SimpleBatch

**Contents:**

1. What is simple\_batch?
2. Simple\_batch feature
3. Simple\_batch syntax and schedule information
4. Simple\_batch configuration and testing
5. **What is simple\_batch?**

Simple\_batch is a basic of batch system that process on operating system. It is similar to a shell script, a text file containing a series of cammands to be executed by the command interpreter.

1. **Simple\_batch feature:**

SimpleBatch’s feature are:

+ Unix cron schedule such that you can declare.

Unix cron schedule is a crontab(schedule information for check and run batch file.

+ Config file modification takes effect immediately.

+ Does not require complex batch system

1. **Simple\_batch syntax and schedule information**
   1. Simple\_batch syntax:

name = name of the test batch batch #

batchClass = jex.test.batch.JexSimpleBatchTest @ execute # class @ batch method

inputClass = jex.mmaa.domain.ido.mmmaap\_ct\_memchek.DMMAAP\_CT\_MEMCHEKIn # batch input domain

schedule = \* \* \* \* \* # batch schedule information

[INPUT] # input value declaration

I\_JUMIN\_NO = 750823111112 # KEY = VALUE declared

* 1. Simple\_batch schedule information (schedule=\* \* \* \* \*) is used to manage the process of batch file.





\*(first star) Minutes sub-pattern: the limit time between each task should be started. Values ​​are 0-59 and default value is 1min.

\* \*(second star) Hours sub-pattern: What time of the day indicates whether the task should be started. Values ​​are 0-23

\* \* \*(third star) Days of month sub-pattern: a few days of the month indicates whether the task should be started. Values ​​are from 1 to 31. The last day of the month, you can use L to represent.

\* \* \* \*(forth star) Months sub-pattern: some months of the year indicates that the task should be started. The values ​​range from 1 (January) to 12 (December) to have a can and "jan," feb "," mar "," apr "," may "," jun "," jul "," aug ", "sep", "oct", "nov", "dec" is also acceptable, such as alias.

\* \* \* \* \*(fifth star) Days of week sub-pattern: any day of the week indicates that the task should be started. Value is 0 (Sunday) to 6 (Saturday) can have, "sun", "mon", tue "," wed "," thu "," fri "," sat "is the alias is allowed.

\* Character as a wildcard, which also allows the use of its characters each, depending on the sub-pattern "every minute of the hour", "all during the day," "every day of the month", "every month of the year "," any day of the week "means.

When you start the scheduler schedules all five parts of the pattern when the work will start conditions are satisfied.

Example:

Schedule=5 \* \* \* \*

This pattern is starting with five minutes to run the task every hour. (00:05, 01:05, 02:05, etc)

\* \* \* \* \*

This pattern is the task to run every minute.

\* 12 \* \* Mon

This pattern is at 12 noon Monday to run the job every minute.

\* 12 16 \* Mon

This pattern is similar to the above on Monday 12:00 hour job to run every minute, day of the month, only one day run the 16.

Each sub-pattern is a comma-separated values ​​that can have more than one.

59 11 \* \* 1,2,3,4,5

This pattern is a Mon, Tue, Wed, Thu, Fri 11:59 am to execute the task.

You can also specify the range of values ​​'-' characters.

59 11 \* \* 1-5

This pattern is the same as the pattern of meaning.

The '/' character to use, you can set the repeat interval in the range. \* / C and ab / c you can use the form. In this case, each sub-pattern is from 0 to the maximum value or a sheet of b to c for each matched.

\* / 5 \* \* \* \*

This pattern is the task to runs every 5 minutes. (0:00, 0:05, 0:10, 0:15, etc.)

3-18/5 \* \* \* \*

This pattern every hour let job to runs every 5 minutes starting at 18 minutes to 3 minutes repeated. (0:03, 0:08, 0:13, 0:18, 1:03, 1:08, etc.)

\* / 15 9-17 \* \* \*

This pattern is every 15 minutes 9:00 to 17:00 during the task execution. (9:00, 9:15, 9:30, 9:45, etc. The last execution time is 17:45)

They can be used in conjunction with these rules.

\* 12 10-16/2 \* \*

This pattern is at 12 o’clock let job to run every minute date is the 10th of the month, 12 days, 14 days, only one day run the 16.

