING_WITH_DIRECTIVES_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Kommentare zu einem Script im Script-Editor hinzufügen

Ein einzeiliger Kommentar wird wie folgt eingegeben

- Geben Sie ' (Apostroph) am Anfang einer Zeile ein. Danach geben Sie die Kommentarzeile ein.
- Geben Sie Rem am Anfang einer Zeile ein; danach geben Sie den Kommentartext ein.

Ein Kommentar wird wie folgt zu einer Scriptzeile hinzugefügt

Am Ende einer Zeile:

- Fügen Sie eine Rem-Anweisung hinzu.
- Geben Sie ' (Apostroph) ein und geben dann den Kommentar ein.

Gehen Sie folgendermaßen vor, um einen Block mit Kommentarzeilen einzugeben

Verwenden Sie die %Rem...%End Rem-Anweisungen:

1. Geben Sie %Rem am Anfang einer Zeile ein.
2. Geben Sie die Zeilen mit Kommentartext ein.
3. Am Anfang einer neuen Zeile geben Sie %End Rem ein.

Scripts im Script-Editor formatieren

Script-Zeilen automatisch einrücken
Mit dem Befehl Script-Editor Vorgaben im Menü Datei legen Sie die Tabulatorbreite fest und aktivieren die Funktion Smart Indenting. Wenn Sie dann Script-Zeilen eingeben, werden diese automatisch eingerückt.

Script-Blöcke automatisch erstellen
Geben Sie eine gültige Anfangs-Anweisung für eine Blockstruktur ein, und drücken Sie Return. Die entsprechende End-Anweisung wird automatisch auf einer Folgezeile eingefügt.

Anfang- und End-Anweisungen werden unten aufgeführt:

Sub	End Sub
Function	End Function
Property Get	End Property
Property Set	End Property
Do	Loop
For	Next
ForAll	End ForAll
While	Wend
If	End If
Select	End Select
With	End With
Type	End Type
Class	End Class

Wenn Sie das IDE schließen, den Fokus vom IDE entfernen oder die Script-Syntax prüfen, schließt das IDE auch nicht abgeschlossene %Rem-Blöcke, mehrzeilige Strings, Objekt-Subroutinen, Funktionen und Eigenschaftenblöcke sowie benutzerdefinierte Typen und Klassenblöcke ab. Außerdem werden Scripts neu formatiert, um außerhalb der Reihenfolge eingegebene oder nicht richtig eingerückte Zeilen einzupassen. Schließlich werden alle Scripts für das Dokument gespeichert.

{button ,AL(`H_IDE_FORMATTING_CODE_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_MANAGING_TEXT_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Überblick: (Declarations) im Script-Editor erstellen

Die folgenden Anweisungen werden in (Declarations) gesetzt: Dim-Anweisungen auf Modulebene, Private-, Type-, Class- und Declare (external C Calls)-Anweisungen sowie außerhalb von (Options) eingegebene Const-Anweisungen. Public-Anweisungen werden in (Globals) (Declarations) gesetzt.

Wenn Sie eine Type-, Class-, Declare (external C Calls)-, Dim-, Public- oder Private-Anweisung in ein Script für ein Objekt eingeben, setzt das IDE die Anweisung automatisch in (Declarations) für das Objekt. Das IDE setzt eine Const-Anweisung in (Declarations), wenn Sie die Const-Anweisung außerhalb von (Declarations) oder (Options) eingeben.

Das IDE deklariert auch Objekteigenschaften, Subroutinen und Funktionen für Sie.

{button ,AL(`H_IDE_CREATING_DECLARATIONS_IN_THE_SCRIPT_EDITOR_OVER_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_CLASS_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_ENTERING_DIM_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_ENTERING_TYPE_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS;',0)} Siehe Verwandte Themen

Type-Anweisungen in den Script-Editor eingeben

Geben Sie eine gültige Type-Anweisung in ein beliebiges Script für ein Objekt ein, bzw. fügen Sie die Type-Anweisung ein.

Das IDE formatiert die Type-Anweisung und setzt die Anweisung an das Ende von (Declarations) für das Objekt, wenn Sie die Anweisung außerhalb von (Declarations) eingeben.

Hinweis Sie müssen mindestens eine gültige Type-Anfangsanweisung eingeben. Das IDE generiert eine Type-Endanweisung, wenn Sie sie nicht angeben.

{button ,AL(`H_IDE_ENTERING_TYPE_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_DECLARATIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_OPTIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_VARIABLES_PANEL_OVER;',0)} Siehe Verwandte Themen

Dim-Anweisungen in den Script-Editor eingeben

Eine Variable für eine Subroutine wird folgendermaßen deklariert
Geben Sie eine gültige Dim-Anweisung in die Subroutine ein, bzw. fügen Sie sie ein.

Eine Variable für alle Scripts, die mit einem Objekt verknüpft sind, wird wie folgt deklariert
Geben Sie eine gültige Dim- oder Private-Anweisung in den Abschnitt (Declarations) für das Objekt ein.

Eine Variable für alle Scripts in einem Dokument wird folgendermaßen deklariert
1. Prüfen Sie, ob (Options) für (Globals) die Anweisung Option Public enthält.
2. Wählen Sie (Globals) und danach (Declarations).
3. Geben Sie eine gültige Dim-Anweisung in (Declarations) für (Globals) ein, bzw. fügen Sie sie ein.

{button ,AL(`H_IDE_ENTERING_DIM_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_DECLARATIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_OPTIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_VARIABLES_PANEL_OVER;',0)} Siehe Verwandte Themen

Class-Anweisungen in den Script-Editor eingeben

Geben Sie eine gültige Class-Anweisung in ein beliebiges Script für ein Objekt ein, bzw. fügen Sie sie ein.

Das IDE formatiert die Class-Anweisung und setzt sie an das Ende von (Declarations) für das Objekt, wenn Sie die Anweisung außerhalb von (Declarations) eingeben.

Hinweis Sie müssen mindestens eine gültige Class-Anfangsanweisung eingeben. Das IDE generiert eine Class-Endanweisung, wenn Sie sie nicht angeben.

{button ,AL(`H_IDE_ENTERING_CLASS_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_DECLARATIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_OPTIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_VARIABLES_PANEL_OVER;',0)} Siehe Verwandte Themen

Überblick: Options in den Script-Editor eingeben

Die folgenden Anweisungen werden in den Abschnitt (Options) gesetzt: Option, Def*typ*, Use und UseLSX sowie mit Use und UseLSX verknüpfte Const-Anweisungen.

Wenn Sie eine gültige Option-, Def*typ*-, Use- oder UseLSX-Anweisung in ein Script für ein Objekt (einschließlich (Globals)) eingeben, setzt das IDE die Anweisung automatisch in den Abschnitt (Options) des Objekts.

Wenn Sie eine Const-Anweisung in (Options) setzen möchten, müssen Sie sie explizit in diesen Abschnitt eingeben; sonst setzt das IDE die Const-Anweisung in (Declarations).

Die Option- und Def*typ*-Anweisungen, die Sie in (Globals) (Options) eingeben, wirken sich nur auf (Globals) Scripts aus; Option- und Def*typ*-Anweisungen, die Sie in den Abschnitt (Options) eines Objekts eingeben, wirken sich nur auf die Scripts dieses Objekts aus.

{button ,AL(`H_IDE_ENTERING_OPTIONS_IN_THE_SCRIPT_EDITOR_OVER_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_DEFTYPE_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_ENTERING_OPTION_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_ENTERING_OPTION_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS;',0)} Siehe Verwandte Themen

Option-Anweisungen in den Script-Editor eingeben

Geben Sie eine gültige Option Base-, Option Compare- oder Option Declare-Anweisung in einen beliebigen (Globals)-Abschnitt oder ein Objektscript ein, bzw. fügen Sie die Anweisungen ein.

Das IDE formatiert die Option-Anweisung und setzt sie an das Ende des Abschnitts (Options) für das Objekt.

Option Public

Das IDE nimmt die Option Public-Anweisung automatisch als erste Anweisung in (Options) für (Globals) auf. Mit dieser Anweisung werden alle (Declarations) und Prozeduren, die Sie für (Globals) erstellen, vorgabegemäß öffentlich.

Die Option Public-Anweisung ist im Abschnitt (Options) eines Objekts nicht zulässig.

{button ,AL(`H_IDE_ENTERING_OPTION_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_OPTIONS_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Deftyp-Anweisungen in den Script-Editor eingeben

Geben Sie eine gültige DefCur-, DefDbl- oder eine andere Def*typ*-Anweisung in ein beliebiges Script für ein Objekt ein, bzw. fügen Sie sie ein.

Das IDE formatiert die Anweisung und setzt sie an das Ende des Abschnitts (Options) des Objekts.

{button ,AL(`H_IDE_ENTERING_DEFTYPE_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_OPTIONS_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Use- und UseLSX-Anweisungen in den Script-Editor eingeben

Geben Sie eine gültige Use- oder UseLSX-Anweisung in ein beliebiges Script für ein Objekt ein, bzw. fügen Sie sie ein.

Das IDE formatiert die Anweisung und setzt sie an das Ende des Abschnitts (Options) des Objekts.

{button ,AL(`H_IDE_ENTERING_USE_AND_USELSX_STATEMENTS_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_ENTERING_OPTIONS_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Überblick: Prozeduren im Script-Editor erstellen

Das IDE stellt automatisch leere Sub-Anweisungsblöcke für die Subroutinen Initialize und Terminate bereit, die mit (Globals) und mit allen script-fähigen Objekten in einem Dokument verknüpft sind. Außerdem stellt das IDE leere Sub-Anweisungsblöcke für alle produkt-definierten Objekt-Events bereit, an die Sie Scripts anhängen können.

