



**OX2OX Migration Framework Scheduler Technical
Documentation for
2.1.0**

2022-01-05

Copyright notice

©2022 by OX Software GmbH. All rights reserved. Open-Xchange and the Open-Xchange logo are trademarks or registered trademarks of OX Software GmbH. All other company and/or product names may be trademarks or registered trademarks of their owners. Information contained in this document is subject to change without notice.

Contents

1	General Information	2
1.1	Warnings	2
1.2	Delivery Comment	2
1.3	Install Package Repository	2
1.4	Build Dependencies	2
2	Shipped Packages and Version	2
2.1	Package open-xchange-omf-orchestrator	2
2.1.1	Installation	2
2.1.2	Configuration	2
2.2	Package open-xchange-omf-scheduler	3
2.2.1	Installation	3
2.2.2	Configuration	3
A	Configuration Files	3

1 General Information

1.1 Warnings



Warning

This preview delivery is not for productive usage and not affected by service-level agreements.



Warning

Custom configuration or template files are potentially not updated automatically. After the update, please always check for files with a **.dpkg-new** or **.rpmnew** suffix and merge the changes manually. Configuration file changes are listed in their own respective section below but don't include changes to template files. For details about all the configuration files and templates shipped as part of this delivery, please read the relevant section of each package.

1.2 Delivery Comment

This delivery was requested with following comment:

OMF Scheduler 2.1.0 Preview Delivery 12

1.3 Install Package Repository

This delivery is part of a restricted preview software repository:

<https://software.open-xchange.com/components/omf-scheduler/preview/2.1.0/RHEL7>
<https://software.open-xchange.com/components/omf-scheduler/preview/2.1.0/DebianStretch>
<https://software.open-xchange.com/components/omf-scheduler/preview/2.1.0/DebianBuster>

1.4 Build Dependencies

This delivery was build with following dependencies:

RedHat:RHEL-7,Debian:Stretch,Debian:Buster

2 Shipped Packages and Version

2.1 Package open-xchange-omf-orchestrator

OMF Orchestrator CLI to interoperate with the OX2OX Migration Framework.

Version: 2.1.0-12

Type: Other

2.1.1 Installation

Install on nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-omf-orchestrator
```

2.1.2 Configuration

For details, please see appendix [A](#)

/opt/open-xchange/omf/orchestrator/etc/omf-orchestrator.yml (page [4](#))

2.2 Package open-xchange-omf-scheduler

OMF Scheduler OX2OX Migration Framework Scheduler.

Version: 2.1.0-12

Type: Other

2.2.1 Installation

Install on nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-omf-scheduler
```

2.2.2 Configuration

For details, please see appendix [A](#)

/opt/open-xchange/omf/scheduler/etc/omf-scheduler.yml (page [9](#))

A Configuration Files

File 1 /opt/open-xchange/omf/orchestrator/etc/omf-orchestrator.yml

```

1 micronaut:
2   application:
3     name: omf
4   http:
5     services:
6       # The OMF Scheduler HTTP REST API service configuration
7       scheduler-admin-source:
8         url: "${omf.scheduler.url}/omf/scheduler/admin/source/"
9         read-timeout: ${omf.http.read-timeout}
10        connect-timeout: ${omf.http.connect-timeout}
11        ssl:
12          enabled: ${omf.http.ssl.enabled}
13          trust-store:
14            path: ${omf.http.ssl.trust-store.path}
15            password: ${omf.http.ssl.trust-store.password}
16            type: ${omf.http.ssl.trust-store.type}
17
18       scheduler-admin-target:
19         url: "${omf.scheduler.url}/omf/scheduler/admin/target/"
20         read-timeout: ${omf.http.read-timeout}
21         connect-timeout: ${omf.http.connect-timeout}
22         ssl:
23           enabled: ${omf.http.ssl.enabled}
24           trust-store:
25             path: ${omf.http.ssl.trust-store.path}
26             password: ${omf.http.ssl.trust-store.password}
27             type: ${omf.http.ssl.trust-store.type}
28
29       # The OMF Scheduler Migration HTTP REST API service configuration
30       scheduler-migration:
31         url: "${omf.scheduler.url}/omf/scheduler/migration/"
32         read-timeout: ${omf.http.read-timeout}
33         connect-timeout: ${omf.http.connect-timeout}
34         ssl:
35           enabled: ${omf.http.ssl.enabled}
36           trust-store:
37             path: ${omf.http.ssl.trust-store.path}
38             password: ${omf.http.ssl.trust-store.password}
39             type: ${omf.http.ssl.trust-store.type}
40
41       # The OMF Scheduler Monitoring HTTP REST API service configuration
42       scheduler-monitoring:
43         url: "${omf.scheduler.url}/ws/omf/scheduler/workers/monitor/"

