
Batch processors

Real Life Examples
How to install
Scheduler Plugin
Set up a structure
Sequential/parallel scheduling
Demo

Real Life Examples

- action reminders
 - send daily (email)message to users with top 10 todo list
- reminders for subscription renewals
 - 6, 4, 2 weeks prior to a date
- monthly email reports for sales, finance
 - send email as html/pdf attachm.
- sales follow up on product downloads
 - ask for evaluation comments after x days.
- monthly invoicing of Service License Agreements
 - fixed price, basically only order date is variable
- daily currency rates
 - Populates a central table used for invoicing and accounting
- clean up data
 - removal of double contact entries
 - removal of invalid/non confirmed orders/proposals
 - move data to history tables


Conclusion

- Besides handling bulk data, batch processors are very useful to perform REPETITIVE tasks
 - Relief of workload for co workers
 - The system handles those BORING tasks
 - accurate
 - much faster
 - on time

How to install

- a. Build a solution with scripts that do specific tasks
- b. Attach the solution as a batch processor to the Servoy server (admin page)

Menu



Servoy Server Home

- Network Settings
- Multi Developer
- Server Plugins
- Database Servers
- Clients
- Solutions
- Locks
- Transactions
- Users
- Batch Processors
- Upload Library

- Licenses
- Server Log
- Performance Data

Batch process clients (auto started on server startup, not in developer)

To create a new batch process (headless)client, fill in the form below. To remove an existing batch process click delete after the process name.

The Arguments must be a comma separated list which will be passes to solution start method , Username and Password must be a valid user having access to the solution

Create new batch processor client

Solution:

Arguments:

Username:

Password:

Password (again):

Batch process list

| | | |
|-------------|------------------------|---------------------------------------|
| Solution | batchprocessor1 | <input type="button" value="Remove"/> |
| Args | -none- | |
| User | admin | |
| Startup Msg | -empty- | |
| Status | Is running | <input type="button" value="Stop"/> |

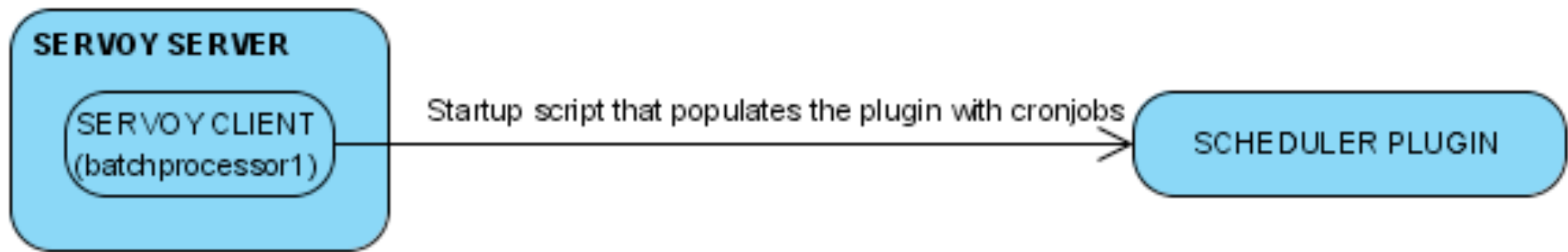
How to install

- Keep in mind that batch processor solutions
 - ..run as a client on the server without a GUI
 - ..automatically restart together with a server reboot
 - ..use the scheduler plugin to create time triggers. (cronjobs)
 - cronjobs run in client memory and are destroyed when ever client restarts.
- need a start up script that resets the cronjobs with the scheduler plugin
- start up script can either be (hard coded or table driven)

Scheduler plugin

- Hard coded startup script

```
// startup script
//plugins.scheduler.addCronJob(jobname , cronTimings , Function globalMethod , [Date startDate], [Date endDate], [arguments])
plugins.scheduler.addCronJob("myJob1", 5/10 * * * * ?, globals.SCDscript1 , null , null , ["Hi There", 5])
plugins.scheduler.addCronJob("myJob2", 0/10 * * * * ?, globals.SCDscript1 , null , null , ["Hello" , 1])
.....
```