Diese Subroutinen werden in dem Drop-down-Listenfeld Script für ein Objekt aufgeführt. Die Event-Subroutinen des Objekts werden nach Event-Name aufgeführt. Der leere Sub-Anweisungsblock wird angezeigt, wenn Sie das Event aus der Liste wählen. Anweisungen, die Sie in eine Subroutine eingeben, werden ausgeführt, wenn das Event für das Objekt aktiviert wird.

Sie können auch Subroutinen, Funktionen und andere Prozeduren für Objekte erstellen, einschließlich (Globals). Die Namen der Prozeduren, die Sie erstellen, werden in die Liste mit benutzerdefinierten Prozeduren am Ende der Script-Liste eingefügt.

Eingabe eines Prozedur-Scripts

Sie können eine Prozedur einmal erstellen, indem Sie eine gültige Sub-, Function-, Property Set- oder Property Get-Anweisung in den Script-Bereich eingeben bzw. einfügen. Das IDE formatiert einen neuen Anweisungsblock und gibt die entsprechende Endanweisung ein. Danach können Sie Anweisungen eingeben, die in der Prozedur aufgenommen werden sollen.

Wenn Sie die Prozedur in (Globals) erstellen und die Anweisung Option Public nicht gelöscht haben, die das IDE in (Globals) (Options) setzt, ist Ihre neue Prozedur vorgabegemäß öffentlich.

Wenn Sie die Prozedur für ein Objekt erstellen, kann die Prozedur aus der Initialize- oder Terminate-Subroutine des Objekts, aus Objekt Event-Subroutinen und anderen Prozeduren aufgerufen werden, die Sie für das Objekt erstellen.

Hinweis Wenn Sie eine Sub-, Function-, Property Get- oder Property Set-Anweisung innerhalb einer Subroutine oder einer Funktion hinzufügen, die eine Klassenmethode oder -eigenschaft definiert, versetzt das IDE die Anweisung nicht, gibt jedoch mit einer Fehlermeldung an, daß die Prozedurdefinition in diesem Umfang unzulässig ist.

Verwendung von Neue Subroutine und Neue Funktion

Sie können eine neue Subroutine bzw. eine neue Funktion auch erstellen, indem Sie Neue Subroutine oder Neue Funktion aus dem Menü Script wählen. Im Dialogfeld Neue Subroutine oder Neue Funktion können Sie den Namen für die Subroutine oder Funktion eingeben und angeben, ob diese global sein soll. Vorgabegemäß wird die Subroutine oder Funktion mit dem aktuellen Objekt verknüpft.

Umbenennen einer bestehenden Event-Subroutine

Das IDE erstellt auch eine Subroutine für ein Objekt, wenn Sie versuchen, eine der Event-Subroutinen des Objekts umzubenennen. Da Event-Subroutinen nicht umbenannt werden können, erstellt das IDE eine benutzerdefinierte Subroutine mit dem von Ihnen angegebenen Namen, versetzt das Script aus der Event-Subroutine in die neue Subroutine und läßt die Event-Subroutine leer.

{button ,AL(`H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_RENAMING_FUNCTIONS_AND_SUBS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_WORKING_WITH_GLOBALS_AND_OBJECTS_IN_THE_IDE_OVER;H_IDE_WORKING_WITH_INITIALIZE_AND_TERMINATE_SUBS_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Funktionen und Subroutinen im Script-Editor umbenennen

Eine benutzerdefinierte Subroutine oder Funktion wird folgendermaßen umbenannt

Geben Sie in der Sub- oder Function-Anweisung den neuen Namen ein, bzw. fügen Sie den neuen Namen ein.

Hinweis Sie können Produkt-Event-Subroutinen nicht umbenennen. Wenn Sie versuchen, eine Event-Subroutine umzubenennen, erstellt das IDE eine neue benutzerdefinierte Subroutine mit dem von Ihnen angegebenen Namen und setzt das Script aus der Event-Subroutine in die neue Subroutine.

{button ,AL(`H_IDE_RENAMING_FUNCTIONS_AND_SUBS_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Überblick: Mit Initialize- und Terminate-Subroutinen im Script-Editor arbeiten

Die Initialize- und Terminate-Subroutinen sind im IDE für (Globals) und für alle scriptfähigen Produktobjekte in einem Dokument vordefiniert.

Anweisungen, die Sie in die Initialize-Subroutine von (Globals) eingeben, werden ausgeführt, bevor auf andere Prozeduren in (Globals) und auf Variablen in (Globals) (Declarations) zugegriffen wird. Anweisungen, die Sie in die Initialize-Subroutine eines Objekts eingeben, werden ausgeführt, bevor die Events des Objekts ausgeführt werden.

Anweisungen in Terminate-Subroutinen werden ausgeführt, wenn ein Script entfernt wird. Ein Objekt-Script wird entfernt, wenn das Script selbst oder wenn (Globals) bearbeitet wird. Ein (Globals)-Script wird entfernt, wenn es bearbeitet wird. Außerdem werden Scripts entfernt, wenn die Anwendung, in der sie ausgeführt werden, geschlossen wird.

Die Initialize- und Terminate-Subroutinen können beliebige LotusScript-Anweisungen enthalten, die in Unterprogrammen gültig sind. Ungültige Anweisungen werden als Kompilierfehler gemeldet.

Einschränkungen bei der Benutzung von Initialize und Terminate

- Es kann nur eine Initialize-Subroutine und eine Terminate-Subroutine pro Objekt vorhanden sein, einschließlich (Globals).
- Sie können Debug-Breakpoints in einer Initialize-Subroutine angeben, jedoch nicht in einer Terminate-Subroutine.

{button ,AL(`H_IDE_WORKING_WITH_INITIALIZE_AND_TERMINATE_SUBS_IN_THE_SCRIPT_EDITOR_OVER_RT;H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Überblick: Mit Direktiven im Script-Editor arbeiten

Sie können %Include- und %Rem-Direktiven in Scripts eingeben, die Sie im IDE erstellen, jedoch keine %If (%Else, %ElseIf, %EndIf)-Direktiven. Wenn Sie %If-Direktiven benutzen möchten, müssen Sie diese in eine Datei eingeben, die Sie mit der %Include-Direktive aufrufen. Wenn sie direkt in IDE-Scripts eingegeben werden, führen %If-Direktiven zu Syntax- und Kompilierfehlern.

%Rem und %End Rem eingeben

Sie können die %Rem-Direktive verwenden, um einen Kommentarblock in einen beliebigen (Globals)-Abschnitt oder ein Objektscript einzugeben.

Wenn Sie %End Rem nicht eingeben, um den Block zu beenden, beendet das IDE den Block für Sie, wenn Sie das IDE schließen, den Fokus vom IDE entfernen oder die Script-Syntax prüfen.

Mit %Include arbeiten

Wenn das IDE einen Fehler in einer Datei feststellt, die mit %Include aufgerufen wird, meldet es den Fehler in der Zeile, die die Direktive enthält. Nachdem Sie eine Included-Datei geändert haben, müssen Sie F2 drücken, um alle Scripts für das Objekt neu zu kompilieren, bzw. Umschalttaste+F2, um alle Scripts für das Dokument neu zu kompilieren, da das IDE nicht feststellen kann, ob diese Änderungen vorgenommen wurden.

{button ,AL(`H_IDE_WORKING_WITH_DIRECTIVES_IN_THE_SCRIPT_EDITOR_OVER_RT;H_IDE_COMPILING_AND_TESTING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Überblick: Text im Script-Editor verwalten

In den Script-Bereich des Script-Editors geben Sie Script-Anweisungen für ein Dokument und für Objekte in dem Dokument ein. In diesem Bereich des Editors können Sie Aufgaben ausführen, die im allgemeinen in den meisten Texteditoren ausgeführt werden: Sie können Text ausschneiden, kopieren und einfügen, Text eingeben und löschen, Änderungen widerrufen und, wenn Ihre Anwendung dies zuläßt, gelöschte Änderungen wiederherstellen, Text importieren sowie Text suchen und ersetzen.

Mit dem Befehl Script-Editor Vorgaben im Menü Datei können Sie außerdem die folgenden Script-Eigenschaften festlegen: Abstand der Tabulatorstopps und Zeileneinrückung in Scripts sowie die Schriftarten und Farben, mit denen Sie Script-Text anzeigen.

Automatische Formatierung

Das IDE formatiert jedes Script automatisch so, daß:

- LotusScript-Schlüsselwörter mit großem Anfangsbuchstaben geschrieben werden; Direktiven und das Schlüsselwort Rem werden in Großbuchstaben angezeigt
- Kennungen und mit Kommentaren versehene Zeilen genau so angezeigt werden, wie Sie sie eingeben
- Zeilen, die mit Labels beginnen, linksbündig ausgerichtet werden
- Wenn die Funktion Smart Indenting aktiviert ist, Zeilen innerhalb eines Blocks um einen Tabulatorstopp eingerückt werden.

Script-Anweisungen eingeben

Im allgemeinen kann eine Zeile in einem Script eine Script-Anweisung enthalten. Sie können mehrere Anweisungen auf einer einzelnen Zeile eingeben, indem Sie die Anweisungen durch einen Doppelpunkt trennen. Wenn Sie aus der Zeile gehen, werden die Doppelpunkte in Zeilenschaltungen umgesetzt und die Anweisungen in separate Zeilen aufgeteilt. Die Zeilen werden automatisch formatiert.

