

# ActiveX Component SoftwareAG.NaturalX.Utilities

This chapter covers the following topics:

- Purpose
  - Interfaces
- 

## Purpose

The ActiveX component `SoftwareAG.NaturalX.Utilities` provides a number of methods that are useful in the context of NaturalX and Natural Studio plug-ins.

As an example, the general usage of the component in a Natural application is in the following way.

```
define data
local
1 #util handle of object
1 #studio handle of object
end-define
*
* First create an instance of the class SoftwareAG.NaturalX.Utilities.
create object #util of 'SoftwareAG.NaturalX.Utilities.4'
if #util eq null-handle
  escape routine
end-if
*
* Now call the individual methods of the component, for instance
* to get access to the Natural Studio Automation Interface.
*
send 'GetThisNaturalStudio' to #util return #studio
if #studio eq null-handle
  escape routine
end-if
*
end
```

This example will run successfully only if it is executed in a Natural Studio session. If it is executed in a Natural runtime session or under the Natural debugger, the call to `GetThisNaturalStudio` will return `NULL-HANDLE`. This is because only a Natural Studio session has an `INatAutoStudio` interface. If the program is running under the Natural debugger, it is effectively executed in a Natural runtime session outside Natural Studio; therefore, the call to `GetThisNaturalStudio` will also return `NULL-HANDLE`.

To make this example run also under the Natural debugger, modify it as follows, in order to retrieve the `INatAutoStudio` interface of the Natural Studio session:

```
define data
local
1 #util handle of object
1 #studio handle of object
1 #rot handle of object
1 #ro (a) dynamic
```

```

end-define
*
* First create an instance of the class SoftwareAG.NaturalX.Utilities.
create object #util of 'SoftwareAG.NaturalX.Utilities.4'
if #util eq null-handle
  escape routine
end-if
*
* Now call the individual methods of the component, for instance
* to get access to the Natural Studio Automation Interface.
*
send 'GetThisNaturalStudio' to #util return #studio
if #studio eq null-handle
* We might be in a debugging session.
* Try to locate the Natural Studio session
* from which the debugger has been started.
* Retrieve the running objects table.
  send 'GetRunningObjects' to #util return #rot
  if #rot eq null-handle
    escape routine
  end-if
* Iterate across the running objects table.
  repeat
    send 'Next' to #rot return #ro
    if #ro eq ' '
      escape bottom
    end-if
* If we hit a running Natural Studio session, we access it.
    if substring(#ro,1,13) eq 'NaturalStudio'
      send 'BindToObject' to #util
      with #ro (ad=o) return #studio
      escape bottom
    end-if
  end-repeat
end-if
*
end

```

## Interfaces

The individual interfaces, their methods and their usage are described in detail in separate documents.

The component provides the following interfaces:

- Interface INaturalXUtilities
- Interface IRunningObjects