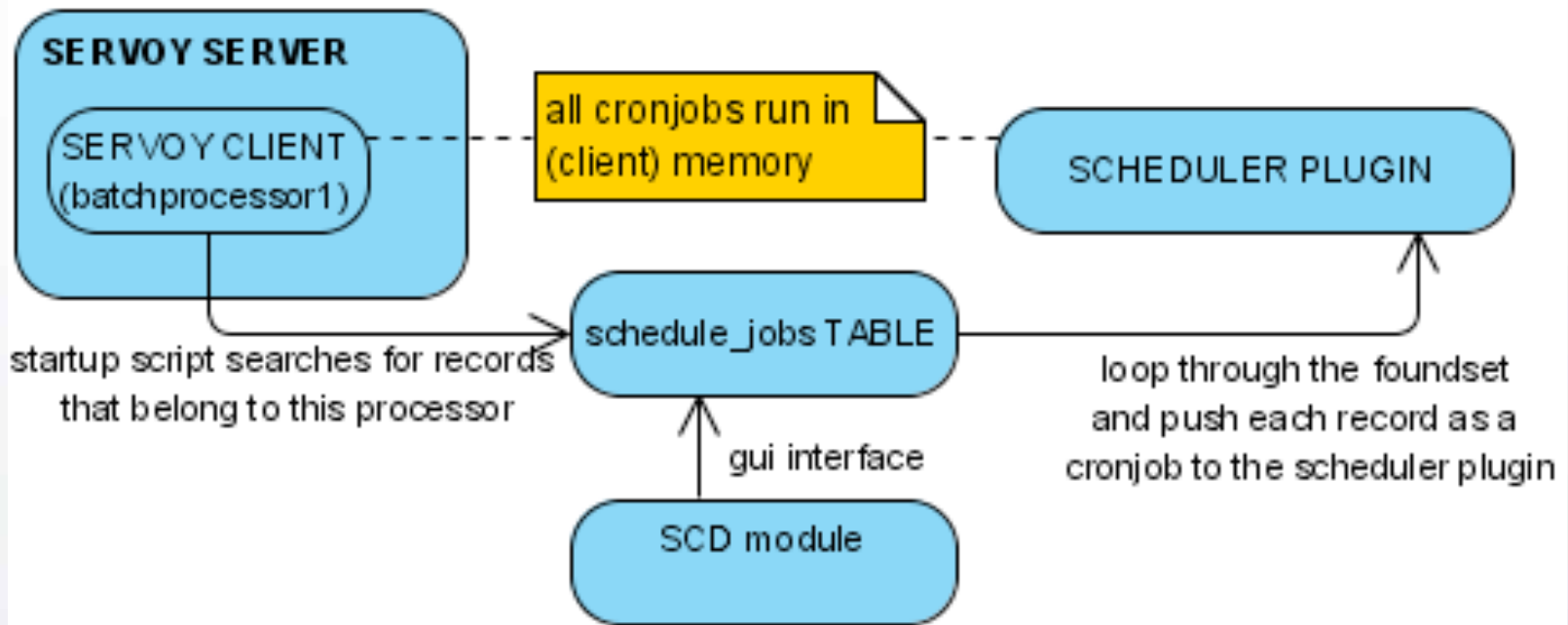


Disadvantages

- static
- no visual feedback (error handling, logging, arguments sent etc)
- encapsulated inside a script (no user accessibility)

Scheduler plugin

- Table driven startup



| ID | | solution name | job name | methodname | method arguments | cron timings |
|-----|---|-----------------|-------------------|---------------------|------------------|----------------|
| 183 | ▶ | batchprocessor1 | abc | SCDscript1 | 183 Hi There | 5/10 * * * * ? |
| 188 | ▶ | batchprocessor1 | short description | SCDscript1 | 188 Hello | 0/10 * * * * ? |
| 187 | ■ | batchprocessor2 | | svySCDsubscriptions | | 0 27 09 ? * * |




