



**OX2OX Migration Framework Scheduler Technical
Documentation for
2.1.0-rev17**

2022-09-08

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 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 17

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 Version

2.1 Package open-xchange-omf-orchestrator

OMF Orchestrator CLI to interoperate with the OX2OX Migration Framework.

Version: 2.1.0-17

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 [5](#))

2.2 Package open-xchange-omf-scheduler

OMF Scheduler OX2OX Migration Framework Scheduler.

Version: 2.1.0-17

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 [10](#))

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: ${omf.http.ssl.trust-store.type}
40
41       scheduler-batch:
42         url: "${omf.scheduler.url}/omf/scheduler/batch/"
43         read-timeout: ${omf.http.read-timeout}

```

```

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

```

```

116     prettyJson: false
117     highlightJson: false
118     shell:
119         prettyJson: true
120         highlightJson: true
121         fancyPrompt: true
122         rightHandPrompt: true
123     history.file: ${user.dir}/.omf_history
124
125 logger:
126     levels:
127         # change this to TRACE to see a detailed log of the HTTP traffic between the
128         # Orchestrator and the Scheduler
129         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:

```

```

49         - isAnonymous()
50     - pattern: /rapidoc/**
51       access:
52         - isAnonymous()
53     - pattern: /redoc/**
54       access:
55         - isAnonymous()
56 # https://docs.micronaut.io/latest/guide/index.html#_configuring_caches
57 #caches:
58   #example:
59     #charset: UTF-8
60     #expire-after-access: 1h
61 metrics:
62   enabled: true
63   export:
64     # Creates an endpoint like http://host/prometheus - uses basic auth from
65     # credentials under scheduler.http.admin
66     prometheus:
67       enabled: true
68       step: PT1M
69       descriptions: true
70 router:
71   # 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:

```



```

116     # - fill-first: create batches up to the batch size then create the next batch
117     # - fill-equal: create batches of equal size
118     strategy: fill-equal
119   cutover:
120     size: 10
121     strategy: fill-equal
122   preprovisioning:
123     size: 10
124     strategy: fill-equal
125   kafka:
126     enabled: true
127     wait: false
128     queues:
129       batch: "omf-batch"
130       response: "omf-response"
131     resize:
132       batch: true
133       response: true
134     record.header.enhance: true
135     topic:
136       list:
137         timeout: -1s
138       describe:
139         timeout: -1s
140       retry:
141         attempts: 3
142         wait: 3s
143     create:
144       # whether topics should be created when a source is created or synced (true) or
145       # whether we make use of auto-creation instead (old behaviour prior to 2.1.0-6):
146       enabled: true
147       timeout: -1s
148       partitions: 2
149       replication.factor: 0
150       config:
151         retention.ms: 432000000
152     rest:
153       lastNextBatches: 10
154   workers:
155     allow:
156       # allow Worker shutdown via the REST API
157       shutdown: false
158       # allow overriding the Sources Workers subscribe to via the REST API
159       changeSources: false
160   http:
161     users:
162       # Users defined here can have roles OMF_ADMIN or OMF_USER. Users without a role
163       # automatically have role OMF_USER
164       # assigned. Users with the role OMF_ADMIN have access to every REST API method.
165       # OMF_USER role is restricted to
166       # a subset of the REST API.
167       admin:
168         password: secret
169         role: OMF_ADMIN
170       omfuser:
171         password: secret
172         role: OMF_USER
173     workers:
174       # a list of valid tokens for workers to use
175       # can generate some with `pwgen -n1 64 1`
176       - 'bei90hchie8nai5em5asee9wohz6uu0ahshaigh0bia1isi4liKi0iwo8bu2niey'
177       - 'eeW5moi6eleik0ziw7ivaen3phoi6oolae9aht2ox9uY0ebiVaht0gashoof1rai'
178       - 'oogheePhaeB5iezairu6ongee8Ee6faePashi9thietahG0bieghiixeivahroco'
179     admin.controller.path: /omf/scheduler/admin
180     migration.controller.path: /omf/scheduler/migration
181     user.controller.path: /omf/scheduler/user
182     batch.controller.path: /omf/scheduler/batch
183   metrics:
184     migration:
185       enabled: true
186     leadership:
187       enabled: true

```

