



OX2OX Migration Framework Source Technical Documentation
for
2.1.0-rev26

2023-03-17

Copyright notice

©2023 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	3
1.5	Notice	3
2	Shipped Version	3
2.1	Package open-xchange-omf-source	3
2.1.1	Installation	3
2.1.2	Configuration	3
2.2	Package open-xchange-omf-source-dualprovisioning	3
2.2.1	Installation	4
2.3	Package open-xchange-omf-source-dualprovisioning-cloudplugins	4
2.3.1	Installation	4
2.4	Package open-xchange-omf-source-guard	4
2.4.1	Installation	4
2.5	Package open-xchange-omf-source-mailfilter	4
2.5.1	Installation	5
A	Configuration Files	5

1 General Information

1.1 Warnings

Warning

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

Warning

It is mandatory to restart the **open-xchange** service on all middleware nodes after performing the update.

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 Source 2.1.0 Preview Delivery 26

1.3 Install Package Repository

This delivery is part of a restricted preview software repository:

<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianBullseye-7.10.6>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianBuster-7.10.6>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianStretch-7.10.6>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL7-7.10.6>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianBuster-7.10.5>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianStretch-7.10.5>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL7-7.10.5>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianBuster-7.10.4>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianStretch-7.10.4>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL7-7.10.4>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianStretch-7.10.3>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL6-7.10.3>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL7-7.10.3>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianStretch-7.10.2>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL6-7.10.2>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL7-7.10.2>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianStretch-7.10.1>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL7-7.10.1>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/DebianJessie-7.8>

4

<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL6-7.8.4>
<https://software.open-xchange.com/components/omf-source/preview/2.1.0/RHEL7-7.8.4>

1.4 Build Dependencies

This delivery was build with following dependencies:

```
cloud-plugins-master-1.6,frontend-master-7.8.4,guard-master-2.8.0,
backend-master,guard-master,backend-master-7.10.5,guard-master-2.10.5,backend-master-
7.10.4,guard-master-2.10.4,cloud-plugins-master-1.10,backend-master-7.10.3,guard-master-
2.10.3,backend-master-7.10.2,guard-master-2.10.2,backend-master-7.10.1,guard-master-
2.10.1
```

1.5 Notice



Info

Some configurations can be changed without restarting the service, please call following command for getting a list of supported settings.

```
/opt/open-xchange/sbin/listreloadables
```

Please use following command to enable capable and changed configurations on a running system.

```
/opt/open-xchange/sbin/reloadconfiguration
```

2 Shipped Version

2.1 Package open-xchange-omf-source

OMF Source Bundles Source component of the OX Migration Framework.

Version: 2.1.0-26

Type: OX Middleware Plugin

Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.8.4)
open-xchange-rest (<<7.10.7)
open-xchange-rest (>=7.8.4)
```

2.1.1 Installation

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

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

2.1.2 Configuration

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

/opt/open-xchange/etc/omf-source.properties (page [12](#))

2.2 Package open-xchange-omf-source-dualprovisioning

OMF Dual-Provisioning Feature

Version: 2.1.0-26

Type: OX Middleware Plugin

Depends on:

```
open-xchange-admin (<<7.10.7)
open-xchange-admin (>=7.8.4)
open-xchange-omf-source (>=2.1.0)
```

2.2.1 Installation

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

```
<package installer> install open-xchange-omf-source-dualprovisioning
```

2.3 Package open-xchange-omf-source-dualprovisioning-cloudplugins

OMF Dual-Provisioning Addon for Cloud-Plugins Sources

Version: 2.1.0-26

Type: OX Middleware Plugin

Depends on:

```
open-xchange-admin (<<7.10.7)
open-xchange-admin (>=7.8.4)
open-xchange-cloudplugins (<<2.0.0)
open-xchange-cloudplugins (>=1.6.12)
open-xchange-omf-source (>=2.1.0)
open-xchange-omf-source-dualprovisioning (>=2.1.0)
```

2.3.1 Installation

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

```
<package installer> install open-xchange-omf-source-dualprovisioning-cloudplugins
```

2.4 Package open-xchange-omf-source-guard

OMF Support to migrate Guard data

Version: 2.1.0-26

Type: OX Middleware Plugin

Depends on:

```
open-xchange-guard
open-xchange-omf-source (>=2.1.0)
```

2.4.1 Installation

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

