NewgenONE OmniOMS

Utility

User Guide

Version: 12.0

Contents

Disclaimer

This document contains information proprietary to Newgen Software Technologies Ltd. User may not disclose or use any proprietary information or use any part of this document without written permission from Newgen Software Technologies Ltd.

Newgen Software Technologies Ltd. makes no representations or warranties regarding any software or to the contents or use of this guide. It also specifically disclaims any express or implied warranties of merchantability, title, or fitness for any particular purpose. Even though Newgen Software Technologies Ltd. has tested the hardware and software and reviewed the documentation, it does not guarantee or imply that this document is error free or accurate regarding any particular specification. As a result, this product is sold as it is and user, the purchaser, is assuming the entire risk as to its quality and performance. Further, Newgen Software Technologies Ltd. reserves the right to revise this publication and make changes in its content without any obligation to notify any person, of such revisions or changes. Newgen Software Technologies Ltd. authorizes no Newgen agent, dealer or employee to make any modification, extension, or addition to the above statements.

Newgen Software Technologies Ltd. has attempted to supply trademark information about company names, products, and services mentioned in this document. Trademarks indicated below were derived from various sources.

Newgen Software, Registered Office, New Delhi
E-44/13
Okhla Phase - II
New Delhi 110020
India
Phone: +91 1146 533 200
info@newgensoft.com

Preface

This section provides information about the revision history, about this guide, details on the intended audience, related documents for this guide, and documentation feedback.

Revision history

Revision date	Description
October 2024	Initial publication

About this guide

The guide provides information on the OmniOMS Health Check Application responsible for checking the health of the OmniOMS Webservices and Tracking Services.

To ensure you are referring to the latest and most recent revision of this guide, download it from one of the following locations:

- Newgen Internal Doc Portal, if you are a Newgen employee.
- Newgen Partner Portal, if you are a Newgen partner.

Intended audience

This guide is intended for system administrators, developers, and all other users who are seeking information on the functioning of the various features of OmniOMS and to aid them in configuration, maintenance, and development. The reader must be comfortable in understanding the computer terminology.

Related documents

The following documents provide additional information about OmniOMS Composition Designer.

- NewgenONE OmniOMS Release Notes
- NewgenONE OmniOMS Composition Designer Installation and Configuration Guide
- NewgenONE OmniOMS Composition Designer User Guide
- NewgenONE OmniOMS Installation and Configuration Guide
- NewgenONE OmniOMS Admin Workspace User Guide

Documentation feedback

To provide feedback or any improvement suggestions on technical documentation, write an email to docs.feedback@newgensoft.com.

To help capture your feedback effectively, share the following information in your email:

- Document name
- Version
- Chapter, topic, or section
- Feedback or suggestions

OmniOMS health check monitoring application

OmniOMS Health Check Application is a proactive monitoring application that is responsible for checking the health (responsiveness) of OmniOMS Webservices APIs and Timer Services. This ensures that the users are aware in case of any failure of the functioning of these services.
The responsiveness of OmniOMS Webservices and Timer Services is checked at user-defined intervals and will wait for the response till the configured timeout interval.
In case any unresponsiveness is encountered in these services, an email notification is triggered to users from this application to take necessary actions.

There are two properties files to be configured by the users to run the OmniOMS Health Check Monitoring Application.

- OMSHealthCheck.properties
- MailConfig.xml

OmniOMS health check properties

The OmniOMS Health Check Properties file is used to configure the parameters around the URLs, Type of APIs, Frequency, Exception Count, and Timeout, and more.
The following are the details of the parameters present in the *OMSHealthCheck.properties* file:

Parameter	Details	Default values
url	Information related to URLs - http/https, ip, port. For Example - http://localhost:8080 If the check has to be done for multiple cabinets associated with the same application server, then http://localhost:8080#Cabinet1#Cabinet2 (each cabinet should start with #) If the application needs to monitor different cabinets on different App Servers, then *http://localhost:8080#Cabinet1, http://localhost:8080,http://localhost:8081* (Comma-separated)	*http://localhost:8080*
apiType	Run health check for the type of Webservices APIs. Valid values are: • R- REST • S- SOAP • B – Both REST & SOAP	B
freqInSec	Frequency of time in seconds to check the API's health. For Example, 120 (every 120 seconds the application checks for APIs health)	14400
mailAfterExceptionCount	Number of times to retry to check the webservices APIs responsiveness before sending mail to the user. For Example 3 (The application waits for 3 unsuccessful responses and then mail the users about the unresponsiveness)	3
timeOut	Time in seconds to check the connection/read timeout response from the OmniOMS Webservice APIs For Example 60 (The application waits for 60 seconds whenever it checks for API health, in case it doesn't get a response by 60 seconds then the application treats the APIs are unresponsive)	120
mailSend	Y to send mail else N	N
startMonitoring	Users should be set to Y to start the monitoring, else set to N to stop monitoring. By default, it is set to N.	N