Wenn Sie im Anschluß an eine Kennung einen Doppelpunkt eingeben, wird die Kennung als Label bezeichnet.

{button ,AL(`H_IDE_MANAGING_TEXT_IN_THE_SCRIPT_EDITOR_OVER_RT;H_IDE_CUTTING_COPYING_AND_PASTING_TEXT_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_ENTERING_AND_DELETING_TEXT_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FORMATTING_CODE_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_SELECTING_TEXT_IN_THE_SCRIPT_EDITOR_STEPS;',0)} Siehe Verwandte Themen

Text im Script-Editor markieren

Sie können Text in einem Script oder einem Drop-down-Listenfeld mit der Maus oder der Tastatur markieren.

Markieren Sie Text wie folgt mit der Maus

- Klicken Sie auf das Element, wenn Sie ein Element in einem Drop-down-Listenfeld markieren möchten.
- Doppelklicken Sie auf das Wort, wenn Sie ein Wort markieren möchten.
- Klicken Sie auf den Anfang des Blocks und ziehen die Markierung bis zum Ende des Blocks, um einen Script-Block zu markieren.
- Drücken Sie die Umschalttaste und klicken Sie, um Text ab der Einfügestelle bis zum Mauszeiger zu markieren.

Text wird wie folgt mit der Tastatur markiert

Umschalttaste+→	Markiert das Zeichen rechts neben der Einfügestelle.
Umschalttaste+←	Markiert das Zeichen links neben der Einfügestelle.
Umschalttaste+	Markiert eine Textzeile links neben der Einfügestelle.
Umschalttaste+↓	Markiert eine Textzeile rechts neben der Einfügestelle.
Umschalttaste+Pos1	Markiert Text ab der Einfügestelle bis zum Anfang der Zeile.
Umschalttaste+Ende	Markiert Text ab der Einfügestelle bis zum Ende der Zeile.
Umschalttaste+Strg+→	Markiert das nächste Wort oder den Rest des Worts, in dem die Einfügestelle steht.
Umschalttaste+Strg+←	Markiert das vorherige Wort oder den Anfang des Worts, in dem die Einfügestelle steht.
Umschalttaste+Strg+	Markiert Text ab der Einfügestelle bis zum Anfang des aktuellen Script-Blocks.
Umschalttaste+Strg+↓	Markiert Text ab der Einfügestelle bis zur letzten Zeile im Textanzeigebereich.

{button ,AL(`H_IDE_SELECTING_TEXT_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_FORMATTING_CODE_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_MANAGING_TEXT_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Text im Script-Editor ausschneiden, kopieren und einfügen

Gehen Sie folgendermaßen vor, um Text aus einem Script auszuschneiden oder zu kopieren
Markieren Sie den Text, und wählen Sie Bearbeiten Ausschneiden oder Bearbeiten Kopieren.

Gehen Sie folgendermaßen vor, um Text in ein Script einzufügen
Setzen Sie die Einfügestelle an die Stelle, an der der Text eingefügt werden soll, und wählen Sie Bearbeiten Einfügen.

Gehen Sie folgendermaßen vor, um ein Schlüsselwort aus einer Browser-Liste in ein Script einzufügen
Setzen Sie die Einfügestelle an der Stelle in das Script, an der Sie das Schlüsselwort einfügen möchten, markieren Sie das Schlüsselwort im Browser, und klicken Sie auf Namen einfügen oben rechts im Feld Browser.

{button ,AL(`H_IDE_CUTTING_COPYING_AND_PASTING_TEXT_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_FORMATTING_CODE_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_MANAGING_TEXT_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Text im Script-Editor eingeben und löschen

Geben Sie Text wie folgt in den Script-Bereich ein

- Setzen Sie die Einfügestelle an die Stelle, an der Sie Text hinzufügen möchten, und geben den Text ein.
- Setzen Sie die Einfügestelle an die gewünschte Position, und wählen Sie Bearbeiten Einfügen.

Wenn die Anwendung, mit der Sie arbeiten, das Importieren von Dateien unterstützt, können Sie möglicherweise Scripts in den Script-Editor importieren und diese Scripts mit bestehenden Scripts mischen.

Löschen Sie Text wie folgt

Markieren Sie den Text und wählen Sie Bearbeiten Löschen, oder drücken Sie Entf.

{button ,AL(`H_IDE_ENTERING_AND_DELETING_TEXT_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_FORMATTING_CODE_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_MANAGING_TEXT_IN_THE_SCRIPT_EDITOR_OVER;',0)} Siehe Verwandte Themen

Überblick: Scripts im Script-Editor kompilieren und testen

Während Sie Scripts für ein Dokument erstellen, können Sie die Scripts testen, um zu prüfen, ob sie wie gewünscht ausgeführt werden. Nachdem Sie jede Script-Zeile eingegeben und die Einfügestelle aus der Zeile genommen haben, testet das IDE Syntax und Struktur der Zeile und meldet Fehler im Drop-down-Listenfeld Fehler.

Sie können auch Subroutinen testen und alle Scripts für ein Objekt oder Dokument überprüfen.

Subroutinen testen

Sie können den Befehl Aktuelle Subroutine ausführen im Menü Script wählen, um eine Subroutine auszuführen, die keine Parameter hat, und deren Struktur und Syntax testen, ohne daß Sie ein Objekt erstellen und die Subroutine aus einem Objekt-Event aufrufen müssen.

Alle Scripts für ein Objekt oder Dokument prüfen

Sie können alle Scripts für ein Objekt oder Dokument prüfen, indem Sie Scripts nach Objekten durchsuchen oder Scripts nach Dokumenten durchsuchen im Menü Script wählen. Mit diesen Befehlen führen Sie eine Kompilierung durch, nachdem Sie eine Datei geändert haben, die aus einer %Include-Direktive aufgerufen wird. Das IDE erhält nur auf diesem Wege Kenntnis über Änderungen und kann Änderungen nur auf diesem Wege prüfen.

Wenn Sie die Befehle Scripts durchsuchen verwenden, formatiert das IDE Scripts automatisch neu, um nicht folgerichtig eingegebene oder nicht richtig eingerückte Zeilen einzupassen, und schließt nicht abgeschlossene %Rem-Anweisungen ab.

Fehlermeldungen anzeigen

Das IDE meldet alle Kompilierfehler im Drop-down-Listenfeld Fehler. Die Anzahl von gefundenen Fehlern wird auch in einem Meldungsfeld angezeigt, wenn Sie das Feld mit dem Befehl Script-Editor Benutzervorgaben im Menü Datei aktivieren.

{button ,AL(`H_IDE_COMPILING_AND_TESTING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER_RT;H_IDE_CHECKING_A_SINGLE_LINE_OF_SCRIPT_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_CHECKING_SCRIPTS_FOR_AN_OBJECT_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_CHECKING_SCRIPTS_FOR_A_DOCUMENT_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_TESTING_A_SUB_IN_THE_SCRIPT_EDITOR_STEPS;H_IDE_USING_THE_IDE_OUTPUT_PANEL_OVER;',0)} Siehe Verwandte Themen

Subroutinen im Script-Editor testen

Wählen Sie die Subroutine im Drop-down-Listenfeld Script im Script-Editor, und wählen Sie Aktuelle Subroutine ausführen im Menü Script, um eine Subroutine auszuführen, die keine Parameter hat, und deren Struktur und Syntax zu überprüfen.

Das IDE meldet Syntax- und Kompilierfehler im Drop-down-Listenfeld Fehler.

Hinweis Sie können den Befehl Aktuelle Subroutine ausführen nicht verwenden, um eine Subroutine zu testen, die über Parameter verfügt.

{button ,AL(`H_IDE_TESTING_A_SUB_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_COMPILING_AND_TESTING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER;H_IDE_WORKING_WITH_GLOBALS_AND_OBJECTS_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

Eine einzelne Script-Zeile im Script-Editor prüfen

Nehmen Sie die Einfügestelle aus der Zeile, um die Syntax der aktuellen Script-Zeile zu prüfen.

Syntax- und Kompilierfehler werden im Drop-down-Listenfeld Fehler gemeldet.

{button ,AL(`H_IDE_CHECKING_A_SINGLE_LINE_OF_SCRIPT_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_COMPILING_AND_TESTING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER;H_IDE_WORKING_WITH_GLOBALS_AND_OBJECTS_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

Scripts für ein Objekt im Script-Editor prüfen

Wählen Sie Scripts nach Objekten durchsuchen im Menü Script, um alle nicht kompilierten Scripts für das aktuelle Objekt zu prüfen.

Syntax- und Kompilierfehler werden im Drop-down-Listenfeld Fehler gemeldet. Außerdem wird die Anzahl von gefundenen Fehlern in einem Meldungsfeld angezeigt, wenn Sie das Meldungsfeld mit dem Befehl Script-Editor Vorgaben im Menü Datei aktiviert haben.

{button ,AL(`H_IDE_CHECKING_SCRIPTS_FOR_AN_OBJECT_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_COMPILING_AND_TESTING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER;H_IDE_WORKING_WITH_GLOBALS_AND_OBJECTS_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

Scripts für ein Dokument im Script-Editor prüfen

Wählen Sie Scripts nach Dokumenten durchsuchen im Menü Script, um alle nicht kompilierten Scripts für das aktuelle Dokument zu prüfen.

Syntax- und Kompilierfehler werden im Drop-down-Listenfeld Fehler gemeldet. Außerdem wird die Anzahl von gefundenen Fehlern in einem Meldungsfeld angezeigt, wenn Sie das Meldungsfeld mit dem Befehl Script-Editor Vorgaben im Menü Datei aktiviert haben.

