iWD FAQ

iWD FAQs

5/4/2025

iWD FAQs

General

General

Q: What does MQ Service do?

[+] SHOW

A: MQ Service delivers messages between iWD services. For example when a task is completed, the distribution point informs capture point about it via a message.

The reason services do not call each other directly is to avoid one part of the system bringing down another part. For example if we have a DB capture point and its database connection is not available at some point, it will not be able to processes any messages from the distribution point. If the distribution point was calling DB capture directly, it would fail and this would lead to data inconsistency and/or bring processing to a complete halt. Instead, MQ Service takes care of the message and the distribution point can continue doing its work without worrying if/when the capture point will be functional again. Meanwhile MQ Service will wait until the DB capture point is OK and then process all the collected messages one by one.

In order to not lose any messages in the case of a restart or some general system failure, MQ Services stores the messages in the database.

Q: When two tasks have the same priority, which is distributed first?

[+] SHOW

A: When two tasks have the same priority, the one that is due first will be distributed first.

Q: What is the Restart Queue for?

[+] SHOW

A: When an interaction is placed into a restart queue, the link between the iWD Application task and the interaction in Genesys Interaction Server is severed. Like the "outbound" queue, the interaction will remain in the restart queue until a strategy stops that interaction after a small delay. In the iWD Application, the task will go through a new cycle in which all the rules, including classification, are applied, and will eventually be distributed to Interaction Server where it will now have a new interaction ID. At this point, the task history in Global Task List will show, for that one task, two distribution events, each with its own unique ID for the interaction in Interaction Server.

Q: Which JDBC driver for MSSQL should I use?

[+] SHOW

A: In iWD 7.6 we use the `sqljdbc4.jar` driver that is valid for JDK 1.6 and higher. For JDK 1.5 we use the `sqljdbc.jar` driver instead of `sqljdbc4.jar`.

To replace the JDK 1.5 driver:

1. Download `sqljdbc_<latest_version>_enu.tar.gz` from http://www.microsoft.com/en-us/download/confirmation.aspx?id=11774 .
2. Extract `sqljdbc_<latest_version>enu.tar.gz` to the `<sqljdbc<latest_version>_enu directory>`.
3. Shut down the iWD Tomcat.
4. Remove the `<iWD Tomcat installation directory>\webapps\evo.cmc.web\WEB-INF\lib\sqljdbc4-4.0.jar` file.
5. Remove <iWD Tomcat installation directory>\webapps\evo.runtime.web\WEB-INF\lib\sqljdbc4-4.0.jar
6. Copy the `<sqljdbc_<latest_version>_enu directory>\sqljdbc_4.1\enu\sqljdbc.jar` to the `<iWD Tomcat installation directory>\webapps\evo.cmc.web\WEB-INF\lib\` folder.
7. Copy the `<sqljdbc_<latest_version>_enu directory>\sqljdbc_4.1\enu\sqljdbc.jar` to the `<iWD Tomcat installation directory>\webapps\evo.runtime.web\WEB-INF\lib\` folder.
8. Re-start the iWD Tomcat.

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.

Q: What is the difference between the iWD Extended Statistics and Aggregate Statistics services?

[+] SHOW

A: The Extended Statisticss service is used to retrieve generated statistics from the iWD Data Mart database and push them to Genesys Statistics Server Adapter. The actual statistics are created by the iWD Aggregate Statistics job that populates the stat tables in Data Mart. So, the Extended Statistics service consumes the data prepared by the Aggregate Statistic service.

Q. Which database should I select when configuring the GTL Extended Statistics Service - Runtime or Data Mart?

[+] SHOW

A: Data Mart. The service is consuming statistics stored in the statistics tables created in iWD Data Mart.

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_reprioriteDateTime` 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: How do I map several instances of a parameter to a single parameter definition?

[+] SHOW

A: The underscore character in parameter names has a special meaning when building rule templates. It is used to specify an index of the parameter, where the rule expression requires more than one instance of the particular parameter - the most common example is range definitions.

Example: You need to create a condition, which has to check whether the task's due date is in the range from `date1` to `date2` or in the range `date3` to `date4`. There are two possible approaches:

- Create a condition like this:

`Due is in "{dueDT1}" to "{dueDT2}" or in "{dueDT3}" to "{dueDT4}"` This will mean defining four parameters with type `InputDate` in the `Parameters` section, which is not very convenient, especially if there is more than one occurrence of the condition/action like this.