```
<package installer> install open-xchange-omf-source-guard
```

2.5 Package open-xchange-omf-source-mailfilter

OMF Source Mail Filter only Part of the Source component of OMF that supplies a core Mail Filter Provider.

Version: 2.1.0-26

Type: OX Middleware Plugin

Depends on:

```
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.8.4)
open-xchange-mailfilter (<<7.10.7)
```

```
open-xchange-mailfilter (>=7.8.4)
open-xchange-omf-source (<<3.0.0)
open-xchange-omf-source (>=2.1.0)
```

2.5.1 Installation

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

```
<package installer> install open-xchange-omf-source-mailfilter
```

A Configuration Files

File 1 /opt/open-xchange/etc/omf-source.properties

```
1  #
2  # The login of the user allowed to access the webservices
3  # Parameter is mandatory
4  #
5  com.openexchange.omf.source.webserviceLogin=
6
7  #
8  # The password of the user allowed to access the webservices
9  # Parameter is mandatory
10 #
11 com.openexchange.omf.source.webservicePassword=
12
13 # ldap connection pool setting, see
14 # https://docs.ldap.com/ldap-sdk/docs/getting-started/connection-pools.html
15 # all other ldap parameters are read from /opt/open-xchange/etc/cloudplugins.properties
16 # com.openexchange.cloudplugins.read.*
17 com.openexchange.omf.source.ldappool.initialConnections=10
18
19 # ldap connection pool setting, see
20 # https://docs.ldap.com/ldap-sdk/docs/getting-started/connection-pools.html
21 # all other ldap parameters are read from /opt/open-xchange/etc/cloudplugins.properties
22 # com.openexchange.cloudplugins.read.*
23 com.openexchange.omf.source.ldappool.maxConnections=20
24
25 # Location of the JKS trust store file that contains the certificates of the target HTTPS
26 # endpoint.
27 # Note that this configuration setting is only applied when the URL the
28 # target AppSuite endpoints are using the HTTPS protocol.
29 #
30 # The default value is empty, which causes the use of the CA certificates that are bundled
31 # with the Java Runtime Environment.
32 #
33 # Example:
34 # com.openexchange.omf.source.ssl.truststore.file=/opt/open-xchange/omf/source-keystore.
35 # jks
36 #
37 # Example for using the bundled CA certificates:
38 # com.openexchange.omf.source.ssl.truststore.file=
39 # com.openexchange.omf.source.ssl.truststore.file=
40
41 # The password to use to open the JKS trust store file.
42 # Only relevant when the configuration parameter above has been set.
43 # Leave empty if no password is necessary (which is the common practice and, hence, the
44 # default).
45 #
46 # Example:
47 # com.openexchange.omf.source.ssl.truststore.password=
48 # com.openexchange.omf.source.ssl.truststore.password=secret
49 # com.openexchange.omf.source.ssl.truststore.password=
50
51 # Should the dual provisioning be enabled?
52 # Dual provisioning ensures send mail related provisioning requests to the migration
53 # target
54 # Default: false
```