Parameter	Details	Default values
loggerFilePath	Log Filepath to store the log file for the OmniOMS Health Check Monitoring Application If the specified file path does not exist, then the application will create the log file accordingly in the specified path. If this is empty, then we are creating the log file using the path of the user.dir For Example C:*jboss-eap-7.0**bin**HealthCheck**Logs**HealthCheck.log*	*D:\\JBOSS**Build_136**jboss-eap-7.0**bin**HealthCheck*\

Mail configuration

The mail configuration file contains the parameters related to the mail account, mail server, mail body, and so on. The following are the details of the parameters present in the *MailConfig.xml*.

Parameters	Details
AccountUser	Username of the Mail Account
AccountPassword	Password of the Mail Account
AccountEmailId	Email-Id for sending mail i.e., sender's email address
MailProtocol	Mail Protocol (SMTP, PoP3 etc.,)
MailServerAddress	Mail Server IP
MailServerPort	Mail Server Port
AuthFlag	Set to true if authentication to mail server is required
TLSFlag	Set to true if TLS is enabled on mail server
SSLFlag	Set to true if SSL is enabled on mail server
MailSubject	Subject of the Mail to the user
MailBody	Body of the Mail to the user
MailFooter	Footer of the Mail to the user
To	Email of the users to whom the health check monitoring status update mail should be sent (Comma separated)
MailCC	Email of the users to whom the health check monitoring status update mail should be in CC (Comma separated)
MailBCC	Email of the users to whom the health check monitoring status update mail should be in BCC (Comma separated)

Enabling delivery of Facebook and X communications

This section outlines the process for enabling the delivery of Facebook and X Communications, covering the deployment of the *NewgenONESocialConnector.war* and necessary configuration updates.

Two tags need to be updated in the *tracker.properties* file for Facebook communication:

- MyVerifyToken - Token from Facebook.
- FBTrackingURL - Facebook tracking URL.

After enabling the delivery of Facebook and X communications, register the appropriate services in the Admin Workspace for communication delivery.

Tracking configuration steps

Following are the tracking configuration steps involved:

Mail

1. Enable Reports (PDF and HTML) for Tracking in the Designer and Baseline it.
2. Create a mail Job and associate the baselined report.
3. Associate the mail Job with the schedule.
4. Create SMTP Channel for mail service with the below details:
   Additionally, for tracking bounced emails – create Bounce Channel similar to SMTP Channel.
5. Create a Mail Service using the SMTP Channel Configured in step 4 and run the service.
   If bounced emails need to be tracked, then create a Bounce Mail Service by associating the Bounce Channel created in step 4.
6. Enable the Mail Tracking in the Admin Console
   a. Go to Connections -> Trackers
   b. Enable the toggle for Enable Mail Tracker
   c. Provide the Servlet URL & Click on Test Connection
      Servlet URL: <IP>:<Port>/OMSTrackingServlet/
      Once the successful toast message appears, it means tracking is enabled for all the Mail Communications.
   d. Initiate the Mail Communication generation and delivery (either Batch or On-demand mode).
   e. Upon successful generation of communications go to Admin Console -> Monitor.
   f. Choose the appropriate schedule and check for the generation and delivery statistics.

Message

1. Enable Message Report for Tracking in the Designer and Baseline it.
2. Create a Message Job and associate the baselined report along with Webservice URL for Output Channel Mapping.
3. Associate the Message Job with the schedule.
4. Create a Message Service using the App Server.
5. Enable the Message Tracking in the Admin Console.
6. Go to Connections -> Trackers
7. Enable the toggle for Enable Message Tracker.
8. Initiate the Message communication generation and delivery (either Batch or On-demand mode).
9. Upon successful generation of communications go to Admin Console -> Monitor.
10. Choose the appropriate schedule and check for the generation and delivery statistics.
11. Go to Admin Console -> Tracker
12. Choose the filters Tracking Type: Message
    Now users can see the tracking details of the Message communications accordingly. Additionally, users can even filter on Jobs and Product Name.

