

21.1 Klasse Process

Die Klasse *Process* wird für die Verwaltung von Prozessen genutzt, die durch die Instruktionen EXEC oder SHELL gestartet werden. Wenn Sie eine der beiden Instruktionen aufrufen, übersetzt das der Interpreter in den Aufruf einer Routine aus einer Systembibliothek mit der er verlinkt ist. Der Interpreter startet einen Child-Prozess.

Aus der Process-Klasse können Sie selbst *kein* Objekt erstellen, wie es zum Beispiel mit anderen Klassen und der NEW-Instruktion möglich ist. Dies gelingt ausschließlich über den Aufruf der Instruktionen EXEC und SHELL. Mit einer Anweisung wie der folgenden können Sie zum Beispiel einen Prozess starten:

```
ProzessVariable = SHELL sCommand FOR READ WRITE AS "MyProcess"
```

um dann u.a. mit *MyProcess_Read* und *MyProcess_Error* die Ausgaben dieses Prozesses abzufangen, auszuwerten und anzuzeigen sowie darauf zu reagieren, weil Sie ihn *selbst* gestartet haben. Da die Klasse *Process* von der (Basis-)Klasse *Stream* abgeleitet ist, können Sie mit deren Input/Output-Anweisungen arbeiten, um von der Standard-Ausgabe zu lesen oder direkt Daten an den Prozess schicken. Um die Fehlerausgaben des gestarteten Prozesses zu lesen und auszuwerten, müssen Sie das *Error-Ereignis* nutzen.

Die Klasse *Process* besitzt neben dynamischen Eigenschaften auch statische, die von allen Instanzen der Klasse geteilt werden. Diese sind:

- *LastState* → Der Status des zuletzt beendeten Process-Objektes
- *LastValue* → Rückgabe-Wert des zuletzt beendeten Process-Objektes

Die dynamischen, jedem Objekt eigenen, Eigenschaften der Klasse *Process* sind in der folgenden Tabelle erklärt:

Eigenschaft	Beschreibung
Handle	Die Prozess-ID (PID) des Prozesses auf dem System
Id	Synonym für Handle
Status	Der aktuelle Status des Prozesses (4 Werte)
Value	Der Rückgabe-Wert des Prozesses, sobald dieser beendet wurde (0=erfolgreich, Werte größer als Null weisen auf einen Fehler hin)

Tabelle 21.1.1: Eigenschaften der Klasse Process

Die Eigenschaft *State* zeigt den Status eines Prozesses an. Die Klasse *Process* stellt die folgenden symbolischen Konstanten bereit, um verschiedene Prozess-Zustände zu reflektieren:

Status-Konstante	Numerischer Wert	Beschreibung
Stopped	0	Ein Prozess wurde gestoppt
Running	1	Ein Prozess wurde gestartet und läuft
Crashed	2	Ein Prozess ist abgestürzt
Signaled	2	Synonym für Crashed

Tabelle 21.1.2: Prozesszustände – Konstanten und numerische Werte

Als Methoden zur Prozesssteuerung stellt Ihnen die Klasse nur zwei nutzbare bereit:

Methode	Beschreibung
Kill	Beenden eines Prozesses über das Signal SIGKILL
Wait	Warten auf das Ende eines Prozesses

Tabelle 21.1.3: Methoden der Klasse Process

Ein Prozess kennt die Ereignisse:

Ereignis	Beschreibung
Kill	Das Ereignis wird ausgelöst, wenn der Prozess beendet wurde.
Error	Dieses Ereignis wird ausgelöst, wenn ein Fehler auftrat und Daten von der Standard-Fehlerausgabe des Prozesses (indirekt) gelesen werden können.
Read	Das Ereignis wird ausgelöst, wenn Daten von der Standard-Ausgabe des Prozesses gelesen werden können.

Tabelle 21.1.4: Ereignisse der Klasse Process