```

51 #
52 # Note that the dual provisioning feature also requires the open-xchange-omf-source-
    dualprovisioning
53 # package to be installed.
54 com.openexchange.omf.source.dualprovision.enabled=false
55
56 # Migration target URL
57 # Parameter is mandatory
58 # Example: https://provisioning.example.com
59 com.openexchange.omf.source.dualprovision.targetUrl=
60
61 # Migration source name
62 # Parameter is mandatory
63 # An alias for this this system used as migration source. The name must be known to the
    target.
64 # Example: foo.customer.site
65 com.openexchange.omf.source.dualprovision.sourceName=
66
67 # Ignore dual provisioning failures when the context cannot be found by the Target.
68 # This is useful when enabling dual provisioning before all contexts have been provisioned
69 # on the Target.
70 # It should not be enabled if you expect all contexts to already be provisioned.
71 com.openexchange.omf.source.dualprovision.ignoreContextNotFound=false
72
73 # Ignore dual provisioning failures when the user cannot be found by the Target.
74 # This is different from ignoreContextNotFound which will only ignore when the Target
75 # context cannot be found. This property will ignore the error when the Target context is
76 # found, but the user is not.
77 # If dual-provisioning is enabled prior to provisioning all contexts, and
    ignoreContextNotFound
78 # is enabled, it's possible that a context may be created but a user create event executed
79 # at the same time does not get propagated to the Target. That means that a user existing
80 # on Source may not exist on Target yet. This would be eventually be resolved, by follow
    up
81 # pre-provisioning runs or identified during presync or cutover, however, it would cause
82 # provisioning errors in the mean time.
83 com.openexchange.omf.source.dualprovision.ignoreUserNotFound=false
84
85 # Brand usernames and passwords to use to authenticate when provisioning users
86 # using the Dual-Provisioning feature.
87 #
88 # Format:
89 # com.openexchange.omf.source.dualprovision.targetBrandPassword.<brandName>=<password>
90 #
91 # Example:
92 com.openexchange.omf.source.dualprovision.targetBrandPassword.acme=secret
93 # com.openexchange.omf.source.dualprovision.targetBrandPassword.name.of.the.brand=
    super_secret
94 #
95 # Note that when the com.openexchange.omf.source.metadata.brandProvider property is
96 # set to 'single', then there should only be a single such entry, as exemplified below:
97 # com.openexchange.omf.source.metadata.brandProvider=single
98 # com.openexchange.omf.source.dualprovision.targetBrandPassword.acme=secret
99 # With that configuration above, new contexts will be provisioned into the target
100 # brand "acme", using "acme:secret" as the basic authentication credentials to do so.
101 #
102 # This setting is mandatory and has no default value.
103
104 # com.openexchange.omf.source.dualprovision.targetBrandPassword.<brandName>=<password>
105
106 # Login resolver strategy: how should logins be determined when exporting metadata.
107 #
108 # Two strategies are available:
109 # * cloudplugins: mailLogin = {userId}@{contextId}
110 # * loginsource: mailLogin = depends on the value of the config cascade aware
    configuration
111 #     property com.openexchange.mail.loginSource
112 #
113 # The parameter is optional and defaults to loginsource.
114 #
115 # Examples:
116 # com.openexchange.omf.source.loginResolver=cloudplugins

```



```

117 # com.openexchange.omf.source.loginResolver=loginsource
118 #
119 com.openexchange.omf.source.metadata.loginResolver=loginsource
120
121 # The method to use to determine the brand to use when creating contexts
122 # on the Target tier, which is used in two mechanisms:
123 # - in the Source metadata, which is used for the pre-provisioning operations,
124 # - in the Dual-Provisioning feature.
125 #
126 # For Source metadata and pre-provisioning operations, this setting determines the
127 # name of the plugin to use to push the brand to use for provisioning contexts and
128 # users on the target side by including it in the metadata.
129 # Not populated if left empty.
130 #
131 # For Dual Provisioning, that brand both determines the authentication credentials
132 # to use (specifically the basic authentication username) as well as in which target
133 # brand the context ought to be created.
134 # The matching password is configured using the property
135 # com.openexchange.omf.source.dualprovision.targetBrandPassword.<brandName>=<password>
136 #
137 # Built-in providers:
138 # - configCascade: looks up a configurable property via config cascade, optionally maps it
139 #   and
140 #   injects that as the brand
141 # - single: uses the brand credentials configuration properties to determine the brand,
142 #   requiring
143 #   that one and only one set of credentials is configured that way
144 #
145 # Example:
146 # com.openexchange.omf.source.metadata.brandProvider=configCascade
147 # com.openexchange.omf.source.metadata.brandProvider=single
148 #
149 # This configuration setting is optional and there is not a default
150 com.openexchange.omf.source.metadata.brandProvider=
151
152 # When using the configCascade brand provider, configures the name
153 # of the property that contains the brand information.
154 #
155 # Is mandatory when using the configCascade brand provider and has
156 # no default value.
157 #
158 # Example:
159 # com.openexchange.omf.source.metadata.brandProvider.configCascade.propertyName=config/io.
160 #   ox/core//theme
161 com.openexchange.omf.source.metadata.brandProvider.configCascade.propertyName=
162
163 # When using the configCascade brand provider, configures the optional
164 # default value that should be used when there is not a value found at the context
165 # level for the property configured in com.openexchange.omf.source.metadata.brandProvider.
166 #   configCascade.propertyName
167 #
168 # This property is optional. When not used, the brand will not be provided.
169 #
170 # Example:
171 # com.openexchange.omf.source.metadata.brandProvider.configCascade.default=defaultBrand
172 com.openexchange.omf.source.metadata.brandProvider.configCascade.default=
173
174 # When using the configCascade brand provider, optionally maps
175 # the values from the property defined above to another value that is
176 # then exported as the brand information.
177 #
178 # Syntax:
179 # com.openexchange.omf.source.metadata.brandProvider.configCascade.map.<from>=<to>
180 #
181 # Example:
182 # com.openexchange.omf.source.metadata.brandProvider.configCascade.map.pink_theme=brand1
183 #
184 # com.openexchange.omf.source.metadata.brandProvider.configCascade.map.<from>=
185
186 # The following property is only used when installing the open-xchange-omf-source-

```

