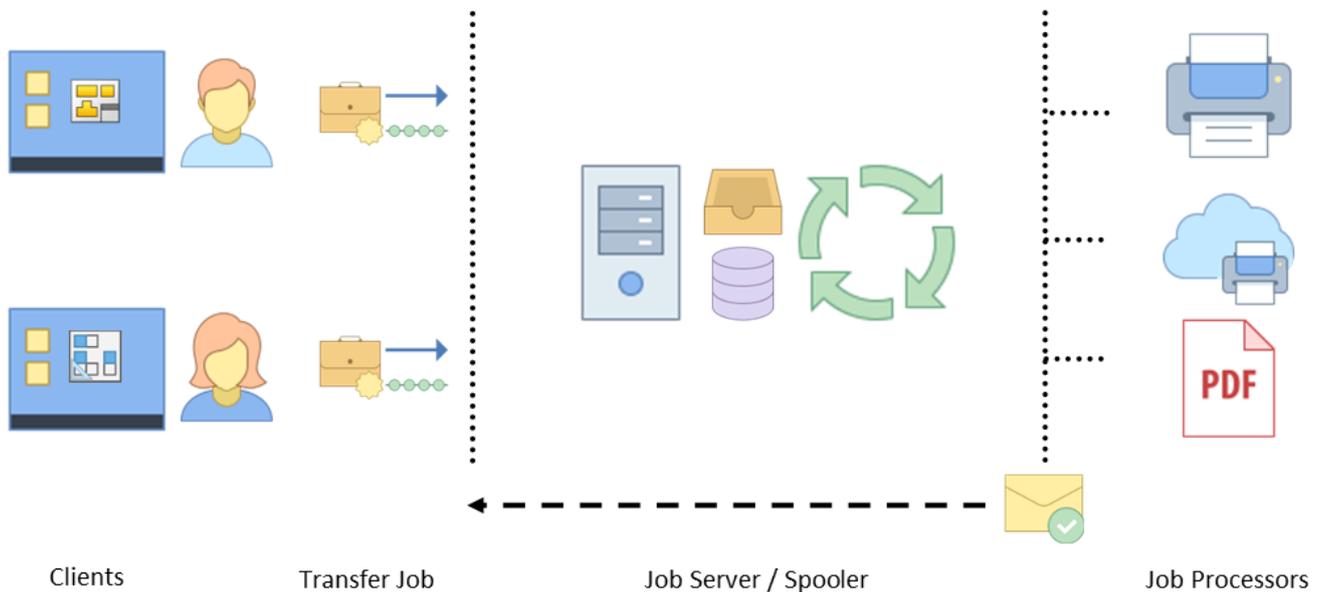


Jobserver

Der speedy-Jobserver erledigt zeitaufwendige Druckaufträge oder erstellt Vorschaubilder für speedWEB an einer zentralen Stelle.



Um einen Jobserver einzurichten wird der Spooler-Mechanismus genutzt. Jobs können nur abgearbeitet werden wenn die entsprechende Konfiguration erstellt wurde und die entsprechenden Module vorhanden sind. Es wird eine Transferdatei genutzt um die Jobs an den Jobserver zu übergeben. Diese Datei hat die Endung *.speedyjob. Wird ein Jobserver verwendet werden die Jobs auf dem Server ausgeführt.

Dies bedeutet:

- Bei Druckjobs muss das verwendete CAD System vorhanden sein
- Entsprechende Leistung (CPU, Arbeitsspeicher, Grafik, ...) muss auf dem Server vorhanden sein

Momentan verfügbare Jobs:

- jsPlot: Stapeldruck auf dem Jobserver
- jsRendition: Bildgenerierung/Vorschaugenerierung auf dem Jobserver
- weitere Jobs auf Anfrage bzw. in Vorbereitung

Ablauf des Jobserver

Der Jobserver überwacht den Posteingang. In den meisten Fällen ein entsprechendes freigegebenes Verzeichnis.

Sendet ein Client eine Jobdatei (*.speedyjob) in den Posteingang des Jobservers wird diese eingelesen und in der Warteschleife (js_jobs in der Datenbank) eingetragen.

Die Warteschleife wird abgearbeitet in dem zu jedem Job der passende Processor gestartet wird, wenn der Job an der Reihe ist.

Im Fall von Druckjobs ist die Stapeldruck-Einstellung im Job enthalten. Der Druckjob wird wie der Stapeldruck lokal nur auf dem Server ausgeführt. Die entsprechenden Anwendungen müssen auf dem Server zur Verfügung stehen.

Konfiguration

Um den Spooler als Jobserver zu Nutzen ist folgende Konfiguration in der dwSpool.ini zu machen:

[WATCHDIR]

WatchDirX = „Pfad für die Ablage der Jobdateien“ (siehe auch [*.jobserver.path])

[WatchDirX.SPEEDYJOB]

Filter=*.speedyjob

Description=

Command=%1

ShowWindow = 1

Dies dient als Beispiel und kann ihrer Konfiguration angepasst werden.

Desweiteren werden 3 Tabellen in der Datenbank benötigt:

- js_jobs
- js_proc
- js_types

Diese 3 Tabellen nutzt der Jobserver um die Jobs abzuarbeiten.

In der **js_types** Tabelle werden die zur Verfügung stehenden Jobs definiert

Schlüssel	Beschreibung
types_id	ID des Typs
types_name	Name des Types. Beispiel: plot→Beschreibt den Druckjob
types_parallelcount	Beschreibt wieviele Prozesse von diesem Typ gleichzeitig gestartet werden können
types_maxtime	Maximale Zeit die ein Job brauchen darf(in Millisekunden). Nach dieser Zeit wird der Job beendet. Dies kann genutzt werden um aufgehängte Jobs neu starten zu können
types_killable	Gibt an ob nach abgelaufener maxtime der Prozess beendet werden kann oder nicht
types_maxrestart	Max Wert wie oft ein Prozess erneut gestartet werden kann
types_settinghead	Gibt den Knoten der Eigenschaften an die zu diesem Job gehören. Beispiel: plot→ dadurch werden die Einstellungen plot.jobserver eingelesen und verwendet. *.jobserver.available gibt an ob der Jobserver aktiv ist. *.jobserver.path gibt an welches Verzeichnis überwacht werden soll.
types_exepath	Vollständiger Pfad zur Job-Anwendung.
types_flag	Allgemeines Flag

Beispielkonfiguration eines Drucktyps:

types_id	types_name	types_parallelcount	types_maxtime	types_killable	types_maxrestart	types_settinghead	types_exepath	types_flag
2	plot	1	50.000	1	1	plot	\\.\pipe\plot\jobserver\jsPlot.exe	0

Konfigurationsparameter

Name	Beschreibung
jobserver.database.keepAll	
jobserver.archiveDir	
jobserver.errorDir	

From:

<https://wiki.speedy-pdm.de/> - **speedyPDM - Wiki**

Permanent link:

https://wiki.speedy-pdm.de/doku.php?id=speedy:30_modules:jobserver&rev=1623325515

Last update: **2021/06/10 13:45**