```

```

44     read-timeout: ${omf.http.read-timeout}
45     connect-timeout: ${omf.http.connect-timeout}
46     ssl:
47         enabled: ${omf.http.ssl.enabled}
48         trust-store:
49             path: ${omf.http.ssl.trust-store.path}
50             password: ${omf.http.ssl.trust-store.password}
51             type: ${omf.http.ssl.trust-store.type}
52
53 omf:
54     http:
55         read-timeout: 30s
56         connect-timeout: 10s
57         ssl:
58             enabled: true
59             # If the scheduler does not have a valid public certificate
60             # (e.g. uses a self-signed certificate), then its certificate can be
61             # configured here.
62             trust-store:
63                 path: file:/opt/open-xchange/omf/certs/scheduler.p12
64                 password: secret
65                 type: PKCS12
66     readonly: false
67     shell:
68         start.dir:
69         config:
70             user.dir: ${user.dir}/.omf/config
71             app.dir: /opt/open-xchange/omf/lib/scripts
72     scheduler:
73         # Credentials for the scheduler
74         # On multi-user systems, specifying the password in a configuration file
75         # with proper file system permissions is preferred to specifying it on
76         # the command line, since the command line is visible to all local users.
77         #
78         # Example:
79         # username: admin
80         # password: secret
81
82         # Location of the scheduler. Only the protocol and host name need to be
83         # specified.
84         url: "https://localhost:8443"
85     ui:
86         color: true
87         color.theme: DARK
88         unicode: true
89         expandIds: false
90         tz: UTC
91         showTz: false
92         showAgo: false
93         prettyJson: false
94         highlightJson: false
95         shell:
96             prettyJson: true
97             highlightJson: true
98             fancyPrompt: true
99             rightHandPrompt: true
100     history.file: ${user.dir}/.omf_history
101
102 logger:
103     levels:
104         # change this to TRACE to see a detailed log of the HTTP traffic between the
105         # Orchestrator and the Scheduler
106         io.micronaut.http.client: INFO

```

File 2 /opt/open-xchange/omf/scheduler/etc/omf-scheduler.yml

```

1 # https://docs.micronaut.io/latest/guide/config.html#configurationProperties
2 ---

```

```

3 micronaut:
4   # SSL configuration
5   # Required for production environments.
6   # See https://docs.micronaut.io/latest/guide/index.html#https for details.
7   ssl:
8     enabled: true
9     port: 8443
10    key-store:
11      path: file:/opt/open-xchange/omf/certs/keystore.p12
12      type: PKCS12
13      password: secret
14    server:
15      dual-protocol: true
16      port: 8080
17
18  http:
19    services:
20      # The omf-source service is use to collect health status and metrics from the Source
21      # OMF nodes. The HTTP client can be configured (ex: ssl) here by referencing
22      # https://docs.micronaut.io/latest/guide/configurationreference.html#io.micronaut.
23      # http.client.ServiceHttpClientConfiguration
24      # and the subsequent sections related to micronaut.http.services.*
25      omf-source:
26        connect-timeout: 30s
27        read-timeout: 120s
28        # Example SSL configuration in case a source uses a private certificate
29        # ssl:
30        #   trust-store:
31        #     path: file:/opt/open-xchange/omf/certs/source.p12
32        #     type: PKCS12
33        #     password: secret
34        omf-target:
35          connect-timeout: 30s
36          read-timeout: 120s
37      application:
38        name: omf-scheduler
39      # Configure security including basic auth: https://micronaut-projects.github.io/
40      # micronaut-security/latest/guide/#basicAuth
41      # Must be set to true or the Source Controller is not secure
42      security:
43        enabled: true
44        # Change the security of the open api views to anonymous so that they can be viewed
45        # without credentials
46        intercept-url-map:
47          - pattern: /swagger/**
48            access:
49              - isAnonymous()
50          - pattern: /swagger-ui/**
51            access:
52              - isAnonymous()
53          - pattern: /rapidoc/**
54            access:
55              - isAnonymous()
56          - pattern: /redoc/**
57            access:
58              - isAnonymous()
59      # https://docs.micronaut.io/latest/guide/index.html#_configuring_caches
60      #caches:
61        #example:
62          #charset: UTF-8
63          #expire-after-access: 1h
64      metrics:
65        enabled: true
66        export:
67          # Creates an endpoint like http://host/prometheus - uses basic auth from
68          # credentials under scheduler.http.admin
69          prometheus:
70            enabled: true
71            step: PT1M
72            descriptions: true
73      router:
74        # Adds api versioning: https://docs.micronaut.io/latest/guide/index.html#apiVersioning