```

mailfilter
185 # package. Otherwise, you can ignore it.
186 #
187 # The master authentication password to use when using the core mailfilter provider.
188 # If the core mail filter properties are already configured to use master password,
189 # or per user passwords will be supplied by OMF, then there is no need to use this.
190 com.openexchange.omf.source.mailfilter.masterPassword=
191
192 # The following property is only used when installing the open-xchange-omf-source-
mailfilter
193 # package. Otherwise, you can ignore it.
194 #
195 # When using SASL PLAIN authentication, it's possible to specify an authentication user
which
196 # is normally selected from the user's credentialSource, however, the auth user can be
197 # overridden here. This is useful when master auth uses a master user.
198 # This property is optional.
199 com.openexchange.omf.source.mailfilter.masterUser=
200
201 # The following property is only used when installing the open-xchange-omf-source-
mailfilter
202 # package. Otherwise, you can ignore it.
203 #
204 # Format the username with printf style format. This is useful when a separate pair of
credentials
205 # have been setup for mail migration. For instance, if the username is jon.doe@domain and
you
206 # specify a userFormat of "mig-%s" then the username will be "mig-jon.doe@domain".
207 # This property is optional.
208 com.openexchange.omf.source.mailfilter.userFormat=
209
210 # The following property is only used when installing the open-xchange-omf-source-
mailfilter
211 # package. Otherwise, you can ignore it.
212 #
213 # Format the authname with printf style format. For instance, if the authname is jon.
doe@domain
214 # and you specify an authFormat of "mig-%s" then the authname will be "mig-jon.doe@domain
".
215 # This property is optional.
216 com.openexchange.omf.source.mailfilter.authFormat=
217
218 # The following property is only used when installing the open-xchange-omf-source-
mailfilter
219 # package. Otherwise, you can ignore it.
220 #
221 # Override the configured mailfilter server.
222 # This property is optional.
223 com.openexchange.omf.source.mailfilter.server=
224
225 # The following property is only used when installing the open-xchange-omf-source-
mailfilter
226 # package. Otherwise, you can ignore it.
227 #
228 # Override the configured mailfilter port. Only used when server is also overridden.
229 # This property is optional.
230 # Default: 4190
231 com.openexchange.omf.source.mailfilter.port=
232
233 # If the mail filter hierarchy separator character (delimiter) that is used on
234 # the Source IMAP server differs from the character used on the Target, then
235 # the Source character will need to be mapped to the Target, and any reference
236 # of the Target character will need to be escaped.
237 # Reference: https://datatracker.ietf.org/doc/html/rfc3501#section-5.1.1
238 #
239 # If mail filters will not be migrated by OMF or the delimiters are the same
240 # on each system, then there is no need to use these properties.
241 #
242 # Specify the Source delimiter with:
243 # com.openexchange.omf.source.mailfilter.sourceDelimiter
244 # Specify the Target delimiter with:
245 # com.openexchange.omf.source.mailfilter.targetDelimiter

```