{button ,AL(`H_IDE_CHECKING_SCRIPTS_FOR_A_DOCUMENT_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_COMPILING_AND_TESTING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_PROCEDURES_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER;H_IDE_WORKING_WITH_GLOBALS_AND_OBJECTS_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

Kompilierfehler im Script-Editor suchen und beheben

Kompilierfehler werden im Drop-down-Listenfeld in folgendem Format gemeldet:

Objekt: *Prozedur*: *Zeile#*: *Meldung*

Korrigieren Sie einen Fehler wie folgt

1. Klicken Sie auf die Fehlermeldung im Drop-down-Listenfeld Fehler.

   Das IDE nimmt einen Bildlauf zu der Script-Zeile vor, die den Fehler enthält.
2. Beheben Sie den Fehler.

Beheben Sie einen Fehler in einer enthaltenen Datei wie folgt

1. Doppelklicken Sie auf die Fehlermeldung im Drop-down-Listenfeld Fehler.

   Das IDE nimmt einen Bildlauf zu der Zeile vor, die die %Include-Direktive für die Datei enthält.
2. Öffnen Sie die Datei, und beheben Sie den Fehler.

{button ,AL(`H_IDE_FINDING_AND_FIXING_COMPILE_TIME_ERRORS_IN_THE_SCRIPT_EDITOR_STEPS_RT;H_IDE_COMPILING_AND_TESTING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

Überblick: Script-Debugger

Im IDE Script-Debugger können Sie Breakpoints setzen, löschen, aktivieren und deaktivieren und schrittweise durch Scripts gehen, um die Ursache von Problemen bei der Ausführung von Scripts zu ermitteln. Der Debugger wird geöffnet, wenn der erste aktivierte Breakpoint während der Ausführung des Scripts angetroffen wird. Außerdem wird der Debugger für ein Dokument geöffnet, wenn eine der zugehörigen Prozeduren aus einem ausführenden Script in einem anderen Dokument aufgerufen wird.

Wenn der Script-Debugger geöffnet ist:

- Wird das Menü Debug im Hauptmenü angezeigt
- Kann auf die Felder Browser, Ausgabe, Breakpoints und Variablen zugegriffen werden
- Wird das Script mit dem Breakpoint im Debugger angezeigt, und die Breakpoint-Zeile ist die aktuelle Zeile.

Benutzung des Debug-Menüs

Das Debug-Menü enthält Befehle, mit denen Sie im IDE:

- Breakpoints für das Debuggen setzen, löschen, aktivieren und deaktivieren können
- Schrittweise durch ein Script gehen und die Script-Ausführung stoppen können.

{button ,AL(`H_IDE_THE_SCRIPT_DEBUGGER_OVER_RT;H_IDE_CLOSING_THE_SCRIPT_DEBUGGER_OVER;H_IDE_EXECUTING_SCRIPTS_IN_THE_DEBUGGER_OVER;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_SWITCHING_TO_ANOTHER_SCRIPT_IN_THE_DEBUGGER_OVER;H_IDE_USING_BREAKPOINTS_IN_DEBUGGING_OVER;',0)} Siehe Verwandte Themen

Überblick: Breakpoints beim Debuggen verwenden

Sie können Breakpoints im Script-Editor und Script-Debugger in Zeilen in allen Subroutinen setzen, mit Ausnahme der Terminate-Subroutine. Wenn Sie einen Breakpoint setzen, ist er vorgabegemäß aktiviert. Wenn Sie ein Script bearbeiten oder debuggen, können Sie den Breakpoint deaktivieren, aktivieren und löschen. Sie können auch andere Breakpoints setzen.

Hinweis Bei einer über zwei Zeilen gehenden Zeile wird der Breakpoint in die letzte Zeile gesetzt.

Breakpoint-Symbole


Breakpoint


Aktueller Breakpoint


Deaktivierter Breakpoint


Aktueller deaktivierter Breakpoint


Kein Breakpoint

Indikator der aktuellen Zeile


Oberste aktuelle Zeile


Verschachtelte aktuelle Zeile

Kombinierte Breakpoint-Symbole und Indikatoren der aktuellen Zeile


Deaktivierter Breakpoint, oberste aktuelle Zeile


Aktueller deaktivierter Breakpoint, oberste aktuelle Zeile


Deaktivierter Breakpoint, verschachtelte aktuelle Zeile


Aktueller deaktivierter Breakpoint, verschachtelte aktuelle Zeile


Breakpoint, oberste aktuelle Zeile


Aktueller Breakpoint, oberste aktuelle Zeile


Breakpoint, verschachtelte aktuelle Zeile


Aktueller Breakpoint, verschachtelte aktuelle Zeile

{button ,AL(`H_IDE_USING_BREAKPOINTS_IN_DEBUGGING_OVER_RT;H_IDE_ENABLING_AND_DISABLING_BREAKPOINTS_IN_THE_IDE_STEPS;H_IDE_SETTING_AND_CLEARING_BREAKPOINTS_IN_THE_IDE_STEPS;H_IDE_USING_THE_IDE_BREAKPOINTS_PANEL_OVER;',0)} Siehe Verwandte Themen

Breakpoints im IDE setzen und löschen

Gehen Sie folgendermaßen vor, um Breakpoints im Script-Editor oder Script-Debugger zu setzen und zu löschen

- Klicken Sie neben der Zeile in der Breakpoint-Spalte, um einen Breakpoint in einer Zeile zu setzen.
- Klicken Sie auf das Breakpoint-Symbol in der Breakpoint-Spalte, um einen Breakpoint zu löschen.

Gehen Sie folgendermaßen vor, um Breakpoints mit dem Menü Debug zu setzen und zu löschen

- Klicken Sie an einer beliebigen Stelle in der Zeile, und wählen Sie Breakpoint setzen im Menü Debug, um einen Breakpoint in einer Zeile zu setzen.
- Klicken Sie an einer beliebigen Stelle in der Zeile, und wählen Sie Breakpoint entfernen im Menü Debug, um einen Breakpoint in einer Zeile zu löschen.
- Wählen Sie Breakpoints für Dokument entfernen im Menü Debug, um alle Breakpoints für ein Dokument zu löschen.

Gehen Sie folgendermaßen vor, um Breakpoints mit der Liste im Feld Breakpoints zu löschen

Markieren Sie einen Breakpoint in der Breakpoint-Liste, drücken Sie Entf, oder wählen Sie Clear Breakpoint im Menü Debug.

{button ,AL(`H_IDE_SETTING_AND_CLEARING_BREAKPOINTS_IN_THE_IDE_STEPS_RT;H_IDE_USING_BREAKPOINTS_IN_DEBUGGING_OVER;H_IDE_USING_THE_IDE_BREAKPOINTS_PANEL_OVER;',0)} Siehe Verwandte Themen

Breakpoints im IDE aktivieren und deaktivieren

Gehen Sie folgendermaßen vor, um Breakpoints mit dem Menü Debug zu aktivieren oder zu deaktivieren

- Klicken Sie an einer beliebigen Stelle in der Zeile, und wählen Sie Breakpoint aktivieren im Menü Debug, um einen Breakpoint in einer Zeile zu aktivieren.
- Klicken Sie an einer beliebigen Stelle in der Zeile, und wählen Sie Breakpoint deaktiveren im Menü Debug, um einen Breakpoint in einer Zeile zu deaktivieren.
- Wählen Sie Breakpoints aktivieren oder deaktiveren im Menü Debug, um alle Breakpoints für ein Dokument zu aktivieren oder zu deaktivieren.

Gehen Sie folgendermaßen vor, um Breakpoints mit der Liste im Feld Breakpoints zu aktivieren oder zu deaktivieren

Markieren Sie einen Breakpoint in der Breakpoint-Liste, und wählen Sie Breakpoint aktivieren oder Breakpoint deaktivieren im Menü Debug.

Gehen Sie folgendermaßen vor, um Breakpoints mit der Tastatur und der Maus zu aktivieren oder zu deaktivieren

Klicken Sie bei gedrückter Strg-Taste auf das Breakpoint-Symbol in der Breakpoint-Spalte.

{button ,AL(`H_IDE_ENABLING_AND_DISABLING_BREAKPOINTS_IN_THE_IDE_STEPS_RT;H_IDE_USING_BREAKPOINTS_IN_DEBUGGING_OVER;H_IDE_USING_THE_IDE_BREAKPOINTS_PANEL_OVER;',0)} Siehe Verwandte Themen

Überblick: Scripts im Debugger ausführen

Der Script-Debugger wird geöffnet, wenn bei der Ausführung eines Scripts auf einen aktivierten Breakpoint gestoßen wird. Sobald der Debugger geöffnet ist, können Sie mit den Befehlen im Menü Debug schrittweise durch ein Script gehen, in eine Prozedur gehen, die in einer Breakpoint-Zeile aufgerufen wird, bzw. diese Prozedur übergehen, die Prozedur beenden, die Ausführung des Scripts fortsetzen oder beenden.

Sie können auch Informationen über das ausführende Script in den Feldern Ausgabe und Variablen anzeigen.

Mit mehr als einem Debugger arbeiten

Wenn ein Script, das Sie in einem Dokument debuggen, in ein Script geht, das zu einem anderen, nicht geöffneten Dokument geht, wird der Debugger für das zweite Dokument geöffnet und dessen Script angezeigt.