- Use underscore and index:

`Due is in "{dueDT_1}" to "{dueDT_2}" or in "{dueDT_3}" to "{dueDT_4}"`

and in Parameters you need to specify only one parameter, with name `dueDT` and type `InputDate`.

> ### Important
> Rules will not be executed properly if you use underscore while defining Parameters in the Rule Template (e.g. `dueDT` or `thisIsMyTextParam` will work fine, whereas `due_DT` or `this_Is_My_Text_Param` will not be recognized). Only use underscore when referring to the multiple instances of the same parameter when creating conditions and actions, as described above.

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}"`

Q: How do I create and use a custom rule template?

[+] SHOW

A: When you create a custom rule template, it must be explicitly added to your iWD Solution, through iWD Manager. You do this by following these steps:

1. A Solution is made up of Modules, which are made up of Components. When you save your new custom rule template, it becomes a new Component. You must then add this Component to a new or existing Module. Do this by going to `Modules &Components > Modules`. You should see your new Rule Template in the Unassigned Components list. Check it and click the arrow to move it to the Assigned Components list. Save the change.
2. If you used an existing Module in the previous step, and that Module is already assigned to your Solution, you should be done. If you created a new Module in the previous step, then you need to add this Module to your Solution. Go to `Services`, select your Managed Tenant, and then click on the name of your Solution. There you will see a list of Assigned Modules and Unassigned Modules. Select your new Module (with your Rule Template in it) from the list of Unassigned Components. Click the arrow to move it to the Assigned Modules list. Save the change.
3. If you added the Rule Template to an existing Module, you do not need to redeploy your Solution in order to use the new template. If you created a new Module earlier,you will be prompted to deploy the Solution first.

Capture

Capture

Q: How do I customize standard transformation scripts?

[+] SHOW

1. Create a new input and output transformation scripts under your tenant (not SYSTEM). **Do not try to import `gtl_transformat.xml` file into your tenant.**
2. Copy content from the standard transformation scripts in to the new scripts.
3. Make the necessary customizations in the new scripts.
4. Create a new module under your tenant and add to it the newly created scripts.
5. Add the module to your solution where all the services are configured.
6. Add a Scripting Service to your solution.
7. Set `Scripting Service` as the value for `transformationScriptingService` of your capture point
8. Select newly created scripts as values for `TransformInputScript` and `TransformOutputScript` of your capture point

Q: How do I specify optional WebService parameters?

[+] SHOW

A: Every specified input parameter must be passed to each request to WebService Capture Point. If a parameter is optional it may be skipped by passing a null value like this:

```
<priority xsi:nil="true"/>
```

Q: What versions of .NET framework are supported as WebService clients?

[+] SHOW

A: Webservice Capture Point works with .NET 2.0 and above. Earlier versions of the .NET framework do not support the WebService "nillable" flag which is used for optional parameters.

Q: Does WebService Capture Point support high availability?

[+] SHOW

A: WebService Capture Point has no HA-specific functionality built in. However HA support can be achieved by creating two or more WebService Capture Points and configuring a load balancer in front of them (for example Apache's **mod_proxy_balancer**).

WebService Capture Point is stateless (it does not maintain a client session) so for client applications generally it doesn't matter which WebService Capture Point will handle the call. There are a couple of limitations that need to be considered:

- Update/hold/resume/complete/cancel functions will work properly only with broker ID (not capture ID)
- Capture point level classification rules need to be duplicated for each capture point

Q: What is the format of the SOAP messages to use with the Web Service Capture Point?