```

246 # Specify the Target delimiter escape character with:
247 #   com.openexchange.omf.source.mailfilter.targetEscapeChar
248
249 # Configure Standard folder names that should be mapped from one value to another during
250 # the mail filter migration process.
251 # For example, if you instruct OMF to map folder Draft to Drafts, then a reference in a
252 # mail filter of Draft.Subfolder would end up as Drafts.Subfolder.
253 # Note: this only takes care of cases when the Standard folder is at the root, so
254 #   cases like INBOX.Draft.Subfolder will stay as is.
255 #
256 # There is no default value.
257 #
258 # Format:
259 #   com.openexchange.omf.source.mailfilter.folder.<fromfoldername>=<tofoldername>
260 #
261 # Example:
262 #   com.openexchange.omf.source.mailfilter.folder.Draft=Drafts
263 #   com.openexchange.omf.source.mailfilter.folder.Drafts=Draft
264
265 # When usernames (uid) are not unique, a different strategy needs to be used to determine
266 # what to send as the username (uid) as part of the metadata.
267 #
268 # Built-in strategies are:
269 # - uid: it's the default and uses the 'uid' field from the 'login2user' table
270 # - email: uses the 'mail' field from the 'user' table
271 # - brand: concatenates the following fields with '_':
272 #   * 'uid' field from the 'login2user' table
273 #   * the numeric user id
274 #   * the numeric context id
275 #   * the source name
276 #   * the target brand
277 # - brandcontext: concatenates the following fields with '_':
278 #   * 'uid' field from the 'login2user' table
279 #   * the numeric user id
280 #   * the numeric context id
281 #   * the context name
282 #   * the source name
283 #   * the target brand
284 # - format: creates a username based on property 'com.openexchange.omf.source.metadata.
    username.format'
285 #
286 # Note that using 'brand' or 'brandcontext' requires a target brand strategy to be
287 # enabled and selected using the property com.openexchange.omf.source.metadata.
    brandProvider,
288 # as well as a source name to be set in com.openexchange.omf.source.dualprovision.
    sourceName.
289 #
290 # Using 'brandcontext' also incurs a slight performance penalty since it requires
    resolving
291 # the context name from the context id using the 'context' table in the configdb.
292 #
293 # It can also be configured as a chain of strategies to attempt, with the
294 # leftmost strategy winning (first hit wins), e.g.:
295 #   com.openexchange.omf.source.metadata.usernameStrategy=email, uid
296 #
297 # Note that this property is optional, defaulting to using the 'uid' column.
298 #
299 # If the architecture of the Source is Cloud-Plugins, then this property must be empty or
    'uid'
300 #
301 # Example:
302 #   com.openexchange.omf.source.metadata.usernameStrategy=email
303 #   com.openexchange.omf.source.metadata.usernameStrategy=
304
305 # Define the username with defined keys within the formatter "${}" when using
306 #   com.openexchange.omf.source.metadata.usernameStrategy=format
307 #
308 # Possible Keys:
309 #   * uid: field from the 'login2user' table
310 #   * userId: the numeric user id
311 #   * contextId: the numeric context id
312 #   * email

```

```

313 # * source: the source name
314 # * brand: the target brand name
315 # * sourceContextName
316 # * targetContextName: might be the same as sourceContextName
317 #
318 # Key brand requires property: com.openexchange.omf.source.metadata.brandProvider
319 # Key source requires property: com.openexchange.omf.source.dualprovision.sourceName
320 #
321 # Example:
322 # com.openexchange.omf.source.metadata.username.format=customer_${userId}_${contextId}_${
    uuid}
323 # where userId = 4 and contextId=87494 would result in: customer_4_87494_8f3078a8-9bf9-4
    d37-ad2e-da003ac0ac68
324 com.openexchange.omf.source.metadata.username.format=
325
326 # When you need to modify the context name that will be used on the Target
327 #
328 # Built-in strategies are:
329 # - name: takes the context name as-is (default)
330 # - string-replace: uses the context.replace properties to perform string replace on the
    existing context name
331 #
332 #
333 # Example:
334 # com.openexchange.omf.source.metadata.contextNameStrategy=name
335 com.openexchange.omf.source.metadata.contextNameStrategy=
336
337 # To control the "string-replace" contextNameStrategy that is configured with property
338 # com.openexchange.omf.source.metadata.contextNameStrategy, use the following properties.
339 #
340 # Property prefix "com.openexchange.omf.source.metadata.context.replace.with." is used to
341 # replace a regex in the existing context name with a string (can be empty).
342 # Example:
343 # com.openexchange.omf.source.metadata.context.replace.with.replaceme=withme
344 # This will replace the string "replaceme" with the string "withme". Other examples of the
345 # regex to be replaced are: "guid\+", "^replaceit", "test$"
346 # You can use the special string "|empty" to replace with an empty string.
347 #
348 # Property prefix "com.openexchange.omf.source.metadata.context.replace.first." is used to
349 # inform the contextStrategy that the regex should be replaced only once in case there are
350 # multiple instances of it in the context name.
351 # Example:
352 # com.openexchange.omf.source.metadata.context.replace.first.replaceme=true
353 # This will replace the regex "replaceme" only one time. The result of this for context
    name
354 # "replacemereplaceme" will be "withmerekplaceme".
355 # This property is optional and defaults to false.
356 #
357 # Property "com.openexchange.omf.source.metadata.context.replace.order" determines the
    order
358 # of replacement when there are multiple strings to replace. Each regex is delimited with
    the
359 # character '|'
360 # Example:
361 # com.openexchange.omf.source.metadata.context.replace.with.replaceme=withme
362 # com.openexchange.omf.source.metadata.context.replace.with.replaceyou=replaceme
363 # com.openexchange.omf.source.metadata.context.replace.order=replaceyou|replaceme
364 # This will first replace "replaceyou" and then "replaceme". So the context name "
    replaceyou"
365 # will change to "replaceme" and finally "withme".
366 # This property is optional, but if it is used, all replacements must be specified. When
    it is
367 # not used, replacement order is random.
368 #
369 # Comma separated list of user attributes that should be included in source metadata
370 #
371 # Example:
372 # com.openexchange.omf.source.metadata.userAttributes=foo,bar,spam
373 #
374 # Comma separated list of database table names that should be excluded from the migration
375 # Example:
376 # com.openexchange.omf.source.database.export.excludedTableNames=myTable1,anotherTable2
377 com.openexchange.omf.source.database.export.excludedTableNames=

```