\* 12 1-15,17,20-25 \* \*

This pattern is at 12 o’clock let job to run every minute, and the date is 1, 15, or 17, or 20 to 25 when executed.

Finally, the '|' character using a different scheduling patterns can be combined into one.

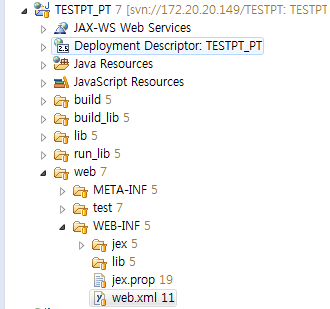
0 5 \* \* \* | 8 10 \* \* \* | 22 17 \* \* \*

This pattern is a daily 05:00, 10:08, 17:22 and run the job.

1. **Simple\_batch configuration and testing**

**4.1. Configuration:**

**4.1.1**: WAS(tomcat project) start it need to declare Servlet to web.xml config file



Copy this servlet to web.xml

<servlet>

<servlet-name>JexSimpleBatch</servlet-name>

<servlet-class>jex.simplebatch.JexSimpleBatchServlet</servlet-class>

<init-param>

<param-name>BATCH\_INFO</param-name>

<param-value>TESTPT</param-value>

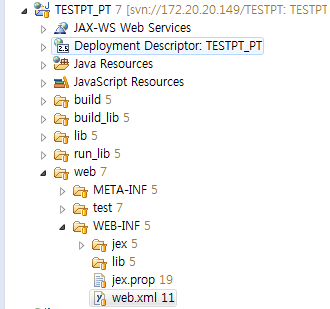
</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<param-value>project\_name</param\_value>

**4.2**: Jex property need to declare JEX\_BATCH=true



Add this statement to jex.prop: JEX\_BATCH=true

**4.3**: If we want to print out on console and create a batch\_log files we need to config in jex config (jex/config/jex.project\_name.xml) as the following code

<simple\_batch\_log level=*"DEBUG"* filePath=*"${JEX\_HOME}/logs"* fileName=*"simple\_batch\_log."* dateFormat=*"yyyymmdd"*>

