6.2.2 Named Pipe

Eine Named Pipe bietet einen unidirektionalen Kommunikationkanal (halb-duplex) zwischen Prozessen innerhalb des selben Systems. Es ist ein wesentliches Merkmal einer Named Pipe, dass die fließenden Daten transient (flüchtig) sind. Sobald Daten gelesen wurden, können diese danach nicht noch einmal gelesen werden.

```
hNamedPipe = PIPE sNamedPipePath FOR [ READ ] [ WRITE ] [ WATCH ]
hNamedPipe = OPEN PIPE sNamedPipePath FOR [ READ ] [ WRITE ] [ WATCH ]
```

Die beiden gleichwertigen Anweisungen öffnen eine Named Pipe zum Lesen, zum Schreiben oder für beides. Wenn die Pipe nicht existiert, wird sie automatisch erzeugt.

- Wenn das READ-Schlüsselwort angegeben ist, wird die Pipe zum Lesen geöffnet.
- Wenn das Schlüsselwort WRITE angegeben wurde, dann, wird die Pipe zum Schreiben geöffnet.
- Wenn das Schlüsselwort WATCH angegeben wird, wird die Pipe vom Interpreter überwacht. Wenn Daten aus der Pipe gelesen werden können, dann wird der Eventhandler File_Read() aufgerufen. Wenn Daten in die Pipe geschrieben werden können, so wird der Eventhandler File_Write() aufgerufen.
- Wenn die Named Pipe erfolgreich geöffnet wurde, wird ein Stream-Objekt zurückgegeben.

Beispiel:

```
Public hNamedPipe As File
Public sNamedPipePath As String

Public Sub Form_Open()
  sNamedPipePath = Temp("FIFO1")
  hNamedPipe = Pipe sNamedPipePath For Read Watch
  ...
End
```

Hinweise:

- Gambas implementiert eine Named Pipe nach dem First-In-First-Out-Prinzip (FIFO-Speicher).
- Eine Named Pipe ist eine spezielle Datei, die einen Pfad (Namen) im Dateisystem hat. Zwei Prozesse können sich zu dieser Datei verbinden und Daten unidirektional versenden oder empfangen.
- Das Öffnen einer Named Pipe zum Lesen blockiert normalerweise die Anwendung, bis ein anderer Prozess die gleiche Named Pipe zum Schreiben öffnet. Allgemein gilt: Eine Named Pipe muss an beiden Enden gleichzeitig geöffnet sein.

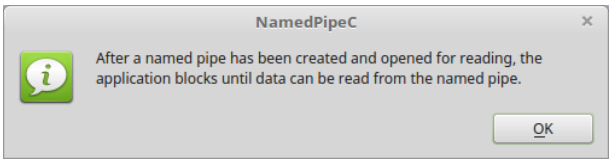

Abbildung 6.2.2.1: Hinweis

Die Arbeitsweise von Pipes kennen Sie bereits von der sequentiellen Abarbeitung von Befehlen in einer Konsole. Die Ausgabe des Befehls ls im folgenden Beispiel ist eine Liste aller Dateien im Ordner BildTon/Blumen einschließlich aller Unterordner (-R), die als Eingabe für den Befehl grep verwendet wird. Der Befehl grep zählt die Anzahl der png- und PNG-Dateien – festgelegt über die Optionen -i und -c und dem Filter .png. Damit das funktioniert, steht zwischen beiden Befehlen das Pipe-Symbol '|':

```
hans@mint-183 ~ $ ls -R ~/BildTon/Blumen | grep -ic '\.png'
78
```

6.2.2.1 Projekt

Das folgende Projekt NamedPipeC repräsentiert eine Named Pipe als Objekt, das Daten der Ausgabe eines Prozesses in einer Konsole als Eingabe – ähnlich zum oberen Beispiel – liest und verarbeitet.