```
378
379 # An export gate which controls access to context data. This enhances security to allow
    only
380 # the data of selected contexts to be exported. By default, all contexts may be exported
381 # so that this setting can be deployed without breaking existing migrations. Note:
    metadata
382 # API's which only export contextId to schema mappings do not use this gate, therefore you
383 # can still learn of the existence of a contextId on the system, even if its data is gated
    .
384 #
385 # Base options:
386 #   <empty> or "allowall": a noop which allows export of all contexts on this system
387 #   "allownone": denies export of all contexts. This is set when this configuration option
    fails to load
388 #
389 # In addition to base options, a plugin may provide additional named ContextExportGates
390 # Example:
391 # com.openexchange.omf.source.context.export.gate.name=customer-x-gate
392 com.openexchange.omf.source.context.export.gate.name=
393
394 # Configure a cache for the context export gate. This is only useful when actually using a
    context
395 # export gate other than allowall. You can invalidate the cache with API
396 # /omf/source/metadata/1/gate/invalidate/{contextId} where no contextId invalidates the
    whole cache
397 #
398 # Enable the cache
399 # com.openexchange.omf.source.context.export.gate.cache.enable=
400 #
401 # Hours until entry is evicted
402 # com.openexchange.omf.source.context.export.gate.cache.hours=
403 #
404 # Max number of entries in the cache
405 # com.openexchange.omf.source.context.export.gate.cache.max=
406 #
407 # Defaults:
408 # com.openexchange.omf.source.context.export.gate.cache.enable=false
409 # com.openexchange.omf.source.context.export.gate.cache.hours=24
410 # com.openexchange.omf.source.context.export.gate.cache.max=100000
411
412
413 # The connect timeout for all outbound HTTP/REST requests.
414 #
415 # Example:
416 # com.openexchange.omf.http.connect.timeout=2m
417 #
418 # Defaults to 1m.
419 com.openexchange.omf.http.connect.timeout=1m
420
421 # The read timeout for all outbound HTTP/REST requests.
422 #
423 # Example:
424 # com.openexchange.omf.http.read.timeout=10m
425 #
426 # Defaults to 5m.
427 com.openexchange.omf.http.read.timeout=5m
428
429 # The write timeout for all outbound HTTP/REST requests.
430 #
431 # Example:
432 # com.openexchange.omf.http.write.timeout=10m
433 #
434 # Defaults to 5m.
435 com.openexchange.omf.http.write.timeout=5m
436
437 # The read timeout for slow outbound HTTP/REST requests.
438 #
439 # Example:
440 # com.openexchange.omf.http.slow.read.timeout=20m
441 #
442 # Defaults to 30m.
443 com.openexchange.omf.http.slow.read.timeout=30m
```

```
444
445 # The write timeout for slow outbound HTTP/REST requests.
446 #
447 # Example:
448 # com.openexchange.omf.http.slow.write.timeout=12m
449 #
450 # Defaults to 30m.
451 com.openexchange.omf.http.slow.write.timeout=30m
```