Falls schon für mehrere Dokumente Script-Editoren geöffnet sind, wenn ein Breakpoint im ausführenden Dokument erreicht wird, werden die Debugger anstelle der Editoren angezeigt, unabhängig davon, ob Scripts für diese Dokumente ausgeführt werden oder nicht. In dem Drop-down-Feld Calls für jeden Debugger werden alle ausführenden Prozeduren aufgeführt, unabhängig von den Dokumenten, zu denen sie gehören.

Wenn die Ausführung des Scripts zu einer Prozedur führt, bei der die Quelle nicht verfügbar ist, wird weiter schrittweise durch das Script gegangen, bis eine anzeigbare Anweisung erreicht oder das Script beendet wird.

{button ,AL(`H_IDE_EXECUTING_SCRIPTS_IN_THE_DEBUGGER_OVER_RT;H_IDE_CONTINUING_AND_STOPPING_SCRIPT_EXECUTION_IN_THE_DEBUGGER_STEPS;H_IDE_STEPPING_THROUGH_A_SCRIPT_IN_THE_DEBUGGER_STEPS;H_IDE_USING_THE_IDE_OUTPUT_PANEL_OVER;H_IDE_USING_THE_IDE_VARIABLES_PANEL_OVER;',0)} Siehe Verwandte Themen

Im Debugger schrittweise durch ein Script gehen

Gehen Sie wie folgt schrittweise durch die Prozeduren in einem Script
Wenn die Ausführung des Scripts bei einer Zeile stoppt, wählen Sie Gehe zu im Menü Debug. Danach wird die nächste Prozedur ausgeführt. Wenn die Prozedur einen Aufruf enthält, geht die Script-Ausführung in die aufgerufene Prozedur.

Gehen Sie folgendermaßen aus der aktuellen Prozedur
Wenn die Ausführung einer Prozedur stoppt, wählen Sie Beenden im Menü Debug, um aus der Prozedur zu gehen. Wenn Sie eine aufgerufene Prozedur beenden, geht die Ausführung des Scripts in die Zeile, die auf die Zeile mit dem Aufruf folgt. Sonst wird die Ausführung des Scripts fortgesetzt, bis der nächste Breakpoint erreicht oder das Script beendet wird.

Übergehen Sie die Prozedur, die in einer Breakpoint-Zeile aufgerufen wird, wie folgt
Wenn die Ausführung des Scripts in der Zeile stoppt, wählen SieÜberspringen im Menü Debug. Die Zeile und die Prozedur, die in der Zeile aufgerufen wird, werden ausgeführt, und die Ausführung des Scripts geht dann zu der Zeile, die auf die Breakpoint-Zeile folgt.

Die Ausführung eines Scripts im Debugger fortsetzen und stoppen

Gehen Sie folgendermaßen vor, um die Ausführung des Scripts ab einem Breakpoint fortzusetzen
Wählen Sie Ausführung fortsetzen im Menü Debug. Die Script-Ausführung wird bis zum nächsten Breakpoint oder bis zum Ende des Scripts fortgesetzt.

Stoppen Sie die Ausführung des Scripts wie folgt
Wählen Sie Ausführung stoppen im Menü Debug. Die Ausführung des Scripts wird bei dem aktuellen Breakpoint gestoppt. Wenn der Script-Editor beim Starten des Debug-Vorgangs geöffnet war, wird er erneut angezeigt, wenn das Debuggen gestoppt wird; sonst wird der Script-Debugger geschlossen.

{button ,AL(`H_IDE_CONTINUING_AND_STOPPING_SCRIPT_EXECUTION_IN_THE_DEBUGGER_STEPS_RT;H_IDE_CLOSING_THE_SCRIPT_DEBUGGER_OVER;H_IDE_EXECUTING_SCRIPTS_IN_THE_DEBUGGER_OVER;H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER;H_IDE_SWITCHING_TO_ANOTHER_SCRIPT_IN_THE_DEBUGGER_OVER;H_IDE_USING_BREAKPOINTS_IN_DEBUGGING_OVER;',0)}
Siehe Verwandte Themen

Überblick: Zu einem anderen Script im Debugger wechseln

Wenn beim Debuggen ein aktivierter Breakpoint erreicht wird, wird die Script-Ausführung gestoppt, und das Script für die Prozedur, die den Breakpoint enthält, wird im Script-Debugger angezeigt. Sie können eine andere Prozedur anzeigen, indem Sie sie aus dem Drop-down-Listenfeld Calls am Ende des Ausschnitts Debugger wählen, bzw. aus dem Drop-down-Listenfeld Script oben rechts.

Verwendung des Drop-down-Listenfelds Calls

Sie können zu einer anderen Prozedur im Script-Debugger-Ausführungsstack wechseln, indem Sie diese aus dem Drop-down-Listenfeld Calls aufrufen.

In dem Feld Calls werden die Prozeduren aufgeführt, die gerade ausgeführt werden, einschließlich Prozeduren, die zu anderen Dokumenten gehören. Jedes Listenelement enthält den Namen des Objekts der Prozedur, mit der es verknüpft ist, sowie den Namen der Prozedur. Wenn Sie eine Prozedur in der Liste wählen, wird das zugehörige Script im Script-Bereich des Debuggers angezeigt.

Eine Prozedur, die zu einem anderen Dokument gehört, wird in der Liste grau angezeigt. Eine Prozedur, deren Quelle nicht verfügbar ist (z. B. ein externes Script, das aus der Prozedur aufgerufen wird), wird ebenfalls grau angezeigt. Sie können diese Prozeduren nicht wählen.

Wenn Sie eine Prozedur wählen, werden deren Variablen und Werte im Feld Variablen angezeigt. Mit dem Feld Variablen können Sie Informationen über eine Variable anzeigen oder deren Wert in der ausführenden Prozedur ändern.

Verwendung des Drop-down-Listenfelds Script

Wenn Sie zu einer Prozedur wechseln möchten, die nicht im Ausführungsstack steht, können Sie mit den Drop-down-Listenfeldern Objekt und Script zu der Prozedur gehen. Das gewählte Script wird im Script-Debugger angezeigt. Im Drop-down-Feld Calls und dem Feld Variablen werden jedoch weiter Informationen für die ausführende Prozedur angezeigt, die als letztes im Feld Calls aufgerufen wurde.

{button ,AL(`H_IDE_SWITCHING_TO_ANOTHER_SCRIPT_IN_THE_DEBUGGER_OVER_RT;H_IDE_SELECTING_A_SCRIPT_FROM_THE_CALLS_DROP_DOWN_LIST_STEPS;H_IDE_SELECTING_A_SCRIPT_FROM_THE_SCRIPT_LIST_IN_THE_DEBUGGER_STEPS;',0)} Siehe Verwandte Themen

Ein Script aus der Drop-down-Liste Calls im Debugger aufrufen

1. Klicken Sie auf das Drop-down-Listenfeld Calls, um eine Liste der Prozeduren aufzurufen, die im Ausführungsstack stehen.
2. Klicken Sie auf die gewünschte Prozedur im Script-Debugger.

   Das Prozedurscript wird im Script-Bereich Debugger angezeigt. Im Feld Variablen werden Informationen über Prozedurvariablen angezeigt.

Hinweis Sie können ein Script im Script-Bereich Debugger nicht ändern; Sie müssen die Script-Ausführung stoppen und Änderungen im Script-Editor durchführen. Sie können Variablenwerte im Feld Variablen ändern.

{button ,AL(`H_IDE_SELECTING_A_SCRIPT_FROM_THE_CALLS_DROP_DOWN_LIST_STEPS_RT;H_IDE_EXECUTING_SCRIPTS_IN_THE_DEBUGGER_OVER;H_IDE_SWITCHING_TO_ANOTHER_SCRIPT_IN_THE_DEBUGGER_OVER;',0)} Siehe Verwandte Themen

Ein Script aus der Script-Liste im Debugger wählen

1. Klicken Sie auf das Drop-down-Listenfeld Objekt, um eine Liste der Objekte im Dokument anzuzeigen.
2. Wählen Sie das Objekt, dessen Script angezeigt werden soll.
3. Klicken Sie auf das Drop-down-Listenfeld Script, um eine Liste der Scripts für das Objekt anzuzeigen.
4. Wählen Sie das Script, das Sie betrachten möchten.

   Das Script wird im Script-Bereich Debugger angezeigt.

Hinweis Sie können ein Script im Script-Bereich Debugger nicht ändern; Sie müssen die Ausführung des Scripts stoppen und Änderungen im Script-Editor vornehmen. Variablenwerte können im Feld Variablen geändert werden.

{button ,AL(`H_IDE_SELECTING_A_SCRIPT_FROM_THE_SCRIPT_LIST_IN_THE_DEBUGGER_STEPS_RT;H_IDE_EXECUTING_SCRIPTS_IN_THE_DEBUGGER_OVER;H_IDE_SWITCHING_TO_ANOTHER_SCRIPT_IN_THE_DEBUGGER_OVER;',0)} Siehe Verwandte Themen

Überblick: Im Script-Debugger gemeldete Fehler beheben

Wenn es zu einem Laufzeitfehler kommt, wird der Fehler in einem Meldungsfeld gemeldet, und der Indikator für die aktuelle Zeile im Script-Debugger wird in die Zeile gesetzt, die den Fehler enthält. Das Format der Meldung wird unten dargestellt:

Objekt: *Prozedur*: *Zeile#*: *Meldung*

Drücken Sie F1, um einen Hilfetext für die Meldung anzuzeigen.

Alle Optionen des Menüs Debug sind deaktiviert, mit Ausnahme von Ausführung stoppen. Wenn Sie Ausführung stoppen wählen, wird das Script mit dem Fehler im Script-Editor angezeigt, wenn der Editor beim Starten des Debug-Vorgangs geöffnet war; sonst wird der Script-Debugger geschlossen.