Der Quelltext wird vollständig angegeben:

```
[1]  ' Gambas class file
[2]
[3]  Public sNamedPipePath As String
[4]  Public hNamedPipe As File
[5]
[6]  Public Sub Form_Open()
[7]    FMain.Resizable = True
[8]    sNamedPipePath = "/tmp/LS2FIFO"
[9]    If Exist(sNamedPipePath) Then Kill sNamedPipePath
[10] End
[11]
[12] Public Sub btnCreateNamedPipe_Click()
[13]   txaOutput.Clear()
[14]   btnCreateNamedPipe.Picture = Picture["icon:/16/apply"]
[15]   btnCreateNamedPipe.Text = "Named Pipe was created and waits for data to read ..."
[16]   Wait
[17]   CreateNamedPipe()
[18] End
[19]
[20] Public Sub CreateNamedPipe()
[21]
[22]   Dim sLine As String
[23]
[24]   hNamedPipe = Open Pipe sNamedPipePath For Read
[25]   Do
[26]     Read #hNamedPipe, sLine, -256
[27]     If Not sLine Then Break
[28]     txaOutput.Insert(sLine)
[29]   Loop
[30]   If Exist(sNamedPipePath) Then Kill sNamedPipePath
[31]   btnCreateNamedPipe.Picture = Picture["icon:/16/package"]
[32]   btnCreateNamedPipe.Text = "Create Named Pipe"
[33]   If hNamedPipe Then Close hNamedPipe
[34] End
[35]
[36] Public Sub btnGetInformation_Click()
[37]   Message.Info("After a named pipe has been created and opened for reading, the application blocks un-
       til data can be read from the named pipe.")
[38] End
[39]
[40] Public Sub Form_Close()
[41]   If Exist(sNamedPipePath) Then Kill sNamedPipePath
[42] End
```

Hinweise:

- Zuerst starten Sie das Programm, indem Sie die ausführbare Datei *read_from_namedpipe.-gambas* öffen. In der Prozedur *CreateNamedPipe()* in den Zeilen 20 bis 36 wird eine Named Pipe zum Lesen erzeugt (Zeile 24). Damit wird das Programm jedoch blockiert, da keine Daten von der Named Pipe gelesen werden können. Der Hinweis 'Nachdem eine Named Pipe angelegt und zum Lesen geöffnet wurde blockiert die Anwendung bis Daten aus der Named Pipe gelesen werden können.' weist auf diese Besonderheit hin! Sie erkennen das Blockieren unter anderem daran, dass Sie das Programm nicht beenden können.

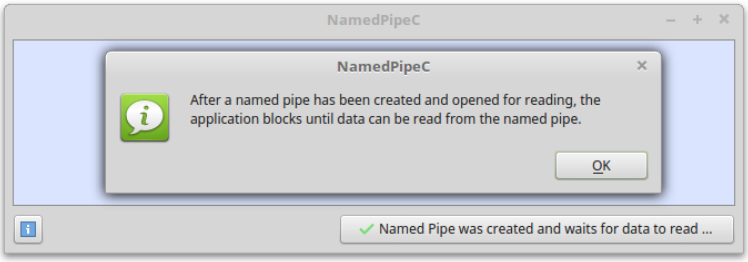

Abbildung 6.2.2.1.1: Hinweis

- Dann wechseln Sie in ein Verzeichnis Ihrer Wahl in Ihrem Home-Verzeichnis. Öffnen Sie dort eine Konsole und geben Sie ls -l > /tmp/LS2FIFO ein. Damit wird die Ausgabe von ls -l in die Pipe umgeleitet und in der Named Pipe – in der speziellen Datei LS2FIFO – werden die Daten gespeichert.

- Abschließend sehen Sie die Daten aus dem ersten Prozess in der TextArea des Programms

NamedPipeC, da nun Daten von der Pipe gelesen werden konnten (Zeilen 25 bis 29) und das Programm nicht mehr blockiert:

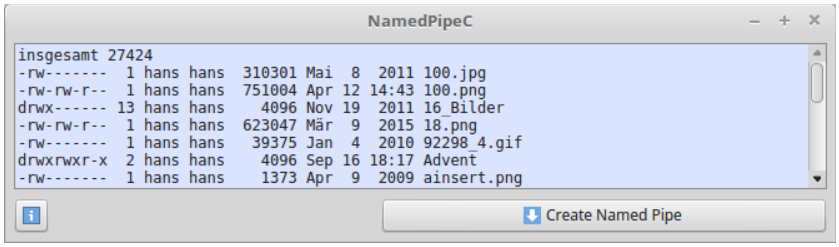

Abbildung 6.2.2.1.2: Anzeige der Daten aus der Named Pipe

- Eine Named Pipe ist ein flüchtiger FIFO-Speicher, deshalb ist er nach dem Auslesen von Daten leer!