WhatsApp

1. Enable Report for Tracking in the Designer and Baseline it.
2. Create a WhatsApp Job and associate the baselined report.
3. Associate the WhatsApp Job with the schedule.
4. Create the WhatsApp Channel with the below details:
    a. WhatsApp URL
    b. Source Number
    c. API Key
    d. Source Name
        i. Create and start the WhatsApp Service using the App Server and WhatsApp Channel
        ii. Initiate the WhatsApp Message communication generation and delivery (either Batch or On-demand mode)
        iii. Enable the WhatsApp Tracking in the Admin Console.

- Go to Connections -> Trackers
- Enable the toggle for Enable WhatsApp Tracker
- Provide the WhatsApp Servlet URL and click on Test Connection
  Servlet URL: <IP>:<Port Number>/OMSTrackingServlet/

Once the successful toast message appears then, it means tracking is enabled for all the WhatsApp Communications.

> The Tracker and the App Server Port Number should be different.
> If it has to be changed then go to the file path
> *C:\jws 6.0\jws-6.0.0-application-server\jws-6.0\tomcat\conf*
> and edit the file (change the port number in case the OMS Utility is installed on the same App Server.

5. Call the generation API for WhatsApp communication generation and delivery
6. Upon successful generation of communications go to Admin Console -> Monitor
7. Choose the appropriate schedule and check for the generation and delivery statistics.
8. Go to Admin Console -> Tracker
9. Choose the filters Tracking Type: WhatsApp
   Now users can see the tracking details of the WhatsApp communications accordingly.
   Additionally, users can even filter on Jobs and Product Name.

Promotional tracking

1. Go to CDG and create an HTML report with an image.
2. Enable Dynamic Content for that Image.
3. Add a decision table having target URL on the image and enable tracking to the HTML Report.
4. Preview the Report and Save it.
5. Enable Tracking for the report in the Report Properties.
6. Check-In and Baseline the report.
7. Create a mail Job and associate the baselined report.
8. Associate the mail Job with the schedule.
9. Create SMTP Channel for mail service with the below details:
   a. Additionally, for tracking bounced emails – create Bounce Channel similar to SMTP Channel.
   b. Create a Mail Service using the SMTP Channel Configured in step 4 and run the service.

   c. If bounced emails need to be tracked, then create a Bounce Mail Service by associating the Bounce Channel.
   d. Enable the Mail Tracking in the Admin Console
   e. Go to Connections -> Trackers
   f. Enable the toggle for Enable Mail Tracker.
      Provide the Servlet URL & Click on Test Connection
      Servlet URL:<IP>:<Port Number>/OMSTrackingServlet/
10. Once the successful toast message appears then, it means tracking is enabled for all the Mail Communications.
11. Initiate the Mail communication generation and delivery (either Batch or On-demand mode)
12. Upon successful generation of communications go to Admin Console -> Monitor
13. Choose the appropriate schedule and check for the generation and delivery statistics.
14. Go to Admin Console -> Tracker
15. Choose the filters Tracking Type: Promotional
    Now users can see the tracking details of the Mail communications accordingly. Additionally, users can even filter on Jobs and Product Name.

Facebook DM

1. Enable Facebook Report for Tracking in the Designer and Baseline it.
2. Create a Facebook DM Job and associate the baselined report.
3. Associate the Facebook DM Job with the schedule.
4. Create and test the Facebook Channel using details
   a. Social Connector URL
   b. Access Token
5. Create a Facebook Service using the App Server.
6. Update the FBTrackingURL *<domainAddress>* in the properties file present in Utility-Installer.
   FBTrackingURL: *<domainAddress>/OMSTrackingServlet/**.

   - Utility-Installer must be deployed on a globally accessible network.
   - *domainAddress* is Utility-Installer deployment path.

7. Initiate the Facebook communication generation and delivery (either Batch or On-demand mode).
8. Upon successful generation of communications go to Admin Console -> Monitor.

9. Choose the appropriate schedule and check for the generation and delivery statistics.
10. Go to Admin Console -> Tracker
11. Choose the filters Tracking Type: Facebook
    Now users can see the tracking details of Facebook communications accordingly. Additionally, users can even filter on Jobs and Product Name.