



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

iWD FAQ

iWD FAQs

iWD FAQs

Data Mart

Data Mart

Q: When are tasks moved from Intraday to Historical Tables?

[+] SHOW

A: A task must have reached a "final" state before it is moved from intraday to historical tables. A "final" state is where the task has reached a completed, canceled or reject state in iWD. Until then, the task will remain in intraday tables even if it spans a 24-hour period.

Rules

Rules

Q: How can I trigger a rule based on an update from my source system?

[+] SHOW

A: Rules cannot be triggered directly by an update action. However it's possible to update the `reprioritizeDateTime` attribute during the same update and trigger execution of prioritization rules for the given task as soon as possible. Prioritization rules are triggered for every task that has the `IWD_reprioritizeDateTime` interaction property set to some moment in the past.

If it's important to know whether prioritization rules were triggered via an update action or not, a constant timestamp could be used for `reprioritizeDateTime` (for example `2000/01/01 00:00:00`). Then in the prioritization rules, add a condition which verifies if `IWD_reprioritizeDateTime` equals `2000/01/01 00:00:00`.

Q: When are rules compiled & loaded?

[+] SHOW

A: Rules are compiled by the iWD Rules service right after service deployment or redeployment. In the case of complex rules it may take a minute or more (of course it also depends on hardware). During this time, all requests to the Rules service are being queued. After rules are compiled and loaded, the Rules service is ready for their execution. The rules service is executing rules on behalf of

the Business Context Management service.

Q: Why does using a custom function in a rule language mapping expression sometime generate an error?

[+] SHOW

A: If you encounter a rule parsing error and the rule language mapping expression and the custom function look semantically correct, the log file can provide additional details of the problem. There is a known problem if nested custom functions are used: for example;

```
- someCustomFunction( otherCustomFunction( $task.getActivationDate() ), "input" )
```

Example errors in log:

```
...
rule compilation problem: Class "evo.gtl.broker.Rule_Archiviere_KB_KFZ_D038__RULE_1338_8__0"
has no method named "getYoungerDate"
...
rule compilation problem: Instance method "int
evo.gtl.broker.Rule_Archiviere_KB_KFZ_D038__RULE_1338_8__0.getYoungerDate()" cannot be
invoked in static context
...
```

As all custom functions are generated as static methods of a class with the same name (with the first letter capitalized), the rule engine appends class names automatically, but in some cases with nested custom functions it is omitted. A workaround is to manually prefix your custom function with the classname.

Here is an example of a manually-added classname prefix for a nested function:

```
someCustomFunction( OtherCustomFunction.otherCustomFunction(
$task.getActivationDate() ), "input" )
```

Q: How can I output messages from rule functions to a custom log?

[+] SHOW

A: You can use log4j in rule functions like this:

```
import org.apache.log4j.Logger;

function ... {
    Logger LOG = Logger.getLogger("some.custom.category.name");
    ...
    LOG.info("this is example message");
    ...
}
```

To output this to the log file C:\iwd\logs\custom.log, update log4j.properties (in iwd_node/WEB-INF/classes) like this:

```
log4j.category.some.custom.category.name=INFO, custom_log
...
log4j.appender.custom_log=org.apache.log4j.RollingFileAppender
log4j.appender.custom_log.Threshold=TRACE
log4j.appender.custom_log.File=C:/iwd/logs/custom.log
```

```
log4j.appender.custom_log.layout=org.apache.log4j.PatternLayout
log4j.appender.custom_log.layout.ConversionPattern=%d{HH:mm:ss} |%t |%p |%c |%m%n
```

Q: Sometimes the Rules Engine seems to be evaluating a different rule condition or firing a different rule action than the one I expect. What could be happening?

[+] SHOW

A: Check to see if you have two rule conditions (or two actions) with Language Expressions that are identical, except for the rule parameters being used. If this is the case, the Drools Rules Engine may be evaluating the incorrect rule condition or action—that is, not the one you expect based on what the rule author selected when creating the rule.

In this case, you should rewrite the Language Expression in one or both of these conflicting rule conditions or actions. For example, let's say your rule template had two rule conditions that checked the `IWD_dueDateTime`, and the only difference between these two conditions is that one used `minutes/days/hours` and the other used `working minutes/working days/working hours`, based on a business calendar. In this case, you should ensure that the rule conditions do not look identical except for the rule parameters, like this:

Condition 1: Due Time is in "{periodFrom}" to "{periodTo}" "{periodType}"

Condition 2: Due Time is in "{periodFrom}" to "{periodTo}" "{periodTypeBC}"

Instead, you should ensure that the rule conditions have slightly different Language Expressions, like this, even if the rule parameters are identical between the rule conditions:

Condition 1: Due Time is in "{periodFrom}" to "{periodTo}" "{periodType}"

Condition 2: Due Time is in "{periodFrom}" to "{periodTo}" working "{periodType}"

Stat Server Extensions

Stat Server Extensions

Q: How do I configure the Stat Server Java Extension so that my CCPulse+ template will automatically refresh on a frequent basis?

[+] SHOW

A: Set the `refresh-interval` configuration option of the Stat Server Java Extension application to 1 minute. Configure your CCPulse+ template to 'Time-based Notification'. This will allow the CCPulse+ template to refresh with a maximum delay of 1 minute.