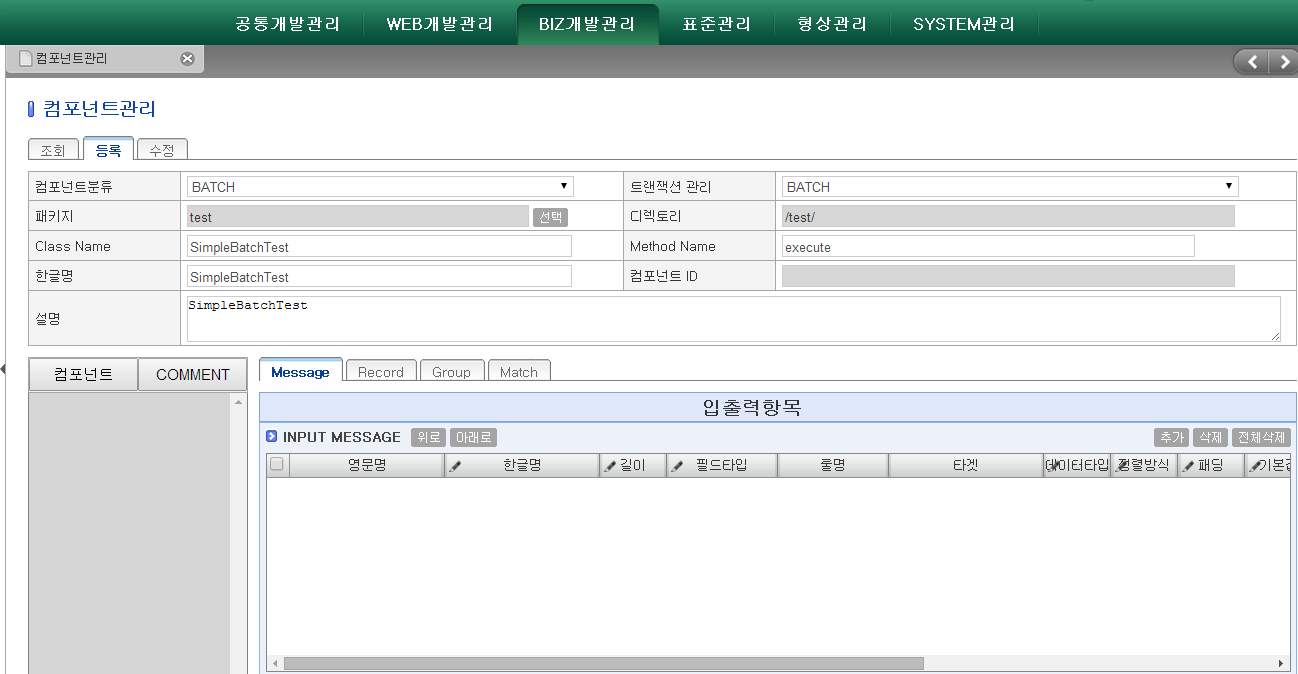
<handler>console,file</handler>

<pattern>[%d{yyyy-mm-dd hh24:mi:ss:ms}][%-5p] : %g%n</pattern>

</simple\_batch\_log>

* 1. **Testing**

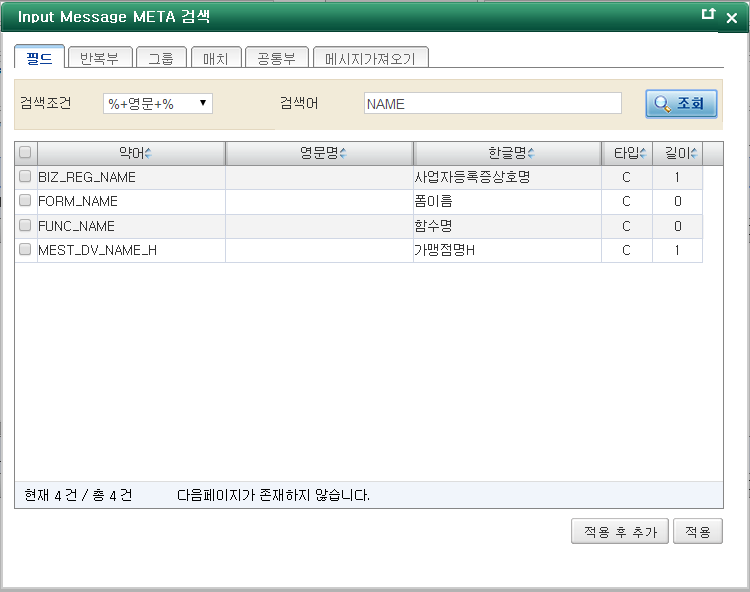
**4.2.1**: Create a batch class in jex\_studio



Click this button to choose input message

Choose BATCH type of component

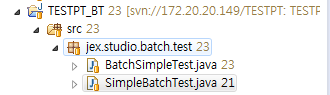
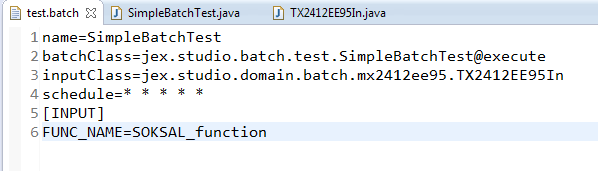
**4.2.2**: Choose input message type



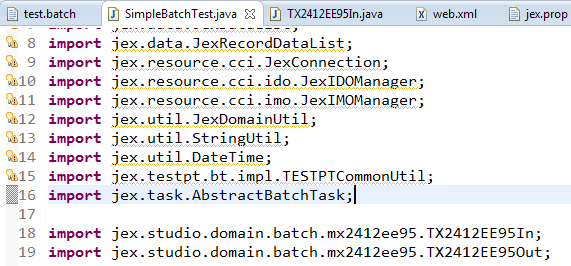
+ First type NAME 🡪click button search🡪check on FUNC\_NAME type🡪save.

**4.2.3**: Move to Eclipse to update project and create a batch file file\_name.batch in this path $ {JEX\_HOME} / simple\_batch /project\_name/file\_name.batch. And $ {JEX\_HOME}=project\_name\_PT/web/WEB-INF/jex/

**4.2.4**: Example of test.batch file



Method in class\_batch java



Package

Class\_namee