[+] SHOW

A: A sample of the `CreateTask` message is shown below. The easiest way to see examples of these messages is to use a SOAP test tool such as the free SOAPUI Toolkit (http://www.soapui.org). You can point to the WebService Capture Point WSDL URL and create requests for each of the methods supported by the Web Service Capture Point.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.capture.gtl.evo" xmlns:brok="http://broker.gtl.evo">
   <soapenv:Header/>
   <soapenv:Body>
      <web:createTask>
         <web:captureId>?</web:captureId>
         <web:channel>?</web:channel>
         <web:category>?</web:category>
         <web:activationDateTime>?</web:activationDateTime>
         <web:dueDateTime>?</web:dueDateTime>
         <web:expirationDateTime>?</web:expirationDateTime>
         <web:businessValue>?</web:businessValue>
         <web:priority>?</web:priority>
         <web:processId>?</web:processId>
         <web:ext>
            <!--Optional:-->
            <brok:customerId>?</brok:customerId>
            <!--Optional:-->
            <brok:customerSegment>?</brok:customerSegment>
            <!--Optional:-->
            <brok:productSubtype>?</brok:productSubtype>
            <!--Optional:-->
            <brok:productType>?</brok:productType>
            <!--Optional:-->
            <brok:requestedAgent>?</brok:requestedAgent>
            <!--Optional:-->
            <brok:requestedSkill>?</brok:requestedSkill>
            <!--Optional:-->
            <brok:resultCode>?</brok:resultCode>
            <!--Optional:-->
            <brok:sourceCreatedDateTime>?</brok:sourceCreatedDateTime>
            <!--Optional:-->
            <brok:sourceDueDateTime>?</brok:sourceDueDateTime>
            <!--Optional:-->
            <brok:sourceFirstCreatedDateTime>?</brok:sourceFirstCreatedDateTime>
            <!--Optional:-->
            <brok:sourceProcessSubtype>?</brok:sourceProcessSubtype>
            <!--Optional:-->
            <brok:sourceProcessType>?</brok:sourceProcessType>
            <!--Optional:-->
            <brok:sourceTenant>?</brok:sourceTenant>
         </web:ext>
         <web:data>
            <!--Zero or more repetitions:-->
            <web:entry>
               <!--Optional:-->
               <web:key>?</web:key>
               <!--Optional:-->
               <web:value>?</web:value>
            </web:entry>
         </web:data>
         <web:hold>?</web:hold>
         <web:reason>?</web:reason>
         <web:actor>?</web:actor>
      </web:createTask>
   </soapenv:Body>
</soapenv:Envelope>
```

Q: How do I encode binary data in iWD XML messages?

[+] SHOW

A: Note: This information is relevant for the XML File Capture Point and the WebSphere MQ Capture Point that are deployed as iWD services. This is not supported for the Integrated Capture Points that are integrated into Interaction Server.

If you have any data in binary format that you want to include as a task attribute, you can do so by using encoding such as Base64 or UUencode. On the host system, you would encode the binary data, and then embed the resulting data within the CDATA section. It would be added to the iWD interaction as a custom data field. For example, if you want to pass through an image file such as a .jpg, in a `createTask` message, it would look like this:

```
<GTLMessage>

<createTask>

...

<data>

<image><![CDATA[add the encoded image here]]></image> </data> ...

</createTask>
```

```
</GTLMessage>
```

This encoded data can then be decoded on the client side (i.e. agent or knowledge worker desktop) and displayed on the user. Be aware that encoding binary data using Base64 or UUencode increases the size of the data at about a 4:3 ratio.

Q: How is the mapping configured between database columns and iWD core attributes ?

[+] SHOW

A: Note: The information described here is relevant for the Database Capture Point that can optionally be deployed as an iWD service. It is not relevant for the Database Capture Point that is integrated into Interaction Server, versions 8.1.0 and later.

Mapping is name-based, and there are two steps that define what database fields will be attached to a task. First, the `captureQuerySql` attribute of the database capture point service, defines what database fields should be selected. It is possible to use aliases, like this:

```
select someColumn as Channel, anotherColumn as dueDateTime …
```

Second, the `fieldsToAttach` attribute defines additional criteria, thus allowing to filter out any unnecessary fields. `fieldsToAttach` can be a comma-separated list of fields:

```
fieldsToAttach=Channel,Category,CustomerID,Priority,dueDateTime,resultCode
```

or a simple pattern, where * equals any number of any characters, and ? equals any single character:

```
fieldsToAttach=SYSTEM_FIELD?, CUSTOM*
```

> Important
>
> Field names in `fieldsToAttach` are case sensitive. They should be specified in exactly the same case as they are returned by `captureQuerySql` query

After the `fieldsToAttach` filter is applied, the resulting collection of database fields is compared with a list of task attribute names. Values of fields, that are recognized as task attributes, are assigned to the respective task attributes, all other values are added to the task as custom attributes.

List of task attributes:

Core attributes

```
activationDateTime
dueDateTime
expirationDateTime
businessValue
category
channel
priority
actor
assignedDateTime
assignedToUser
```

```
completedDateTime
distributedDateTime
distributionId
heldDateTime
queue
queueTarget
queueType
reason
reprioritizeDateTime
captureId
```

Extended attributes

```
customerSegment
customerId
productType
productSubtype
sourceTenant
sourceProcessType
sourceProcessSubtype
sourceFirstCreatedDateTime
sourceCreatedDateTime
sourceDueDateTime
resultCode
requestedAgent
requestedSkill
```