

```

*** System error handling failed ***
Original error: MessageNotUnderstood: ToolRegistry>>openBrowser.
Smalltalk tools debugger error: MessageNotUnderstood: EyeInspector class>>ownerChanged:
MorphicUIManager>>onPrimitiveError:
SpecDebugger class(Object)>>primitiveError:
[ :ex ]
Smalltalk tools debugger
primitiveError:
Original error: ' , title asString
?
Smalltalk tools debugger error: '

[[ ex description ]
on: Error
do: [ 'a ' , ex class printString ] , ':' ] in openDebuggerOn: process context: context label: title contents: contentsStringOrNil fullView: bool
"Open a notifier in response to an error, halt, or notify. A notifier view
just shows a short view of the sender stack and provides a menu that
lets the user open a full debugger."

| errorWasInUIProcess debugger |
errorWasInUIProcess := self spawnNewProcessIfThisIsUI: process.
[
Smalltalk tools debugger logDebuggerStackToFile
ifTrue: [ Smalltalk logError: title inContext: context ] ]
on: Error
do: [ ].
debugger := Smalltalk tools debugger new.
self
defer: [
[
debugger process: process controller: nil context: context."schedule debugger in deferred UI message to address
redraw problems after opening a debugger e.g. from
the testrunner.""self defer: [
bool
ifTrue: [ debugger openFullNoSuspendLabel: title ]
ifFalse: [ debugger openNotifierContents: contentsStringOrNil label: title ].
debugger errorWasInUIProcess: errorWasInUIProcess ]
on: Error
do: [ :ex ]
Smalltalk tools debugger
primitiveError:
Original error: ' , title asString
?
Smalltalk tools debugger error: '

[[ ex description ]
on: Error
do: [ 'a ' , ex class printString ] , ':' ] ].
process suspend.
^ debugger in MorphicUIManager>>openDebuggerOn:context:label:contents:fullView:
BlockClosure>>cull:
[ self exceptionHandlerBlock cull: exception ] in MethodContext(ContextPart)>>handleSignal:
BlockClosure>>ensure:
MethodContext(ContextPart)>>handleSignal:
MethodContext(ContextPart)>>handleSignal:
MethodContext(ContextPart)>>handleSignal:
MethodContext(ContextPart)>>handleSignal:
MethodContext(ContextPart)>>handleSignal:
MessageNotUnderstood(Exception)>>signal
EyeInspector class(Object)>>doesNotUnderstand: #ownerChanged
InspectorNavigator>>basicInspector:
InspectorNavigator>>inspector:
InspectorNavigator>>inspect:
SpecDebugger>>updateInspectorsFromContext:
[ :aContext ]
self updateCodeFromContext: aContext.
self updateInspectorsFromContext: aContext.
self stack
expand;
takeKeyboardFocus"This is needed, as otherwise, the inspectors will take the focus" ] in SpecDebugger>>initializePresenter
BlockClosure>>cull:
BlockClosure>>cull:cull:
-----
Type CR to enter an emergency evaluator.
Type any other character to restart.

```