Scheduler plugin

- Cronjobs timer settings
 - <http://quartz.sourceforge.net/firstTutorial.html#cronTriggers>
 - ? = no specific value
 - * = every possible value
 - / = specify increments to values
 - - = specify a range
 - , = specify specific values

| sec | min | hrs | dayOfMonth | month | dayOfWk |
|---|-----|-------|------------|-------|---------|
| every Wednesday at 12:00 pm | | | | | |
| 0 | 0 | 12 | ? | * | WED |
| every 5 minutes, starting 1 min after the hour | | | | | |
| 0 | 1/5 | * | * | * | ? |
| 10:30, 11:30, 12:30, and 13:30, every Wed and Fri | | | | | |
| 0 | 30 | 10-13 | ? | * | WED,FRI |

Set up a structure

- SCD module as a GUI for controlling cronjobs
 - TableSide scripts:
 - createJobRecord() > controller.newRecord()
 - deleteJobRecord() > controller.deleteRecord()
 - setJobRecordActiveInactive()
 - simple toggle script that sets
 - » job_status = 1(active)
 - » job_status = 0 (inactive)

| ID | | solution name | job name | methodname | method arguments | cron timings |
|-----|---|-----------------|-------------------|---------------------|------------------|----------------|
| 183 |  | batchprocessor1 | abc | SCDscript1 | 183 Hi There | 5/10 * * * * ? |
| 188 |  | batchprocessor1 | short description | SCDscript1 | 188 Hello | 0/10 * * * * ? |
| 187 |  | batchprocessor2 | | svySCDsubscriptions | | 0 27 09 ? * * |

Set up a structure

- resetScheduler()

```
8 //loop through jobRecords and push them to scheduler Plugin
9 for(var i=1 ; i<=foundset.getSize() ; i++)
10 {
11     foundset.setSelectedIndex(i)
12     if(job_status==1)
13     {
14         startJobInScheduler();
15     }
16 }
```

- startJobInScheduler() //subscript

```
6 plugins.scheduler.addCronJob(schedule_job_id+"", cron_timings, globals[methodname],start_date , end_date, vArray)
```

Set up a structure

- General concept
 - A. create, edit, delete a jobRecord (table side)
 - B. push records as cronjobs to the scheduler plugin
 1. resetScheduler()
 - removes all cronJobs from scheduler plugin
 - adds records as cronJobs again to the scheduler plugin

Set up a structure

- “batchprocessor1” solution
 - runs as a client on the servoy server
 - has SCD solution as module attached
 - Has its own startup script that searches for all records with `solution_name = “batchprocessor1”`
 - calls `resetScheduler()`, which removes all its cronJobs, and creates new ones based on the foundset.

Set up a structure

startup script in batchprocessor1

```
1 forms.svySCDjobs.controller.find()
2 forms.svySCDjobs.solution_name = "batchprocessor1"
3 forms.svySCDjobs.controller.search()
4
5 // loop through the foundset and push all record information to the cronjobs in the schedule plugin
6 // cronjobs always run in memory in the batchprocessor (client side)
7 forms.svySCDjobs.resetScheduler();
8
9
10 // refresh the cronjob memory every 7 seconds by running this startup script again
11 // This is done to push changes in the SCD module records to the scheduler plugin again.
12 plugins.scheduler.addCronJob("reset", "0/7 * * * * ?", globals.startup)
```

Sequential/parallel scheduling

- SEQUENTIAL
 - Schedule your jobs in a sequential pattern
 - A batch processor is capable of handling scripts fired at the same time, but will execute them in a sequence.
- PARALLEL
 - use a 2nd batch processor when 1 isn't capable of doing all jobs within a given amount of time
 - watch out for simultaneous editing of same records. Unpredictable results

Important!

- Don't forget to FLUSH the server and RESTART a batch processor whenever you make changes to scripts using Developer. Remember that a batch processor, although started on the server machine, is running in a client!

DEMO