```

186     batchresponse:
187         enabled: true
188         percentiles: true
189     schedulers:
190         enabled: true
191     source:
192         enabled: true
193     target:
194         enabled: true
195     monitor:
196         windows:
197             enabled: true
198             interval: 5m
199             delay: 30s
200         batches:
201             enabled: true
202             interval: 5m
203             delay: 30s
204         sources:
205             enabled: true
206             interval: 5m
207             delay: 30s
208         targets:
209             enabled: true
210             interval: 5m
211             delay: 30s
212         contextmappings:
213             enabled: true
214             interval: 5m
215             delay: 30s
216         usermappings:
217             enabled: true
218             interval: 60m
219             delay: 5m
220         migrationevents:
221             enabled: true
222             interval: 5m
223             delay: 30s
224     workers:
225         enabled: true
226         idle.since: [5m, 10m, 30m]
227     orphan-check:
228         context.batch.size: 50
229     database:
230         migration:
231             allowRead: false
232             allowWrite: false
233         scheduler:
234             allowRead: false
235             allowWrite: false
236             useSkipLocked: false
237     batchSkipList:
238         reap:
239             windows: true
240             batches: true
241 ---
242 jackson:
243     bean-introspection-module: true
244     serialization:
245         indent-output: true
246         writeDatesAsTimestamps: false
247 ---
248 datasources:
249     # Used to persist scheduling data
250     scheduler:
251         # url should use createDatabaseIfNotExist=true if the database will not
252         # already exist: https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-reference-
253         # configuration-properties.html
254         url: jdbc:mysql://localhost:3306/scheduler?createDatabaseIfNotExist=true
255         username: root
256         password: my-secret-pw
257         driverClassName: org.mariadb.jdbc.Driver

```

```

257     dialect: MYSQL
258     maximumPoolSize: 10
259     maxLifetime: 180000
260 # Used to create/drop databases for sources. This is not really the "default" data
    source
261 # but we need to use default because of bug https://github.com/micronaut-projects/
    micronaut-data/issues/598s
262 default:
263     url: jdbc:mysql://localhost:3306/
264     username: root
265     password: my-secret-pw
266     driverClassName: org.mariadb.jdbc.Driver
267     dialect: MYSQL
268     maximumPoolSize: 5
269     maxLifetime: 180000
270 ---
271 endpoints:
272     loggers:
273         enabled: true
274         sensitive: true
275     health:
276         discovery-client:
277             enabled: false
278     sources:
279         enabled: false
280     targets:
281         enabled: false
282     liquibase:
283         # fails with missing transition, might be fixed in later Micronaut releases
284         enabled: false
285     info:
286         enabled: true
287         sensitive: true
288         sourceCodeOrigin:
289             enabled: true
290             location: file:/opt/open-xchange/omf/scheduler/share/SourceCodeOrigin.txt
291 ---
292 zookeeper:
293     server: zookeeper:2181
294     blockUntilConnected: true
295     maxConnectedWaitTime: 30s
296     sessionTimeout: 1m
297     connectionTimeout: 15s
298     maxCloseWait: 15s
299     waitForShutdownTimeout: 15s
300     connectionRetry:
301         baseSleepTime: 5s
302         maxSleepTime: 30s
303         maxRetries: 50
304 ---
305 kafka:
306     bootstrap:
307         servers: kafka-1:9092, kafka-2:9092, kafka-3:9092
308     producers:
309         batch-producer:
310             enable.idempotence: true
311             # This enables transactions for the Batch Producer
312             # The value must be unique per application, but should
313             # not change for the same app after a crash, etc.
314             # Note that if you only want to allow a single Scheduler instance to be capable of
315             # writing Batches into the Kafka topics, then change this to be the same value
            across
316             # all Scheduler instances as Kafka will fence them (see PRODUCER_FENCED).
317             # But if you want all Scheduler instances to be able to write batches into Kafka
            topics,
318             # use a unique value for each Scheduler instance:
319             transactional.id: ${scheduler.id}
320     #consumers:
321         #response-consumer:
322 ---
323 mail:
324     # whether to send emails

```

```
325 enabled: false
326 window:
327     # whether to send emails when a Window succeeds:
328     success: false
329     # whether to send emails when a Window fails:
330     failure: true
331     # mandatory, must be set to be able to send emails and it
332     # must be a valid email address in the form localpart@domain, or sending will fail:
333     from: ${scheduler.id}@example.com
334     # whom to send those mails to (can be a comma separated list):
335     to:
336     cc:
337     bcc:
338     # text to include in the subject line, wrapped in []:
339     subject.id:
340     smtp:
341         host: localhost
342         port: 25
343         # leave empty for no authentication:
344         username:
345         password:
346         # SMTP, SMTPS or SMTP_TLS (SMTP with mandatory StartTLS):
347         transport: SMTP
348         # whether to allow SMTP without StartTLS:
349         smtp.plain: true
350         tls:
351             # whether to trust all SMTP server keys
352             trustall: false
353             # whether to verify SMTP server keys
354             verify: true
355 ---
356 liquibase:
357     datasources:
358         scheduler:
359             change-log: 'classpath:liquibase/scheduler/liquibase-changelog.xml'
360 ---
361 logger:
362     levels:
363         ROOT: INFO
364         com.openxchange: INFO
365         omf: INFO
366         omf.scheduler.admin.AuthenticationProviderUserPassword: WARN
367         org.apache.kafka.clients.consumer.ConsumerConfig: WARN
368         omf.scheduler.security: INFO
369 ---
```