```

```

72     versioning:
73         enabled: true
74     parameter:
75         enabled: true
76         names: 'v'
77     header:
78         enabled: true
79         names: 'X-API-VERSION'
80     # Allows the openapi views to be seen
81     static-resources:
82         swagger:
83             paths: classpath:META-INF/swagger
84             mapping: /swagger/**
85         redoc:
86             paths: classpath:META-INF/swagger/views/redoc
87             mapping: /redoc/**
88         rapidoc:
89             paths: classpath:META-INF/swagger/views/rapidoc
90             mapping: /rapidoc/**
91         swagger-ui:
92             paths: classpath:META-INF/swagger/views/swagger-ui
93             mapping: /swagger-ui/**
94 ---
95     scheduler:
96         # Must be unique for each Scheduler instance.
97         # Be very careful when changing this value, as it is also used to determine the
98         # transactional ID for writing
99         # batches into Kafka topics for each Scheduler node.
100         id: 'scheduler-0'
101         hostname: ''
102         window:
103             # If more than this many batches are created as part of a new Window, don't return the
104             # list of
105             # Batch IDs as part of the Window creation result DTO, as they would just be too many
106             # to display
107             # in the first place (also affects the debug logging):
108             batch.id.threshold: 50
109             # If more than this many context IDs are assigned as part of a new Window, don't
110             # return the list of
111             # context IDs as part of the Window creation result DTO, as they would just be too
112             # many to display
113             # in the first place:
114             context.id.threshold: 50
115         batch:
116             presync:
117                 # Max number of contexts in a batch
118                 size: 10
119                 # Strategy to use when creating batches.
120                 # Current supported strategies:
121                 # - fill-first: create batches up to the batch size then create the next batch
122                 # - fill-equal: create batches of equal size
123                 strategy: fill-equal
124             cutover:
125                 size: 10
126                 strategy: fill-equal
127             preprovisioning:
128                 size: 10
129                 strategy: fill-equal
130         kafka:
131             wait: false
132             queues:
133                 batch: "omf-batch"
134                 response: "omf-response"
135             resize:
136                 batch: true
137                 response: true
138             record.header.enhance: true
139             topic:
140                 list:
141                     timeout: -1s
142             describe:
143                 timeout: -1s