Wenn beim Debuggen ein Kompilierfehler angetroffen wird, wird die Ausführung des Scripts beendet und wird das Script mit dem Fehler im Script-Editor angezeigt. Außerdem wird eine Meldung über den Fehler im Drop-down-Listenfeld Fehler angezeigt. Sie können einen Hilfetext für die Fehlermeldung aufrufen, indem Sie die Meldung markieren und F1 drücken.

{button ,AL(`H_IDE_FIXING_ERRORS_REPORTED_IN_THE_SCRIPT_DEBUGGER_OVER_RT;H_IDE_EXECUTING_SCRIPTS_IN_THE_DEBUGGER_OVER;H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER;H_IDE_SWITCHING_TO_ANOTHER_SCRIPT_IN_THE_DEBUGGER_OVER;',0)} Siehe Verwandte Themen

Überblick: Text im Script-Debugger markieren und kopieren

Sie können Text genau wie im Script-Editor markieren und in die Zwischenablage kopieren, z. B. indem Sie die Maus über den zu kopierenden Text ziehen und Bearbeiten Kopieren wählen.

Im Script-Debugger können Sie keinen Text in ein Script eingeben oder einfügen und können keinen Text aus einem Script ausschneiden.

{button ,AL(`H_IDE_SELECTING_AND_COPYING_TEXT_IN_THE_SCRIPT_DEBUGGER_OVER_RT;H_IDE_EXECUTING_SCRIPTS_IN_THE_DEBUGGER_OVER;H_IDE_USING_THE_IDE_VARIABLES_PANEL_OVER;',0)}
Siehe Verwandte Themen

Überblick: Script-Debugger schließen

Sie können den Script-Debugger für ein Dokument schließen, solange das Dokument nicht das gerade ausführende Script enthält. Wenn Sie versuchen, den Debugger zu schließen, während das zugehörige Script ausgeführt wird, wird mit einer Meldung angegeben, daß die Ausführung des Scripts durch das Schließen des Debuggers gestoppt wird. Dann werden Sie gefragt, ob Sie fortfahren möchten. Wenn Sie auf OK klicken, wird die Ausführung des Scripts gestoppt und der Debugger geschlossen.

Wenn Sie einen Debugger für das ausführende Script schließen, wirkt sich dies wie folgt auf andere geöffnete Debugger aus:

- Debugger, die anstelle von Script-Editoren angezeigt wurden (als der Debugger für das ausführende Script geöffnet wurde), werden wieder zu Editoren.
- Debugger, die automatisch während der Ausführung des Scripts geöffnet wurden, werden geschlossen.

Sie können einen Debugger schließen, indem Sie im Menü Debug Ausführung stoppen wählen, indem Sie auf das Systemmenüfeld in der oberen linken Ecke des Debugger-Fensters doppelklicken oder indem Sie das Dokument schließen, in dem der Debugger geöffnet ist.

{button ,AL(`H_IDE_CLOSING_THE_SCRIPT_DEBUGGER_OVER_RT;H_IDE_CLOSING_THE_IDE_WINDOW_STEPS;H_IDE_CONTINUING_AND_STOPPING_SCRIPT_EXECUTION_IN_THE_DEBUGGER_STEPS;',0)} Siehe Verwandte Themen

Überblick: Die Felder Breakpoints, Browser, Ausgabe und Variablen

In den Feldern Breakpoints, Browser, Ausgabe und Variablen werden Informationen über Scripts angezeigt, die Sie im Script-Editor bzw. Script-Debugger erstellen und testen. Mit diesen Feldern können Sie Scripts ändern und Breakpoints für das Debuggen von Scripts verwalten.

Feld Browser

Im Feld Browser werden Listen mit Schlüsselwörtern, Komponenten von Anwendungs-Scripts und Typ-Bibliotheken und -Klassen für OLE Automation-Objekte angezeigt. Sie können Elemente in diesen Listen markieren und deren Namen in das aktuelle Script einfügen. Das Feld Browser ist vorgabegemäß aktiviert, wenn der Script-Editor aktiv ist. Sie können Browser-Auflistungen anzeigen, während der Script-Editor oder Script-Debugger aktiv ist; Sie können nur Elemente markieren und in das aktuelle Script einfügen, wenn der Script-Editor aktiv ist.

Feld Breakpoints

Im Feld Breakpoints werden Breakpoints aufgeführt, die in Scripts gesetzt sind. Mit diesem Feld können Sie auch zu Breakpoint-Zeilen in Scripts gehen und Breakpoints löschen, aktivieren und deaktivieren. Sie können das Feld Breakpoints benutzen, während der Script-Editor oder Script-Debugger aktiv ist.

Feld Ausgabe

Im Feld Ausgabe wird die Ausgabe angezeigt, die von den LotusScript Drucken-Anweisungen generiert wird, die in Scripts enthalten sind. Diese Ausgabe wird generiert, wenn Scripts ausgeführt werden, und kann angezeigt werden, während der Script-Editor oder Script-Debugger aktiv ist.

Feld Variablen

Im Feld Variablen werden Informationen über Variablen in einem Script angezeigt. In diesem Feld können Sie die Variablenwerte ändern. Das Feld Variablen kann nur aufgerufen werden, wenn der Script-Debugger aktiv ist.

{button ,AL(`H_IDE_THE_BREAKPOINTS_BROWSER_OUTPUT_AND_VARIABLES_PANELS_OVER_RT;H_IDE_USING_THE_IDE_BREAKPOINTS_PANEL_OVER;H_IDE_USING_THE_IDE_BROWSER_PANEL_OVER;H_IDE_USING_THE_IDE_OUTPUT_PANEL_OVER;H_IDE_USING_THE_IDE_VARIABLES_PANEL_OVER;',0)}Siehe Verwandte Themen

Überblick: Das IDE-Feld Breakpoints verwenden

Im Feld Breakpoints wird eine Liste der Breakpoints angezeigt, die für alle Scripts in einem Dokument gesetzt sind. Anhand dieser Liste können Sie sehen, welche Objekte, Prozeduren und Zeilen Breakpoints enthalten, können bestimmen, ob ein Breakpoint aktiviert ist, und können Breakpoints löschen, aktivieren und deaktivieren.

Sie können auch zu einer Breakpoint-Zeile im Script-Editor oder Script-Debugger gehen, indem Sie auf den entsprechenden Eintrag in der Liste Breakpoints klicken. Sie können zu der Breakpoint-Zeile gehen und den Script-Editor oder Script-Debugger aktivieren, indem Sie auf den Eintrag Breakpoints doppelklicken oder den Eintrag markieren und Return drücken.

{button ,AL(`H_IDE_USING_THE_IDE_BREAKPOINTS_PANEL_OVER_RT;H_IDE_MANAGING_BREAKPOINTS_FROM_THE_BREAKPOINTS_PANEL_OVER;H_IDE_USING_BREAKPOINTS_IN_DEBUGGING_OVER;H_IDE_VIEWING_A_LIST_OF_BREAKPOINTS_IN_THE_BREAKPOINTS_PANEL_OVER;',0)} Siehe Verwandte Themen

Überblick: Eine Liste mit Breakpoints im Feld Breakpoints anzeigen

Breakpoints werden in der Reihenfolge aufgeführt, in der sie in Scripts eines Dokuments eingegeben werden. Informationen über jeden Breakpoint werden in folgendem Format angezeigt:

Objekt: *Prozedur*: *Zeile#* (Deaktiviert)

{button ,AL(`H_IDE_VIEWING_A_LIST_OF_BREAKPOINTS_IN_THE_BREAKPOINTS_PANEL_OVER_RT;H_IDE_USING_BREAKPOINTS_IN_DEBUGGING_OVER;H_IDE_USING_THE_IDE_BREAKPOINTS_PANEL_OVER;',0) } Siehe Verwandte Themen

Überblick: Breakpoints im Feld Breakpoints verwalten

Aus dem Feld Breakpoints können Sie zu einer Breakpoint-Zeile im Script-Editor oder Script-Debugger gehen, können einen Breakpoint aus einer Script-Zeile löschen und können einen Breakpoint aktivieren bzw. deaktivieren.

Zu Zeilen im Script-Editor oder Script-Debugger gehen

Sie können auf ein Element in der Liste Breakpoints klicken, um zu der Breakpoint-Zeile im Script-Editor oder Script-Debugger zu gehen:

- Wenn Sie auf ein Breakpoint-Element in der Liste klicken, wird ein Bildlauf zu der Breakpoint-Zeile im Script-Editor oder Script-Debugger vorgenommen, ohne den Editor oder Debugger zu aktivieren.
- Wenn Sie auf ein Breakpoint-Element doppelklicken (oder das Element markieren und Return drücken), wird zu der Breakpoint-Zeile im Script-Editor oder Script-Debugger gegangen, wird die Einfügestelle an den Anfang der Zeile gesetzt und der Editor oder Debugger aktiviert.

Breakpoints löschen

Markieren Sie einen Breakpoint im Feld Breakpoints und drücken Entf oder wählen Breakpoint entfernen im Menü Debug, um einen Breakpoint zu löschen.

Wenn Sie einen Breakpoint löschen, ändert sich der Fokus des Ausschnitts oder das Script nicht, das im Script-Editor oder Script-Debugger angezeigt wird.

Breakpoints aktivieren und deaktivieren

Wenn Sie einen Breakpoint aktivieren oder deaktivieren möchten, markieren Sie ihn im Feld Breakpoints und wählen Breakpoint aktivieren oder Breakpoint deaktivieren im Menü Debug.

{button ,AL(`H_IDE_MANAGING_BREAKPOINTS_FROM_THE_BREAKPOINTS_PANEL_OVER_RT;H_IDE_USING_BREAKPOINTS_IN_DEBUGGING_OVER;H_IDE_USING_THE_IDE_BREAKPOINTS_PANEL_OVER;',0)} Siehe Verwandte Themen

Überblick: Das IDE-Feld Browser verwenden

Im Feld Browser werden LotusScript-Schlüsselwörter sowie Anweisungssyntax, Komponenten von Anwendungsscripts und Typ-Bibliotheken und -Klassen für OLE Automation-Objekte angezeigt. In diesem Feld können Sie Schlüsselwörter, Komponentennamen und OLE Application-Class-Kennungen in das aktuelle Script einfügen.

Das Feld Browser wird aus dem Script-Editor und Script-Debugger aufgerufen, Sie können jedoch nur Elemente wählen und sie in das aktuelle Script einfügen, wenn der Script-Editor aktiv ist.

Wenn Sie im Browser arbeiten, können Sie einen Hilfetext für ein LotusScript-Schlüsselwort oder ein anderes Listenelement anzeigen, indem Sie das Schlüsselwort oder Element markieren und F1 drücken.

{button ,AL(`H_IDE_USING_THE_IDE_BROWSER_PANEL_OVER_RT;H_IDE_PASTING_FROM_THE_BROWSER_INTO_A_SCRIPT_STEPS;H_IDE_VIEWING_INFORMATION_ABOUT_APPLICATION_COMPONENTS_STEPS;H_IDE_VIEWING_LOTUSSCRIPT_KEYWORDS_IN_THE_BROWSER_STEPS;H_IDE_VIEWING_REGISTERED_TYPE_LIBRARIES_FOR_OLE_CLASSES_STEPS;',0)} Siehe Verwandte Themen

LotusScript-Schlüsselwörter im Browser anzeigen

Sie können die Syntax von LotusScript-Schlüsselwörtern im Browser anzeigen, wenn der Script-Editor oder Script-Debugger aktiv ist. Außerdem können Sie einen Hilfetext für ein Schlüsselwort anzeigen, indem Sie dieses markieren und F1 drücken.

Wenn der Script-Editor aktiv ist, können Sie ein Schlüsselwort markieren, um es in das aktuelle Script einzufügen.

Die Syntax von Schlüsselwörtern wird wie folgt angezeigt

1. Im Drop-down-Listenfeld Kategorie wählen Sie LotusScript Sprache.
2. Wenn Sie eine Liste der LotusScript-Schlüsselwörter in alphabetischer Reihenfolge anzeigen möchten, blenden Sie Alle ein.

   Wenn Sie eine Liste mit Schlüsselwörtern für eine andere Kategorie anzeigen möchten, blenden Sie die Kategorie ein.

Ein Schlüsselwort wird wie folgt in das aktuelle Script eingefügt

1. Markieren Sie einen Schlüsselwort-Eintrag in einer Kategorie-Liste.
2. Klicken Sie auf Namen einfügen oben rechts im Feld Browser.

   Das Schlüsselwort wird in das aktuelle Script im Script-Editor eingefügt.

{button ,AL(`H_IDE_VIEWING_LOTUSSCRIPT_KEYWORDS_IN_THE_BROWSER_STEPS_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_BROWSER_PANEL_OVER;',0)}Siehe Verwandte Themen

Informationen über Anwendungskomponenten im Browser anzeigen

Sie können Listen mit Anwendungsklassen, Konstanten, Variablen, Subroutinen und Funktionen im Browser anzeigen, wenn der Script-Editor oder Script-Debugger aktiv ist. Sie können deren Namen nur in das aktuelle Script einfügen, wenn der Script-Editor aktiv ist.

Gehen Sie folgendermaßen vor, um eine Liste mit Klassen, Konstanten, Variablen oder Subroutinen und Funktionen anzuzeigen

1. Im Drop-down-Listenfeld Kategorie wählen Sie eine Anwendungskategorie.
2. Wenn es Unterkategorien gibt, blenden Sie die Kategorien ein, die Sie betrachten möchten.

Fügen Sie einen Komponentennamen wie folgt in das aktuelle Script ein

1. Markieren Sie ein Element aus einer eingeblendeten Kategorie.
2. Klicken Sie auf Namen einfügen oben rechts im Feld Browser.

   Der Komponentenname wird in das aktuelle Script im Script-Editor eingefügt.

{button ,AL(`H_IDE_VIEWING_INFORMATION_ABOUT_APPLICATION_COMPONENTS_STEPS_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_BROWSER_PANEL_OVER;',0)} Siehe Verwandte Themen

Registrierte Typ-Bibliotheken für OLE-Klassen im Browser anzeigen

Sie können Listen mit OLE Automation-Objektklassen im Browser anzeigen, wenn der Script-Editor oder Script-Debugger aktiv ist. Wenn eine Klasse in Scripts aktiviert ist, können Sie die Kennung der Anwendungsklasse auch in das aktuelle Script einfügen, wenn der Script-Editor aktiv ist.

Hinweis Wenn mit einer OLE-Klasse ein Hilfetext verknüpft ist, können Sie den Hilfetext für die Klasse anzeigen, indem Sie den Klassennamen im Script-Editor, Script-Debugger oder Browser markieren und F1 drücken.

Informationen über OLE-Klassen werden wie folgt angezeigt

1. Im Drop-down-Listenfeld Kategorie wählen Sie OLE-Klassen.
2. Aus der Liste mit registrierten OLE-Typ-Bibliotheken markieren Sie die gewünschte Bibliothek.
3. Aus der Klassenliste in der Typ-Bibliothek wählen Sie die Klasse, für die Sie Informationen anzeigen möchten; danach wählen Sie Eigenschaften oder Methoden, um eine Liste mit Eigenschaften oder Methoden für die Klasse anzuzeigen.

Eine OLE-Klassenkennung wird wie folgt in das aktuelle Script eingefügt

Wenn eine OLE-Klasse mit der Funktion CreateObject in einem Script aktiviert werden kann, wird die Kennung der Anwendungsklasse in Klammern neben dem Klassennamen angezeigt.

1. Markieren Sie die gewünschte Klasse im aktuellen Script.
2. Klicken Sie auf Namen einfügen oben rechts im Feld Browser.

   Die Kennung der Anwendungsklasse wird in das Script eingefügt.

{button ,AL(`H_IDE_VIEWING_REGISTERED_TYPE_LIBRARIES_FOR_OLE_CLASSES_STEPS_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_BROWSER_PANEL_OVER;',0)}
Siehe Verwandte Themen

Elemente aus dem Browser in ein Script einfügen

Wenn der Script-Editor aktiviert ist, können Sie ein Listenelement aus dem Browser in das aktuelle Script einfügen.

1. In einer Browser-Liste markieren Sie einen Eintrag für ein Schlüsselwort der LotusScript Sprache, eine OLE-Klasse oder Anwendungskomponente.
2. Klicken Sie auf Namen einfügen oben rechts im Feld Browser.

   Das Schlüsselwort, die Kennung der OLE-Anwendungsklasse oder der Name der Anwendungskomponente wird in das aktuelle Script im Script-Editor eingefügt.

In dem Script führen Sie die Anweisung für das Schlüsselwort, die Kennung oder den Namen aus.

{button ,AL(`H_IDE_PASTING_FROM_THE_BROWSER_INTO_A_SCRIPT_STEPS_RT;H_IDE_CREATING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_BROWSER_PANEL_OVER;',0)} Siehe Verwandte Themen

Überblick: Das IDE-Feld Ausgabe verwenden

Wenn Sie Scripts in einem Dokument ausführen, werden maximal 1024 Zeichen der Ausgabe aus jeder Drucken-Anweisung, die in den Scripts enthalten ist, im Feld Ausgabe angezeigt. Diese Ausgabe wird in der Reihenfolge angezeigt, in der sie in dem Dokument generiert wurde.

Sie können die Ausgabeliste überprüfen, um festzustellen, ob Scripts wie gewünscht ausgeführt werden.

Wenn das Dokument geschlossen wird, werden maximal 500 Zeilen der Ausgabe für das Dokument gespeichert.

{button ,AL(`H_IDE_USING_THE_IDE_OUTPUT_PANEL_OVER_RT;H_IDE_COMPILING_AND_TESTING_SCRIPTS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_EXECUTING_SCRIPTS_IN_THE_DEBUGGER_OVER;',0)} Siehe Verwandte Themen

Überblick: Das IDE-Feld Variablen verwenden

Im Feld Variablen werden Informationen über Variablen in einem Script angezeigt. In diesem Feld können Sie Variablenwerte ändern. Dieses Feld ist nur beim Debuggen aktiv und enthält Informationen über Variablen im aktuellen Script (d. h. das Script für das Objekt und die Prozedur, die in der Drop-down-Liste Calls im Script-Debugger angezeigt wird).

Variablen werden nach Variablennamen in der Reihenfolge aufgeführt, in der sie deklariert wurden. Außerdem werden der Wert der Variablen und der aktuelle Datentyp aufgeführt. Wenn es sich bei einer Variablen um einen Typ mit Members handelt, wird die Variable als einblendbarer Eintrag dargestellt, und ihr Name und Typ werden aufgelistet. Blenden Sie den Eintrag ein, um die Members und deren Werte sowie den Datentyp anzuzeigen.