```



```
139     retry:
140         attempts: 3
141         wait: 3s
142     create:
143         # whether topics should be created when a source is created or synced (true) or
144         # whether we make use of auto-creation instead (old behaviour prior to 2.1.0-6):
145         enabled: true
146         timeout: -1s
147         partitions: 2
148         replication.factor: 0
149         config:
150             retention.ms: 432000000
151 workers:
152     allow:
153         # allow Worker shutdown via the REST API
154         shutdown: false
155         # allow overriding the Sources Workers subscribe to via the REST API
156         changeSources: false
157 http:
158     users:
159         # Users defined here can have roles OMF_ADMIN or OMF_USER. Users without a role
160         # automatically have role OMF_USER
161         # assigned. Users with the role OMF_ADMIN have access to every REST API method.
162         # OMF_USER role is restricted to
163         # a subset of the REST API.
164         admin:
165             password: secret
166             role: OMF_ADMIN
167         omfuser:
168             password: secret
169             role: OMF_USER
170         admin:
171             controller:
172                 path: /omf/scheduler/admin
173         migration:
174             controller:
175                 path: /omf/scheduler/migration
176 metrics:
177     migration:
178         enabled: true
179     leadership:
180         enabled: true
181     batchresponse:
182         enabled: true
183     percentiles: true
184     schedulers:
185         enabled: true
186     source:
187         enabled: true
188     target:
189         enabled: true
190     monitor:
191         windows:
192             enabled: true
193             interval: 5m
194             delay: 30s
195         batches:
196             enabled: true
197             interval: 5m
198             delay: 30s
199         sources:
200             enabled: true
201             interval: 5m
202             delay: 30s
203         targets:
204             enabled: true
205             interval: 5m
206             delay: 30s
207     contextmappings:
208         enabled: true
209         interval: 5m
210         delay: 30s
```

```

209     migrationevents:
210         enabled: true
211         interval: 5m
212         delay: 30s
213     workers:
214         enabled: true
215         idle.since: [5m, 10m, 30m]
216 ---
217 jackson:
218     bean-introspection-module: true
219     serialization:
220         indent-output: true
221         writeDatesAsTimestamps: false
222 ---
223 datasources:
224     # Used to persist scheduling data
225     scheduler:
226         # url should use createDatabaseIfNotExist=true if the database will not
227         # already exist: https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-reference-
228         # configuration-properties.html
229         url: jdbc:mysql://localhost:3306/scheduler?createDatabaseIfNotExist=true
230         username: root
231         password: my-secret-pw
232         driverClassName: org.mariadb.jdbc.Driver
233         dialect: MYSQL
234         maximumPoolSize: 10
235         maxLifetime: 180000
236     # Used to create/drop databases for sources. This is not really the "default" data
237     # source
238     # but we need to use default because of bug https://github.com/micronaut-projects/
239     # micronaut-data/issues/598s
240     default:
241         url: jdbc:mysql://localhost:3306/
242         username: root
243         password: my-secret-pw
244         driverClassName: org.mariadb.jdbc.Driver
245         dialect: MYSQL
246         maximumPoolSize: 5
247         maxLifetime: 180000
248 ---
249 endpoints:
250     loggers:
251         enabled: true
252         sensitive: true
253     health:
254         discovery-client:
255             enabled: false
256         sources:
257             enabled: false
258         targets:
259             enabled: false
260     liquibase:
261         # fails with missing transition, might be fixed in later Micronaut releases
262         enabled: false
263     info:
264         enabled: true
265         sensitive: true
266         sourceCodeOrigin:
267             enabled: true
268             location: file:/opt/open-xchange/omf/scheduler/share/SourceCodeOrigin.txt
269 ---
270 zookeeper:
271     server: zookeeper:2181
272     blockUntilConnected: true
273     maxConnectedWaitTime: 30s
274     sessionTimeout: 1m
275     connectionTimeout: 15s
276     maxCloseWait: 15s
277     waitForShutdownTimeout: 15s
278     connectionRetry:
279         baseSleepTime: 5s
280         maxSleepTime: 30s

```

```

278     maxRetries: 50
279 ---
280 kafka:
281   bootstrap:
282     servers: kafka-1:9092, kafka-2:9092, kafka-3:9092
283   producers:
284     batch-producer:
285       enable.idempotence: true
286       # This enables transactions for the Batch Producer
287       # The value must be unique per application, but should
288       # not change for the same app after a crash, etc.
289       # Note that if you only want to allow a single Scheduler instance to be capable of
290       # writing Batches into the Kafka topics, then change this to be the same value
291       # across
292       # all Scheduler instances as Kafka will fence them (see PRODUCER_FENCED).
293       # But if you want all Scheduler instances to be able to write batches into Kafka
294       # topics,
295       # use a unique value for each Scheduler instance:
296       transactional.id: ${scheduler.id}
297   #consumers:
298   #response-consumer:
299 ---
300 mail:
301   # whether to send emails
302   enabled: false
303   window:
304     # whether to send emails when a Window succeeds:
305     success: false
306     # whether to send emails when a Window fails:
307     failure: true
308   # mandatory, must be set to be able to send emails and it
309   # must be a valid email address in the form localpart@domain, or sending will fail:
310   from: ${scheduler.id}@example.com
311   # whom to send those mails to (can be a comma separated list):
312   to:
313   cc:
314   bcc:
315   # text to include in the subject line, wrapped in []:
316   subject.id:
317   smtp:
318     host: localhost
319     port: 25
320     # leave empty for no authentication:
321     username:
322     password:
323     # SMTP, SMTPS or SMTP_TLS (SMTP with mandatory StartTLS):
324     transport: SMTP
325     # whether to allow SMTP without StartTLS:
326     smtp.plain: true
327     tls:
328       # whether to trust all SMTP server keys
329       trustall: false
330       # whether to verify SMTP server keys
331       verify: true
332 ---
333 liquibase:
334   datasources:
335     scheduler:
336       change-log: 'classpath:liquibase/scheduler/liquibase-changelog.xml'
337 ---
338 logger:
339   levels:
340     ROOT: INFO
341     com.openxchange: INFO
342     omf: INFO
343     omf.scheduler.admin.AuthenticationProviderUserPassword: WARN
344     org.apache.kafka.clients.consumer.ConsumerConfig: WARN
345 ---

```