{button ,AL(`H_IDE_USING_THE_IDE_VARIABLES_PANEL_OVER_RT;H_IDE_CHANGING_VARIABLE_VALUES_IN_THE_IDE_STEPS;H_IDE_VIEWING_GLOBALS_FOR_THE_CURRENT_SCRIPT_STEPS;H_IDE_VIEWING_VARIABLES_FOR_THE_CURRENT_SCRIPT_STEPS;',0)} Siehe Verwandte Themen

Variablen für das aktuelle Script anzeigen

1. Im Drop-down-Listenfeld Calls im Script-Debugger wählen Sie die Prozedur, deren Variablen angezeigt werden sollen.
2. Im Feld Variablen blenden Sie die Variableneinträge (falls erforderlich) ein, um deren Members anzuzeigen.

   Member-Variablen und deren Werte und Datentypen werden angezeigt.

Wenn ein Wert mehr als 42 Zeichen enthält, werden die ersten 40 Zeichen plus ... (Auslassungspunkten) angezeigt.

Hinweis Wenn in einer Liste der Inhalt eines Bereichs angezeigt wird, stehen neben den Werten Zahlen. Diese Zahlen stellen die Reihenfolge des Inhalts dar; sie können nicht als Indizes für den Bereich verwendet werden.

Globals für das aktuelle Script anzeigen

Blenden Sie den Eintrag Globals im Feld Variablen ein.

Die folgenden Variablen und deren Werte und Datentypen werden angezeigt:

- In (Globals) definierte und im aktuellen Script referenzierte Variablen
- In Modulen definierte Variablen, die mit Use- und UseLSX-Anweisungen im aktuellen Script aufgerufen werden
- Im Abschnitt (Declarations) des aktuellen Objekts deklarierte Variablen.

{button ,AL(`H_IDE_VIEWING_GLOBALS_FOR_THE_CURRRENT_SCRIPT_STEPS_RT;H_IDE_CREATING_DECLARATIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_VARIABLES_PANEL_OVER;',0)}
Siehe Verwandte Themen

Variablenwerte im IDE ändern

Sie können den Wert einer Variablen beim Debuggen eines Scripts ändern.

Ein Variablenwert wird wie folgt beim Debuggen geändert

1. Im Drop-down-Listenfeld Calls im Script-Debugger wählen Sie die Prozedur, die die Wertdefinition enthält.
2. Im Feld Variablen wählen Sie die Variable, deren Wert geändert werden soll.

   Wenn der Variablenwert geändert werden kann, ist das Textfeld Wert aktiviert.
3. Geben Sie in das Textfeld Wert einen neuen Wert für die Variable ein.

   Wenn eine Variable mehr als 1024 Zeichen enthält, können Sie nur die ersten 1024 Zeichen im Textfeld anzeigen und bearbeiten.
4. Drücken Sie Return, oder klicken Sie auf das Kontrollkästchen neben dem Textfeld.

   Der Wert für die Variable wird geändert.

{button ,AL(`H_IDE_CHANGING_VARIABLE_VALUES_IN_THE_IDE_STEPS_RT;H_IDE_CREATING_DECLARATIONS_IN_THE_SCRIPT_EDITOR_OVER;H_IDE_USING_THE_IDE_VARIABLES_PANEL_OVER;',0)} Siehe Verwandte Themen

Überblick: Online-Hilfe im IDE

Das Hilfesystem enthält Informationen über das IDE. Im Hilfesystem wird beschrieben, wie Sie Scripts erstellen und testen. Sie können das Hilfesystem aufrufen, indem Sie mit F1 einen kontextbezogenen Hilfetext für den gerade aktiven Teil des IDE-Fensters aufrufen oder indem Sie im Menü ? ein Thema wählen.

Kontextbezogene Hilfe

Im IDE können Sie für folgende Elemente kontextbezogene Hilfetexte aufrufen:

- Die Ausschnitte Script-Editor und Script-Debugger
- Die Felder Browser, Breakpoints, Ausgabe und Variablen
- LotusScript-Schlüsselwörter und Anwendungskomponenten (sowie OLE Automation-Klassen, für die Hilfetexte verfügbar sind), die im Script-Editor, Script-Debugger und Browser angezeigt werden
- Kompilier-Fehlermeldungen, die im Drop-down-Listenfeld Fehler im Script-Editor angezeigt werden
- Laufzeitmeldungen, die in Meldungsfeldern im Script-Debugger angezeigt werden

Hilfe Übersicht und Suchen

Sie können auch das IDE-Hilfethema Übersicht und Suchen verwenden, um Hilfetexte für Fehlermeldungen und Beschreibungen dafür anzuzeigen, wie Sie Menüs, die Tastatur und Maus, die Drop-down-Listenfelder Objekt und Script sowie die Felder Breakpoints, Browser, Ausgabe und Variablen verwenden, um Scripts zu erstellen und zu verwalten.

{button ,AL(`H_IDE_ONLINE_HELP_IN_THE_IDE_OVER_RT;H_IDE_GETTING_CONTEXTSENSITIVE_HELP_STEPS;H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER;H_IDE_GETTING_HELP_ON_THE_IDE_STEPS;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} Siehe Verwandte Themen

LotusScript-Hilfe im IDE öffnen

Im Script-Editor oder Script-Debugger

1. Wählen Sie ein LotusScript-Schlüsselwort.
2. Drücken Sie F1.

   Daraufhin wird ein Hilfetext für das Schlüsselwort angezeigt.

Im Browser

1. Wählen Sie einen Eintrag für ein LotusScript-Schlüsselwort.
2. Drücken Sie F1.

   Daraufhin wird ein Hilfetext für das Schlüsselwort angezeigt.

{button ,AL(`H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS_RT;H_IDE_GETTING_CONTEXTSENSITIVE_HELP_STEPS;H_IDE_ONLINE_HELP_IN_THE_IDE_OVER;',0)} Siehe Verwandte Themen

Kontextbezogene Hilfetexte im Script-Editor, Debugger und Browser aufrufen

Im Script-Editor oder Script-Debugger

1. Wählen Sie ein LotusScript-Schlüsselwort oder den Namen einer Anwendungsklasse, Methode, Eigenschaft oder anderen Komponente.
2. Drücken Sie F1.

   Daraufhin wird ein Hilfetext für das Schlüsselwort oder die Anwendungskomponente angezeigt.

Im Browser

1. Wählen Sie einen Eintrag für ein LotusScript-Schlüsselwort oder einen Eintrag für eine Anwendungsklasse, Methode, Eigenschaft oder andere Komponente.
2. Drücken Sie F1.

   Daraufhin wird ein Hilfetext für das Schlüsselwort oder die Komponente angezeigt.

{button ,AL(`H_IDE_GETTING_CONTEXTSENSITIVE_HELP_STEPS_RT;H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER;H_IDE_GETTING_HELP_ON_THE_IDE_STEPS;H_IDE_ONLINE_HELP_IN_THE_IDE_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} Siehe Verwandte Themen

Überblick: Hilfetexte zu Fehlermeldungen im IDE aufrufen

Für Kompilier- und Laufzeitfehler, die im IDE angezeigt werden, können Sie kontextbezogene Hilfetexte aufrufen. Kompiliermeldungen werden im Drop-down-Listenfeld Fehler im Script-Editor angezeigt. Markieren Sie die Meldung, und drücken Sie F1, um den Hilfetext für eine dieser Meldungen anzuzeigen.

Laufzeitmeldungen werden in Meldungsfeldern angezeigt. Drücken Sie F1, während das Meldungsfeld angezeigt wird, um den Hilfetext für eine dieser Meldungen anzuzeigen.

Sie können auch in der IDE-Hilfe Hilfetexte für diese Fehlermeldungen aufrufen, wenn Sie nach "Messages" suchen und zu dem Hilfethema IDE Error Messages gehen. Dieses Thema enthält eine Liste mit Kompilier- und Laufzeitmeldungen. Wenn Sie eine Meldung wählen, wird ein Hilfetext für die Meldung angezeigt.

{button ,AL(`H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER_RT;H_IDE_GETTING_CONTEXTSENSITIVE_HELP_STEPS;H_IDE_IDE_ERROR_MESSAGES_OVER;H_IDE_ONLINE_HELP_IN_THE_IDE_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} Siehe Verwandte Themen

Hilfetexte im IDE aufrufen

Allgemeine Informationen für den Script-Editor oder Script-Debugger werden folgendermaßen angezeigt
Klicken Sie an einer beliebigen Stelle im Ausschnitt Script-Editor oder Script-Debugger (außer in einem Script), und drücken Sie F1.

Allgemeine Informationen für die Felder Breakpoints, Browser, Ausgabe und Variablen werden folgendermaßen angezeigt
Klicken Sie an einer beliebigen Stelle in einem Feld, und drücken Sie F1.

Wenn Sie auf ein Element in einer Browser-Liste klicken und F1 drücken, wird ein Hilfetext für das Element angezeigt.

Informationen über Menüs, Drop-down-Listenfelder und Aktivitäten, die Sie ausführen können, werden folgendermaßen angezeigt
Wählen Sie Hilfe Suchen oder das IDE-Hilfethema Übersicht.

{button ,AL(`H_IDE_GETTING_HELP_ON_THE_IDE_STEPS_RT;H_IDE_GETTING_CONTEXTSENSITIVE_HELP_STEPS;H_IDE_GETTING_HELP_FOR_ERROR_MESSAGES_IN_THE_IDE_OVER;H_IDE_ONLINE_HELP_IN_THE_IDE_OVER;H_IDE_OPENING_LOTUSSCRIPT_HELP_IN_THE_IDE_STEPS;',0)} Siehe Verwandte Themen